

**IBM solidDB
IBM solidDB Universal Cache
バージョン 6.5**

IBM solidDB Universal Cache ユーザー・ガイド



注記

本書および本書で紹介する製品をご使用になる前に、169 ページの『特記事項』に記載されている情報をお読みください。

本書は、バージョン 6 リリース 5 の IBM solidDB (製品番号 5724-V17) および IBM solidDB Universal Cache (製品番号 5724-W91)、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。

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： SC23-9877-01
IBM solidDB
IBM solidDB Universal Cache
Version 6.5
IBM solidDB Universal Cache User
Guide

発行： 日本アイ・ビー・エム株式会社

担当： トランスレーション・サービス・センター

第1版第1刷 2010.10

© Solid Information Technology Ltd. 1993, 2010

目次

図	v
表	vii
変更の要約	ix
本書について	xi
書体の規則	xi
構文表記法の規則	xii
1 IBM solidDB Universal Cache の概要	1
solidDB Universal Cache のアーキテクチャーの概要	2
Universal Cache の構成	4
機能	8
制限事項	10
データベースの制限事項	10
InfoSphere CDC の制限事項	11
セキュリティと認証	12
2 solidDB Universal Cache のインストールと構成	13
solidDB Universal Cache のインストールおよび構成手順の概要	13
solidDB Universal Cache のインストール・パッケージ	15
solidDB Universal Cache 用のユーザー・アカウントおよびデータベース接続データ	16
solidDB Universal Cache 用の solidDB サーバーのインストールと構成	19
solidDB Universal Cache 用の solidDB サーバーのインストール	19
solidDB for solidDB Universal Cache の構成	19
InfoSphere CDC for solidDB のインストールと構成	21
バックエンド・データ・サーバーのインストールおよび構成	22
バックエンド・データ・サーバー用の InfoSphere CDC のインストールと構成	22
InfoSphere CDC Access Server のインストールと構成	23
InfoSphere CDC Management Console のインストールと構成	24
レプリケーション・サブスクリプションのセットアップ	24
3 InfoSphere CDC 用の solidDB Universal Cache の固有の設定およびタスク	29
solidDB Universal Cache 固有の InfoSphere CDC システム・パラメーター設定 - 要約	29
外部キーの使用の有効化 (参照整合性)	30

solidDB ソース表のドロップおよび再作成	31
データを同期化しないミラーリングの開始	31
solidDB Universal Cache での Unicode データベースおよび部分的 Unicode データベースの使用	32
高速リフレッシュの使用可能化	33
solidDB Universal Cache での共有メモリー・アクセス (SMA) の使用	34

4 パフォーマンスのチューニングおよびモニター	37
solidDB Universal Cache のパフォーマンスに影響を与える要因	37
パフォーマンスのモニター	40

5 SQL パススルー	41
操作の原理	42
SQL パススルーを使用したアプリケーション開発に関する考慮事項	44
SQL パススルーの構成と使用	47
SQL パススルーのセットアップ	47
SQL パススルー・モードの設定と変更	58
SQL パススルーのトレースとモニター	59
SQL パススルーおよび solidDB 高可用性	61
SQL パススルーの障害の処理	61

6 データ・エージング	63
動作原理	63
データ・エージングの使用	66

7 ツールおよびユーティリティ	69
Perl 自動化フレームワーク	70
インスタンスおよびサブスクリプション管理ツール	71
ucdeploy - 構成およびセットアップ・サンプル	71
ucpassthrough - SQL パススルー・セットアップ・サンプル	72
uchsbmonitor - HSB サブスクリプション・モニター・サンプル	72
データ・エージングとリフレッシュのための SQL ストアード・プロシージャ	72
エージング・ストアード・プロシージャの使用	73
リフレッシュ・ストアード・プロシージャの使用	76
例: 双方向サブスクリプションのためのデータ・エージングの自動化	79

8 solidDB Universal Cache の障害シナリオ	81
スタンドアロン solidDB サーバーの障害	81
InfoSphere CDC インスタンスの障害	81
HA モード (HotStandby) の solidDB サーバーの障害	82

1 次 solidDB サーバーと InfoSphere CDC for solidDB インスタンス間の通信リンクの障害	83
バックエンド・サーバーまたはバックエンド・ノードの障害	83
バックエンド 1 次サーバーの障害	84

9 トラブルシューティング 85

10 InfoSphere CDC for solidDB (エンド・ユーザー向け資料) 89

このセクションについて	89
InfoSphere CDC for IBM solidDB について	89
システム要件	89
必須のデータベース、ユーザー・アカウント、およびスキーマ	90
1 バイト文字およびマルチバイト文字のサポート	91
InfoSphere CDC のインストール	93
InfoSphere CDC の対話式インストール	93
InfoSphere CDC のサイレント・インストール	94
InfoSphere CDC の構成 (Windows)	94
InfoSphere CDC インスタンスの構成 (Windows)	95
InfoSphere CDC の構成 (UNIX および Linux)	98
InfoSphere CDC インスタンスの構成 (UNIX および Linux)	98
InfoSphere CDC の開始と停止	102
InfoSphere CDC の開始	102
InfoSphere CDC の停止	102
Management Console での SQL ステートメントの使用可能化	103
InfoSphere CDC がサポートするデータ型	104
サポートされているデータ型	104
サポートされているマッピング	105
InfoSphere CDC メタデータ表	106
InfoSphere CDC のコマンド	107
InfoSphere CDC コマンドの使用	107
TSINSTANCE 環境変数の設定	108
レプリケーション・コマンドの制御	108

データベース・トランザクション・ログ・コマンド	111
レプリケーション・コマンドに関する表の管理	115
レプリケーション・コマンドのモニター	121
構成コマンドのエクスポートとインポート	125
その他のコマンド	126
InfoSphere CDC のユーザー・出口	132
表レベルおよび行レベルの操作のためのストアード・プロシージャ・ユーザー・出口	133
ストアード・プロシージャ・ユーザー・出口の定義	133
ストアード・プロシージャ・ユーザー・出口のデータベース接続	133
ストアード・プロシージャ・ユーザー・出口でのデータのリトリート	133
ストアード・プロシージャ・ユーザー・出口の例	139
InfoSphere CDC のサンプル・ユーザー・出口	141
競合解決監査表	143
Management Console 管理ガイドの付録	147
ユーザー・出口の構成	147
InfoSphere CDC for solidDB のシステム・パラメーター	149

付録 A. ログ・リーダーのパラメーター 157

付録 B. SQL パススルー・パラメーター 159

付録 C. SQL パススルーでの ODBC データ型のサポート 163

付録 D. バックエンド ODBC ドライバー接続ストリング用のフォーマット規則 (RemoteServerDSN パラメーター) 167

特記事項 169



1.	solidDB Universal Cache アーキテクチャー	2	5.	例: 3 つのパーティション化モデルを使用した solidDB Universal Cache のセットアップ	39
2.	solidDB Universal Cache のデフォルト・デプロイメント	5	6.	SQL パススルー	41
3.	複数の solidDB サーバーを備えた solidDB Universal Cache のデプロイメント	6	7.	SQL パススルーのアーキテクチャー	42
4.	solidDB 高可用性を備えた solidDB Universal Cache のデプロイメント	7	8.	データ・エージング・アーキテクチャー	64

表

1. 書体の規則	xi	13. InfoSphere CDC for solidDB パッケージ - ツールおよびユーティリティー	69
2. 構文表記法の規則	xii	14. solidDB パッケージ - ストアード・プロシージャ	70
3. solidDB のインストール・イメージ	15	15. AUX_AUTOMATIC_DELETES 表の定義	73
4. InfoSphere CDC のインストール・イメージ	15	16. AUX_AUTOMATIC_DELETES_BREAK 表の定義	74
5. solidDB のユーザー・アカウントおよびネットワーク接続データ	16	17. エージング・プロシージャを作成および実行するスクリプト	75
6. InfoSphere CDC for solidDB のユーザー・アカウントおよびネットワーク接続データ	17	18. TS_REFRESH 表の定義	76
7. バックエンド・データ・サーバー用の InfoSphere CDC のユーザー・アカウントおよびネットワーク接続データ	18	19. ハードウェア要件	90
8. Access Server のユーザー・アカウントおよびネットワーク接続データ	18	20. ポート要件	90
9. Management Console のユーザー・アカウントおよびネットワーク接続データ	18	21. 文字データ型列およびワイド文字データ型列のデフォルト・エンコード設定 (部分的 Unicode) および Unicode エンコード設定	92
10. solidDB Universal Cache 固有の InfoSphere CDC システム・パラメーター設定	29	22. ログ・リーダーのパラメーター	157
11. dmconfigurets を使用した SMA 接続の使用可能化	34	23. SQL パススルー・パラメーター	159
12. Perfmon カウンター	60	24. サポートされているデータ型	163
		25. 変換されるデータ型	164
		26. サポートされていない SQL 標準データ型	164

変更の要約

改訂 01 での変更点

- 以下のパラメーターが、セクション『SQL パススルー・パラメーター』に新たに追加されました。
 - **Passthrough.ComplexNumNonindexedConstr**
 - **Passthrough.ComplexNumOrderedRows**
 - **Passthrough.ComplexNumTables**
- セクション『SQL パススルーのためのバックエンド ODBC ドライバーのインストールと構成』（ステップ 1）で、SQL パススルー用の *IBM Data Server Driver for ODBC and CLI* をインストールおよび構成するための AIX® 固有の新手順が追加されました。
- 追加された新しいセクションは、以下のとおりです。
 - solidDB® Universal Cache 固有の InfoSphere™ CDC システム・パラメーター設定
 - solidDB Universal Cache での Unicode データベースおよび部分的 Unicode データベースの使用
 - 高速リフレッシュの使用可能化
 - solidDB Universal Cache での共有メモリー・アクセス (SMA) の使用
 - Perl 自動化フレームワーク
 - `solid_fast_refresh_apply_pipes` (システム・パラメーター)
 - `solid_fast_refresh_on` (システム・パラメーター)
- 新規パラメーター **Passthrough.Force32bitODBCHandles** が、セクション『SQL パススルー・パラメーター』に追加されました。
- セクション『SQL パススルー・パラメーター』の以下のパラメーターのアクセス・モードが `RW/Startup` に変更されました (パラメーター値の変更を有効にするには、サーバーを再始動する必要があります)。
 - **Passthrough.PassthroughEnabled**
 - **Passthrough.RemoteServerDSN**
- セクション『エージング・ストアード・プロシージャの使用』のサンプル構文が修正されました。
- 新しい例が、UTF-8 のサポート付きで、SQL パススルーのための IBM® Data Server Driver for ODBC and CLI のインストールおよび構成用に追加されました。
- バックエンド・ログイン・データを入力するための構文については、セクション『バックエンドのログイン・データの手動設定』で詳しく説明しています。
- 最も一般的な solidDB Universal Cache タスクのスクリプトを記述するための自動化ツールについての情報が、セクション『ツールおよびユーティリティー』に追加されています。

- コード・ページ・サポートについての情報が、セクション『SQL パススルーを使用したアプリケーション開発に関する考慮事項』に追加されています。

本書について

IBM solidDB Universal Cache は、従来のディスク・ベースの SQL データ・サーバーを高速化するためのソリューションであり、1 つ以上の solidDB インメモリ・データベース・インスタンスを使用して、アプリケーションとデータ・サーバー間のデータ・トラフィックをキャッシュに入れます。solidDB インスタンスとデータ・サーバー・インスタンス間のデータ・レプリケーションは、IBM InfoSphere Change Data Capture テクノロジーを使用してインプリメントされています。

本書では、solidDB Universal Cache の概要を示し、さらに solidDB Universal Cache のインストールと構成について説明します。また、障害とトラブルシューティングのシナリオに対処するためのガイドラインも記載します。CDC for solidDB の章に、InfoSphere CDC for solidDB をインストールし、構成する方法の詳しい説明が記載されています。このセクションは、solidDB Universal Cache の構成時に必要になります。ご使用のバックエンド・データ・サーバーに関して、「InfoSphere Change Data Capture のエンド・ユーザー向け資料」に対応する情報を提供します。

このマニュアルでは、読者が一般的なデータベース管理システム (DBMS) に関する知識を持ち、SQL と solidDB に習熟していることを想定しています。

書体の規則

solidDB の資料では、以下の書体の規則を使用します。

表 1. 書体の規則

フォーマット	用途
データベース表	このフォントは、すべての通常テキストに使用します。
NOT NULL	このフォントの大文字は、SQL キーワードおよびマクロ名を示しています。
solid.ini	これらのフォントは、ファイル名とパス式を表しています。
SET SYNC MASTER YES; COMMIT WORK;	このフォントは、プログラム・コードとプログラム出力に使用します。SQL ステートメントの例にも、このフォントを使用します。
run.sh	このフォントは、サンプル・コマンド行に使用します。
TRIG_COUNT()	このフォントは、関数名に使用します。
java.sql.Connection	このフォントは、インターフェース名に使用します。
LockHashSize	このフォントは、パラメーター名、関数引数、および Windows® レジストリー項目に使用します。

表 1. 書体の規則 (続き)

フォーマット	用途
<i>argument</i>	このように強調されたワードは、ユーザーまたはアプリケーションが指定すべき情報を示しています。
管理者ガイド	このスタイルは、他の資料、または同じ資料内の他の章の参照に使用します。新しい用語や強調事項もこのように記述します。
ファイル・パス表示	特に明記していない場合、ファイル・パスは UNIX® フォーマットで示します。スラッシュ (/) 文字は、インストール・ルート・ディレクトリーを表します。
オペレーティング・システム	資料にオペレーティング・システムによる違いがある場合は、最初に UNIX フォーマットで記載します。UNIX フォーマットに続いて、小括弧内に Microsoft® Windows フォーマットで記載します。その他のオペレーティング・システムについては、別途記載します。異なるオペレーティング・システムに対して、別の章を設ける場合があります。

構文表記法の規則

solidDB の資料では、以下の構文表記法の規則を使用します。

表 2. 構文表記法の規則

フォーマット	用途
INSERT INTO <i>table_name</i>	構文の記述には、このフォントを使用します。置き換え可能セクションには、このフォントを使用します。
solid.ini	このフォントは、ファイル名とパス式を表しています。
[]	大括弧は、オプション項目を示します。太字テキストの場合には、大括弧は構文に組み込む必要があります。
	垂直バーは、構文行で、互いに排他的な選択項目を分離します。
{ }	中括弧は、構文行で互いに排他的な選択項目を区切ります。太字テキストの場合には、中括弧は構文に組み込む必要があります。
...	省略符号は、引数が複数回繰り返し可能なことを示します。
• • •	3 つのドットの列は、直前のコード行が継続することを示します。

1 IBM solidDB Universal Cache の概要

IBM solidDB Universal Cache は、従来のディスク・ベース・データベースを高速化するソリューションです。高速の solidDB フロントエンド・インメモリー・データベースを取り込むことで、アプリケーション負荷の処理を、わずかな応答時間で、バックエンド・データベースで実行できます。これにより、パフォーマンス、速度、および待ち時間が改善されます。

solidDB フロントエンド・データベースに加えて、このソリューションは IBM InfoSphere Change Data Capture (InfoSphere CDC と呼ばれる) テクノロジーを使用して、データ・レプリケーションを行います。フロントエンド・データベースとバックエンド・データベースの両方に、内蔵タイプのデータベース管理システム (DBMS) が存在します。

注: solidDB Universal Cache を solidDB の内部キャッシュ (バッファ・プール) と混同しないでください。内部キャッシュは、solidDB データベース・サーバーで効率的な入出力操作を容易にするために使用されます。

solidDB Universal Cache のアーキテクチャーの概要

以下の図は、標準的構成の solidDB Universal Cache のアーキテクチャーとキー・コンポーネントを示しています。

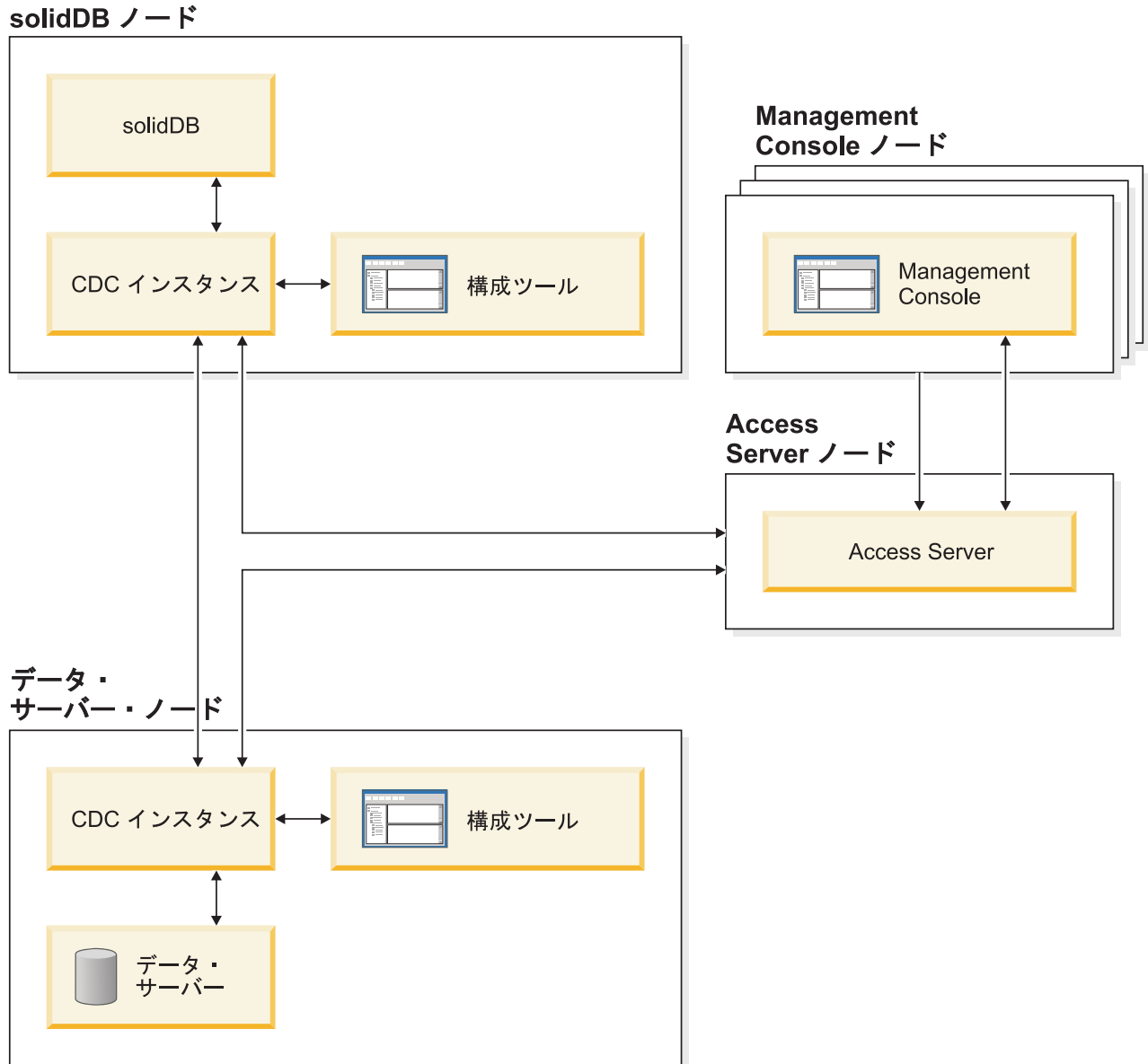


図 1. solidDB Universal Cache アーキテクチャー

コンポーネントの役割と機能を以下に説明します。

solidDB: データの複製先または複製元のフロントエンド・データベース。レプリケーション・モデルに応じて、solidDB をソース・データベースまたはターゲット・データベースのいずれか、あるいは両方のデータベースにすることができます。

データ・サーバー: データの複製先または複製元のバックエンド・データベース。レプリケーション・モードに応じて、バックエンド・データベースをソース・データベースまたはターゲット・データベース、あるいは両方のデータベースにすることができます。

InfoSphere CDC インスタンス — 指定された DBMS 用の InfoSphere CDC エンジンのランタイム・インスタンス。InfoSphere CDC インスタンスをセットアップするには、対応する InfoSphere CDC エージェント・ソフトウェア (InfoSphere CDC for DB2[®]、InfoSphere CDC for solidDB など) が、該当する DBMS が稼働しているのと同じノードにインストールされている必要があります。例外として solidDB では、solidDB が稼働するノードに接続された任意のノードに InfoSphere CDC for solidDB エージェントをインストールし、セットアップできます。

構成ツール: InfoSphere CDC インスタンスの構成と作成に使用する GUI ベースまたはコンソール・ベースのツール。インスタンスの構成中に、InfoSphere CDC システムの他の部分との通信のためのポート番号、およびデータベースに接続するためのログイン情報を入力します。

Access Server: Management Console ユーザーが InfoSphere CDC インスタンスにアクセスし、構成するためのプロセス (一般にサービスまたはデーモンとして稼働)。さまざまなユーザーが、さまざまなインスタンスにアクセスできます。Access Server のインストール中に、Management Console で使用するポート番号および管理者ログイン情報の入力を要求されます。

Management Console: レプリケーションの構成とモニターに使用できる GUI を備えた対話式アプリケーション。各種のサーバー上でのレプリケーションの管理、レプリケーション・パラメーターの指定、クライアント・ワークステーションからのリフレッシュ操作とミラーリング操作の開始が行えます。

アクセス・マネージャー・パースペクティブでデータ・ストアを作成することにより、Management Console の使用を開始します。データ・ストアは、データベースとそれに関連する InfoSphere CDC インスタンスの論理エンティティです。定義されたデータ・ストアを使用することにより、1 つのデータ・ストアから別のデータ・ストアへのデータ・レプリケーションを実現するサブスクリプションをセットアップできます。

データ・ストアには、ソース、ターゲット、およびデュアルという 3 つのタイプがあります。デュアル・データ・ストアは、ソースおよびターゲットの両方としてサブスクリプションに関係させることができます。通常、デュアル・データ・ストアは、solidDB Universal Cache で使用され、対称マッピング・ペアを双方向レプリケーションに定義します。フロントエンド・データ・サーバーでレプリカが読み取り専用の場合は、単一のアップロード・サブスクリプションを定義します。

レプリケーションのセットアップ完了後、ソースとターゲットのサーバー間のアクティブなデータ・レプリケーション・アクティビティに影響を及ぼすことなく、クライアント・ワークステーション上で Management Console を閉じることができます。Management Console にはイベント・ログとモニターも含まれています。イベント・ログでは、生成された InfoSphere CDC イベント・メッセージを確認できます。モニターは、レプリケーション操作と待ち時間を継続的にモニターするために必要なサポートを提供します。レプリケーション構成のコンポーネントを示す図

は、グラフィカル・オブジェクトの直接操作によって構成されます。Management Console のモニターは、データの移動を継続的に分析する必要がある、時間制限の厳しい作業環境で使用するためのものです。

Universal Cache の構成

solidDB Universal Cache は、さまざまな構成をサポートしています。標準的な構成では、データ・サーバーと対応する InfoSphere CDC エンジン (インスタンス) は同じノードに配置され、Management Console は別のノードで実行されます。solidDB 高可用性 (ホット・スタンバイ) を使用する構成では、InfoSphere CDC インスタンスを solidDB サーバーとは異なるノードで実行する必要があります。

以下のセクションでは、よく用いられるデプロイメント・トポロジーについて説明し、異なるマシンで実行する異なる製品コンポーネントを示します。他のインプリメンテーションも可能です。

一般原則

InfoSphere CDC インスタンスのデプロイメントは、使用するデータベース製品によって異なります。

- 一般的に、InfoSphere CDC インスタンスは、InfoSphere CDC レプリケーションに関係している各ノードで作成されます。
- solidDB サーバーと、InfoSphere CDC for solidDB インスタンスは、同じノードに配置する必要はありません。

これは、InfoSphere CDC for solidDB は、ローカルおよびリモートの両方の JDBC 接続を使用して、solidDB データベースからデータを読み取ることも、このデータベースにデータを挿入することもできるためです。

- solidDB 高可用性 (ホット・スタンバイ) を使用する構成では、CDC インスタンスを solidDB サーバーとは異なるノードで実行する必要があります。
- バックエンド・サーバーの InfoSphere CDC インスタンスは、通常、データ・サーバー自身と同じノードにのみ配置できます。solidDB とは異なり、すべてのデータベースが、リモート JDBC 接続を介して機能するログ・リーダー API を提供するわけではありません。

Universal Cache デプロイメントでは、複数の solidDB サーバーを配置できますが、バックエンド・データ・サーバーは 1 つしか配置できません。

solidDB Universal Cache のデフォルト・デプロイメント

最も単純なセットアップは、バックエンド・サーバーと単一の solidDB データベースです。これらは、それぞれ専用のハードウェア上で実行されることを前提としています。各データベースに関連付けられた InfoSphere CDC インスタンスを、データベースと同じノードで実行します。実稼働環境では、InfoSphere CDC Management Console を個別の管理ノードで実行し、いずれかのデータベース・サーバーのパフォーマンスへの影響を制限します。

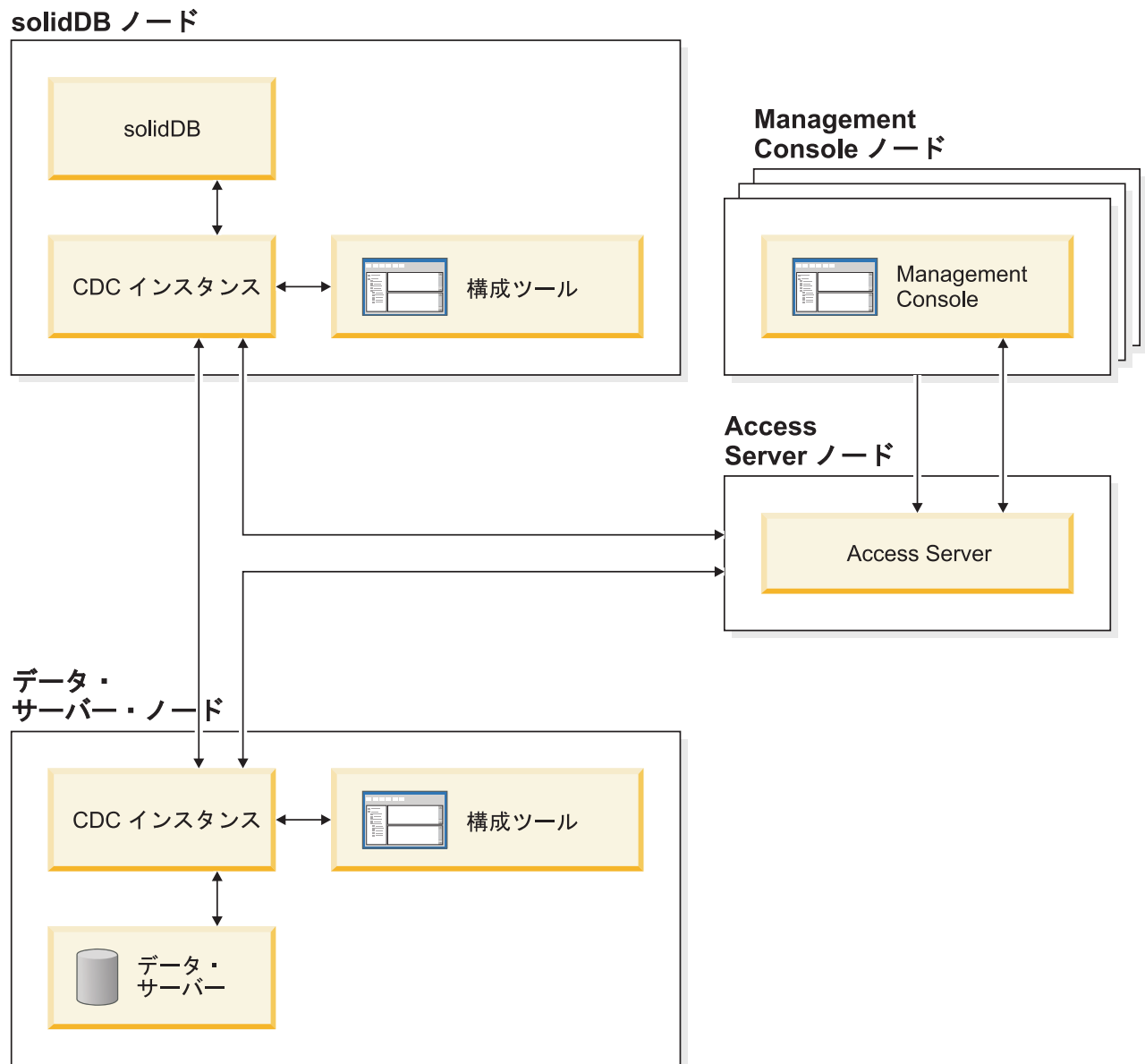


図2. solidDB Universal Cache のデフォルト・デプロイメント

複数の solidDB サーバーを備えた solidDB Universal Cache のデプロイメント

複数の solidDB サーバーを使用することができ、例えば、フロントエンドの複数の solidDB サーバー上でバックエンドをパーティション化する場合などに使用します。ただし、このようなデプロイメントでは、各 solidDB サーバーは自律型であり、他の solidDB サーバーのデータにアクセスすることなく、アプリケーション要求を処理します。

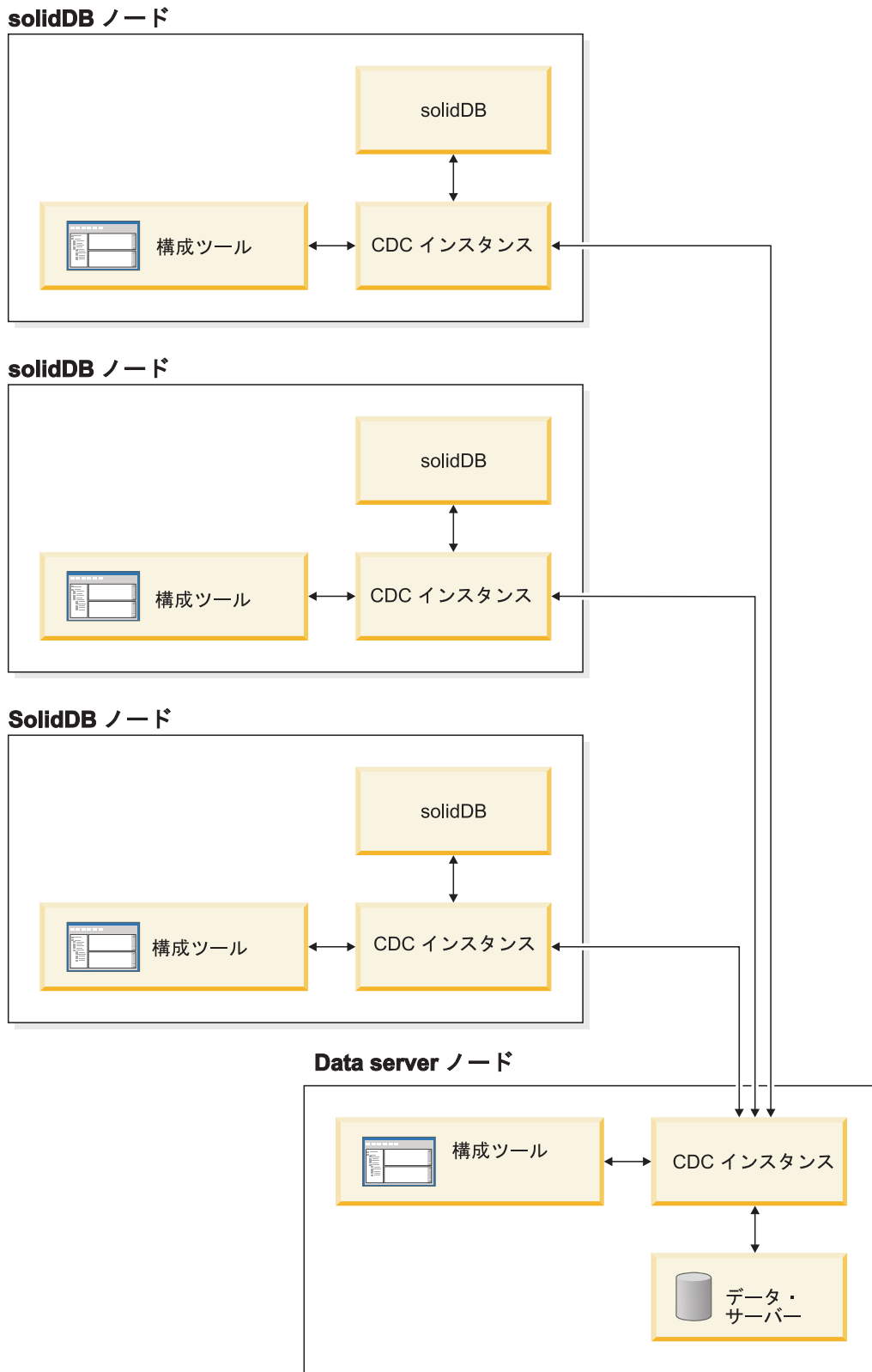


図 3. 複数の solidDB サーバーを備えた solidDB Universal Cache のデプロイメント

solidDB 高可用性を備えた solidDB Universal Cache のデプロイメント

フロントエンド・アプリケーションが中断のないデータ・アクセスを必要としている場合は、solidDB HotStandby テクノロジーを使用して、高可用性を提供できます。複数の持続性セマンティクスや待機読み取りなど、すべての HotStandby の側面は、solidDB Universal Cache の高可用性セットアップに適用できます。この場合、すべての InfoSphere CDC インスタンスはバックエンド・ノードで実行され、solidDB HotStandby ペアへの接続がリモート側で確立されます。Management Console は別のノードで実行されます。

solidDB 高可用性を使用することにより、フロントエンド層の単一障害から Universal Cache の操作が保護されます。

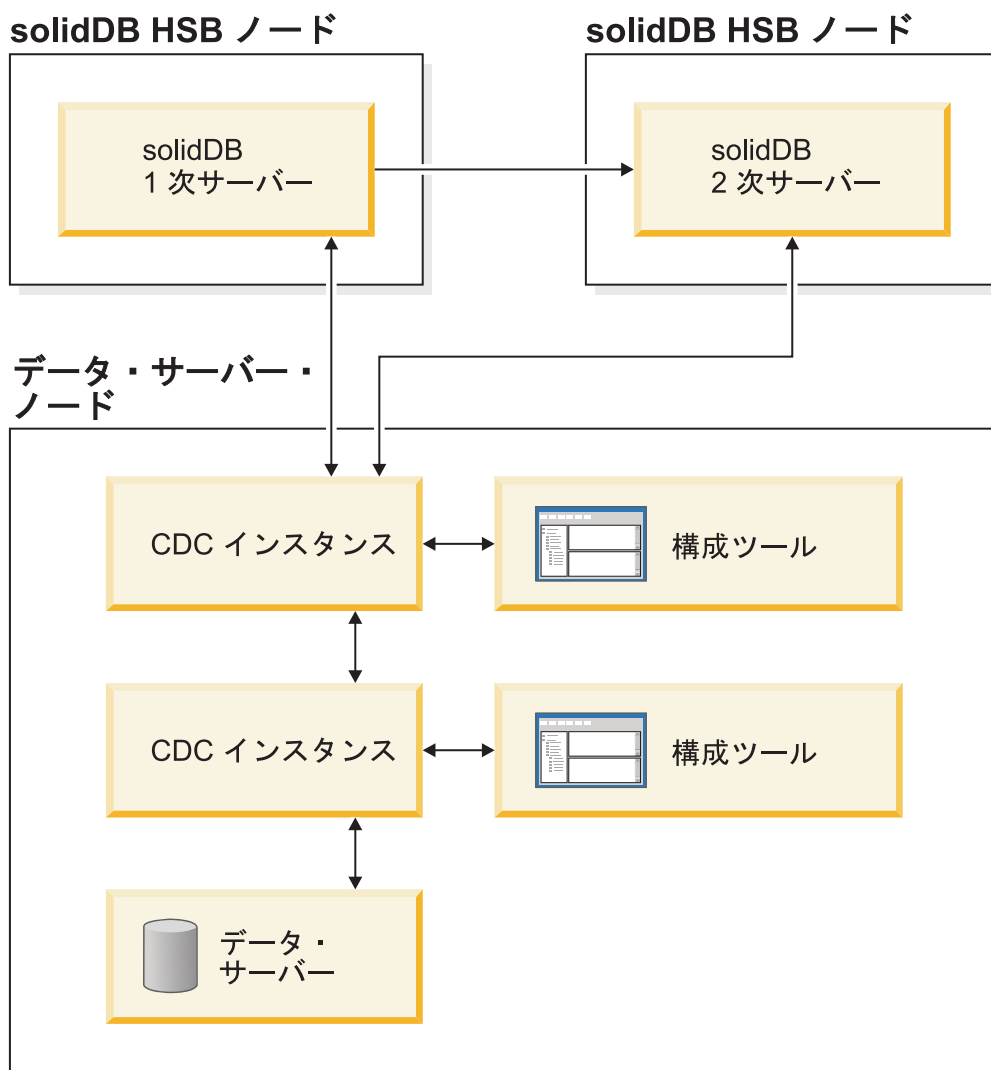


図 4. solidDB 高可用性を備えた solidDB Universal Cache のデプロイメント

solidDB HA を備えたデプロイメントを設計する場合、環境内での障害シナリオにどのように対処するかを計画してください。solidDB Universal Cache システムの障害に対処する方法については、81 ページの『8 章 solidDB Universal Cache の障害シ

ナリオ』セクションを参照してください。

バックエンド高可用性を備えた solidDB Universal Cache のデプロイメント

アプリケーションは、バックエンド・データ・サーバーにも追加の高可用性または災害時回復要件を設定することができます。異なるデータ・サーバーが異なるテクノロジーを使用して、このような機能をインプリメントしています。バックエンド・データ・サーバーは、依然として単一の論理エンティティですが、データや実行中のプロセスは複数のノードに分散することができます。

この場合は、InfoSphere CDC インスタンスは、1 次バックエンド・データ・サーバー・ノードで実行されます。ただし、バックエンドに障害が発生すると、バックエンド・データ・サーバーの完全なフェイルオーバーの一環として、これらの InfoSphere CDC インスタンスを別のマシンで再開する必要があります。関連インスタンスを再接続するため、サブスクリプションも再構成する必要があります。

機能

InfoSphere CDC テクノロジーをデプロイする構成では、以下の solidDB 機能が使用可能です。

- **共有メモリー・アクセス (SMA) およびリンク・ライブラリー・アクセス (LLA)**

アプリケーションを作成して、共有メモリー・アクセス・ドライバー・ライブラリーまたはリンク・ライブラリー・アクセス・ライブラリーにリンクすると、インメモリー・データベースのパフォーマンスと反応性をフル活用することができます。

詳しくは、*IBM solidDB 共有メモリー・アクセスおよびリンク・ライブラリー・アクセス・ユーザー・ガイド*を参照してください。

- **SQL パススルー**

solidDB Universal Cache で、SQL パススルーを使用すると、アプリケーションからフロントエンドとバックエンドの両方のデータ・サーバーのデータに単一接続でアクセスできます。例えば、SQL パススルーを有効にして、solidDB フロントエンドでは実行できない SQL ステートメントをバックエンドに渡すように設定することができます。

- **アプリケーション主導型データ・エージング**

solidDB Universal Cache では、データ・エージング機能により、アプリケーションで solidDB フロントエンド・データベースから古くなったデータを削除する一方、バックエンドでそのデータを保持することができます。データ・エージングを使用すると、solidDB フロントエンドのメイン・メモリーの使用量を制御することができます。つまり、solidDB フロントエンドのデータの一部が不要になった場合、および、アプリケーションで使用されなくなった場合に、そのデータを削除できます。不要なデータを削除すると、フロントエンドのメイン・メモリー・リソースが解放されます。

- **高速リフレッシュ**

高速リフレッシュ機能により、バックエンド・データ・サーバーから solidDB フロントエンドへの大容量のデータの複製にかかる時間が削減されます。大部分のパフォーマンス向上は、簡単なセットアップで実現されます。例えば、表あたり行あたりのデータ量、コード・ページ変換、および列マッピングなどの要因が、高速リフレッシュ機能のパフォーマンスに影響を及ぼします。

高速リフレッシュは、以下の InfoSphere CDC 機能をサポートしません。

- 競合検出
- 要約
- 行の統合
- アダプティブ・アプライ
- ユーザー出口

• solidDB の高可用性サポート

solidDB 高可用性機能 (solidDB サーバーでのフェイルオーバーを含む) はサポートされます。詳しくは、「*IBM solidDB 高可用性ユーザー・ガイド*」を参照してください。

• スロットル

レプリケーションによって、solidDB サーバーでの持続的な負荷に対応できない場合は、処理が減速 (低速化) します。アプリケーションの観点からいうと、これは応答時間が長くなることを意味しています。レプリケーション・トラフィックがバッファーに入れられ、負荷のバーストに対応できるようにします。対応するインメモリー・バッファーのサイズは、構成パラメーター **LogReader.MaxSpace** を使用して制御します。

• オフライン操作とログのオーバーフロー

レプリケーションが停止、あるいはレプリケーションに失敗しても、solidDB サーバーは負荷の処理を続行し、後で転送するためにデータを蓄積することができます。蓄積するデータの限度は、構成パラメーター **LogReader.MaxLogSize** を使用して設定します。蓄積したデータの量が **LogReader.MaxLogSize** パラメーターの値を超えると、ログのオーバーフローが発生し、それ以降はレプリケーション・キャッチアップが実行できない状態になります。その場合は、サブスクリプションのリフレッシュが必要です。

• ログ・リーダーの診断

solidDB のキャプチャー・プロセス (solidDB をソースとして) およびアプライ・プロセス (solidDB をターゲットとして) の操作をモニターするために、いくつかのランタイム・カウンターが使用可能です。セクション 10 ページの『データベースの制限事項』の LOBS 関連の制限事項も参照してください。

• データ型のサポート

レプリケーションでは、solidDB のすべてのデータ型がサポートされています。詳しくは、104 ページの『サポートされているデータ型』のセクションを参照してください。

制限事項

データベースの制限事項

ソースおよびターゲットのデータ・サーバーとして、solidDB には以下の制限事項が適用されます。

- **参照整合性** (solidDB をソースおよびターゲットとして)

連続ミラーリングの場合、ソースとターゲットの両方に対して参照整合性制約 (外部キー) を使用できます。必須要件は、参照整合性関連がサブスクリプション内に限定されていること、つまり外部キーがサブスクリプション外の表を指さないことです。このルールに違反すると、ミラーリング中にターゲットで参照整合性エラーが発生し、レプリケーション・サブスクリプションが終了することがあります。

参照整合性は、表の自動作成ではサポートされていません。予備手段については、11 ページの『InfoSphere CDC の制限事項』のセクションを参照してください。

- **データ型のサポート**

- ディスク・ベース表の LOB はサポートされない (solidDB をソースとして)

使用可能なサイズ制限内のインメモリー表で維持しているすべての LOB は miniLOB と呼ばれ、許可されます。サイズ制限は、行サイズとブロック・サイズによって異なります。1 行に 1 つの LOB と想定すると、サイズ制限はブロック・サイズに近くなります。ブロック・サイズが 32 KB に設定されている場合は、miniLOB の実際のサイズ制限は約 30 KB になります。

solidDB のディスク・ベース表の大きなサイズ (最大 2 GB) の LOB (maxiLOB) は、ソースではサポートされていません。ログ・リーダー・パーティションの一部であるディスク・ベース表に maxiLOB を書き込もうとすると失敗し、アプリケーションにエラーが返されるという方法で、この制限事項は実施されます。

- LOB サポートの制限 (solidDB をターゲットとして)

LOB がインメモリー表に書き込まれ、miniLOB サイズ制限を超えた場合は、エラーが返され、サブスクリプションのレプリケーションが終了します。

- **TRUNCATE** (solidDB をソースとして)

サブスクリプションの一部となっている表には、TRUNCATE TABLE ステートメントは許可されません。このルールに違反すると、アプリケーションにエラーが返されます。

- **主キー制約** (solidDB をソースとして)

主キーは推奨されていますが、必須ではありません。表に主キーが定義されていない場合、主キーが定義されている場合と比較して、挿入と更新の実行効率が低下します。主キーの更新は、以下のように制限されます。

- 単一の列に主キーが定義されている場合、主キーの複数行の更新は許可されません。

- 複数列の主キーの場合、主キーの一部にのみ影響する場合に限って、複数行の更新が許可されます。

上記のルールの内いずれかに違反すると、エラーが発生し、サブスクリプションのレプリケーション (ミラーリング) が終了します。

- **トランジエント表とテンポラリー表 (solidDB をソースとして)**

solidDB をソースとして使用すると、トランジエント表およびテンポラリー表をサブスクリプションの一部とすることはできません。

- **UNIQUE 列の複数の NULL (solidDB をターゲットとして)**

solidDB では、UNIQUE として定義された列は、NULL インスタンスを 1 つだけ含むことができます。さらに NULL の挿入の伝搬を試行すると、UNIQUE 制約違反が発生し、サブスクリプションのレプリケーション (ミラーリング) が終了します。

- **複数の solidDB サーバーを使用したデータとワークロードのパーティション化**

複数の solidDB サーバーを使用して、データとワークロードをパーティション化できます。例えば、フロントエンドの複数の solidDB サーバー上でバックエンド・データを分散 (パーティション化) することができます。ただし、各 solidDB サーバーは自律型であり、他の solidDB サーバー (パーティション) のデータにアクセスすることなく、保持しているパーティション内でのみアプリケーション要求を処理します。

参照整合性制約も適用されるため、パーティションには、パーティションの外側との参照を行う表を収めることはできません。

InfoSphere CDC の制限事項

他のデータ・サーバー用に InfoSphere CDC コンポーネントで使用可能な以下の機能は、InfoSphere CDC for solidDB ではサポートされません。

- **リフレッシュ用の高速ロード**

InfoSphere CDC for solidDB は、リフレッシュ用の高速ロード機能をサポートしていません。

- **ターゲット表の自動作成**

ミラーリングされる表が参照整合性制約に関連付けられているときは、新しいサブスクリプションの定義時に、ターゲット表の作成を自動的に行うオプション (「**Create new target tables**」) を使用することはできません。代わりに、オプション「**Map to existing tables**」を使用します。このルールに違反すると、サブスクリプションは作成されません。

この制限は、他の DBMS 製品も含め、すべての構成に適用されます。

- **行フィルター**

行フィルター (水平パーティショニング) が完全に機能するのは、主キーがソース表で定義されている場合だけです。

- **solidDB がソース・データ・ストアである場合の、表のドロップと再作成**

solidDB がソース・データ・ストアであるサブスクリプションの表をドロップして再作成する必要がある場合は、表マッピングを再構成する必要があります。詳しくは、31 ページの『solidDB ソース表のドロップおよび再作成』を参照してください。

セキュリティと認証

レプリケーションに関係するデータベースは、通常の名/パスワードのメカニズムによって保護されます。

- InfoSphere CDC インスタンスによるデータベース・アクセスに使用するユーザー名には、データとメタデータのアクセスと変更に関して、必要なすべての資格情報を与えておく必要があります。データベース管理者のユーザー名を使用することを推奨します。
- InfoSphere CDC システムは、InfoSphere CDC インスタンスとそれぞれのパーシスタント・リポジトリに、ユーザー名とパスワードを格納します。格納する認証データにはスクランブルがかけられます。つまり、弱い暗号化方式で暗号化されます。
- InfoSphere CDC インスタンスとデータベース間のトラフィックは、どのような場合においても暗号化されます。暗号化は、製品固有の JDBC ドライバーで提供されるか、使用可能です。
- InfoSphere CDC インスタンス間のトラフィックは暗号化されません。

2 solidDB Universal Cache のインストールと構成

solidDB Universal Cache をデプロイするには、solidDB、バックエンド・データ・サーバー、および関連の InfoSphere CDC コンポーネントをインストールし、構成する必要があります。

solidDB Universal Cache のインストールおよび構成手順の概要

このセクションでは、solidDB Universal Cache のインストールおよび構成手順の概要を説明します。

注:

- このインストールおよび構成の説明では、使用する構成に 1 つの solidDB サーバーだけが含まれていることを想定しています。構成内に追加の solidDB サーバーがある場合は、これらの手順を繰り返してください。
- 各コンポーネントは、以下に述べる順にインストールしてください。これは、各コンポーネントのインストールと構成の要件を確実に満たすためです。

1. **solidDB Universal Cache** コンポーネントのインストール・イメージを見つけます。

さまざまなプラットフォーム用のインストール・イメージのリストについては、15 ページの『solidDB Universal Cache のインストール・パッケージ』で詳細を参照してください。

2. **solidDB Universal Cache** をインストールするときに必要になる、以下のすべての **Documentation** パッケージにアクセスできることを確認します。

- オンライン・インフォメーション・センター、または *IBM solidDB 6.5 Documentation* パッケージ
- *InfoSphere Change Data Capture Documentation* パッケージ

Documentation パッケージとその配信方法の詳細については、「*IBM solidDB スタートアップ・ガイド*」の『*Documentation* パッケージ』のセクションを参照してください。

3. **solidDB Universal Cache** コンポーネントをインストールするすべてのノードに対して、システム管理者 (またはそれと同等の) アクセス権限を持っていることを確認します。

ヒント: solidDB Universal Cache をセットアップする間、さまざまなコンポーネントが互いに通信できるよう、ユーザー・アカウント、データベース、およびネットワーク接続識別データを作成する (または既存のものを使用する) 必要があります。主要な識別データの要約は、16 ページの『solidDB Universal Cache 用のユーザー・アカウントおよびデータベース接続データ』に記載されています。

4. **IBM solidDB** サーバーをインストールし、構成します。

詳しくは、19 ページの『solidDB Universal Cache 用の solidDB サーバーのインストールと構成』を参照してください。

結果: 正常に機能する solidDB インストール済み環境と solidDB データベースが得られます。このデータベースには、バックエンド・データ・サーバーとの間で複製するデータが入っています。

5. **InfoSphere CDC for IBM solidDB をインストールし、構成します。**

詳しくは、21 ページの『InfoSphere CDC for solidDB のインストールと構成』を参照してください。

結果: 正常に機能するインストール済み環境と、少なくとも 1 つの InfoSphere CDC インスタンスが作成されています。このインスタンスは、solidDB データベースに接続しています。

6. **バックエンド・データ・サーバーをインストールし、構成します。**

詳しくは、22 ページの『バックエンド・データ・サーバーのインストールおよび構成』を参照してください。

結果: 正常に機能するバックエンド・サーバーのインストール済み環境とデータベースが得られます。このデータベースには、solidDB との間で複製するデータが入っています。

7. **バックエンド・データ・サーバー用の InfoSphere CDC をインストールし、構成します。**

詳しくは、22 ページの『バックエンド・データ・サーバー用の InfoSphere CDC のインストールと構成』を参照してください。

結果: 正常に機能するインストール済み環境と、少なくとも 1 つの InfoSphere CDC インスタンスが作成されています。このインスタンスは、バックエンド・データベースに接続しています。

8. **InfoSphere CDC Access Server をインストールし、構成します。**

詳しくは、23 ページの『InfoSphere CDC Access Server のインストールと構成』を参照してください。

結果: 正常に機能するインストール済み環境が得られ、Management Console にログインするためのシステム管理者アカウントが作成されています。

9. **InfoSphere CDC Management Console をインストールし、構成します。**

詳しくは、24 ページの『InfoSphere CDC Management Console のインストールと構成』を参照してください。

結果: 正常に機能するインストール済み環境が得られ、システム管理者アカウントを使用して Management Console にログインできます。

10. **レプリケーション・サブスクリプションをセットアップします。**

詳しくは、24 ページの『レプリケーション・サブスクリプションのセットアップ』を参照してください。

結果: solidDB とバックエンド・データ・サーバーとの間にレプリケーション・サブスクリプションが作成されています。

solidDB Universal Cache のインストール・パッケージ

solidDB Universal Cache の各コンポーネントは、別々のインストール・イメージとして配信されます。

表3. solidDB のインストール・イメージ

コンポーネント名	インストール・パッケージ	内容
IBM solidDB 6.5	Linux[®] および UNIX: solidDB-6.5-<platform>.bin Windows: solidDB-6.5-<platform>.exe	「IBM solidDB スタートアップ・ガイド」の『solidDB サーバー・パッケージ』のセクションで、内容の説明を参照してください。

表4. InfoSphere CDC のインストール・イメージ

InfoSphere CDC コンポーネント名	インストール・パッケージ	内容
InfoSphere Change Data CapturesolidDB (InfoSphere CDC for solidDB)	Linux および UNIX: setup-<platform>-solid.bin 以下に例を示します。 setup-linux-x86-solid.bin Windows: setup-x86-solid.exe	<ul style="list-style-type: none"> 構成ツールおよび InfoSphere CDC for solidDB インスタンスのためのソフトウェア solidDB JDBC ドライバー (/lib ディレクトリー内の SolidDriver2.0.jar) ツール、ユーティリティ、およびサンプル (/samples ディレクトリー) <ul style="list-style-type: none"> 最も一般的な InfoSphere CDC タスクのスクリプトを記述するための自動化ツール、ユーティリティ、およびサンプル (ucutils、ucpassthrough、および uchsmonitor ディレクトリー) Java™ ユーザー出口および SQL スクリプト用の一般的な InfoSphere CDC サンプル InfoSphere CDC API 文書 (/docs ディレクトリー)
バックエンド・データ・サーバー用 InfoSphere Change Data Capture 注: solidDB Universal Cache にのみ含まれる	Linux および UNIX: setup-<platform>-<back-end_dataserver>.bin 例えば、以下のように指定します。 setup-aix-power-udb.bin Windows: setup-x86-<back-end_dataserver>.exe	<ul style="list-style-type: none"> 構成ツール用ソフトウェアおよびバックエンド・データ・サーバー用の InfoSphere CDC インスタンス PDF フォーマットの InfoSphere Change Data Capture のエンド・ユーザー向け資料 (/docs ディレクトリー) サンプルの Java ユーザー出口および SQL スクリプト (/samples ディレクトリー) InfoSphere CDC API 文書 (/docs ディレクトリー)

表 4. InfoSphere CDC のインストール・イメージ (続き)

InfoSphere CDC コンポーネント名	インストール・パッケージ	内容
InfoSphere Change Data Capture Access Server	<p>Linux および UNIX:</p> <p>setup-<platform>-accessserver.bin</p> <p>例えば、以下のように指定します。</p> <p>setup-solaris-sparc-accessserver.bin</p> <p>Windows:</p> <p>setup-win-x86-accessserver.exe</p>	<ul style="list-style-type: none"> レプリケーション環境へのアクセスを制御するためのソフトウェア
InfoSphere Change Data Capture Management Console	<p>Linux および UNIX:</p> <p>適用外 - Windows の場合のみ</p> <p>Windows:</p> <p>setup-x86-dmclient.exe</p>	<ul style="list-style-type: none"> solidDB Universal Cache および InfoSphere CDC Replication 構成セットアップで、InfoSphere CDC ユーザー・アクセスとレプリケーション・サブスクリプションを構成およびモニターするためのソフトウェア PDF フォーマットの「InfoSphere Change Data Capture Management Console 管理ガイド」(/documentation ディレクトリー) オンライン・ヘルプ (Management Console ユーザー・インターフェースで「Help」メニューからアクセス可能) IBM Java SDK and Runtime Environment Guides (/docs ディレクトリー)

solidDB Universal Cache 用のユーザー・アカウントおよびデータベース接続データ

solidDB Universal Cache をインストールおよび構成するとき、さまざまなコンポーネントが互いに通信できるよう、ユーザー・アカウントとデータベースおよび接続情報を作成するか、既存のそれらを使用する必要があります。このセクションの表に、solidDB Universal Cache をセットアップするときに作成されるユーザー・アカウントおよびデータベース接続データが要約されています。

デフォルト値がある場合は、それも示してあります。

solidDB

表 5. solidDB のユーザー・アカウントおよびネットワーク接続データ

solidDB	値の例 (可能な場合はデフォルト)	使用法
サーバー接続データ (サーバー名およびポート番号)	tcp 1964	<ul style="list-style-type: none"> solid.ini 構成ファイル内で定義される InfoSphere CDC for solidDB インスタンスを作成する場合に必要
データベース・ログイン・データ	<p>ユーザー名: soliduser</p> <p>パスワード: admin</p>	<ul style="list-style-type: none"> solidDB データベースの作成時に定義される InfoSphere CDC for solidDB インスタンスを作成する場合に必要

表 5. solidDB のユーザー・アカウントおよびネットワーク接続データ (続き)

solidDB	値の例 (可能な場合はデフォルト)	使用法
システム・カタログ名	dba	<ul style="list-style-type: none"> solidDB データベースの作成時に定義される <p>データベース・オブジェクト階層の solidDB 構文は、以下のとおりです。</p> <p>catalog_name.schema_name.database_object</p> <p>詳細については、「IBM solidDB SQL ガイド」の『データベース・オブジェクトの管理』のセクションを参照してください。</p> <p>重要: サブスクリプションには、システム・カタログに含まれている表だけを含めることができます。</p>
スキーマ名	soliduser	<ul style="list-style-type: none"> デフォルトのスキーマ名はユーザー名です。CREATE SCHEMA ステートメントを使用して、新しいスキーマを作成できます。 InfoSphere CDC for solidDB インスタンスを作成する場合に必要な

InfoSphere CDC for solidDB

表 6. InfoSphere CDC for solidDB のユーザー・アカウントおよびネットワーク接続データ

InfoSphere CDC for solidDB	値の例 (可能な場合はデフォルト)	使用法
インスタンス名	solidDB_source	<ul style="list-style-type: none"> InfoSphere CDC for solidDB の作成時に定義される dm コマンドでインスタンスを管理するときに使用される
サーバー・ポート	11101 (デフォルト)	<ul style="list-style-type: none"> InfoSphere CDC for solidDB の作成時に定義される Management Console/アクセス・マネージャーからインスタンスへの接続時に必要
Windows Service ユーザー・アカウント		<ul style="list-style-type: none"> InfoSphere CDC for solidDB の作成時に定義される InfoSphere CDC サービスを管理するとき (例えば、インスタンスの開始) に必要
データベース・ログイン・データ	ユーザー名: soliduser パスワード: admin メタデータ・スキーマ: SOLIDUSER	<ul style="list-style-type: none"> solidDB データベースへのログイン・データと、InfoSphere CDC メタデータ表に使用するスキーマ名を指定する
サーバー接続データ	solidnode 1964	<ul style="list-style-type: none"> solidDB サーバーへの接続データを指定する ホスト名は、ネットワーク名または IP アドレスにすることができます。InfoSphere CDC for solidDB が、solidDB と同じノードにある場合、ホスト名も localhost にすることができます。 ポート番号は、solidDB サーバーが listen するポート (solid.ini 構成ファイル内で定義) でなければならない

バックエンド・データ・サーバー用の InfoSphere CDC

表7. バックエンド・データ・サーバー用の InfoSphere CDC のユーザー・アカウントおよびネットワーク接続データ

バックエンド・データ・サーバー用の InfoSphere CDC	値の例 (可能な場合はデフォルト)	使用法
インスタンス名	backend_target	<ul style="list-style-type: none"> InfoSphere CDC for solidDB の作成時に定義される dm コマンドでインスタンスを管理するときに使用される
サーバー・ポート	10901 (デフォルトはバックエンド・データ・サーバーによって異なる)	<ul style="list-style-type: none"> InfoSphere CDC for solidDB の作成時に定義される Management Console/アクセス・マネージャーからインスタンスへの接続時に必要
Windows Service ユーザー・アカウント		<ul style="list-style-type: none"> InfoSphere CDC for solidDB の作成時に定義される InfoSphere CDC サービスを管理するとき (例えば、インスタンスの開始) に必要
データベース・ログイン・データ	バックエンド・データ・サーバーによって異なる	<ul style="list-style-type: none"> 使用するバックエンド・データベースのログイン・データおよび接続設定を指定する <p>詳しくは、使用するバックエンド・データ・サーバー用の「InfoSphere Change Data Capture のエンド・ユーザー向け資料」の『インストールの前に: 必要なデータベース、ユーザー・アカウント、およびスキーマ』のセクションを参照してください。</p>

Access Server

表8. Access Server のユーザー・アカウントおよびネットワーク接続データ

Access Server	値の例 (可能な場合はデフォルト)	使用法
ポート番号	10101 (デフォルト)	<ul style="list-style-type: none"> Access Server のインストール時 (Windows) または構成時 (Linux および UNIX) に定義される Management Console にログインするときに必要
ログイン・データ (システム管理者)	ユーザー名: Admin (デフォルト) パスワード: solid123	<ul style="list-style-type: none"> Access Server のインストール時 (Windows) または構成時 (Linux および UNIX) に定義される Access Server システム管理者のユーザー名を指定する Management Console にログインするときに必要

Management Console

表9. Management Console のユーザー・アカウントおよびネットワーク接続データ

Management Console	値の例 (可能な場合はデフォルト)	使用法
ログイン・データ (システム管理者)	ユーザー名: Admin (デフォルト) パスワード: solid123	<ul style="list-style-type: none"> Access Server のインストール時 (Windows) または構成時 (Linux および UNIX) に定義される Access Server システム管理者のユーザー名を指定する

表9. Management Console のユーザー・アカウントおよびネットワーク接続データ (続き)

Management Console	値の例 (可能な場合はデフォルト)	使用法
サーバー名		<ul style="list-style-type: none"> Access Server を実行するワークステーションのホスト名 (システム名) または完全な IP アドレスを指定する Management Console を Access Server に接続するために使用される
ポート番号	10101 (デフォルト)	<ul style="list-style-type: none"> Access Server のインストール時 (Windows) または構成時 (Linux および UNIX) に定義される

solidDB Universal Cache 用の solidDB サーバーのインストールと構成

solidDB Universal Cache 用の solidDB サーバーのインストール

手順

1. IBM solidDB をインストールします。

手順については、「*IBM solidDB スタートアップ・ガイド*」の『*solidDB のインストールのインストール*』のセクションを参照してください。

2. ご使用の solidDB を確認し、solidDB の基本的な操作に習熟しておいてください。

詳細については、「*IBM solidDB スタートアップ・ガイド*」の『*solidDB のインストールの検証*』のセクション、および「*IBM solidDB 管理者ガイド*」を参照してください。

次のタスク

『solidDB for solidDB Universal Cache の構成』

solidDB for solidDB Universal Cache の構成

InfoSphere CDC テクノロジーで solidDB を使用するには、InfoSphere CDC for solidDB が solidDB データベースに接続して、データを複製できるように、構成の設定を変更する必要があります。

始める前に

このセクションでは、読者が solidDB の管理に習熟しており、例えば、「*IBM solidDB 管理者ガイド*」の『*solidDB の管理*』、および『*solidDB の構成*』のセクションを読了していることを想定しています。

手順

1. データベース環境をセットアップするために、作業ディレクトリー、solidDB データベース、およびユーザー・アカウントをセットアップします。

手順については、「*IBM solidDB 管理者ガイド*」の『*データベースの新規作成*』を参照してください。

ヒント:

solidDB をインストールした後、solidDB インストール・ディレクトリーに以下のディレクトリーがあります。

```
<installation directory>
  bin¥
  ..
  eval_kit¥
    standalone¥
    cdc¥
  ..
  samples
  ..
```

solidDB インストール・ディレクトリー内の eval_kit/cdc ディレクトリーを作業ディレクトリーとして使用できます。ここでは、solidDB を solidDB Universal Cache または InfoSphere CDC レプリケーションと一緒に使用するためのサンプルの solid.ini ファイルが入っています。

2. solid.ini 構成ファイルの LogReader セクションにある構成パラメーターを変更することにより、ログ・リーダーを構成します。

a. LogReaderEnabled 構成パラメーターを「yes」に設定します。

```
[LogReader]
LogReaderEnabled=yes
```

これは、InfoSphere CDC レプリケーションで solidDB をソース・データベースとして使用するために必要です。ファクトリー値は「no」です。

b. MaxLogSize 構成パラメーターで、トランザクション・ログの保存スペースのサイズを設定します。

```
[LogReader]
MaxLogSize=<MB>
```

MaxLogSize パラメーターは、キャッチアップの実行に使用可能なログ・ファイルの量 (サイズ) を設定します。ログ・ファイルの最大サイズは、使用可能なディスク・スペースおよびキャッチアップが必要になるまでのダウン時間によって異なります。ファクトリー値は 10240 (10 GB) です。

ログ・リーダーが有効な場合、指定されたログ・ファイル保存スペースは、常にいっぱいまで使用されます。バックアップが実行されていない場合、またはパラメーター **CheckpointDeleteLog** が「Yes」に設定されていない場合には、ログ・ファイルはより大きなスペースを占有することがあります。

c. MaxSpace 構成パラメーターを指定して、ログ・レコードのインメモリー・バッファー・サイズを設定します。

```
[LogReader]
MaxSpace=<ログ・レコード数>
```

MaxSpace パラメーターは、スロットルで使用されるインメモリー・ログ・リーダー・バッファーのサイズを (ログ・レコードの数で) 設定します。ログ・レコードの最大数は、予想される負荷バースト・サイズによって異なります。ファクトリー値は、100000 ログ・レコードです。

ログ・レコードのサイズは、(バイナリーの) 行サイズに追加メタデータ・オーバーヘッドの数バイトを加算したものになります。バッファーがいっぱい

になると、スループット・スロットルが適用され、ログ・リーダー・バッファに空きができるまで、操作がブロックされます。

- 必要に応じて、パフォーマンスとデータベース・セットアップに関連するその他の構成パラメーターを変更します。

- **Logging.DurabilityLevel**

デフォルトでは、solidDB サーバーの持続性レベルはリラックス (**Logging.DurabilityLevel=1**) に設定されています。この場合、solidDB サーバーに予期しない障害が発生すると、最新のトランザクションが失われる状態になる可能性があります。

これを防止するために、以下の solid.ini ファイルの設定で、持続性レベルをストリクトに設定します。

```
[Logging]
DurabilityLevel=3
```

注: ストリクト持続性設定では、リラックス持続性と比較して、パフォーマンスで不利な条件が生じます。solidDB HA (HotStandby) 構成が 2-Safe レプリケーション・プロトコル (デフォルト) で適用されている場合、リラックス持続性はデータ損失のリスクなしに使用できます。

- **General.DefaultStoreIsMemory**

デフォルトでは、solidDB 表のストレージ・タイプはインメモリー表 (**General.DefaultStoreIsMemory=yes**) に設定されています。

- **Sql.IsolationLevel**

デフォルトでは、solidDB 分離レベルは、READ COMMITTED (**Sql.IsolationLevel=1**) に設定されています。

InfoSphere CDC for solidDB のインストールと構成

InfoSphere CDC for solidDB をインストールするには、インストール・ウィザードの手順に従います。インストール後、InfoSphere CDC 構成ツールを使用して、InfoSphere CDC for solidDB インスタンスを構成します。

始める前に

以下を確認してください。

- solidDB サーバーが稼働している。
- solidDB データベースの作成が完了している。
- solidDB データベースのユーザー名とパスワードが分かっている。
- solidDB サーバーが listen するネットワーク・アドレスとポート番号が分かっている。
- InfoSphere CDC for solidDB によるメタデータ表の作成用のスキーマを新規に作成してあるか、既存のスキーマを選択してある。

手順

1. InfoSphere CDC for solidDB をインストールします。

InfoSphere CDC for solidDB をインストールする方法の段階的な手順は、93 ページの『InfoSphere CDC のインストール』のセクションに記載されています。インストールの終わりに、InfoSphere CDC 構成ツールの起動を選択して、InfoSphere CDC for solidDB インスタンスを構成します。

2. 構成ツールを使用して、InfoSphere CDC for solidDB の新しいインスタンスを作成します。

新しい InfoSphere CDC for solidDB インスタンスを作成する方法の段階的な手順は、94 ページの『InfoSphere CDC の構成 (Windows)』および 98 ページの『InfoSphere CDC の構成 (UNIX および Linux)』のセクションに記載されています。

注: 構成に solidDB High Availability をデプロイしてある場合は、1 次および 2 次 solidDB サーバーのホスト・アドレスとポート番号を定義する 1 つの InfoSphere CDC インスタンスを作成する必要があります。

次のタスク

『バックエンド・データ・サーバーのインストールおよび構成』に進みます。

バックエンド・データ・サーバーのインストールおよび構成

バックエンド・データ・サーバーに付属する説明書に従って、バックエンド・データ・サーバーのインストールと構成を行います。その際、使用するバックエンド・データ・サーバー用の InfoSphere CDC の「InfoSphere Change Data Capture のエンド・ユーザー向け資料」に特別な要件が設定されている場合は、それに注意してください。

手順

1. InfoSphere CDC に使用するバックエンド・データ・サーバーのインストールの前提条件を確認します。

インストール要件については、バックエンド・データ・サーバー用の「InfoSphere Change Data Capture のエンド・ユーザー向け資料」の『インストールの前に』のセクションに説明があります。

2. バックエンド・データ・サーバーを、製品に付属する説明書に従ってインストールします。

次のタスク

『バックエンド・データ・サーバー用の InfoSphere CDC のインストールと構成』

バックエンド・データ・サーバー用の InfoSphere CDC のインストールと構成

バックエンド・データ・サーバー用の InfoSphere CDC をインストールするには、インストール・ウィザードの手順に従います。インストール後、InfoSphere CDC 構成ツールを使用して、InfoSphere CDC インスタンスを構成します。

始める前に

- バックエンド・データ・サーバーが稼働していることを確認します。
- バックエンド・データベースを作成しておきます。
- バックエンド・データベース用のユーザー名とパスワードを確認します。
- バックエンド・データ・サーバーが listen するネットワーク・アドレスとポート番号を確認します。
- InfoSphere CDC がメタデータ表を作成する際に使用するスキーマを新規に作成しておくか、既存のスキーマを選択しておきます。

手順

1. インストールの前提条件を確認します。

インストール要件については、バックエンド・データ・サーバー用の「InfoSphere Change Data Capture のエンド・ユーザー向け資料」の『インストールの前に』のセクションに説明があります。

2. バックエンド・データ・サーバー用 InfoSphere CDC をインストールします。

詳しくは、バックエンド・データ・サーバー用の「InfoSphere Change Data Capture のエンド・ユーザー向け資料」の『InfoSphere CDC のインストール』のセクションを参照してください。インストールの終わりに、InfoSphere CDC 構成ツールの起動を選択して、InfoSphere CDC インスタンスを構成します。

3. 構成ツールを使用して、バックエンド・データ・サーバー用の InfoSphere CDC の新しいインスタンスを作成します。

詳しくは、バックエンド・データ・サーバー用の「InfoSphere Change Data Capture のエンド・ユーザー向け資料」の『InfoSphere CDC の構成』のセクションを参照してください。

次のタスク

『InfoSphere CDC Access Server のインストールと構成』

InfoSphere CDC Access Server のインストールと構成

Access Server をインストールするには、インストール・ウィザードの手順に従います。インストール後、ネットワークで通信用に静的ポートを必要とするファイアウォールまたはその他のセキュリティー・メカニズムを使用している場合は、他のコンピューターが Access Server サービスとの通信に使用できるポートを指定する必要があります。

手順

1. 「InfoSphere Change Data Capture Access Server および Management Console のインストレーション・ガイド」の説明に従って、Access Server をインストールします。

重要: Access Server アカウントは、インストール時に作成されます。Access Server アカウントは、以下のために使用されます。

- Management Console から Access Server にログオンする。

- Management Console でユーザーとデータ・ストアを管理する。
2. 環境に必要であれば、他のコンピューターが Access Server サービスとの通信に使用できるポートを指定します。

手順については、「*InfoSphere Change Data Capture Access Server* および *Management Console* のインストール・ガイド」の『Access Server のインストール後』のセクションを参照してください。

次のタスク

『InfoSphere CDC Management Console のインストールと構成』

InfoSphere CDC Management Console のインストールと構成

Management Console をインストールするには、インストール・ウィザードの手順に従います。インストール後、Access Server のインストール時に作成したシステム管理者アカウントを使用して、Management Console にログインします。

手順

1. 「*InfoSphere Change Data Capture Access Server* および *Management Console* のインストール・ガイド」の説明に従って、Management Console をインストールします。
2. Access Server のインストール時に作成したシステム管理者アカウントを使用して、Management Console にログインします。

次のタスク

- 「**Help**」 → 「**Help Contents**」メニュー・パスを使用して、Management Console のヘルプ文書を表示します。
- 『レプリケーション・サブスクリプションのセットアップ』セクションの説明に従って、solidDB Universal Cache のセットアップを続行します。

レプリケーション・サブスクリプションのセットアップ

レプリケーション・サブスクリプションは、Management Console で作成されます。このセクションでは、作成処理の概要と、「*InfoSphere Change Data Capture Management Console* 管理ガイド」での詳細な説明の参照箇所を示します。

始める前に

- 複製する表が solidDB とバックエンド・データベースの両方に存在することを確認してください。また、表に外部キーが含まれない場合には、レプリケーション中に表を作成することもできます。
- solidDB およびバックエンド・データベースが稼働していることを確認します。
- solidDB およびバックエンド・データ・サーバー用の InfoSphere CDC インスタンスが稼働していることを確認します。
- データベースに対する十分なアクセス権を取得しておきます。
- ビジネス・ルールに従って、望ましいレプリケーション原則を定義しておきます。

手順

1. **Management Console** にログインし、**Access Server** に接続します。

詳しくは、「*InfoSphere Change Data Capture Management Console 管理ガイド*」の『*Management Console* へのログイン (*Access Server* への接続)』のセクションを参照してください。

ヒント: **Management Console** のアクセス・マネージャー・パースペクティブで作業するには、データ・ストアとユーザー・アカウントの管理特権を持つシステム管理者でなければなりません。システム管理者のアカウントは、**Management Console** のインストール時に作成済みです。

2. **solidDB** およびバックエンド・データベース用のデータ・ストアをセットアップします。
 - a. 新しいデータ・ストアを追加します。詳しくは、「*InfoSphere Change Data Capture Management Console 管理ガイド*」の『*新規データ・ストアを追加する*』のセクションを参照してください。
 - b. 接続パラメーターを設定します。詳しくは、「*InfoSphere Change Data Capture Management Console 管理ガイド*」の『*データ・ストアの接続パラメーターを設定する*』のセクションを参照してください。
3. ユーザーにデータ・ストアを割り当てます。

solidDB データ・ストアとバックエンド・データ・ストアの両方を同じユーザーに割り当てる必要があります。

- a. 必要に応じて、新しいユーザーを追加します。新しいユーザーを追加するには、ユーザー・アカウントの管理特権を持つシステム管理者でなければなりません。詳しくは、「*Management Console 管理ガイド*」の『*ユーザー・アカウントの管理*』のセクションを参照してください。
 - b. ユーザーにデータ・ストアを割り当てます。詳しくは、「*Management Console 管理ガイド*」の『*データ・ストアへのユーザーの割り当て*』のセクションを参照してください。
 - c. 変更を有効にするには、**Access Server** から切断して、再接続します。
 - 1) 「File」 > 「Access Server」 > 「Disconnect」をクリックします。
 - 2) 「File」 > 「Access Server」 > 「Connect」をクリックします。
4. オプションで、データ・ストアへの接続用の接続設定を行います。デフォルト設定をそのまま使用することもできます。

詳しくは、「*InfoSphere Change Data Capture Management Console 管理ガイド*」の『*接続の設定*』のセクションを参照してください。

5. レプリケーション用のデータ・ストアをセットアップします。
 - a. **solidDB** のデータ・ストアに接続します。
 - b. バックエンド・データ・ストアに接続します。

詳しくは、「*InfoSphere Change Data Capture Management Console 管理ガイド*」の『*レプリケーション用データ・ストアのセットアップ*』のセクションを参照してください。

6. オプションで、**solidDB** およびバックエンド・データ・ストア上でシステム・パラメーターを設定します。

Universal Cache 固有のシステム・パラメーターについて詳しくは、29 ページの『solidDB Universal Cache 固有の InfoSphere CDC システム・パラメーター設定 - 要約』を参照してください。

システム・パラメーターの設定方法について詳しくは、「InfoSphere Change Data Capture Management Console 管理ガイド」の『ソース・データ・ストアおよびターゲット・データ・ストアでのシステム・パラメーターの設定』のセクションを参照してください。

7. サブスクリプションをセットアップします。

- a. solidDB がソースで、バックエンド・データ・ストアがターゲットのサブスクリプションを追加します。
- b. バックエンド・データ・ストアがソースで、solidDB がターゲットのサブスクリプションを追加します。

詳しくは、「InfoSphere Change Data Capture Management Console 管理ガイド」の『サブスクリプションのセットアップ』のセクションを参照してください。

8. 両方のサブスクリプションで、レプリケーション用の表をマップします。

詳しくは、「InfoSphere Change Data Capture Management Console 管理ガイド」の『表のマッピング』のセクションを参照してください。

ヒント: Universal Cache をインプリメントするには、レプリケーション方式として「Mirror (Change Data Capture)」を選択する必要があります。

9. 表マッピングごとに、ビジネス・ルールに従って競合検出と解決を設定します。

詳しくは、「InfoSphere Change Data Capture Management Console 管理ガイド」の『競合検出および解決の設定』のセクションを参照してください。

10. オプションで、ソース列の文字セット変換を設定します。

solidDB データベース・モードが Unicode (**General.InternalCharEncoding=UTF8**) の場合、文字データ型 (CHAR、VARCHAR など) の列のエンコードを UTF-8 に設定します。

詳しくは、32 ページの『solidDB Universal Cache での Unicode データベースおよび部分的 Unicode データベースの使用』を参照してください。

11. 両方のサブスクリプションでレプリケーションを開始します。

詳しくは、「InfoSphere Change Data Capture Management Console 管理ガイド」の『サブスクリプションでのレプリケーションの開始と終了』のセクションを参照してください。

次のタスク

- データ・ストアおよびサブスクリプションの管理方法の一般的な説明については、「InfoSphere Change Data Capture Management Console 管理ガイド」を参照してください。

- Universal Cache 固有の設定および管理タスクについては、29 ページの『3 章 InfoSphere CDC 用の solidDB Universal Cache の固有の設定およびタスク』を参照してください。
- solidDB Universal Cache のパフォーマンスを最適化およびモニターする方法に関する solidDB 固有の説明については、37 ページの『4 章 パフォーマンスのチューニングおよびモニター』を参照してください。

3 InfoSphere CDC 用の solidDB Universal Cache の固有の設定およびタスク

このセクションでは、InfoSphere CDC テクノロジーを solidDB Universal Cache で使用するための具体的な説明を行います。InfoSphere CDC インスタンスおよびレプリーション・サブスクリプションの管理に関する一般的な説明は、「InfoSphere Change Data Capture Management Console 管理ガイド」に記載されています。

solidDB Universal Cache 固有の InfoSphere CDC システム・パラメーター設定 - 要約

バックエンド・データ・サーバーおよびデータベースの設定によっては、InfoSphere CDC のシステム・パラメーター設定を変更する必要があることがあります。

表 10. solidDB Universal Cache 固有の InfoSphere CDC システム・パラメーター設定

コンポーネント	システム・パラメーター	変更するタイミング
InfoSphere CDC for solidDB	refresh_with_referential_integrity	サブスクリプションに外部キーを持つ表が含まれる場合は、InfoSphere CDC for solidDB の refresh_with_referential_integrity システム・パラメーターを「true」に設定します。 詳しくは、30 ページの『外部キーの使用の有効化 (参照整合性)』を参照してください。
	retrieve_credentials	SQL パススルーを使用しており、バックエンド・データ・サーバーが DB2 for iSeries [®] 、または DB2 for z/OS [®] である場合は、InfoSphere CDC for solidDB の retrieve_credentials を「false」に設定します。
	solid_fast_refresh_on solid_fast_refresh_apply_pipes	solidDB がソース・データ・ストアである場合に、サブスクリプションの高速リフレッシュを有効にする場合は、 solid_fast_refresh_on を「true」に設定し、 solid_fast_refresh_apply_pipes に、システム内のプロセッサ (コア) の数 (デフォルトは 2) を設定します。 詳しくは、33 ページの『高速リフレッシュの使用可能化』を参照してください。

表 10. solidDB Universal Cache 固有の InfoSphere CDC システム・パラメーター設定 (続き)

コンポーネント	システム・パラメーター	変更するタイミング
InfoSphere CDC for DB2 Linux、UNIX、および Windows	refresh_allow_fast_loader	サブスクリプションに外部キーを含む表が含まれており、バックエンド・データ・サーバーが DB2 for Linux、UNIX、および Windows である場合は、InfoSphere CDC for DB2 の refresh_allow_fast_loader システム・パラメーターを「false」に設定します。 詳しくは、『外部キーの使用の有効化 (参照整合性)』を参照してください。
	ddl_awareness	双方向レプリケーションを使用しており、バックエンド・データ・サーバーが DB2 for Linux、UNIX、および Windows である場合は、InfoSphere CDC for DB2 の ddl_awareness システム・パラメーターを「false」に設定します。
InfoSphere CDC for Oracle	refresh_allow_fast_loader ts_fast_loader_disable_constraint	サブスクリプションに外部キーを持つ表が含まれ、バックエンド・データ・サーバーが Oracle の場合は、InfoSphere CDC for Oracle の refresh_allow_fast_loader および ts_fast_loader_disable_constraint システム・パラメーターを「false」に設定します。 詳しくは、『外部キーの使用の有効化 (参照整合性)』を参照してください。

外部キーの使用の有効化 (参照整合性)

サブスクリプションに外部キーを持つ表が含まれている場合、InfoSphere CDC for solidDB のシステム・パラメーター **refresh_with_referential_integrity** を「true」に設定する必要があります。また、バックエンド・データ・サーバーが Oracle、または DB2 for Linux、UNIX、および Windows の場合は、高速ローダーを無効にする必要があります。

このタスクについて

以下を使用して、システム・パラメーターを設定できます。

- `dmset -I <INSTANCE_NAME> <parameter_name>=<parameter_value>` コマンド
- または、Management Console を使用して以下を行います。
 1. Management Console の「**Configuration**」パースペクティブで、データ・ストアを選択します。
 2. データ・ストアを右クリックし、「**Properties**」 → 「**System Parameters**」を選択します。

アクティブ・レプリケーション時にシステム・パラメーターを変更した場合は、変更を有効にするために、レプリケーションを停止して再始動する必要があります。

手順

1. InfoSphere CDC for solidDB のシステム・パラメーター **refresh_with_referential_integrity** を「true」に設定します。

例えば、以下のようにします。

```
dmset -I solidDB_1 refresh_with_referential_integrity=true
```

2. バックエンド・データ・サーバーが DB2 for Linux、UNIX、および Windows の場合は、InfoSphere CDC for DB2 の **refresh_allow_fast_loader** システム・パラメーターを次のように「false」に設定します。

例えば、以下のようにします。

```
dmset -I DB2_1 refresh_allow_fast_loader=false
```

3. バックエンド・データ・サーバーが Oracle の場合、InfoSphere CDC for Oracle の **refresh_allow_fast_loader** および **ts_fast_loader_disable_constraint** システム・パラメーターを「false」に設定します。

例えば、以下のようにします。

```
dmset -I Oracle_1 refresh_allow_fast_loader=false
```

```
dmset -I Oracle_1 ts_fast_loader_disable_constraint=false
```

solidDB ソース表のドロップおよび再作成

solidDB がソース・データ・ストアであるサブスクリプションの表をドロップして再作成する必要がある場合は、表マッピングを再構成する必要があります。

手順

1. solidDB がソース・データ・ストアであるサブスクリプションで、レプリケーションを停止します。
2. ソース表を再マップします。
3. サブスクリプションのレプリケーション (ミラーリング) を再始動します。

表をマップしてサブスクリプションを開始および停止する方法については、

「*InfoSphere Change Data Capture Management Console 管理ガイド*」を参照してください。

データを同期化しないミラーリングの開始

サブスクリプションに対するミラーリングを開始すると、レプリケーション方式が「Mirror」で、状況が「Refresh」のすべての表は、最初にサブスクリプションでリフレッシュされます。これにより、ソース表とターゲット表が同期します。ミラーリングを開始するキャプチャー・ポイントを手動で設定することにより、それができます。」→「ミラーリングを開始するキャプチャー・ポイントを手動で設定します。これは、例えば、既にフロントエンドとバックエンドのデータベースが同期していることが分かっている場合などに便利です。サブスクリプションに多数のデータが含まれている場合は、リフレッシュせずにミラーリングを開始すると、時間の節約になります。

手順

1. ソース表を含んでいるサブスクリプションに対するアクティブ・レプリケーションが終了していることを確認します。
2. Management Console または `dmmarktablecapturepoint` コマンドを使用して、表キャプチャー・ポイントにマークを付けます。

- Management Console を使用して表キャプチャー・ポイントにマークを付ける方法については、「*InfoSphere Change Data Capture Management Console 管理ガイド*」の『ソース表での表キャプチャー・ポイントのマーキング』のセクションを参照してください。
- `dmmarktablecapturepoint` コマンドの使用方法については、117 ページの『`dmmarktablecapturepoint`: ソース表に表キャプチャー・ポイントのマークを付ける』のセクションを参照してください。

solidDB Universal Cache での Unicode データベースおよび部分的 Unicode データベースの使用

solidDB のデータベース・モード (Unicode または部分的 Unicode) に応じて、文字データ型 (CHAR、VARCHAR など) の列のエンコードを指定する必要が生じることがあります。

このタスクについて

- solidDB のデータベース・モードが *Unicode* (`General.InternalCharEncoding=UTF8`) の場合、solidDB 文字データ型 (CHAR、VARCHAR など) の列のエンコードを UTF-8 に設定します。
- solidDB データベース・モードが *部分的 Unicode* (`General.InternalCharEncoding=Raw`) の場合、solidDB 文字データ型 (CHAR、VARCHAR など) の列のエンコードを、アプリケーション環境で使用されているエンコードに設定します。

重要: デフォルトで、文字データ型列のエンコードは、ISO Latin1 に設定されます。アプリケーションで Latin1 エンコードが使用されている場合、エンコードを明示的に設定する必要はありません。

手順

1. Management Console で、「**Configuration**」 → 「**Subscriptions**」とクリックします。
2. サブスクリプションを選択します。
3. 「**Table Mappings**」ビューをクリックし、表マッピングを選択します。
4. 「**Edit Mapping Details**」を右クリックし、選択します。
5. 「**Translation**」タブをクリックします。
6. 文字データ型 (CHAR、VARCHAR など) のソース列を選択します。これにより、「**Encoding Conversion**」領域が有効になります。
7. 「**Source**」リストから文字エンコードを選択します。
 - Unicode データベース: UTF-8
 - 部分的 Unicode データベース: アプリケーションのエンコード
8. 「**Target**」リストから、変換する文字エンコードを選択します。例えば、バックエンド・データ・サーバーに、UCS-2 ビッグ・エンディアン・フォームの文字データ型が保管されている場合があります。
9. 「**Apply**」をクリックします。

10. solidDB がソース・データ・ストアまたはターゲット・データ・ストアであるすべてのサブスクリプションに対して、上記の手順を繰り返します。

タスクの結果

サブスクリプションでレプリケーションを開始すると、InfoSphere CDC はソース列の文字エンコードを指定したエンコードに変換し、マップされたターゲット列に新規エンコードでデータを入力します。

高速リフレッシュの使用可能化

高速リフレッシュ機能により、バックエンド・データ・サーバーから solidDB フロントエンドへの大容量のデータの複製にかかる時間が削減されます。高速リフレッシュを使用可能にするには、InfoSphere CDC for solidDB のシステム・パラメーター **solid_fast_refresh_on** を「true」に設定します。パフォーマンスをさらに向上させるには、**solid_fast_refresh_apply_pipes** システム・パラメーターを、システム内のプロセッサ（コア）の数に設定します。

始める前に

高速リフレッシュ機能は、solidDB がターゲット・データ・ストアであるサブスクリプションにおいてのみ使用できます。

大部分のパフォーマンス向上は、簡単なセットアップで実現されます。例えば、表あたり行あたりのデータ量、コード・ページ変換、および列マッピングなどの要因が、高速リフレッシュ機能のパフォーマンスに影響を及ぼします。

このタスクについて

以下を使用して、システム・パラメーターを設定できます。

- `dmset -I <INSTANCE_NAME> <parameter_name>=<parameter_value>` コマンド
- または、Management Console を使用して以下を行います。
 1. Management Console の「**Configuration**」パースペクティブで、データ・ストアを選択します。
 2. データ・ストアを右クリックし、「**Properties**」 → 「**System Parameters**」を選択します。

アクティブ・レプリケーション時にシステム・パラメーターを変更した場合は、変更を有効にするために、レプリケーションを停止して再始動する必要があります。

手順

1. InfoSphere CDC for solidDB システム・パラメーター **solid_fast_refresh_on** を「true」に設定します（デフォルトは「false」です）。

例えば、以下のように指定します。

```
dmset -I solidDB_1 solid_fast_refresh_on=true
```

2. InfoSphere CDC for solidDB システム・パラメーター **solid_fast_refresh_apply_pipes** を、システム内のプロセッサ（コア）の数（デフォルトは「2」）に設定します。

例えば、以下のようにします。

```
dmset -I solidDB_1 solid_fast_refresh_apply_pipes=4
```

solidDB Universal Cache での共有メモリー・アクセス (SMA) の使用

solidDB Universal Cache で SMA を使用するには、SMA サーバーを始動し、InfoSphere CDC for solidDB インスタンスと SMA サーバー間のローカル SMA 接続を有効にする必要があります。

始める前に

SMA 接続の場合、SMA を使用する solidDB サーバーと InfoSphere CDC for solidDB は同じノードに配置する必要があります。

手順

1. SMA ドライバー・ライブラリーの場所が、LD_LIBRARY_PATH or LIBPATH (Linux および UNIX の場合) または PATH (Windows の場合) 環境変数に含まれていることを確認します。

詳しくは、「*IBM solidDB 共有メモリー・アクセスおよびリンク・ライブラリー・アクセス・ユーザー・ガイド*」の『Java を使用する場合の SMA 用の環境の構成』を参照してください。

2. ファイル・タイプ拡張子を使用せずに、SMA ドライバー・ライブラリーのシンボリック・リンク (ssolidisma65) を <solidDBインストール・ディレクトリー>/bin ディレクトリーに作成します。

例えば、Linux オペレーティング・システムの場合、以下のコマンドを使用します。

```
ln -s ssolidisma65.so ssolidisma65
```

3. コマンド・プロンプトにコマンド solidsma を入力して、SMA を始動します。
4. InfoSphere CDC for solidDB インスタンスを構成して、solidDB サーバーに接続するときに SMA 接続が使用されるようにします。

InfoSphere CDC for solidDB 構成ツール (dmconfigurets) を使用して、SMA 接続を有効にします。

表 11. dmconfigurets を使用した SMA 接続の使用可能化

オペレーティング・システム	dmconfigurets を使用して SMA 接続を使用可能にする方法
Linux および UNIX	<ol style="list-style-type: none">1. 「Single server」構成タイプを選択します。2. 「Enable SMA」オプションで、y を入力し、Enter キーを押します。
Windows	「 New instance 」または「 Edit instance 」ダイアログの「 Server 」領域で、「 Enable SMA 」チェック・ボックスを選択します。

「**Enable SMA**」を選択すると、solidDB 固有の接続プロパティー `solid_shared_memory=yes` が接続ストリングに追加されます。

関連概念

98 ページの『InfoSphere CDC の構成 (UNIX および Linux)』

94 ページの『InfoSphere CDC の構成 (Windows)』

4 パフォーマンスのチューニングおよびモニター

solidDB Universal Cache のパフォーマンスは、システムおよびセットアップに関する多くの考慮事項に応じて異なります。Management Console のモニター機能および solidDB パフォーマンス・カウンターは、パフォーマンス・レベルのモニターと分析のための手段を提供します。

solidDB Universal Cache のパフォーマンスに影響を与える要因

solidDB Universal Cache のパフォーマンスは、例えば、システムおよびネットワーク・セットアップの最適化や、構成セットアップへの並列処理の導入などによって、改善することができます。

この章のガイドラインは、solidDB (「*IBM solidDB 管理者ガイド*」を参照) およびバックエンド・データ・サーバーに固有のチューニング情報を補足することです。

システムおよびネットワーク・セットアップの最適化

- 各 InfoSphere CDC インスタンスには、少なくとも 256 MB のメモリー量を割り振る必要があります。

InfoSphere CDC インスタンスのメモリーの割り振りは、構成ツールを使用してインスタンスを作成するときに定義されます (オプション **Maximum Memory Allowed**)。最小割り振り量は 64 MB です。デフォルトは、32 ビット・インスタンスでは 512 MB で、64 ビット・インスタンスでは 1024 MB です。

- 使用可能な CPU

InfoSphere CDC によるデータ処理は、CPU 集中型であるため、solidDB Universal Cache のセットアップに組み込まれた全ノードで十分な CPU 容量が使用可能であることを確認してください。

- ネットワークの待ち時間とスループット

ネットワークを最適化してスループットを高め、待ち時間を短縮すると、solidDB Universal Cache のパフォーマンスが向上します。

並列処理によるパフォーマンスの向上

標準的なセットアップでは、InfoSphere CDC エンジンは、パフォーマンスに以下の制限を設定します。

- ソース・エンドでは、パフォーマンスは、すべてのサブスクリプションに対して 1 秒当たり合計約 15000 回の操作に制限されます。
- ターゲット・エンドでは、パフォーマンスは、各サブスクリプションごとに毎秒約 5000 回の操作に制限されます。

複数のサブスクリプションまたは複数の solidDB フロントエンドを使用して、データとワークロードをパーティション化することで、パフォーマンスを改善することができます。これは、複数のサブスクリプションがフロントエンドとバックエンドの両方で並列に処理されるためです。

例えば、サブスクリプションの外部とは相互に参照しない自律型の表に対する個別のサブスクリプションを作成できます。このような表/サブスクリプションは、個別の solidDB フロントエンドにも配置することができます。

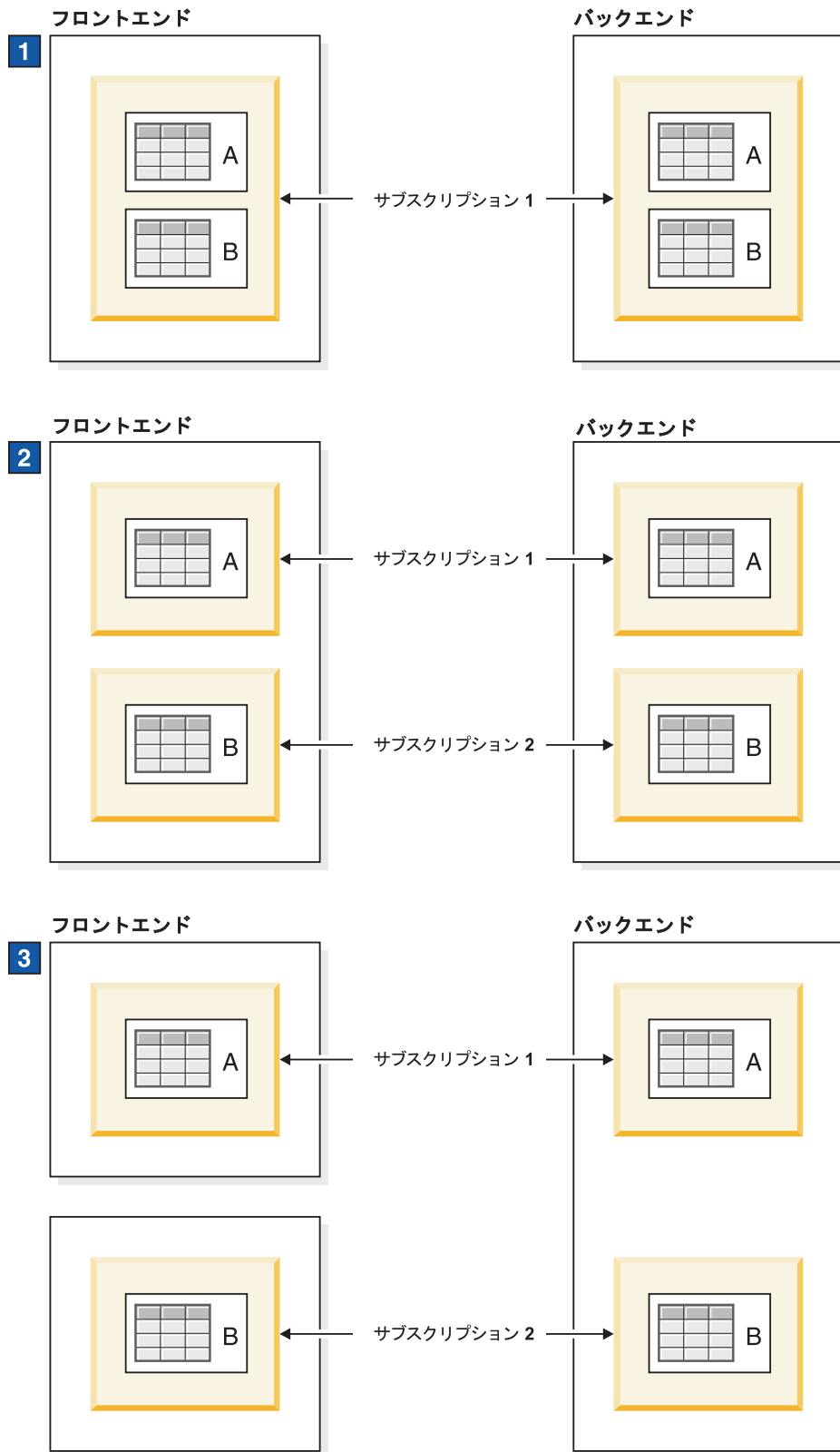


図 5. 例: 3 つのパーティション化モデルを使用した *solidDB Universal Cache* のセットアップ

1. 単一 *solidDB* フロントエンドと 1 つのサブスクリプション (パーティション化なし)

2. 単一 solidDB フロントエンドと 2 つのサブスクリプション (2 つの並列サブスクリプションによるパーティション化)
3. 2 つの solidDB フロントエンドと、それぞれ 1 つずつのサブスクリプション (2 つのフロントエンドによるパーティション化)

その他の考慮事項

- ターゲット・データベースは、InfoSphere CDC エンジンが生成する負荷に対処できる必要があります。
- あらゆるデータ処理でボトルネックが発生する可能性があります。以下に例を示します。
 - 行レベル・フィルター
 - データ形式変更およびデータ式
 - コード・ページ変換

パフォーマンスのモニター

Management Console の「**Monitoring**」および「**Statistics**」ビューを使用すると、サブスクリプションのパフォーマンス統計を収集できます。solidDB パフォーマンス・カウンターは、solidDB フロントエンドのパフォーマンス・データを提供します。

Management Console でのパフォーマンスのモニター

Management Console では、待ち時間、スループット、およびレプリケーション操作の回数とサイズに関する統計を収集できます。この統計は、Management Console に表示するか、あるいは .csv 形式で保存してエクスポートすることができます。待ち時間の通知としきい値を設定することもできます。

Management Console でのモニターと統計の使用法の詳細については、「*InfoSphere Change Data Capture Management Console 管理ガイド*」のセクション『サブスクリプションの監視』を参照してください。

solidDB フロントエンド・パフォーマンスのモニター

solidDB には、solidDB を InfoSphere CDC とともに使用する場合に専用のいくつかのパフォーマンス・カウンターが用意されています。

- 名前が *Logreader* から始まる変数を持つカウンター

例えば、*Logreader commits sent* は、1 秒あたりに InfoSphere CDC インスタンスに送信されたコミット数を追跡します。

- *TS applied transactions*

TS applied transactions カウンターは、solidDB がターゲット・データ・ストアである場合に、InfoSphere CDC インスタンスによって solidDB に適用されるトランザクションの数を追跡します。

solidDB パフォーマンス・カウンターの詳細なリストおよびその使用方法については、「*IBM solidDB 管理者ガイド*」の『*solidDB のモニター*』を参照してください。

5 SQL パススルー

solidDBSQL パススルーを使用すると、アプリケーションからフロントエンドとバックエンドの両方のデータ・サーバーのデータに単一接続でアクセスできます。例えば、SQL パススルーを有効にして、solidDB フロントエンド・サーバーでは実行できない SQL ステートメントをバックエンドに渡すように設定することができます。SQL パススルー・モードは、セッション単位またはトランザクション単位で設定できます。デフォルトでは、SQL パススルーは有効ではありません。

フロントエンドとバックエンドの間の接続は、バックエンド互換の ODBC ドライバーを使用して確立されます。このドライバーは、solidDB サーバーに動的にロードされます。solidDB は、このドライバーを使用して、バックエンド・データ・サーバーでパススルー・ステートメントを直接実行します。

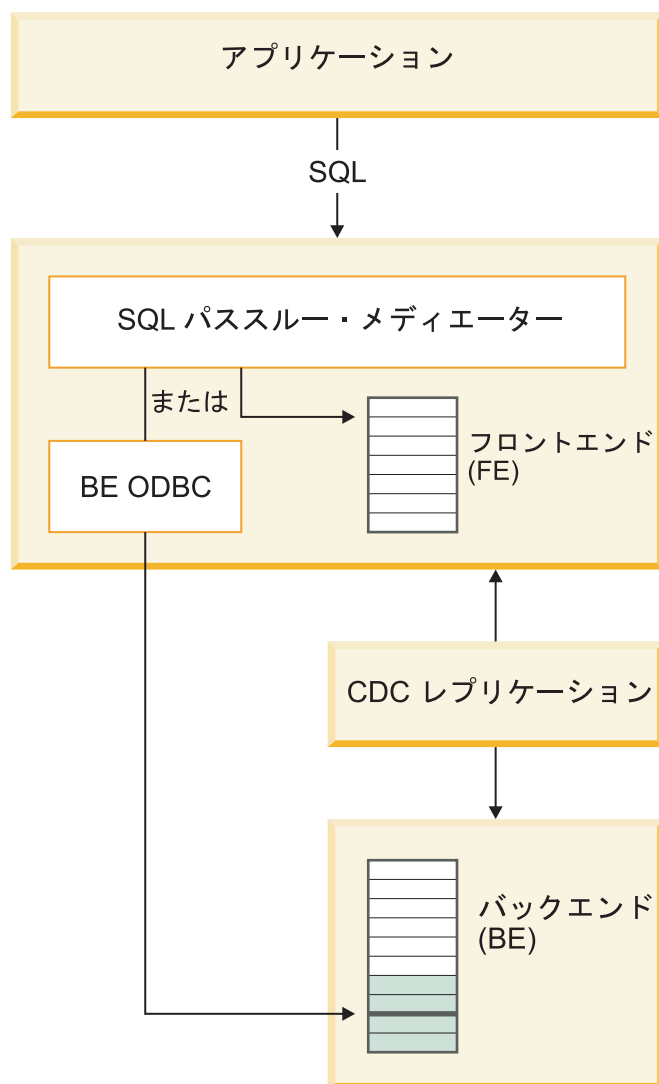


図 6. SQL パススルー

操作の原理

solidDB サーバー内の SQL パススルー・メディエーター と呼ばれる層は、選択されたパススルー・モードに従って、バックエンドへの SQL ステートメントのパススルーを処理します。SQL パススルー・モードは、実行時に動的に変更できます。バックエンド・サーバーへのアクセスは、solidDBサーバーにリンクしたバックエンド ODBC ドライバーを使用すると容易になります。バックエンドのログイン・データ (ユーザー名とパスワード) は、InfoSphere CDC コンポーネントを介して転送されます。

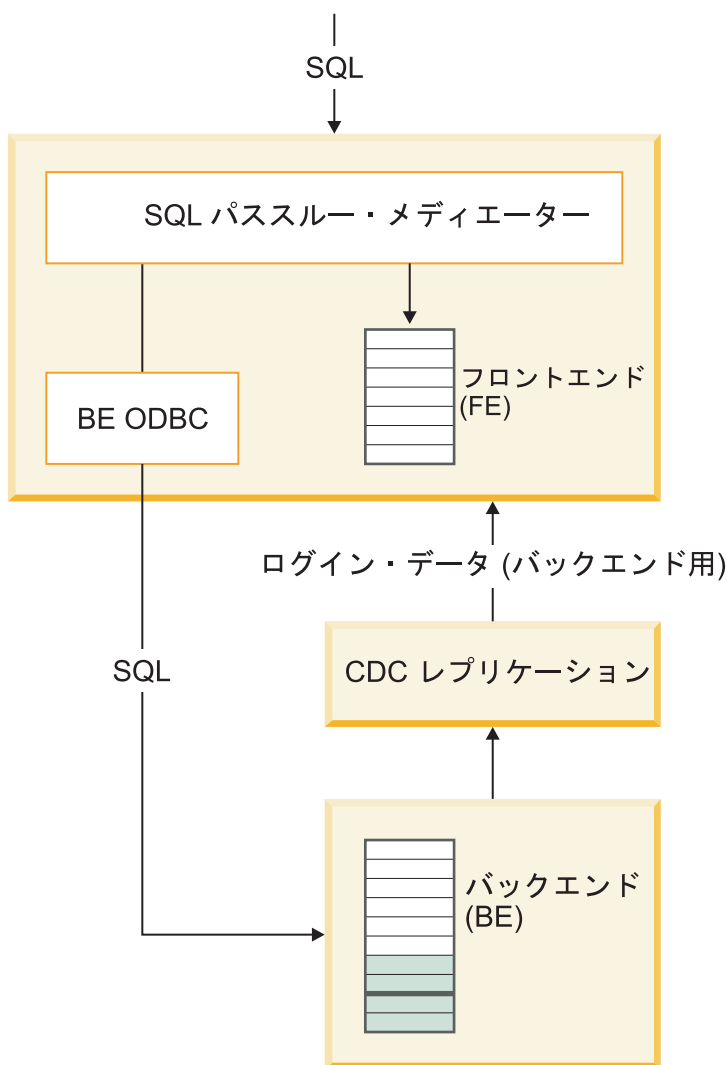


図7. SQL パススルーのアーキテクチャー

パススルー・モード

パススルー・モードは、読み取り/書き込みステートメントをバックエンドに渡す方法を定義します。パススルー・モードは、読み取りステートメント (SELECT) およ

び書き込みステートメント (INSERT、UPDATE、DELETE などの読み取り以外のステートメント) に対して個別に設定します。以下の 3 つのパススルー・モードを使用できます。

- **FORCE:** すべての読み取り/書き込みステートメントが、バックエンドに渡されず。
- **NONE (デフォルト):** 読み取り/書き込みステートメントは、バックエンドに渡されません。
- **CONDITIONAL:** ステートメントで、例えば、表欠落エラーや構文エラーなどが発生した場合、そのステートメントがバックエンドに渡されます。

CONDITIONAL パススルー・モードは、エラー・メッセージを使用するロジックに基づきます。

- 表欠落エラーまたは構文エラーは、パススルーを実行します (準備段階)。

ステートメントが準備された後でパススルー・モードが変更されると、必要に応じてステートメントが新しい位置で再び準備されます。

- 特権違反エラーは、パススルーを実行しません。
- 実行段階で発生するエラーは、パススルーを実行しません。例えば、書き込みステートメントが保全性制約違反で失敗した場合、パススルーは実行されません。

さらに、SQL ステートメントを複合して使用すると、長時間にわたって実行されるステートメントが常にバックエンドにパススルーされるように定義することができます。

注: SQL パススルーは、フロントエンド・データベースとバックエンド・データベースのデータ範囲またはデータ設定に関する情報を使用しません。特に、フロントエンドで照会が正常に実行されたが、データが返されなかった (あるいは、わずかなデータしか返されなかった) 場合、データが存在する可能性があっても、照会はバックエンドにリダイレクトされません。

トランザクションと分離レベル

SQL パススルーのトランザクション・モデルは、バックエンド・データベースの整合性を保持するように設計されており、バックエンド・トランザクションは最高の分離レベル (REPEATABLE READ または SERIALIZABLE) を満たすことができます。SQL パススルーの使用時にバックエンド・データベースの整合性を保持するには、フロントエンドの分離レベルをバックエンドと同一 (または、ほぼ同等) に設定するか、より高く設定します。

一般に、個別のトランザクションは、フロントエンドまたはバックエンドで、完全に実行およびコミットされます。これらは、ローカル・トランザクションとして、データベースの整合性を保持し、目的の分離レベルの制限が与えられます。ただし、連続トランザクションでは、バックエンドからフロントエンドへの非同期レプリケーションによる遅延のために、一時的に相互に不整合になる可能性があります。例えば、後続のトランザクションがパススルー書き込み操作をバックエンド・データベースに実行した場合、トランザクションは先行トランザクションの複製結果を確認できない場合があります。

パススルーされるのは全部のステートメントのみで、ステートメントがフロントエンドとバックエンドの両方にまたがることはできません。つまり、照会を分散させることはできません。

場合によっては、トランザクションはフロントエンドまたはバックエンドのいずれかのデータベースから読み取り、他方のデータベースに書き込みます。このようなトランザクションは、2つのサブトランザクションから構成されているものと見なされる場合があります。そのようなトランザクションがコミットされるのは、書き込みのサブトランザクションが正常にコミットされときに限られます。書き込みサブトランザクションのコミットが失敗した場合は、そのトランザクション全体も失敗します。

バックエンドのデータのアクセス

フロントエンドとバックエンドの間の接続は、バックエンド ODBC ドライバーを使用して確立されます。このドライバーは、solidDB ノードにインストールされ、solidDB サーバーに動的にロードされます。solidDB は、このドライバーを使用して、バックエンド・データ・サーバーでパススルー・ステートメントを直接実行します。

ほとんどの場合、InfoSphere CDC テクノロジーを使用して、バックエンドのログイン・データがフロントエンドに転送されます。solidDB からバックエンド・データ・サーバーへの最初のサブスクリプションでミラーリングまたはリフレッシュが開始されたときに、InfoSphere CDC for solidDB インスタンスは、ログイン・データをバックエンド InfoSphere CDC インスタンスからリトリートし、そのデータを solidDB システム表 SYS_SERVER に CREATE REMOTE SERVER ステートメントと一緒に保管します。SYS_SERVER 表に保管されたパスワードは隠蔽されます。

InfoSphere CDC テクノロジーは、以下の場合にはバックエンドのログイン・データを転送しません。

- バックエンドの InfoSphere CDC インスタンスが、データベースに自動的にアクセスできるユーザー ID を使用して実行されている。このため、ログイン・データを保管する必要はありません。
- バックエンド・データ・サーバーが DB2 for z/OS または DB2 for iSeries である。
- ご使用の InfoSphere CDC for solidDB インストール済み環境およびサブスクリプションを 6.3 からアップグレードしてある。6.3 バージョンの InfoSphere CDC for solidDB は、バックエンドのログイン・データを SYS_SERVER 表に保管しません。

上記のような場合は、CREATE REMOTE SERVER (または ALTER REMOTE SERVER) ステートメントを使用して、ログイン・データを手動で定義します。

SQL パススルーを使用したアプリケーション開発に関する考慮事項

アクセス権限

- パススルー機能は、検証済みユーザーのみが使用できます。この検証メカニズムは、GRANT PASSTHROUGH ステートメントに基づきます。

新規データベースでは、管理者がパススルー検証を行います。

- 表に対するユーザー・アクセスは制限される場合があります。フロントエンド・データベースで、ステートメントが特権制限によって制限される場合、バックエンド・データベースに特権制限がない場合であっても、対応するステートメントはバックエンド・データベースに渡されません。

SQL ステートメント

- SET [TRANSACTION] PASSTHROUGH 自体、SET [TRANSACTION] ISOLATION LEVEL ステートメント、およびデータ・エージング関連ステートメントである SET DELETE CAPTURE を除き、すべての SQL ステートメントをバックエンドに渡すことができます。また、以下の制限事項も適用されます。
 - SELECT ステートメントでは、前方カーソルのみがサポートされます。
 - UPDATE/DELETE ... WHERE CURRENT OF ステートメントはサポートされていないため、使用すると予期しない結果が生じる場合があります。
 - 次の行以外をフェッチすると、エラーが返されます。
 - データベース全体にわたるメタデータ照会は、常にフロントエンドで実行されます。

例えば、SQL パススルー・モードが FORCE に設定されていても、SELECT * FROM TABLES などの照会や、getTables などの JDBC 関数呼び出しによって、solidDB の表に関する情報が返されます。

ステートメント固有のメタデータ照会 (例えば、ODBC SQLColAttr()) は、可能な場合は solidDB から受け取ったデータを使用し、可能でない場合は、バックエンド・データベースから受け取ったデータを使用します。

- SQL ステートメントが両方のデータベースで実行できる場合、表定義 (例えば、列タイプ) は、常に solidDB から取得されます。solidDB とバックエンドの定義が異なる場合、フロントエンドおよびバックエンド間でデータは可能であれば変換されます。列の数と列名は一致している必要があります。
- solidDB サーバーが Java Transaction API (JTA) インターフェースを使用して分散トランザクションに参加する場合は、読み取りステートメント (SELECT) のみがパススルーされます。
- 複合 SQL ステートメントが常にバックエンドにパススルーされるように定義できます。複合照会はバックエンドでより効率的に実行される可能性があります。ステートメントがパススルーされる複合レベルは、以下のパラメーターで定義されます。
 - **Passthrough.ComplexNumTables** – 複合ステートメント内の表の最小数を指定します。ステートメントの表の数が、このパラメーターで指定された表の数より少ない場合、このステートメントは複合ステートメントではなく、バックエンドにパススルーされません。
 - **Passthrough.ComplexNumNonindexedConstr** – 複合ステートメント内の非索引 WHERE 節制約の最小数を指定します。ステートメントに含まれる非索引制約 (WHERE 節制約が索引で解決しないか、索引が存在しないか、あるいはオペレーターが異なる索引を制約に選択するかのいずれかのタイプ) の数がその最小数より少ない場合、そのステートメントは複合ステートメントではなく、バックエンドにパススルーされません。

- **Passthrough.ComplexNumOrderedRows** - 複合ステートメントでソートする必要がある行の最小推定数を指定します。ステートメントに含まれるソート可能な行の数がその推定数よりも少ない場合、そのステートメントは複合ステートメントではなく、バックエンドにパススルーされません。

この 3 つのパラメーターのファクトリー値はすべて 0 (ゼロ) です。これは、複合ステートメントであるかどうかの推定時に、所定のプロパティーが使用されないことを意味します。

データ・タイプと列のバインディング

- SQL パススルーは、solidDB がサポートするすべての標準 SQL 標準データ型をサポートします。詳しくは、163 ページの『付録 C. SQL パススルーでの ODBC データ型のサポート』を参照してください。
- アプリケーション・サイドのドライバーでは、列のバインディングは標準の ODBC バインディング方式に基づきます。

コード・ページのサポート

- コード・ページのサポートは、以下のように solidDB のデータベース・モードによって異なります。
 - solidDB のデータベース・モードが Unicode の場合、SQL パススルーは、情報を損失することなく、フロントエンドとバックエンドでさまざまなコード・ページの使用をサポートします。
 - solidDB のデータベース・モードが部分的 Unicode の場合、Latin-1 または ASCII (Latin-1 のサブセット) コード・ページのみがサポートされます。

Latin-1 文字セット以外のエンコードがデータで使用されている場合は、Unicode モードで solidDB データベースを使用することをお勧めします。

- solidDB のデータベース・モードが Unicode (**General.InternalCharEncoding=UTF8**) の場合は、solidDB からのデータが UTF-8 エンコードであることを求めるようにバックエンド ODBC ドライバーを設定する必要があります。この理由は、Unicode モードでは、文字データ型は solidDB に UTF-8 エンコードで保管されるためです。

Unicode モード環境では、バックエンド ODBC ドライバーは、solidDB の UTF-8 エンコードおよびバックエンド・エンコード間の変換を処理します。アプリケーション側では、使用可能なすべてのバインディング方式を使用することができます。その理由は、アプリケーションおよび solidDB フロントエンド・エンコード間の変換は、「*IBM solidDB プログラマー・ガイド*」のセクション『*Unicode の使用*』に説明されているように、solidDB の ODBC または JDBC ドライバーによって処理されるからです。

- solidDB のデータベース・モードが部分的 *Unicode* (**General.InternalCharEncoding=Raw**) であり、アプリケーションおよび solidDB 環境で ASCII または Latin-1 エンコード (西洋言語) が使用されている場合、ODBC ドライバーにコード・ページのサポートを明示的に設定しなくても、バックエンド ODBC ドライバーによって正しく文字変換が処理される可能性が高くなります。

これは、部分的 Unicode モードでは、文字データ型は solidDB に raw (バイナリー) フォーマットで保管され、その際、アプリケーション・エンコードと solidDB の内部表記の間で変換は行われなためです (アプリケーションが、このことを認識して、必要に応じて変換を処理することを想定しています)。

ヒント: デフォルトでは、IBM Data Server Driver for ODBC and CLI をインストールすると、ドライバーはインストール・ノードのシステム・ロケールを使用するように設定されます。

バックエンド・データベースで ASCII または Latin-1 以外のエンコードが使用されている場合は、solidDB からのデータが ASCII または Latin-1 のエンコードであることを求めるようにバックエンド ODBC ドライバーを設定する必要があります。

solidDB ツールでの SQL パススルーのサポート

- SQL パススルーでは、solidDB SQL エディター (solsql) が完全にサポートされています。
- その他の solidDB ツールは、サポートされていません。これらはフロントエンドでのみ使用できます。

エラー・コード

- フロントエンドからのエラーは、常に solidDB のネイティブ・エラー・コードです。
- バックエンドからのエラーの前には SQLSTATE が付き、これはバックエンドのネイティブ・エラー・コードおよびテキストを示します。
- バックエンドのネイティブ・エラー・コードは、マッピング・ファイルを使用して solidDB エラー・コードにマッピングできます。マッピング・ファイルは、**Passthrough.ErrorMapFileName** パラメーターで定義します。

SQL パススルーの構成と使用

SQL パススルーの構成には、solidDB 構成パラメーターの設定と、solidDB フロントエンド・ノードへのバックエンド互換 ODBC ドライバーのインストールおよび構成が含まれます。SQL パススルーの構成が完了したら、この有効/無効を動的に切り替えることができます。

また、solidDB は、トレースおよびモニター機能を備えており、SQL パススルーの接続タイプおよびステートメント・アクティビティに関するデータを収集することができます。

SQL パススルーのセットアップ

SQL パススルーの構成手順は、バックエンドと、使用する ODBC 接続のタイプによって異なります。

始める前に

SQL パススルーを有効にする前に、solidDB Universal Cache を稼働中にする必要があります。

1. solidDB Universal Cache コンポーネントをインストールします。
2. フロントエンドとバックエンドの InfoSphere CDC インスタンスを構成します。
3. フロントエンド・データ・サーバーからバックエンドに、少なくとも 1 つの表マッピングを持つサブスクリプションを最低でも 1 つ定義します。

手順

1. SQL パススルー用のバックエンド ODBC ドライバーをインストールして構成します。
 - バックエンド・データ・サーバーが IBM データ・サーバーである場合は、solidDB Universal Cache インストール・イメージとともに提供される *IBM Data Server Driver for ODBC and CLI* を使用します。
 - バックエンド・データ・サーバーが IBM データ・サーバーではない場合、バックエンド・データ・サーバーとともに提供されるネイティブ ODBC ドライバーを使用します。
2. システムのデフォルト SQL パススルー設定を構成します。

例えば、**Passthrough.PassthroughEnabled=yes** パラメーターを使用してシステムの SQL パススルーを有効に設定し、**Passthrough.SqlPassthroughRead** および **Passthrough.SqlPassthroughWrite** パラメーターを使用してデフォルトのパススルー・モードを定義します。

3. GRANT PASSTHROUGH ステートメントを使用して、該当ユーザーに SQL パススルー権限を付与します。
4. バックエンド・データ・サーバー用のログイン・データが使用可能であることを確認します。
 - a. solidDB およびバックエンドのデータ・ストアに接続し、solidDB がソース・データ・ストアで、バックエンドがターゲット・データ・ストアであるサブスクリプションでレプリケーションを開始します。
 - b. solidDB システム表 SYS_SERVER に正しいログイン・データが含まれていることを確認します。
 - たいていの場合、solidDB からバックエンド・データ・サーバーへの最初のサブスクリプションでミラーリングまたはリフレッシュが開始されたときに、InfoSphere CDC for solidDB インスタンスは、ログイン・データをバックエンド InfoSphere CDC インスタンスからリトリートし、そのデータを solidDB システム表 SYS_SERVER に保管します。
 - SYS_SERVER 表に正しくないログイン・データが含まれている場合や、ログイン・データがない場合は、ログイン・データの手動による追加または変更を行ってください。
5. アプリケーションを開始します。

SQL パススルーのためのバックエンド ODBC ドライバーのインストールと構成

solidDB で SQL パススルーを有効にする前に、solidDB フロントエンド・ノード上にバックエンド ODBC ドライバーをインストールして構成する必要があります。ドライバ・マネージャーを使用するか、または直接 (動的ドライバ・ライブラリーを使用して) ドライバーにリンクすることができます。

始める前に

ODBC ドライバー・インストール・パッケージと、使用するバックエンド・データ・サーバー用のインストールと構成の説明を見つけます。

- バックエンド・データ・サーバーが IBM データ・サーバーである場合は、solidDB Universal Cache インストール・イメージとともに提供される *IBM Data Server Driver for ODBC and CLI* を使用します。
- バックエンド・データ・サーバーが IBM データ・サーバーではない場合、バックエンド・データ・サーバーとともに提供されるネイティブ ODBC ドライバーを使用します。

手順

1. バックエンド ODBC ドライバー (クライアント) を solidDB ノードにインストールします。

- バックエンド・データ・サーバーが IBM データ・サーバーである場合、以下の手順に従います。
 - a. *IBM Data Server Driver for ODBC and CLI* が含まれている圧縮ファイルを、インストール・イメージから solidDB ノードにコピーします。
 - b. solidDB ノードの選択したインストール・ディレクトリーに、そのファイルを圧縮解除します。
 - c. オプション: 圧縮ファイルを削除します。
 - d. フロントエンド solidDB データ・サーバーが AIX で稼働している場合、以下のようにします。
 - 1) 共有ライブラリー (/odbc_cli/clidriver/lib/libdb2.a) を抽出し、64 ビット・オペレーティング・システムに shr_64.o を生成します。混乱を避けるため、ファイル名を libdb2.so に変更します。

以下のコマンドを実行します。

```
cd odbc_cli/clidriver/lib
ar -x -X 64 libdb2.a
mv shr_64.o libdb2.so
```

solidDB はドライバーを動的にロードするため、AIX ではこれらの手順を行う必要があります。

重要: AIX システム上のドライバー・ライブラリーを参照する場合、必ず正しいファイル名 (libdb2.so) を使用するようしてください。

2) DB2NOEXITLIST 環境変数を ON に設定します。

solidDB ノードで、以下のコマンドを実行します。

```
export DB2NOEXITLIST=ON
```

この環境変数によって、ドライバーは、solidDB が既に解放しているリソースの解放をシャットダウン時に試行しなくなります。

- バックエンド・データ・サーバーが IBM データ・サーバーではない場合、バックエンド・データ・サーバーとともに提供される指示に従います。
- ### 2. ODBC ドライバーとバックエンド・データ・サーバーの間に接続設定を定義します。

SQL パススルー用の ODBC ドライバーは、バックエンド・データベースを使用して通常のリモート接続を設定する場合と同じ方法で構成します。ドライバー・マネージャーを使用するか、または直接ドライバーにリンクすることができます。

• 直接リンク

バックエンド・データ・サーバーとオペレーティング・システムによっては、環境変数またはその他のセットアップ・パラメーターを設定して直接リンクを有効にすることが必要になる場合があります。

詳しくは、下の例を参照するか、バックエンド・データ・サーバーに付属する説明書を参照してください。

• ドライバー・マネージャー

使用するバックエンド・データ・サーバー、オペレーティング・システム、およびドライバー・マネージャーによっては、データ・ソース名、ログイン・データ、パフォーマンス・オプション、または接続オプションなどの設定値を構成する必要があります。

詳しくは、下の例を参照するか、バックエンド・データ・サーバーに付属する説明書を参照してください。

3. **solid.ini** 構成ファイルの **[Passthrough]** セクションを変更して、**solidDB** サーバーとドライバーまたはドライバー・マネージャー間に接続設定を定義します。

パラメーター値のフォーマットは、ドライバーに直接リンクするか、ドライバー・マネージャーを使用してリンクするかによって異なります。

直接リンク

- **RemoteServerDriverPath** を使用して、ドライバーのパスを設定します。
- **RemoteServerDSN** を使用して、ドライバーの接続ストリングを設定します。

注: 正確な接続ストリングは、ドライバーによって異なります。

RemoteServerDSN の接続ストリングのフォーマット規則について詳しくは、167 ページの『付録 D. バックエンド ODBC ドライバー接続ストリング用のフォーマット規則 (RemoteServerDSN パラメーター)』を参照してください。

例: Linux オペレーティング・システムの DB2 または IDS を使用した *IBMData Server Driver for CLI and ODBC*

```
[Passthrough]
RemoteServerDriverPath=/home/solid/odbc_cli/clidriver/lib/libdb2.so
RemoteServerDSN="Driver={IBM DB2 ODBC DRIVER};Database=my_ids;Hostname=9.212.253.10;Port=9088;protocol=TCP/IP;"
```

ドライバー・マネージャー

- **RemoteServerDriverPath** を使用して、ドライバー・マネージャーのパスを設定します。
- **RemoteServerDSN** を使用して、データ・ソース名を設定します。

例: DB2 を使用した unixODBC DriverManager

```
[Passthrough]  
RemoteServerDriverPath=/usr/lib/libodbc.so  
RemoteServerDSN=BE_DB2
```

4. **solidDB のデータベース・モード (Unicode または部分的 Unicode) に従って、ODBC ドライバーのコード・ページ・サポートを構成します。**

• **Unicode データベース**

solidDB のデータベース・モードが Unicode (**General.InternalCharEncoding=UTF8**) の場合、solidDB からのデータが UTF-8 でエンコードされるように ODBC ドライバーを構成します。

UTF-8 サポートを構成するための手順は、ドライバーによって異なります。詳しくは、バックエンド・データ・サーバーに付属する説明書を参照してください。

例えば、DB2 for Linux、UNIX、および Windows 環境では、UTF-8 サポートは、環境変数 `DB2CODEPAGE` を 1208 に設定することで構成されます (1208 という値は、DB2 環境における UTF-8 コード・ページの ID です)。

• **部分的 Unicode データベース**

– solidDB のデータベース・モードが部分的 Unicode (**General.InternalCharEncoding=Raw**) であり、アプリケーションおよび solidDB 環境で ASCII または Latin-1 エンコード (西洋言語) が使用されている場合、ODBC ドライバーにコード・ページのサポートを明示的に設定しなくても、バックエンド ODBC ドライバーによって正しく文字変換が処理される可能性は高くなります。

例えば、ASCII エンコードを使用するシステムに *IBM Data Server Driver for ODBC and CLI* をインストールした場合、そのインストールによって、ドライバーは、インストール・ノードのシステム・ロケールを使用するように自動的に設定されます。

– バックエンド・データベースで ASCII または Latin-1 以外のエンコードが使用されている場合は、solidDB からのデータが ASCII または Latin-1 のエンコードであることを求めるようにバックエンド ODBC ドライバーを設定します。

ASCII または Latin-1 サポートを構成するための手順は、ドライバーによって異なります。詳しくは、バックエンド・データ・サーバーに付属する説明書を参照してください。

重要: アプリケーションのエンコードと solidDB サーバー間の変換は、solidDB ODBC ドライバー または solidDB JDBC ドライバー によって処理されます。

• C/ODBC 環境では、アプリケーションと solidDB 間のコード・ページ変換は、サーバー・サイド・パラメーター **Srv.ODBCDefaultCharBinding** またはクライアント・サイド・パラメーター **Client.ODBCCharBinding** によって制御されます。

• Java/JDBC 環境では、設定は不要です。コード・ページ変換は、solidDB JDBC ドライバーによって自動的に処理されます。

パラメーターの設定とsolidDB Unicode の一般的なサポートについては、「*IBM solidDB プログラマー・ガイド*」の『Unicode の使用』を参照してください。

5. バックエンド・データ・サーバーが、62 ビット・システムで稼働中の DB2 であり、IBM Data Server Driver for CLI and ODBC を直接リンクとともに使用している場合、Passthrough.Force32bitODBCHandles を「yes」に設定します。

例: 直接リンクを使用した IBM Data Server Driver for Informix のインストールおよび構成:

この例では、バックエンド・データ・サーバーが Windows 32 ビット・オペレーティング・システムの IBM Informix® Dynamic Server (IDS) V11.50 であるときに、動的ドライバー・ライブラリーを使用して、*IBM Data Server Driver for CLI and ODBC* をインストールおよび構成する方法を示します。

1. ドライバーが入っている **IBM Data Server Driver for CLI and ODBC** のインストール・パッケージ (ibm_data_server_driver_for_odbc_cli_win32_v97.zip) を見つけ、それを、希望するインストール・ディレクトリー (例えば、C:\solid) にコピーします。
2. ibm_data_server_driver_for_odbc_cli_win32_v97.zip を unzip します。

ODBC ドライバー・ライブラリー・ファイル db2cli.dll は clidriver\bin ディレクトリーにあります。

3. IDS バックエンド・データ・サーバーが確実に **drtlitcp** または **drsoctcp** プロトコルを listen しているようにします (DRDA® 接続)。

例えば、**drtlitcp** プロトコルを使用するには、次のようにします。

- a. 新しいサーバー別名を SQLHOSTS ファイルに構成します。

例えば、以下のように指定します。

```
demo_on drtlitcp idshost 9088
```

- b. ONCONFIG ファイルが DRDA 接続をサーバー別名の 1 つとしてリストしていることを検証します。

詳しくは、IDS v11.50 インフォメーション・センターでセクション『*IBM データ サーバ クライアントに接続できるように Dynamic Server を構成する*』

(http://publib.boulder.ibm.com/infocenter/idshelp/v115/index.jsp?topic=com.ibm.admin.doc/ids_admin_0207.htm) を参照してください。

4. solidDB 構成ファイル (solid.ini) の中で、ドライバーのパスおよびドライバーの接続ストリングを IDS バックエンド・データ・サーバー用に定義します。

例えば、以下のように指定します。

```
[Passthrough]
RemoteServerDriverPath=C:\solid\clidriver\bin\db2cli.dll
RemoteServerDSN="Driver={IBM DB2 ODBC DRIVER};Database=my_ids;Hostname=9.252.253.10;Port=9088;protocol=TCPIP;"
```

重要: 接続ストリングは二重引用符で囲んで指定し、最初の等号と二重引用符の間にスペースを入れてはなりません。

例: 直接リンクおよび UTF-8 サポートを使用した IBM Data Server Driver for DB2 のインストールおよび構成:

この例は、バックエンド・データ・サーバーが Linux 32 ビット・オペレーティング・システムの DB2 V9.7 であるときに、動的ドライバー・ライブラリーにリンクすることで、*IBM Data Server Driver for CLI and ODBC* をインストールおよび構成する方法を示しています。また、ドライバーは、solidDB からのデータが UTF-8 エンコードであることを求めるように構成されます。

1. ドライバーが入っている **IBM Data Server Driver for CLI and ODBC** のインストール・パッケージ
(`ibm_data_server_driver_for_odbc_cli_32_linuxia32_v97.tar.gz`) を見つけ、希望するインストール・ディレクトリー (例えば、`$HOME/solid`) にそれをコピーします。

2. `ibm_data_server_driver_for_odbc_cli_32_linuxia32_v97.tar.gz` を解凍します。

```
cd $HOME/solid
uncompress ibm_data_server_driver_for_odbc_cli_32_linuxia32_v97.tar.gz
tar -xvf ibm_data_server_driver_for_odbc_cli_32_linuxia32_v97.tar.gz
```

ODBC ドライバー・ライブラリー・ファイル `db2cli.a` は、`clidriver/bin` ディレクトリーにあります。

3. `DB2NOEXITLIST` 環境変数を ON に設定します。

```
export DB2NOEXITLIST=ON
```

4. solidDB 構成ファイル (`solid.ini`) の中で、ドライバーのパスおよびドライバーの接続ストリングを IDS バックエンド・データ・サーバー用に定義します。

例えば、以下のように指定します。

```
[Passthrough]
RemoteServerDriverPath=C:\solid\clidriver\bin\db2cli.dll
RemoteServerDSN="Driver={IBM DB2 ODBC DRIVER};Database=my_db2;HostIP=9.252.253.10;Port=9088;protocol=TCPIP;"
```

重要: 接続ストリングは二重引用符で囲んで指定し、最初の等号と二重引用符の間にスペースを入れてはなりません。

5. solidDB からのデータが UTF-8 エンコードであることを求めるようにドライバーを構成します。

- a. solidDB データベースが Unicode データベース

(**General.InternalCharEncoding=UTF8**) であることを確認します。

- b. C/ODBC 環境で、solidDB ODBC ドライバーが、文字データ型列に関して、アプリケーションと solidDB の間でコード・ページ変換を処理するように構成されていることを確認します。

例えば、アプリケーションで文字データ型に UTF-8 エンコードが使用されている場合は、以下のパラメーター設定を用いて、文字データ型が UTF-8 エンコードであることを求めるように solidDB ODBC ドライバーを構成することができます。

```
[Srv]
ODBCDefaultCharBinding=utf8
```

Srv.ODBCDefaultCharBinding パラメーターと、solidDB Unicode の一般的なサポートについては、「*IBM solidDB プログラマー・ガイド*」の『Unicode の使用』を参照してください。

- c. DB2 固有の環境変数 DB2CODEPAGE を 1208 に設定します。

1208 という値は、DB2 環境における UTF-8 コード・ページの ID です。

例: unixODBC DriverManager を使用した IBM Data Server Driver for DB2 のインストールおよび構成:

この例では、バックエンド・データ・サーバーが Linux 32 ビット・オペレーティング・システムの DB2 V9.7 であるときに、unixODBC DriverManager を使用して、*IBM Data Server Driver for CLI and ODBC* をインストールおよび構成する方法を示します。

1. unixODBC DriverManager を solidDB ノードにインストールします (まだインストールしていない場合)。

unixODBC DriverManager は、<http://www.unixodbc.org/> からダウンロードできます。

一般に、unixODBC DriverManager のインストール・パスは `/usr/lib/libodbc.so` です。

2. ドライバーが入っている **IBM Data Server Driver for ODBC and CLI for Linux 32-bit operating systems, V9.7** のインストール・パッケージ (`ibm_data_server_driver_for_odbc_cli_32_linuxia32_v97.tar.gz`) を見つけ、それを、希望するインストール・ディレクトリー (例えば、`$HOME/solid`) にコピーします。
3. `ibm_data_server_driver_for_odbc_cli_32_linuxia32_v97.tar.gz` を解凍します。

例えば、以下のように指定します。

```
cd $HOME/solid/odbc_cli
uncompress ibm_data_server_driver_for_odbc_cli_32_linuxia32_v97.tar.gz
tar -xvf ibm_data_server_driver_for_odbc_cli_32_linuxia32_v97.tar
```

4. DB2NOEXITLIST 環境変数を ON に設定します。
`export DB2NOEXITLIST=ON`
5. unixODBC の `/etc/odbcinst.ini` 構成ファイルで、DB2 ドライバーのパスと名前を定義します。

例えば、以下のように指定します。

```
[DB2drv]
Description = DB2 ODBC Driver
Driver = /home/solid/odbc_cli/clidriver/lib/libdb2.so
FileUsage = 1
DontDLClose = 1
```

重要: Driver パスには、絶対パスを指定します。相対パスおよび環境変数は使用しないでください。

6. unixODBC の `/etc/odbc.ini` 構成ファイルで、データ・ソースを定義します。

例えば、以下のように指定します。

```
[BE_DB2]
Description = DB2 back-end database @ myhost
Driver = DB2drv
```

重要: ドライバー名 ([DB2drv] など) には、odbcinst.ini ファイルで定義した名前を指定する必要があります。

7. DB2 ドライバーの /home/solid/odbc_cli/clidriver/cfg/db2cli.ini 構成ファイルで、DB2データ・ソース・パラメーターを定義します。

例えば、以下のように指定します。

```
[BE_DB2]
Database=mydb
Protocol=TCPIP
Hostname=myhost
Port=50000
AutoCommit=0
```

8. solidDB 構成ファイル (solid.ini) の中で、unixODBC DriverManager パスと、バックエンド・データ・サーバーのデータ・ソース名を定義します。

例えば、以下のように指定します。

```
[Passthrough]
RemoteServerDriverPath=/usr/lib/libodbc.so
RemoteServerDSN=BE_DB2
```

システムのデフォルト SQL パススルー設定の構成

SQL パススルーのデフォルトの動作は、solid.ini ファイルの Passthrough セクションに含まれる構成パラメーターを使用して構成されます。

始める前に

バックエンド固有の ODBC ドライバーをまだインストールしていない場合は、solidDB フロントエンド・ノードにインストールします。詳しくは、48 ページの『SQL パススルーのためのバックエンド ODBC ドライバーのインストールと構成』を参照してください。

手順

1. **Passthrough.PassthroughEnabled** パラメーターを「yes」に設定して、SQL パススルーを有効にします (デフォルトは「no」です)。

さらに、**Passthrough.IgnoreOnDisabled** パラメーターを使用して、パススルーが無効である (**PassthroughEnabled=no**) 場合のパススルー・ステートメントの処理方法を設定します。この値が「yes」(デフォルト) の場合は、パススルーに関連するすべてのステートメント (SET PASSTHROUGH ...) が無視されます。この値が「no」の場合は、これらのステートメントを実行しようとしても、エラーが返されます。

2. デフォルトの SQL パススルー・モードを設定します。

- **Passthrough.SqlPassthroughRead** パラメーターを使用して、solidDB サーバーからバックエンドに読み取りステートメントを渡す方法を設定します。
- **Passthrough.SqlPassthroughWrite** パラメーターを使用して、solidDB サーバーからバックエンドに書き込みステートメントを渡す方法を設定します。

両方に対して、値「None」(デフォルト)、「Conditional」、および「Force」を指定できます。

ヒント: SET PASSTHROUGH または SET TRANSACTION PASSTHROUGH コマンド、または ODBC/JDBC 接続設定を使用して、デフォルトの SQL パススルー・モードをオーバーライドできます。詳しくは、58 ページの『SQL パススルー・モードの設定と変更』を参照してください。

3. オプション: ネイティブ・バックエンド・エラー・コードを solidDB エラー・コードにマップするファイルの名前と場所を定義します。
 - a. マッピング・ファイルを作成します。

マッピング・ファイル内の項目のフォーマットは、以下のとおりです。

```
<backend_error> <solidDB error> ; rest of the line is comment
```

例えば、以下のように指定します。

```
; this file maps DB2 native errors to solidDB native errors
-207 13015 ; column not found
-407 13110 ; NULL not allowed for non NULL column
; end of errormappings
```

その他のマッピング・ファイルの例については、solidDB インストール・ディレクトリーの samples/sqlpassthrough ディレクトリーを参照してください。

- b. **Passthrough.ErrorMapFileName** パラメーターで、マッピング・ファイルの名前と場所と定義します。

例えば、以下のように指定します。

```
[Passthrough]
ErrorMapFileName=myfiles/db2tosoliderrors.txt
```

ErrorMapFileName が定義されていない場合、またはエラーがマップされない場合、ネイティブ・バックエンド・エラー・コードは solidDB エラー 13456 (Passthrough backend error: SQLState=<value>, NativeError=<back-end error identifier>, MessageText=<back-end error description>) へマップされません。

4. SQL ステートメントが常にバックエンドにパススルーされるステートメントの複合レベルを定義します。
 - **Passthrough.ComplexNumTables** – 複合ステートメント内の表の最小数を指定します。ステートメントの表の数が、このパラメーターで指定された表の数より少ない場合、このステートメントは複合ステートメントではなく、バックエンドにパススルーされません。
 - **Passthrough.ComplexNumNonindexedConstr** – 複合ステートメント内の非索引 WHERE 節制約の最小数を指定します。ステートメントに含まれる非索引制約 (WHERE 節制約が索引で解決しないか、索引が存在しないか、あるいはオプティマイザーが異なる索引を制約に選択するかのいずれかのタイプ) の数がその最小数より少ない場合、そのステートメントは複合ステートメントではなく、バックエンドにパススルーされません。
 - **Passthrough.ComplexNumOrderedRows** – 複合ステートメントでソートする必要のある行の最小推定数を指定します。ステートメントに含まれるソート可能な行の数がその推定数よりも少ない場合、そのステートメントは複合ステートメントではなく、バックエンドにパススルーされません。

例

ドライバー・マネージャーを使用する Windows 32 ビット環境 (DB2):

```
[Passthrough]
RemoteServerDriverPath = C:¥WINDOWS¥system32¥odbc32.DLL

RemoteServerDSN = BE_DB2

PassthroughEnabled = yes

IgnoreOnDisabled = no

SqlPassthroughRead = Conditional

SqlPassthroughWrite = Conditional
```

バックエンドのログイン・データの手動設定

バックエンドのログイン・データは、solidDB SQL ステートメントを使用して手動で設定することができます。

このタスクについて

ほとんどの場合、InfoSphere CDC テクノロジーを使用して、バックエンドのログイン・データがフロントエンドに転送されます。solidDB からバックエンド・データ・サーバーへの最初のサブスクリプションでミラーリングまたはリフレッシュが開始されたときに、InfoSphere CDC for solidDB インスタンスは、ログイン・データをバックエンド InfoSphere CDC インスタンスからリトリーブし、そのデータを solidDB システム表 SYS_SERVER に CREATE REMOTE SERVER ステートメントと一緒に保管します。SYS_SERVER 表に保管されたパスワードは隠蔽されます。

InfoSphere CDC テクノロジーは、以下の場合にはバックエンドのログイン・データを転送しません。

- バックエンドの InfoSphere CDC インスタンスが、データベースに自動的にアクセスできるユーザー ID を使用して実行されている。このため、ログイン・データを保管する必要はありません。
- バックエンド・データ・サーバーが DB2 for z/OS または DB2 for iSeries である。
- ご使用の InfoSphere CDC for solidDB インストール済み環境およびサブスクリプションを 6.3 からアップグレードしてある。6.3 バージョンの InfoSphere CDC for solidDB は、バックエンドのログイン・データを SYS_SERVER 表に保管しません。

上記のような場合は、CREATE REMOTE SERVER (または ALTER REMOTE SERVER) ステートメントを使用して、ログイン・データを手動で定義します。

手順

• ログイン・データの作成

```
CREATE [OR REPLACE] REMOTE SERVER [USERNAME <username> PASSWORD <password>]
```

デフォルトで、ユーザー名とパスワードは大文字で格納されます。大/小文字の区別を保持するには、ユーザー名とパスワードを単一引用符で囲んで入力します。

例えば、以下のようにします。

```
CREATE REMOTE SERVER USERNAME 'Admin' PASSWORD 'Pwd123'
```

- ログイン・データの削除

```
DROP REMOTE SERVER
```

- ログイン・データの変更

```
ALTER REMOTE SERVER SET USERNAME | PASSWORD <value>
```

SQL パススルー・モードの設定と変更

デフォルトの SQL パススルー・モードは、**SqlPassthroughRead** および **SqlPassthroughWrite** パラメーターを使用して設定します。このパラメーター設定は、SET PASSTHROUGH および SET TRANSACTION PASSTHROUGH コマンドを使用することで、セッション単位またはトランザクション単位でオーバーライドできます。代わりに、ODBC 接続属性または JDBC 接続プロパティにより、パススルー・モードを接続単位で定義することもできます。

以下の 3 つの SQL パススルー・モード (レベル) があります。

- NONE: SQL パススルーは使用されません。コマンドは、フロントエンドからバックエンドに渡されません。
- CONDITIONAL: SQL パススルーは、表欠落エラーまたは構文エラーによってアクティブ化されます。
- FORCE: すべてのステートメントが、フロントエンドからバックエンドに渡されます。

SET TRANSACTION PASSTHROUGH および SET PASSTHROUGH ステートメントの場合、4 番目のオプション DEFAULT も使用できます。これは、パススルー・モードを現行セッションのデフォルトに戻します。

優先順位の階層を高いものから低いものへの順序で示すと、以下のようになります。

1. SET TRANSACTION PASSTHROUGH: トランザクション・レベルの設定
2. SET PASSTHROUGH: セッション・レベルの設定
3. ODBC 接続属性および JDBC 接続プロパティ
4. solid.ini 構成ファイル内の値によって指定されたパラメーター設定
5. solidDB のこのパラメーターのファクトリー値: **SqlPassthroughRead** および **SqlPassthroughWrite** のファクトリー値は「NONE」です。

SET TRANSACTION PASSTHROUGH コマンドを使用したトランザクション・レベルのパススルー・モードの設定

SET TRANSACTION PASSTHROUGH コマンドは、トランザクションの開始時に有効になり、トランザクションがコミットまたは異常終了するまで影響を与えます。このステートメントが、トランザクションの途中で発行された場合、エラーが返されません。

```
SET TRANSACTION PASSTHROUGH {READ <passthrough level> [WRITE <passthrough level>]}  
| {WRITE <passthrough level> | [READ <passthrough level>]}  
| <passthrough level>
```

ここで、

```
passthrough level ::= NONE | CONDITIONAL | FORCE | DEFAULT
```

SET PASSTHROUGH コマンドを使用したセッション・レベルのパススルー・モードの設定

SET PASSTHROUGH ステートメントは、次の SQL ステートメントから即座に有効になり、類似ステートメントまたは SET TRANSACTION PASSTHROUGH によって戻されるまで影響を与えます。

SET PASSTHROUGH コマンドの構文は、以下のとおりです。

```
SET PASSTHROUGH {READ <passthrough level> [WRITE <passthrough level>]}  
| {WRITE <passthrough level> | [READ <passthrough level>]}  
| <passthrough level>
```

ここで、

```
passthrough level ::= NONE | CONDITIONAL | FORCE | DEFAULT
```

ODBC または JDBC の接続レベルの設定

ODBC

SQL パススルー・モードは、以下の接続属性を使用して設定できます。

- SQL_ATTR_PASSTHROUGH_READ、値: 「NONE」、 「CONDITIONAL」、 「FORCE」
- SQL_ATTR_PASSTHROUGH_WRITE、値: 「NONE」、 「CONDITIONAL」、 「FORCE」

JDBC

SQL パススルー・モードは、以下の接続プロパティを使用して設定できます。

- プロパティ名: 「solid_passthrough_read」、値: 「NONE」、 「CONDITIONAL」、 「FORCE」
- プロパティ名: 「solid_passthrough_write」、値: 「NONE」、 「CONDITIONAL」、 「FORCE」

ADMIN COMMAND を使用したデフォルト設定の変更

SqlPassthroughRead および **SqlPassthroughWrite** パラメーターは、読み取り/書き込みタイプ (R/W) であり、パラメーター値は、ADMIN COMMAND を使用して変更することができ、変更は即座に反映されます。

```
ADMIN COMMAND 'parameter Passthrough.<parameter name>=<value>';
```

ここで、

parameter name は SqlPassthroughRead または SqlPassthroughWrite です。

value は、NONE、CONDITIONAL、または FORCE です。

SQL パススルーのトレースとモニター

solidDB には、SQL パススルーをトレースまたはモニターする手段が用意されています。

ADMIN COMMAND 'trace { on | off | } passthrough'

ADMIN COMMAND 'trace on passthrough' を実行すると、SQL パススルー接続と ODBC ドライバーのロードに関する情報がトレースされます。

- ODBC ドライバーのロード: ドライバー名とロードの状況
- バックエンドへの接続状況: 接続/再接続/切断/失敗

ADMIN COMMAND 'passthrough status'

ADMIN COMMAND 'passthrough status' を実行すると、SQL パススルー接続に関する、以下の状況情報を取得できます。

- NO REMOTE SERVER - リモート・サーバー・オブジェクトが定義されていません。
- NOT CONNECTED - 接続されていません。エラーはありません。
- CONNECTED - 接続されています。
- LOGIN FAILED - ログインに失敗しました。
- CONNECTION BROKEN - 接続に失敗しました。

例

```
ADMIN COMMAND 'passthrough status';
RC TEXT
-- ----
0  CONNECTED
```

パフォーマンス・カウンター

以下のパフォーマンス・カウンターは、SQL パススルーの接続とステートメントに関する情報を提供します。

表 12. Perfmon カウンター

Perfmon 変数	説明
Passthru open connections	バックエンドに対する SQL パススルー接続の数
Passthru open statements	バックエンドに対する準備済みステートメントの数
Passthru reads	行を返す実行済みの読み取りタイプ・ステートメント (例えば、SELECT ステートメント) の数
Passthru non reads	行を返す実行済みの書き込みタイプ・ステートメント (例えば、INSERT ステートメント) の数
Passthru commits	コミット済みステートメントの数
Passthru rollbacks	ロールバック・ステートメントの数
Passthru result cnv	バックエンドおよび solidDB のデータ型間で変換が実行されたフェッチ済み (読み取り済み) 行の数。変換は、例えば、バックエンドのデータ型が CHAR(5) で、solidDB では VARCHAR の場合に必要になります。
Passthru param cnv	ステートメント・パラメーター間で変換が実行されたステートメントの数
Passthru failures	バックエンドで準備済みにできなかったステートメントの数
Passthru reprepared	INSERT、UPDATE、および DELETE 以外の書き込みタイプのステートメントがバックエンドで実行されたため再準備されたステートメントの数。再準備は、このような場合に、表定義が変更されてしまわないようにするために必要になり、それが準備済みステートメントでのエラーの原因になります。

パフォーマンス・カウンターの使用方法の詳細については、「IBM solidDB 管理者ガイド」のセクション『solidDB のモニター (Monitoring solidDB)』を参照してください。

SQL パススルーおよび solidDB 高可用性

SQL パススルーは、solidDB 高可用性と一緒に使用することができます。

基本接続

通常の操作では、ロード・バランシングを使用しない限り、すべてのパススルー処理が 1 次サーバーで実行されます。

ロード・バランシングによる透過接続

ロード・バランシングを実行している場合 (PREFERRED_ACCESS=READ_MOSTLY)、2 次サーバーが読み取りステートメントのパススルーを実行できます。通常のトランザクション受け渡しメカニズムを使用することで、読み取り以外のステートメントは、すべて 1 次サーバーに送信されます。

サーバーのフェイルオーバー

SQL パススルーは、通常ルールに従い、ホット・スタンバイ・フェイルオーバーが存在する状態で正しく作動します。

- フェイルオーバーでは、処理中 (アクティブ) であるトランザクションは、すべて異常終了します。
- コミットされているすべてのトランザクションは、バックエンドでも正常にコミットされます。

透過的フェイルオーバーの使用

透過接続を使用する場合、セッションのパススルー・モードを接続フェイルオーバーの間保持するために、障害の透過性レベルを TF_LEVEL=SESSION に設定する必要があります。これ以外の場合 (TF_LEVEL=CONNECTION) では、セッション固有のパススルー・モードは失われ、パススルーは新規接続時のように作動します。

バックエンド・フェイルオーバー

SQL パススルーは、バックエンド・フェイルオーバーをサポートしていません。

SQL パススルーの障害の処理

障害状態では、エラーはパススルー要求時に呼び出し元のユーザーに返されます。

バックエンドの障害またはシャットダウン

バックエンド・データ・サーバーに障害が起こるか、これがシャットダウンする (すべての接続が終了する) と、次のパススルー要求は接続に失敗し、エラーがユーザーに返されます。パススルーの状況は、CONNECTION BROKEN に変更され、solidDB からのアクティブな接続はすべて閉じられます。

これをリカバリーするには、以下の手順を実行します。

1. バックエンド・データ・サーバーを再始動します。
2. パススルーするステートメントを実行します。

最初のパススルー要求は、新しい接続の確立を正常に実行し、パススルー状態は `CONNECTED` に変更されます。

バックエンドでの障害発生後に、**Passthrough.PassthroughEnabled** パラメーターを「no」に設定すると、パススルーするステートメントが実行されたときに、エラーは返されません。

6 データ・エージング

データ・エージングは、バックエンドで表の行を保持しながら、それらの行をフロントエンドからは削除する処置です。これを使用すると、solidDB フロントエンド・データベースの不必要なデータを削除する一方で、そのデータをバックエンド・データベースに保持することによって、フロントエンド・データベースのメイン・メモリーの使用量を制御することができます。

solidDB Universal Cache のデータ・エージング・ソリューションはアプリケーション主導型です。アプリケーションが、エージング対象のデータを制御して、エージングを実行、すなわちフロントエンド・データベースから不要なデータを削除します。

エージングのルールは、アプリケーションによって異なります。データ・エージングを効果的に使用するには、フロントエンド・データベースでエージングが行われたデータが、以降のバックエンド・データベースからフロントエンド・データベースへのリフレッシュまたはミラーリングのときに、フロントエンド・データベースに再び伝搬しないよう、サブスクリプションおよび表のマッピングを設計してください。

動作原理

データ・エージングは、レプリケーション時のデータ削除の処理方法を定義する solidDB SQL ステートメントで制御されます。データ・エージングは、セッションまたはトランザクションの単位で実行できます。

データ・エージングの概要

アプリケーションは、データのエージングを行う準備ができると、フロントエンドからバックエンドへの削除の伝搬を無効にする、solidDB の削除取り込みモードを呼び出します。これは solidDB SQL ステートメント `SET [TRANSACTION] DELETE CAPTURE NONE` で行われます。次のトランザクションから、データの変更 (削除) は、そのセッションまたはトランザクションで伝搬しなくなります。

データが `DELETE` ステートメントにより削除された後で、削除取り込みモードの設定は通常の状態に戻ります。そのデータがバックエンド・データベースから削除されないようにするために、フロントエンドからバックエンドへのリフレッシュは、行のエージングが行われた表について永続的にブロックされます。

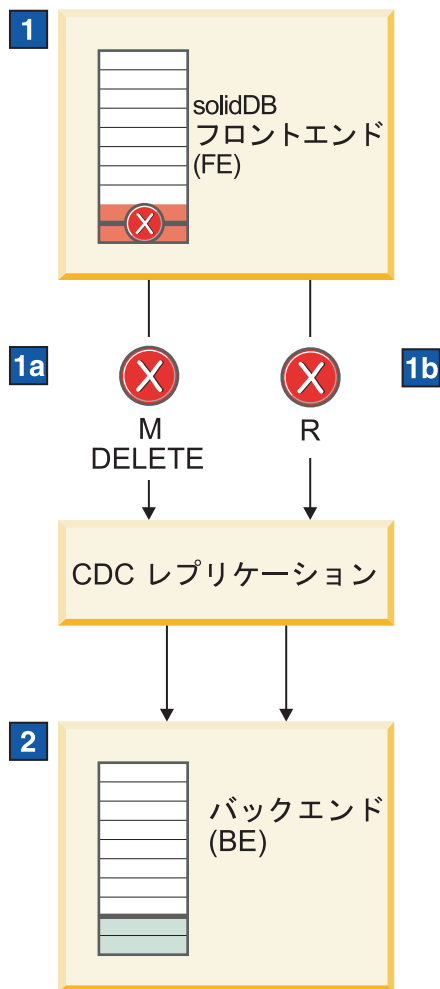


図8. データ・エージング・アーキテクチャー

1. データ・エージング・モードが `SET DELETE CAPTURE NONE` ステートメントで有効になり、データが `DELETE` ステートメントで削除されます。
 - a. `DELETE` ステートメントのミラーリングが無効にされます。
 - b. データのエージングが行われた表について、リフレッシュが永続的に無効にされます。
2. フロントエンド・データベースで削除された行はバックエンド・データ・サーバーに保持されます。

データ・エージング時の solidDB の動作

solidDB では、以下の 2 つのデータ取り込みモードを設定できます。

- `SET [TRANSACTION] DELETE CAPTURE NONE`: データ・エージング・モードをオンに設定します。次のトランザクションから、データの変更 (削除) はこのセッションまたはトランザクションで伝搬しません。
- `SET [TRANSACTION] DELETE CAPTURE CHANGES`: データ・エージング・モードをオフに設定します。次のトランザクションから、データの変更 (削除) はこのセッションまたはトランザクションで伝搬します。

セッションに対してデータ取り込みモードを設定すると、通常モードに変更するまで (DELETE CAPTURE CHANGES)、このモードが有効になります。

フロントエンドおよびバックエンドのデータベースで整合性を保持するには、データ・エージング時に、特定の solidDB ステートメントの実行を制限します。トランザクション・ロギングやデータベース・リカバリーなどの他のデータベース操作は、影響を受けません。

セッションまたはトランザクションが、DELETE CAPTURE NONE モードの場合、solidDB サーバーの動作は、以下ようになります。

- INSERT および UPDATE ステートメントは許可されません。

これは、トランザクションがこれらのステートメントをいずれか 1 つまたは両方とも含む場合、バックエンド・データベースで矛盾する結果を生成する可能性があるためです。同じ制限事項が、プロシージャおよびトリガーで実行されるステートメントにも適用されます。INSERT および UPDATE ステートメントを実行すると、エラーが発生します。

- DELETE ステートメントは許可されます。

削除された行はバックエンドに伝搬しないため、フロントエンド・データのどの部分を削除しても、バックエンド・データベースでそれが削除される危険はありません。

- DDL を除く、他のすべてのステートメントは、正常に実行されます。DDL ステートメントは許可されません。
- DELETE CAPTURE NONE モードは、それが設定された接続にのみ影響を与え、他のフロントエンド接続 (セッション) で実行されるデータベース操作には、影響を与えません。
- DELETE CAPTURE NONE モードは、通常のトランザクション・ロギング、データベース・リカバリー、および高可用性 (ホット・スタンバイ) 操作には、影響を与えません。

データ・エージングおよび solidDB の高可用性

データ・エージングが行われているときに HotStandby が失敗した場合、アクティブ・エージング・トランザクション (DELETE ステートメントによるトランザクション) は中止されます。中止されたトランザクションは再実行する必要があります。それ以外のトランザクションの動作は、接続モードに応じて異なります。

- **基本接続:** 基本接続では、アプリケーションは、新しい 1 次データベースに再接続して、セッションに対し DELETE CAPTURE NONE モードを設定する必要があります。
- **透過接続:**
 - 障害の透過性レベル (TF_LEVEL) が SESSION に設定されている場合、アプリケーションは、準備ステップなしで処理を続行することがあります。
 - 障害の透過性レベル (TF_LEVEL) が CONNECTION に設定されている場合、アプリケーションは、セッションの削除取り込みモードを設定してから、処理を続行する必要があります。
 - ロード・バランシングは、データ・エージングには効果がありません。

データ・エージング時およびその後の InfoSphere CDC の動作

solidDB がソース・データ・ストアであるサブスクリプション

エージングが行われた表のフロントエンドからバックエンドへのリフレッシュは、ブロックされます。

フロントエンドからデータを削除するが、そのデータをバックエンドで保持できるようにするために、行のエージングが行われた表について、フロントエンドからバックエンドへのリフレッシュはブロックされます。これは、サブスクリプションのリフレッシュを開始するとき、InfoSphere CDC がソース表のデータの完全なコピーをターゲット表に送信するためです。リフレッシュが許可されると、フロントエンドから削除されたデータがバックエンドでも削除されてしまいます。

注: 「Refresh」フィーチャーと、「Mirroring」の始めに実行される自動リフレッシュの両方がブロックされます。

solidDB がターゲット・データ・ストアであるサブスクリプション

バックエンドからフロントエンドへのリフレッシュおよびミラーリングは影響を受けません。これは、フロントエンドから削除されたデータは、バックエンドからフロントエンドへのリフレッシュまたはミラーリングが開始されると、フロントエンドへ返されることを意味します。

このようなバックエンドからフロントエンドへのデータの再帰を回避するには、以下のことを行います。

- サブスクリプションと表のマッピングの設計時に、フロントエンドで不要になる可能性のあるデータが、バックエンドからフロントエンドに伝搬するようなマッピングにならないようにしてください。
- このサブスクリプションでは、バックエンドからフロントエンドへのミラーリングを使用しないでください。

エージングが行われたデータをフロントエンドに戻したい場合は、バックエンドからフロントエンドへのリフレッシュを実行する必要があります。これによって、エージングが行われた表について、フロントエンドからバックエンドへのリフレッシュのブロッキングも解除されます。

データ・エージングの使用

solidDB フロントエンドでデータのエージングを行うには、最初に、SQL ステートメントを使用して、データ取り込みモードを、削除がバックエンドまで伝搬されない状態に設定し、次に、古くなったデータまたは不要なデータを持つ行を削除します。データの削除が完了したら、solidDB で、データの伝搬が常に許可されるようにデータ取り込みモードを変更できます。

始める前に

データ・エージングを使用するには、solidDB Universal Cache を稼働中にする必要があります。

バックエンドからフロントエンドへのサブスクリプションが存在しないこと、あるいはエージングが行われたデータがバックエンドからフロントエンドに返されるリスクがないように、サブスクリプションと表のマッピングが設計されていることを前提とします。

手順

1. データ取り込みモードを **DELETE CAPTURE NONE** に設定します。

- セッションに対してデータ・エージングを有効にするには、以下のコマンドを発行します。

```
SET DELETE CAPTURE NONE
```

- 次のトランザクションに対してデータ・エージングを有効にするには、以下のコマンドを発行します。

```
SET TRANSACTION DELETE CAPTURE NONE
```

2. 不要なデータを含む行をドロップ (削除) します。

3. データ取り込みモードを **DELETE CAPTURE CHANGES** に設定して、データ・エージングを無効にします。

- セッションに対してデータ・エージングを無効にするには、以下のコマンドを発行します。

```
SET DELETE CAPTURE CHANGES
```

- `SET TRANSACTION DELETE CAPTURE NONE` ステートメントを使用した場合、トランザクションのコミット後に、モードは `DELETE CAPTURE CHANGES` に変更されます。

次のタスク

自動化されたデータ・エージング・スクリプトの作成については、73 ページの『エージング・ストアード・プロシージャの使用』を参照してください。

7 ツールおよびユーティリティー

solidDB サーバーおよび InfoSphere CDC for solidDB インストール・パッケージには、InfoSphere CDC レプリケーション・テクノロジーのセットアップおよび使用において、共通タスクの自動化およびスクリプト記述を行うためのツールおよびユーティリティーが含まれています。例えば、これらのツールおよびユーティリティーを使用して、InfoSphere CDC インスタンスおよびサブスクリプションのスクリプトを作成したり、データ・エージングおよびリフレッシュを自動化したりすることができます。

ツールはサンプルのアプリケーション、スクリプト、およびストアード・プロシージャのセットとして提供され、オペレーティング・システムのコマンド・プロンプトから使用できます。

InfoSphere CDC for solidDB パッケージ - ツールおよびユーティリティー

表 13. InfoSphere CDC for solidDB パッケージ - ツールおよびユーティリティー

ツールまたはユーティリティー	用途	場所
インスタンスおよびサブスクリプション管理ツール <ul style="list-style-type: none"> • dminstancemanager • dmsubscriptionmanager 	InfoSphere CDC インスタンスおよびサブスクリプションの管理を容易にするコマンド行ユーティリティー	<InfoSphere CDCfor solidDB インストール・ディレクトリー>%samples%ucutils
ucautomation 自動化フレームワーク (Perl)	データ・ストア、サブスクリプション、およびマッピングの作成など、インストールおよび構成タスクを自動化するための Perl ベースのサンプル・スクリプトおよびライブラリー・モジュール	<InfoSphere CDCfor solidDB インストール・ディレクトリー>%samples%ucautomation
ucdeploy サンプル・アプリケーション	スクリプトを使用して InfoSphere CDC レプリケーションをセットアップする方法を示すサンプル・アプリケーション。	<InfoSphere CDCfor solidDB インストール・ディレクトリー>%samples%ucdeploy
uchsbmonitor サンプル・スクリプト	高可用性 (HotStandby) 構成など、サブスクリプションのセットをモニターおよび再開するためのサンプル・スクリプト	<InfoSphere CDCfor solidDB インストール・ディレクトリー>%samples%uchsbmonitor
ucpassthrough サンプル・アプリケーション	SQL パススルー機能を使用して InfoSphere CDC レプリケーションをセットアップする方法を示すサンプル・アプリケーション	<InfoSphere CDCfor solidDB インストール・ディレクトリー>%samples%ucpassthrough

solidDB パッケージ - ストアード・プロシージャ

表 14. solidDB パッケージ - ストアード・プロシージャ

ストアード・プロシージャ	用途	場所
データ・エージング • create_automatic_aging.sql • start_automatic_aging.sql • stop_automatic_aging.sql	ユーザー定義のエージング・ルールに基づいて、solidDB データベース内のデータのエージングを自動化する	<solidDBインストール・ディレクトリ>%procedures
リフレッシュ • create_refresh_package.sql	InfoSphere CDC インスタンスとの対話なしに、アプリケーションからプログラムでリフレッシュを開始できるようにする	<solidDBインストール・ディレクトリ>%procedures

Perl 自動化フレームワーク

Perl 自動化フレームワークによって、Linux および Windows 環境でインストーラ、構成、およびサブスクリプション処理タスクの自動化を行うための、Perl ベースのさまざまなサンプル・スクリプトおよびライブラリー・モジュールを使用することができます。

Perl 自動化フレームワークは、<InfoSphere CDC for solidDB installation directory>%samples%uautomation ディレクトリにあります。

- **include** – Perl モジュール
- **perldoc** – 各ライブラリー・モジュールにある Perl 文書 (POD)
- **samples** – 2 つの solidDB インスタンス間、および solidDB と、DB2 for Linux、UNIX、および Windows インスタンス間のサブスクリプションを作成および制御するためのサンプル・スクリプト

このフレームワークを使用して、以下の自動化を行うことができます。

- solidDB Universal Cache のコンポーネントのインストール、または InfoSphere CDC テクノロジーを使用した solidDB と solidDB の間のレプリケーション
- さまざまなタイプ (solidDB、Informix、DB2 for Linux、UNIX、および Windows) のデータベースの作成、およびそれらに対する SQL ステートメントの実行
- サポートされるすべてのデータベース・タイプに応じた、InfoSphere CDC インスタンスの作成/開始
- データ・ストア、サブスクリプション、およびマッピングの作成と、ミラーリングの開始
- 作成済みの任意のコンポーネントの削除による環境のクリーンアップ

自動化フレームワークを使用するには、以下が必要です。

- Linux または Windows 環境
- Perl の稼働中のインストール済み環境 (<http://www.perl.com> で入手可能)

環境のセットアップ方法、および自動化フレームワークの使用方法について詳しくは、<InfoSphere CDC for solidDB インストール・ディレクトリ>%samples%uautomation ディレクトリにある readme ファイルを参照してください。

インスタンスおよびサブスクリプション管理ツール

dminstancemanager および **dmsubscriptionmanager** ツールを使用して、InfoSphere CDC インスタンスおよびサブスクリプションの作成、除去、および変更のスクリプトを記述することができます。

これらのツールは、<InfoSphere CDCfor solidDB installation directory>%samples%ucutils ディレクトリーにあります。

- **dminstancemanager** – InfoSphere CDC インスタンスの作成、除去、変更、および状況の照会を行います
- **dmsubscriptionmanager** – サブスクリプションの作成および除去、または既存のサブスクリプションへの表マッピングの追加を行います
- **ucenv** – ユーティリティーを使用するための環境を構成します

dminstancemanager および **dmsubscriptionmanager** ツールは、InfoSphere CDC for solidDB とだけでなく、どの InfoSphere CDC エンジンとも使用することができます。 **ucenv** スクリプトは、**dminstancemanager** が使用する InfoSphere CDC エンジンの定義に使用されます。

dminstancemanager および **dmsubscriptionmanager** ユーティリティーを使用するための環境のセットアップ方法、およびそれらの使用方法について詳しくは、<InfoSphere CDC for solidDBインストール・ディレクトリー>%samples%ucutils ディレクトリーにある readme.txt を参照してください。

ucdeploy および **ucpassthrough** のサンプル・アプリケーションも参照してください。これらは、<InfoSphere CDCfor solidDB installation directory>%samples ディレクトリーにあります。これらのサンプルは、**dminstancemanager** および **dmsubscriptionmanager** ツールを使用して、InfoSphere CDC インスタンスとレプリケーション・サブスクリプションを作成します。

ucdeploy – 構成およびセットアップ・サンプル

ucdeploy サンプルは、2 つの solidDB データベース (フロントエンドおよびバックエンド)、対応する InfoSphere CDC インスタンスとデータ・ストア、およびそれらの間のサブスクリプションを作成します。次に、サンプルはサブスクリプションでのミラーリングを開始し、フロントエンド・データベースからバックエンド・データベースにデータが複製される方法が示されます。

ucdeploy サンプルは **dminstancemanager** ユーティリティーを使用して、フロントエンドおよびバックエンドのインスタンスと、**dmsubscriptionmanager** ユーティリティーを作成し、サブスクリプションを作成します。このサンプルは、標準 InfoSphere CDC dm コマンドも使用して、サブスクリプションでのミラーリングの開始などを行います。

ucdeploy サンプルは、<InfoSphere CDC for solidDB installation directory>%samples%ucdeploy ディレクトリーにあります。

サンプルの使用方法について詳しくは、上記のディレクトリーにある readme.txt を参照してください。

ucpassthrough – SQL パススルー・セットアップ・サンプル

ucpassthrough サンプルは、2 つの solidDB データベース (フロントエンドおよびバックエンド)、対応する InfoSphere CDC インスタンスとデータ・ストア、およびそれらの間のサブスクリプションを作成します。次に、このサンプルは SQL パススルー機能を使用して、バックエンド・データベースにデータを挿入します。

ucpassthrough サンプルは **dminstancemanager** ユーティリティを使用して、フロントエンドおよびバックエンドのインスタンスと、**dmsubscriptionmanager** ユーティリティを作成し、サブスクリプションを作成します。このサンプルは、標準 InfoSphere CDC dm コマンドも使用して、インスタンスの開始などを行います。solidDB SQL ステートメントは、バックエンド・データベースにデータを挿入し、それを読み取るパススルー・ステートメントに使用されます。

ucpassthrough サンプル・スクリプトは、<InfoSphere CDCfor solidDB installation directory>%samples%ucpassthrough ディレクトリーにあります。サンプルの使用方法について詳しくは、上記のディレクトリーにある readme.txt を参照してください。

uchsbmonitor – HSB サブスクリプション・モニター・サンプル

uchsbmonitor サンプルは Perl スクリプトであり、高可用性セットアップでサブスクリプションをモニターし、フェイルオーバーまたは切り替えイベントによってサブスクリプションが停止した場合に、サブスクリプション上でミラーリングを再開します。

例えば、ターゲット・データ・ストアである solidDB 1 次サーバーに障害が起これると、サブスクリプション上でのレプリケーションは終了します。リカバリーするには、サブスクリプション上のレプリケーションを再開する必要があります。

uchsbmonitor サンプル・スクリプト `hsbmonitor.pl` は、<InfoSphere CDC for solidDB installation directory>%samples%uchsbmonitor ディレクトリーにあります。

このスクリプトを実行する構文は、以下のとおりです。

```
perl hsbmonitor.pl -s src -t tgt <subscription_name>
```

ここで、

- `src` – ソース・インスタンスの名前
- `tgt` – ターゲット・インスタンスの名前
- `<subscription_name>` – モニター対象のサブスクリプションの名前

データ・エージングとリフレッシュのための SQL ストアード・プロシージャ

solidDB パッケージに含まれているストアード・プロシージャによって、データ・エージングとリフレッシュを自動的に行うことができます。

エージング・プロシージャーは、ユーザー定義のエージング・ルールに基づいて、solidDB データベースの行を削除します。エージング・プロシージャーを solidDB 始動時にアクティブ化して、バックグラウンドで自動データ・エージングを実行することができます。

リフレッシュ・ストアード・プロシージャーを使用すると、InfoSphere CDC インスタンスと対話しなくても、アプリケーションからプログラムでリフレッシュを開始できます。

エージング・ストアード・プロシージャーの使用

エージング・プロシージャー `SQL_START_AUTOMATIC_AGING` は、solidDB SQL ストアード・プロシージャーであり、solidDB データベースでユーザー定義の `DELETE` ステートメントを実行します。ユーザーは、表 `AUX_AUTOMATIC_DELETES` に維持される `DELETE` ステートメントの形式でエージング・ルールを定義します。`AUX_AUTOMATIC_DELETES` 表は、このプロシージャーによって自動的に作成されます。

エージング・ルール

通常の SQL ステートメントを使用して、`AUX_AUTOMATIC_DELETES` のエージング・ルールを作成および変更します。ルールは、実行時に削除、追加、または変更することができます。

表 15. `AUX_AUTOMATIC_DELETES` 表の定義

列	データ型	説明
id	INTEGER PRIMARY KEY	エージング・ルール用の ID
statement	LONG VARCHAR NOT NULL	この値には、完全な <code>DELETE</code> ステートメントを指定する必要があります。それ以外のステートメントを指定すると、プロシージャーはすべて失敗します。 1 行につき 1 つのステートメントのみを指定できます。
exec_period	INTEGER NOT NULL	エージング間隔を秒単位で定義します。
next_exec_date	TIMESTAMP	次にルールを実行する時間を定義します。 プロシージャーは、 <code>exec_period</code> の値を現在の実行時間に加算して、値を計算します。 ユーザーがルールの作成時にこの値を指定する場合、最初の削除操作は指定した時間に実行されます。 この値を指定しない場合は、このステートメントが次の使用可能な機会に実行されます。

いずれのタイプの DELETE ステートメントも、エージング・ルールとして使用することができます。AUX_AUTOMATIC_DELETES 表の各行は、単一のルールに対応します。表に複数のルール挿入して、それぞれのルールを独自の頻度で実行することもできます。

ルールの表記は、アプリケーション設計に応じて異なります。以下に 2 つの例を示します。

- **例 1: エージング・ルールは、エージング状態に関する情報を収める列に基づきます。**

「table_1」という名前の表で、エージングを行う行を列「state」の値「DONE」によって識別できる場合、ルール・ステートメントは、以下のようになります。

```
DELETE FROM table_1 WHERE state='DONE';
```

- **例 2: エージング・ルールは日付に基づきます。**

「table_2」という名前の表で、日付が現在の日付よりも古いすべての行に対してエージングを行うことができる場合、ルール・ステートメントは、以下のようになります。

```
DELETE FROM table_2 WHERE DATE<CURDATE();
```

プロシージャのライフ・サイクル

このプロシージャには、パラメーターがありません。これは内部ループで実行され、各反復においてルールを読み取り、適用可能なルールを実行してから、exec_period の値 (秒単位) を現在の実行時間に加算することで、次のルールの実行時間を計算して更新します。デフォルトでは、プロシージャは各反復間に 1 秒スリープします。スリープ間隔は、プロシージャのコードを編集することで、変更することができます。

このプロシージャは、通常、バックグラウンド・ジョブとして実行します。終了メカニズムは、プロシージャが作成する AUX_AUTOMATIC_DELETES_BREAK という表に基づきます。内部ループの各反復で、プロシージャは AUX_AUTOMATIC_DELETES_BREAK 表に行が存在するかどうかを検査します。表に少なくとも 1 つの行が存在する場合、プロシージャは終了します。次の開始時に、プロシージャは AUX_AUTOMATIC_DELETES_BREAK 表からすべての行を削除します。

表 16. AUX_AUTOMATIC_DELETES_BREAK 表の定義

列	データ型	説明
break	INTEGER	行が存在する場合、エージング・プロシージャを終了します。

エージング・プロシージャを作成および実行するスクリプト

solidDB パッケージには、ストアード・プロシージャを作成および実行するための SQL スクリプトが含まれています。このスクリプトは、solidDB のインストール・ディレクトリ下の procedures ディレクトリに格納されています。

表 17. エージング・プロシージャを作成および実行するスクリプト

スクリプト	使用法
create_automatic_aging.sql	ストアード・プロシージャを作成します。
start_automatic_aging.sql	ストアード・プロシージャを呼び出します。
stop_automatic_aging.sql	ストアード・プロシージャを停止します。

エージング・プロシージャの作成

エージング・プロシージャを作成するには、以下の手順を実行します。

1. フロントエンドでエージングを行う表に関連する、バックエンドからフロントエンドへのサブスクリプションが存在する場合、そのサブスクリプションを削除または停止します。

あるいは、データベースを設計するときに、InfoSphere CDC 行フィルターを使用して、エージングを行ったデータの再帰を回避できるようにします。例については、79 ページの『例: 双方向サブスクリプションのためのデータ・エージングの自動化』を参照してください。

2. スクリプト create_automatic_aging.sql を実行して、プロシージャを作成します。

例えば、solsql を使用して、以下のスクリプトを実行できます。

```
solsql -f "C:\%solidDB%\procedures\create_automatic_aging.sql" "tcp 2315" dba dba
```

エージング・プロシージャの開始および実行

プロシージャを作成したら、プロシージャを開始してエージング・ルールを定義する必要があります。エージング・ルールは、実行時に変更することもできます。

1. エージング・プロシージャを開始します。
 - スクリプト start_automatic_aging.sql を実行します。

これにより、エージング・プロシージャがバックグラウンドで開始されます。

または

- -x executeandnoexit コマンド行オプションを使用して、solidDB の始動時に start_automatic_aging.sql スクリプトを組み込みます。

```
solid -x executeandnoexit:start_automatic_aging.sql
```

2. AUX_AUTOMATIC_DELETES 表にデータを設定することで、エージング・ルールを定義します。

例えば、5 秒ごとに「state」列の値に基づいて表「table_1」のデータのエージングを行うには、以下のコマンドを発行します。

```
INSERT INTO aux_automatic_deletes (id, statement, exec_period) values
(1, 'DELETE FROM table_1 WHERE state='DONE'', 5);
COMMIT WORK;
```


エージング・プロシージャの停止

エージング・プロシージャは、以下の方法で停止できます。

- スクリプト `stop_automatic_aging.sql` を実行します。
- 以下のコマンドを発行することで、`AUX_AUTOMATIC_DELETES_BREAK` 表に行を追加します。

```
INSERT INTO aux_automatic_deletes_break (1);  
COMMIT WORK;
```

- `ADMIN COMMAND 'backgroundjob'` コマンドを使用して、プロシージャを制御します。

リフレッシュ・ストアード・プロシージャの使用

リフレッシュ・プロシージャ `TS_REFRESH_CDC_SUBSCRIPTION` は、`solidDB SQL` ストアード・プロシージャであり、サブスクリプション名に基づいて、サブスクリプションのリフレッシュを開始します。

`solidDB` 接続からリフレッシュを開始できるようにするには、リフレッシュ・プロシージャを開始する前に、`InfoSphere CDC for solidDB` のコマンド `dmsetaccessserverparams` を使用して、`Access Server` のログイン・データを設定する必要があります。

このプロシージャが呼び出されると、サブスクリプションの存在とリフレッシュ状況を検査します。

- リフレッシュを開始できる場合、プロシージャ呼び出しは、リフレッシュが完了するまでブロックされます。リフレッシュするデータのサイズによっては、呼び出しが長時間ブロックされる場合があります。

呼び出しが戻らない場合は、通常のタイムアウトが適用されます。

- リフレッシュを開始できない場合、エラーが返されます。

リフレッシュの状態は、`TS_REFRESH` という表に維持されます。この表は、インスタンスが作成されるときに、その `InfoSphere CDC for solidDB` によって自動的に作成されます。プロシージャを開始すると、状況は「1」（リフレッシュの進行中）に変わります。リフレッシュが完了すると、`InfoSphere CDC for solidDB` は状態を「2」（リフレッシュが完了しました）に更新します。リフレッシュが失敗した場合、`InfoSphere CDC for solidDB` は、表のエラーを報告します。

表 18. `TS_REFRESH` 表の定義

列	データ型	説明
<code>subscription_name</code>	<code>VARCHAR (20) PRIMARY KEY</code>	サブスクリプション名
<code>state</code>	<code>INTEGER NOT NULL</code>	リフレッシュの状態 • -1 — エラー • 0 — リフレッシュが要求されました • 1 — リフレッシュの進行中 • 2 — リフレッシュが完了しました

表 18. TS_REFRESH 表の定義 (続き)

列	データ型	説明
error_description	VARCHAR(255)	<p>エラーの説明</p> <ul style="list-style-type: none"> • Access Server のパラメーターのロードに関する問題 • Access Server のユーザー名が設定されていません • Access Server のパスワードが設定されていません • Access Server のホスト・アドレスが設定されていません • Access Server のポート番号が設定されていません • Access Server への接続の確立に関するエラー • Access Server への接続エラー • Access Server への接続が存在しません • バブリッシャーの取得に失敗しました • 一致するサブスクリプションの検出に失敗しました • サブスクリプションが存在しません • リフレッシュのポーリング・エラー
inserts_performed	BIG INT	<p>リフレッシュ時にコミットされた挿入の行数</p> <p>コミットごとの挿入の数は、InfoSphere CDC システム・パラメーター 152 ページの『refresh_commit_after_max_operations』で設定された値によって異なります。</p> <p>デフォルト値は 0 です。</p>

リフレッシュ・プロシージャの作成

solidDB パッケージには、ストアード・プロシージャを作成するための SQL スクリプトが含まれています。このスクリプトは、solidDB のインストール・ディレクトリ下の procedures ディレクトリに格納されています。

スクリプト	使用法
create_refresh_package.sql	ストアード・プロシージャを作成します。

リフレッシュ・プロシージャを作成するには、以下の手順を実行します。

1. サブスクリプションと、フロントエンドおよびバックエンドのデータ・サーバーが作成されており、InfoSphere CDC コンポーネントが正常に稼働していることを確認します。
2. スクリプト create_refresh_package.sql を実行して、リフレッシュ・プロシージャを作成します。

以下の例で示すように **solsql** を使用して、スクリプトを実行できます。

```
solsql -f "C:\solidDB\procedures\create_refresh_package.sql" "tcp 2315" dba dba
```

3. InfoSphere CDC for solidDB のコマンド dmsetaccessserverpararms を使用して、Access Server 用のログイン・データを定義します。

dmsetaccessserverparams コマンドの構文は、以下のとおりです。

```
dmsetaccessserverparams [-u <username>] [-p <password>] [-H <hostname>] [-P <port>]
```

例えば、以下のように指定します。

```
dmsetaccessserverparams -u dba -p dba - H 192.167.3.3 -P 10101
```

リフレッシュ・プロシージャの実行

リフレッシュ・プロシージャを実行するには、以下の手順を実行します。

1. フロントエンドからバックエンドへのサブスクリプションで、進行中のミラーリングが存在することを確認します。
2. 以下の構文を使用して、リフレッシュ・プロシージャを呼び出します。

```
CALL ts_refresh_cdc_subscription ('subscription_name');
```

例えば、以下のように指定します。

```
CALL ts_refresh_cdc_subscription ('current_invoices');
```

リフレッシュの状況のモニター

リフレッシュの進行状況は、リフレッシュの状態およびリフレッシュされた行数 (inserts_performed) について TS_REFRESH 表を表示することによって確認できます。

例えば、以下のように指定します。

```
SELECT * from TS_REFRESH;
```

SUBSCRIPTION_NAME	STATE	ERROR_DESCRIPTION	INSERTS_PERFORMED
current_invoices	1		2000

1 rows fetched.

ストアド・プロシージャの停止

プロシージャ呼び出しは、リフレッシュが正常に実行されるまでブロックされます。プロシージャを停止する場合は、ADMIN COMMAND 'throwout' を使用して強制終了します。

通常のタイムアウトとして、以下が適用されます。

- 照会タイムアウトが設定されている場合、呼び出しは照会タイムアウトでタイムアウトになります。デフォルトでは、タイムアウトはありません。

例えば、以下のように指定します。

- ODBC では、ODBC ステートメントの属性 SQL_ATTR_QUERY_TIMEOUT を使用して照会タイムアウトを設定します (秒単位)。
- JDBC では、ステートメントのメソッド setQueryTimeout() を使用して照会タイムアウトを設定します (秒単位)。
- 接続タイムアウトが設定されている場合は、タイムアウトの満了時に接続が失われます。

タイムアウト動作の詳細については、「*IBM solidDB プログラマー・ガイド*」の付録『タイムアウト制御』を参照してください。

例: 双方向サブスクリプションのためのデータ・エージングの自動化

この例では、エージング・プロシージャを InfoSphere CDC 行フィルターとともに使用して、双方向サブスクリプション・セットアップでデータ・エージングを自動化する方法について説明します。

セットアップに双方向サブスクリプションが含まれる場合は、アプリケーションとサブスクリプションを設計する際に、フロントエンドから削除される (エージングが行われる) 行が、バックエンドからフロントエンドへのリフレッシュまたはミラーリングの使用時にフロントエンドに返されないようにする必要があります。

可能な方法の 1 つとして、エージング・プロシージャを使用して、フロントエンドのデータを削除すると同時に、InfoSphere CDC 行フィルターをセットアップして、エージングが行われた行のフロントエンドへのレプリケーションを回避します。

この例では、アプリケーションは、エージング対象にできるデータを制御するために、そのデータのエージング状況に関する情報を維持し、削除される行にはフラグを立てます。データの実際の削除処理は、エージング・プロシージャを使用して実行されます。次に、削除フラグが立てられた行がバックエンドからフロントエンドに複製されないように、InfoSphere CDC 行フィルターがセットアップされます。

双方向サブスクリプションによるデータ・エージングの設定例

注: この例では、表に新しい列を追加できることを前提とします。これは必須ではありません。データベース設計によっては、既存の列を使用して、エージングが行われた行を特定することができます。

1. データ・エージングをサポートするために、環境をセットアップします。
 - 値「0」(エージングが行われていない) または「1」(エージングが行われた) を収める列「aged」を追加します。
 - アプリケーションの設計では、エージングが行われる行の「aged」列に値「1」が設定されるようにします。
2. **solidDB** でエージング・プロシージャを作成して開始します。

詳しくは、73 ページの『エージング・ストアド・プロシージャの使用』を参照してください。

3. フロントエンドからバックエンド、およびその逆方向にサブスクリプションをセットアップします。
4. バックエンドからフロントエンドへのサブスクリプションでは、行フィルターをセットアップします。

「aged」列で値が 1 より小さい行のみを複製するという行フィルター・ルールを作成します。

フィルターの設定方法については、「InfoSphere Change Data Capture Management Console 管理ガイド」のセクション『行と列のフィルタリング』を参照してください。

5. **AUX_AUTOMATIC_DELETES** に **DELETE** ステートメントを追加して、エージング・ルールを作成します。

例えば、table_1 で削除フラグが立てられたすべての行を削除するというルールを作成する場合は、以下の INSERT ステートメントを実行します。

```
INSERT INTO aux_automatic_deletes (id, statement, exec_period) values
(1, 'DELETE FROM table_1 WHERE aged=1', 10);
```

結果

アプリケーションが実行中の場合、データベースの所定の行にエージング・フラグ (「aged」=1) が立てられます。それらの行は、変更されたエージング状態とともにバックエンドに複製されます。エージング・プロシージャが実行中の場合、フラグが立てられた行がフロントエンド表から削除されます。これらの行は、行フィルター (「aged」< 1) によって回避されるため、バックエンドからフロントエンドに複製されません。

バックエンドからフロントエンドへのデータの複製

バックエンドからフロントエンドへのすべてのレプリケーション形式を使用できます。つまり、連続ミラーリングも、InfoSphere CDC ツールから開始するリフレッシュも、また「リフレッシュ・プロシージャ」を使用してアプリケーションから開始するリフレッシュも実行できます。

ただし、InfoSphere CDC レプリケーションは実際には非同期であるため (フロントエンドでの変更が即座にバックエンドで有効になりません。また、その逆も同様です)、この例では、以下の制限事項が適用されます。

- エージングを行う行がバックエンドで変更中の場合、バックエンドからフロントエンドへのミラーリングは、エージング・アクティビティが完了するまで許可されません。
- リフレッシュ・プロシージャを使用する場合は、リフレッシュを実行する前に「aging」列の値がバックエンドに複製されていることを確認します。レプリケーションが完了する前にリフレッシュを実行すると、フロントエンドでエージングを行ったデータがバックエンドに返される場合があります。

リフレッシュの進行状況は、TS_REFRESH 表内のリフレッシュされた行の数を表示することによって確認できます。

例えば、以下のように指定します。

```
SELECT * from TS_REFRESH;
```

SUBSCRIPTION_NAME	STATE	ERROR_DESCRIPTION	INSERTS_PERFORMED
current_invoices	1		2000

1 rows fetched.

8 solidDB Universal Cache の障害シナリオ

以下のセクションでは、さまざまな障害シナリオの概要を示し、必要なリカバリー手順があればそれについても説明します。

ヒント: リカバリー手順に手動操作が含まれている場合、その操作は、スクリプトを使用するか、または InfoSphere CDC で使用可能なコマンドを使用することで自動化できる場合がよくあります。

スタンドアロン solidDB サーバーの障害

スタンドアロン solidDB サーバーに障害が発生すると、サブスクリプションのレプリケーションも終了します。リカバリーするには、以下の手順を実行します。

手順

1. solidDB サーバーを手動で再始動し、データベースをリカバリーします。

詳しくは、「*IBM solidDB 管理者ガイド*」の『*solidDB の管理*』セクションを参照してください。

2. InfoSphere CDC インスタンスを再開します。

詳しくは、102 ページの『InfoSphere CDC の開始と停止』のセクションを参照してください。

3. サブスクリプションのレプリケーションを再開します。

詳しくは、「*InfoSphere Change Data Capture Management Console 管理ガイド*」の『サブスクリプションでのレプリケーションの開始と終了』のセクションを参照してください。

タスクの結果

再開すると、サブスクリプションのレプリケーションが再開し、データベースが再同期されます。レプリケーションが正常に続行します。

InfoSphere CDC インスタンスの障害

InfoSphere CDC インスタンスに障害が発生すると、サブスクリプションのレプリケーションも終了します。リカバリーするには、以下の手順を実行します。

1. InfoSphere CDC インスタンスを再開します。

詳しくは、102 ページの『InfoSphere CDC の開始と停止』のセクションを参照してください。

2. サブスクリプションのレプリケーションを再開します。

詳しくは、「*InfoSphere Change Data Capture Management Console 管理ガイド*」の『サブスクリプションでのレプリケーションの開始と終了』のセクションを参照してください。

結果

再開すると、サブスクリプションのレプリケーションが再開し、データベースが再同期されます。レプリケーションが正常に続行します。

この障害が発生すると、solidDB サーバーは **LogReader.MaxLogSize** パラメーターで指定された限度に達するまで、トランザクションの処理を続行します。

HA モード (HotStandby) の solidDB サーバーの障害

以下のセクションでは、solidDB HotStandby 構成での障害シナリオを説明します。

1 次 solidDB サーバーの障害

1 次 solidDB サーバーに障害が発生した場合は、高可用性コントローラー (HAC) などの高可用性マネージャーが、標準的なプロシージャとして 2 次 solidDB サーバーへのフェイルオーバーを実行します。2-Safe プロトコルが使用されている場合、データベースとログの状態が完全に保持されます。アプリケーションが認識するフェイルオーバー時間は、1 秒未満です。

- solidDB がソース・データ・ストア (データがフロントエンドからバックエンドにのみ複製される書き込み専用キャッシュ) である場合、InfoSphere CDC インスタンスは新しい 1 次側に自動的に再接続し、レプリケーションは続行されます。
- solidDB がターゲット・データ・ストア (読み取り専用キャッシュまたは読み取り/書き込みキャッシュ) である場合、サブスクリプションのレプリケーションは終了します。Management Console または InfoSphere CDC コマンド `dmstartmirror` を使用して、サブスクリプションを再開する必要があります。

詳しくは、「*InfoSphere Change Data Capture Management Console 管理ガイド*」の『サブスクリプションでのレプリケーションの開始と終了』のセクションを参照してください。

上記のシナリオ中、InfoSphere CDC インスタンスは常に稼働状態にあります。

ヒント: 高可用性 (HotStandby) 機能と高可用性コントローラー (HAC) について詳しくは、「*IBM solidDB 高可用性ユーザー・ガイド*」を参照してください。

2 次 solidDB フロントエンドの障害

2 次フロントエンドの障害の場合、手操作による介入は不要です。

2 次フロントエンドに障害が発生した場合は、2 次フロントエンド・ノードが、インストール固有の通常の方法でリカバリーされます (例えば、自動的にリポートするなど)。HAC が残りのリカバリーを自動的に行います。障害は、アプリケーションや InfoSphere CDC インスタンスには認識されません。

1 次 solidDB サーバーと InfoSphere CDC for solidDB インスタンス間の通信リンクの障害

1 次 solidDB サーバーと InfoSphere CDC for solidDB インスタンス間の通信リンクに障害が発生すると、サブスクリプションのレプリケーションも終了します。ただし、リンクだけの障害の可能性は低いと考えられます。

リカバリーするには、以下の手順を実行します。

1. InfoSphere CDC インスタンスを再開します。

詳しくは、102 ページの『InfoSphere CDC の開始と停止』のセクションを参照してください。

2. サブスクリプションのレプリケーションを再開します。

詳しくは、「InfoSphere Change Data Capture Management Console 管理ガイド」の『サブスクリプションでのレプリケーションの開始と終了』のセクションを参照してください。

結果

再開すると、サブスクリプションのレプリケーションが再開し、データベースが再同期されます。レプリケーションが正常に続行します。

この障害が発生すると、solidDB サーバーは **LogReader.MaxLogSize** パラメーターで指定された限度に達するまで、トランザクションの処理を続行します。

バックエンド・サーバーまたはバックエンド・ノードの障害

バックエンド・サーバーまたはバックエンド・ノードに障害が発生すると、サブスクリプションのレプリケーションも終了します。リカバリーするには、以下の手順を実行します。

手順

1. バックエンド・サーバーを再始動し、データベースをリカバリーします。
2. InfoSphere CDC インスタンスを再開します。
3. サブスクリプションのミラーリング (レプリケーション) を再開します。

注: バックエンド固有のツールやプロシージャを使用して上記の手順を自動化することが可能な場合があります。

タスクの結果

再始動すると、レプリケーションが再開し、データベースが再同期されます。レプリケーションが正常に続行します。

この障害が発生すると、solidDB フロントエンドは **LogReader.MaxLogSize** パラメーターで指定された限度に達するまで、トランザクションの処理を続行します。

バックエンド 1 次サーバーの障害

バックエンド 1 次サーバーの障害時、またはバックエンド・ノード全体の障害時には、該当するバックエンド製品のルールとツールに従ってリカバリーを処理する必要があります。solidDB サーバーはこの状態を修正するための方法を提供していません。

バックエンド・サーバーが新しい 1 次サーバーとして稼働すると、InfoSphere CDC インスタンスとまったく同じコピーが残存していたノードで再開されます。InfoSphere CDC ツールを使用してサブスクリプションを再構成し、該当する InfoSphere CDC インスタンスを再接続する必要があります。ミラーリングを開始するには、新しいサブスクリプションを（両方向での）フル・リフレッシュから続行する必要があります。

場合によっては、サブスクリプション・レプリケーションの状態が失われ、フル・リフレッシュが必要なこともあります。

9 トラブルシューティング

このセクションでは、solidDB Universal Cache を構成または使用する際の一般的な問題を防止またはトラブルシューティングする方法について説明し、ガイドラインを示します。

- 『初期接続が成功しない』
- 『レプリケーションで使用するコンポーネント間の従属関係』
- 『レプリケーション・サブスクリプションの変更』
- 86 ページの『hsb netcopy に続けて切り替えを実行するとサブスクリプションが失敗する』

初期接続が成功しない

solidDB Universal Cache のコンポーネントのインストールと構成は、『インストールおよび構成手順の概要』のセクションで説明されている順序で行う必要があります。以下の手順を検討し、必ず、これらのインストールと構成の手順に従ってください。

インストールと構成の順序

- フロントエンド solidDB サーバー
- InfoSphere CDC for solidDB
- バックエンド・データ・サーバー
- バックエンド・データ・サーバー用 InfoSphere CDC
- Access Server
- Management Console

レプリケーションで使用するコンポーネント間の従属関係

データベース間のレプリケーションをセットアップするには、互いに依存する各種エンティティとコンポーネントを定義し、作成する必要があります。これらのエンティティとコンポーネントは、以下の順序で作成し、逆の順序で変更または削除する必要があります。詳しくは、「*InfoSphere Change Data Capture Management Console* 管理ガイド」を参照してください。

1. データベース
2. InfoSphere CDC インスタンス
3. データ・ストア
4. サブスクリプション
5. 表マッピング

レプリケーション・サブスクリプションの変更

レプリケーション・サブスクリプションの変更が必要な場合には、まずサブスクリプションのレプリケーションを終了する必要があります。詳しくは、「*InfoSphere Change Data Capture Management Console* 管理ガイド」の『サブスクリプションで

のレプリケーションの終了』のセクションを参照してください。

hsb netcopy に続けて切り替えを実行するとサブスクリプションが失敗する

solidDB 高可用性 (HotStandby) 構成では、ソース・データ・ストアとして solidDB を使用するサブスクリプションは、**hsb netcopy** の直後に切り替えを行うと、失敗する場合があります。

これは、例えば以下のような場合に起こる可能性があります。

1. 障害または保守のために中断が生じた後、1 次サーバー (ノード 1) および 2 次サーバー (ノード 2) が ADMIN COMMAND 'hsb netcopy' を使用して同期される場合。
2. 1 次サーバー (ノード 1) に対するレプリケーションがいくつかのトランザクションについて続行される場合。
3. 1 次サーバー (ノード 1) に障害が生じ、切り替えによって 2 次サーバー (ノード 2) が新規 1 次サーバーに変更される場合。
4. サブスクリプションが失敗し、新規 1 次サーバー (ノード 2) に対するレプリケーションを再開できない場合。

原因

コマンド ADMIN COMMAND 'hsb netcopy' は、どのログ・ファイルもコピーしません。この結果、InfoSphere CDC レプリケーションは本質的に非同期であるため、InfoSphere CDC for solidDB は、**netcopy** が行われた時点まですべてのトランザクションの処理を行っていない可能性があります。これは、切り替え後に InfoSphere CDC for solidDB が使用しようとするログ位置はおそらく有効でなく、つまり **netcopy** 前のノード 1 上の最後のトランザクションに対するログ・エントリーが新規 1 次サーバー (ノード 2) におそらく存在しないことを意味します。

回避策

InfoSphere CDC for solidDB が、切り替えの後、新規 1 次サーバー (ノード 2) の有効なログ・エントリーに確実にアクセスできるようにするには、以下のようになります。

- **netcopy** を実行する前に、1 次サーバー (ノード 1) から 2 次サーバー (ノード 2) にログ・ファイルをコピーします。これによって、InfoSphere CDC for solidDB は、**netcopy** が行われる前に実行されたトランザクションのログ位置に確実にアクセスできるようになります。

または

- **netcopy** の直後に切り替えを行わないでください。つまり、バックエンド・データベースにいくつかのトランザクションが複製されるまで待ってから、切り替えを行ってください。これによって、1 次サーバー (ノード 1) と 2 次サーバー (ノード 2) のログ位置が確実に同期されます。

または

- (例えば、ノード 1 に障害が生じたため) 切り替えが既に行われた場合は、以下の手順を実行します。

1. 古い 1 次サーバー (ノード 1) をリカバリーします。
2. 切り替えを行って、以前の 1 次サーバー (ノード 1) を 1 次サーバーに戻します。
3. サブスクリプションのレプリケーションを再開します。

別の切り替えを (ノード 2 を新規 1 次サーバーにするために) 行うのは、いくつかのトランザクションが複製されるのを待ってからにしてください。これによって、1 次サーバー (ノード 1) と 2 次サーバー (ノード 2) のログ位置が確実に同期されます。

10 InfoSphere CDC for solidDB (エンド・ユーザー向け資料)

このセクションについて

このセクションでは、solidDB Universal Cache と InfoSphere CDC レプリケーションで使用する IBM InfoSphere Change Data Capture for IBM solidDB コンポーネントのインストールおよび構成手順について詳しく説明します。また、このセクションでは、InfoSphere CDC に固有のコマンドおよびその他の参照情報を記載しています。

このセクションは、「*IBM InfoSphere Change Data Capture* のエンド・ユーザー向け資料」という資料に対応しています。この資料は、その他のデータ・サーバーの InfoSphere CDC コンポーネントとともに配信されます。

solidDB Universal Cache または InfoSphere CDC レプリケーションの設定では、システム・レベルのインストールおよび構成の手順に従い、必要に応じてこのセクションを参照してください。

このセクションでは、InfoSphere CDC という用語は、InfoSphere CDC for solidDB を意味しています。

InfoSphere CDC for IBM solidDB について

IBM InfoSphere Change Data Capture (InfoSphere CDC) は、サポート対象のデータベースにデータを複製したり、そのデータベースからデータを複製したりできるレプリケーション・ソリューションです。また、Management Console で構成した表マッピングの詳細に基づいて、サポート対象のデータベースから複製したデータを受け取ることができます。

InfoSphere CDC では、処理オーバーヘッドとネットワーク・トラフィックを減らすのに使用できる複製データベースを維持できます。レプリケーションは、連続的に実施することも、また最終的な変更に応じて定期的にも実施することもできます。ソース・サーバーからデータが転送されると、ターゲット環境で、そのデータの再マップやトランスフォームを行うことができます。

システム要件

ソフトウェア要件

サポートされるプラットフォームおよびオペレーティング・システムについては、<http://www-01.ibm.com/software/data/soliddb/>の solidDB の Web ページを参照してください。

ハードウェア要件

表 19. ハードウェア要件

RAM	ディスク・スペース
500 MB	220 MB インストール・ファイル、データ・キュー、およびログ・ファイル用

ポート要件

InfoSphere CDC では、Management Console と他のサーバーを実行するクライアント・ワークステーションとの通信用のポート・セットを割り振る必要があります。インターネットへのアクセスは必要ありませんが、これらのポートは、ファイアウォール経由でアクセス可能でなければなりません。

表 20. ポート要件

プロトコル	デフォルトのポート	用途
TCP	11101	以下からの接続を受け入れます。 <ul style="list-style-type: none">• Management Console• レプリケーションのソースとして使用する InfoSphere CDC の他のインストール• コマンド行ユーティリティー Management Console のインストール方法の詳細については、「 <i>Management Console のインストール・ガイド</i> 」を参照してください。

必須のデータベース、ユーザー・アカウント、およびスキーマ solidDB データベースの作成

InfoSphere CDC の構成時に、InfoSphere CDC を接続してデータを複製したい solidDB サーバーのホスト名およびポート番号の入力を促すプロンプトが出されます。InfoSphere CDC をインストールする前に、この solidDB データベースが存在することを確認し、それにアクセスできるデータベース・ユーザーの作成とセットアップが完了していることを確認してください。

SYS_ADMIN_ROLE 特権を持つ solidDB アカウントのセットアップ

Sybase ユーザーを作成およびセットアップし、このユーザーに DBA 特権を割り当てます。InfoSphere CDC を solidDB データベースに接続するには、solidDB ユーザー・アカウントを作成し、そのユーザーに SYS_ADMIN_ROLE 特権を割り当てる必要があります。InfoSphere CDC の構成時に、InfoSphere CDC の接続先にしたい solidDB サーバーのホスト名およびポート番号のほか、このデータベースにアクセスできる solidDB ユーザーのユーザー名およびパスワードの入力を促すプロンプトが出されます。

solidDB スキーマの作成

InfoSphere CDC データベース・メタデータ表用に、スキーマを作成するか、または既存のスキーマを選択します。InfoSphere CDC の構成時には、このスキーマを指定する必要があります。

InfoSphere CDC 用の Windows ユーザー・アカウントのセットアップ

InfoSphere CDC を Windows システムにインストールする場合は、InfoSphere CDC のインストール、構成、またはアップグレードに使用する Windows アカウントを新規にセットアップするか、既存のアカウントから選択する必要があります。

InfoSphere CDC 用の Linux または UNIX ユーザー・アカウントのセットアップ

Linux または UNIX システム上に InfoSphere CDC をインストールする場合は、InfoSphere CDC のインストール、構成、またはアップグレードに使用する Linux または UNIX アカウントを新規にセットアップするか、既存のアカウントから選択する必要があります。InfoSphere CDC は任意のディレクトリーにインストールできますが、そのディレクトリーは、その Linux または UNIX アカウントに所有されている必要があります。

1 バイト文字およびマルチバイト文字のサポート

InfoSphere CDC は、1 バイトとマルチバイトの両方の文字セットのレプリケーションをサポートします。

1 バイト文字のサポート

InfoSphere CDC は、1 バイト文字サポート (SBCS) によるコード・ページ変換を透過的に行います。つまり、各システムで使用されているコード・ページをユーザーが認識する必要はありません。InfoSphere CDC は、ユーザーの構成パラメーターを調べることで、この変換を自動的に実行することができます。

マルチバイト文字のサポート

InfoSphere CDC は、1 バイトで表すことができない日本語や中国語などのマルチバイト文字セット (MBCS) のレプリケーションをサポートします。最も一般的な MBCS 実装は、2 バイト文字セット (DBCS) です。

MBCS の指定では、特定の変換を構成したときに、データはターゲット・システム上のマップ対象列にそのまま適用されるように指示されます。これは、データベースに (データの実際の文字セットに関係なく) 1 バイト文字セットが構成されている場合に可能ですが、このことは文字セットがマルチバイトである場合には保証できません。

InfoSphere CDC は、マッピングを考慮し、構成セットに従ってデータを適用します。列に文字セットを正しく挿入できるという確証はありません。ユーザーは、データベース上の文字セットを把握し、それらのデータの文字セット変換を選択する際に、適切な値を選択する必要があります。Management Console にエンコード変換を設定すると、InfoSphere CDC はデータを、そのデータが受信されたときと完全

に同じフォームで、ターゲット・データベースに適用します。

solidDB データベース上でのマルチバイト文字サポートに対する影響

solidDB 文字データ型のエンコードは、データベース・モードが *Unicode* であるか部分的 *Unicode* であるかによって決まります。

Unicode モード (`General.InternalCharencoding = utf8`)

- 文字データ型 (CHAR、VARCHAR など) は、UTF-8 で保管されます。
- ワイド文字データ型 (WCHAR、WVARCHAR など) は、UTF-16 で保管されます。

部分的 Unicode モード (`General.InternalCharencoding = raw`)

- 文字データ型は特定のエンコードを使用しません。代わりに、データはバイト・ストリングで保管されます (ユーザーのアプリケーションがこれを認識し、必要に応じて変換を処理することが想定されています)。
- ワイド文字データ型は、UTF-16 で保管されます。

InfoSphere CDC for solidDB の新しいインスタンスが作成されると、デフォルトの solidDB データベース・モードである部分的 *Unicode* に従って、デフォルトのエンコードが設定されます。デフォルトで、文字データ型列のエンコードは、常に *ISOLatin1* に設定されます。

- 使用するデータベース・モードが *Unicode* の場合、文字データ型 (CHAR、VARCHAR など) の列のエンコードを UTF-8 に設定する必要があります。
- データベース・モードが部分的 *Unicode* であり、アプリケーション・エンコードが *ISOLatin1* に設定されていない場合、文字データ型 (CHAR、VARCHAR など) の列のエンコードを、アプリケーション環境で使用されているエンコードに設定する必要があります。

表 21. 文字データ型列およびワイド文字データ型列のデフォルト・エンコード設定 (部分的 *Unicode*) および *Unicode* エンコード設定

列の型	デフォルト・エンコード (部分的 <i>Unicode</i>)	<i>Unicode</i> データベース に必要なエンコード
文字データ型 (CHAR、VARCHAR など)	ISOLatin1	UTF-8
ワイド文字データ型 (WCHAR、WVARCHAR など)	UTF-16BE	UTF-16BE

ユーザー出口およびマルチバイト文字セット

InfoSphere CDC の Java クラス・ユーザー出口は、マルチバイト文字セット (MBCS) をサポートします。マルチバイト文字セットは、Java ストリング (UTF-16) に変換されます。

InfoSphere CDC のインストール

このセクションでは、InfoSphere CDC のインストールの段階的な手順を説明します。

InfoSphere CDC の対話式インストール

InfoSphere CDC を、Windows サーバーや、UNIX サーバーまたは Linux サーバーにインストールできます。

InfoSphere CDC をインストールするには (Windows)

手順

1. インストール・ファイルをダブルクリックします。InfoSphere CDC インストール・ウィザードが開きます。
2. 「Next」をクリックします。
3. ライセンス条項に同意する場合には、「**I accept the terms in the license agreement**」を選択し、「Next」をクリックします。
4. InfoSphere CDC をインストールするフォルダーを選択し、「Next」をクリックします。
5. 以前の InfoSphere CDC がインストールされている場合には、インストールのアップグレードを促すプロンプトが出されます。「OK」をクリックしてインストール済み環境をアップグレードします。
6. 製品アイコンの位置を選択し、「Next」をクリックします。
7. インストール・サマリーを検討し、「Install」をクリックします。
8. インストール後に、オプションとして「**Launch Configuration Tool**」を選択し、構成ツールを起動します。構成ツールでは、InfoSphere CDC のインスタンスを追加できます。
9. 「Done」をクリックして、インストールを終了します。

InfoSphere CDC をインストールするには (UNIX および Linux)

このタスクについて

注: X Window システムがインストールされている場合、インストール・プログラムは、グラフィック環境で構成ツールを起動します。インスタンスの開始と停止を行う必要がない点を除いて、構成プロセスは Windows に類似しています。

手順

1. InfoSphere CDC 用にセットアップしたアカウントでログオンします。
2. ご使用の Linux プラットフォーム用の InfoSphere CDC インストール・ファイルをコピーします。
3. インストール・プログラムを実行可能にします。
4. インストール・ファイルの名前を入力して、インストール・プログラムを実行します。
5. 「Introduction」画面で Enter キーを押して、使用許諾契約書を表示します。画面の指示に従って、使用許諾契約書をナビゲートします。
6. 使用許諾契約書を受け入れるには、1 を入力します。

7. インストール・ディレクトリーの絶対パスを入力するか、または Enter キーを押してデフォルトを受け入れます。

注: 指定するディレクトリーは、インストールに使用するアカウントが所有するディレクトリーである必要があります。インストール・プログラムがそのディレクトリーを作成できない場合、別のディレクトリーの指定を促すプロンプトが出されます。

8. インストール・サマリーを検討します。Enter キーを押してインストールを開始します。
9. インストールが完了すると、InfoSphere CDC から InfoSphere CDC 用に構成ツールを起動するオプションが表示されます。
10. 1 を入力して構成ツールを起動します。

InfoSphere CDC のサイレント・インストール

サイレント・インストールでは、各種のパラメーターを含むコマンドを指定することにより、InfoSphere CDC を自動的にインストールできます。このタイプのインストール方式は、スクリプトにサイレント・インストール・コマンドを組み込むことにより、InfoSphere CDC の大規模なデプロイメントで使用できます。

InfoSphere CDC のサイレント・インストールを行うには (UNIX および Linux)

手順

1. InfoSphere CDC 用にセットアップしたアカウントでログオンします。
2. InfoSphere CDC インストール・ファイルをコピーします。
3. インストール・プログラムを実行可能にします。
4. 以下のコマンドを実行して、InfoSphere CDC をインストールし、応答ファイルを生成します。

```
<setup.bin> -r <response-file>
```

5. 別のシステムで、以下のコマンドを実行してサイレント・インストールを行います。

```
<setup.bin> -isilent -f <response-file>
```

ここで、

- <response-file> は、インストール・ファイルの絶対パスです。

InfoSphere CDC の構成 (Windows)

InfoSphere CDC のインストール後に、インストール・プログラムが、構成ツールを起動します。構成ツールでは、ご使用の環境用に InfoSphere CDC を構成できます。レプリケーションを開始するには、InfoSphere CDC を構成する必要があります。

InfoSphere CDC インスタンスの構成 (Windows)

InfoSphere CDC のインスタンスを追加、編集、および削除できます。InfoSphere CDC 構成ツールを使用して、インスタンスに対する作業を行います。

InfoSphere CDC の新しいインスタンスを追加するには (Windows) 始める前に

インストール後に InfoSphere CDC の最初のインスタンスを構成する場合は、この手順のステップ 3 に進みます。

手順

1. コマンド・プロンプトで、以下のコマンドを指定されたディレクトリーで実行し、構成ツールを起動します。

```
¥<InfoSphere CDC Installation Directory>¥bin¥dmconfigurets
```
2. ウェルカム・メッセージで、「OK」をクリックして続行します。
3. 「IBM InfoSphere CDC New Instance」ダイアログ・ボックスの「Instance」領域で、以下のオプションを構成できます。

オプション	説明
Name	InfoSphere CDC インスタンスの名前を入力します。この名前はユニークである必要があります。
Server Port	InfoSphere CDC が、Management Console およびその他のサーバーを実行しているクライアント・ワークステーションとの通信に使用するポート番号を入力します。 注: このポート番号は、同じサーバーにインストールされている別のアプリケーションで使用することはできません。このポート番号は、Management Console のアクセス・マネージャー・パースペクティブでデータ・ストアのアクセス・パラメーターを指定するときに使用します。InfoSphere CDC は、デフォルト TCP/IP ポート番号の 11101 を表示します。詳しくは、Management Console の資料を参照してください。 注: 同じノードに複数のインスタンスをインストールする場合、各インスタンスのポート番号はユニークである必要があります。
Auto-Discovery Port	このボックスを選択して、Access Server から送信されるオートディスカバリー・ブロードキャストで使用する UDP ポート番号を入力します。オートディスカバリーについて詳しくは、Management Console の資料を参照してください。

オプション	説明
Maximum Memory Allowed	InfoSphere CDC に割り振る RAM の最大量を入力します。構成する各インスタンスに少なくとも 64 MB を割り振る必要があります。デフォルトでは、32 ビットのインスタンスには 512 MB の RAM が割り振られ、64 ビットのインスタンスには 1024 MB の RAM が割り振られます。
Bit-Version	以下のいずれかのオプションを選択して、データベースのビット・バージョンを選択します。 <ul style="list-style-type: none"> • 32 ビット • 64 ビット InfoSphere CDC を 32 ビット・サーバーにインストールしている場合、これらのオプションは使用できません。

4. 「**Windows Service**」領域で、InfoSphere CDC サービスの開始に使用するアカウントを指定できます。以下のオプションのいずれかを選択します。

オプション	説明
Local System account	ローカル・システム管理者のアカウントで InfoSphere CDC サービスを開始します。
This account	指定したユーザー・アカウントで InfoSphere CDC サービスを開始します。 アカウントは、<domain>%<user name> のフォーマットで指定する必要があります。<domain> は環境のドメイン名で、<user name> は指定したドメインの有効なログイン・ユーザー名です。コンピューターがドメインの一部でない場合は、<computer name>%<user name> と指定できます。 「 Password 」ボックスおよび「 Confirm Password 」ボックスに、選択した Windows ユーザー・アカウントに現在関連付けられているパスワードを入力します。InfoSphere CDC のインストール後に Windows ユーザー・アカウントのパスワードを変更した場合は、「Windows Services」ダイアログを使用して、各 InfoSphere CDC サービスに現在設定されているパスワードを変更する必要があります。

5. 「**Database**」領域で、レプリケーション用の表を含むデータベースへのアクセスを構成できます。この手順を完了するには、システム管理者特権が必要です。これで、Management Console のアクセス・マネージャー・パースペクティブでデ

ータ・ストアを追加し、ユーザーにこのデータベースへのアクセス権限を提供できるようにします。詳しくは、Management Console の資料を参照してください。

オプション	説明
User name	指定したデータベースのユーザー名を入力します。
Password	指定したデータベースのパスワードを入力します。
Metadata Schema	InfoSphere CDC メタデータ表に使用するデータベースのスキーマを選択します。 デフォルトとして、上記で入力したユーザー名が使用されます。インストールされている別の InfoSphere CDC インスタンスがそのデータベースで使用しているスキーマを除いて、任意のスキーマを指定できます。このスキーマは、インストールの前提条件の一部として、セットアップまたは決定する必要があります。 注: メタデータ・スキーマには、必ず大文字を使用してください。デフォルトでは、solidDB 内のすべてのスキーマ名 (カタログ名) は、大文字です。
Advanced	「Advanced」ボタンを使用して、solidDB JDBC ドライバーの構成パラメーターを変更できます。JDBC ドライバー・パラメーターの詳細については、「IBM solidDB プログラマー・ガイド」を参照してください。

6. 「Server」領域で、データの複製先または複製元とし、レプリケーション用のすべての表を含む solidDB サーバーを構成できます。単一サーバーまたは HA 構成 (HotStandby) を構成できます。

オプション	説明
Single server	指定した solidDB サーバーのホスト名とポート番号を入力します。
Enable SMA	このチェック・ボックスは、solidDB で共有メモリー・アクセス (SMA) を使用する場合に選択します。
HA Configuration (HotStandby)	指定した 1 次および 2 次 solidDB サーバーのホスト名とポート番号を入力します。

7. 「OK」をクリックして、InfoSphere CDC インスタンスの構成設定を保存します。
8. 「Apply」をクリックして、構成ツールの変更を保存します。

次のタスク

構成を完了した後、InfoSphere CDC を開始できます。

InfoSphere CDC のインスタンスを編集するには (Windows) 手順

1. InfoSphere CDC が開始している場合は、`dmshutdown` コマンドを使用して停止します。
2. コマンド・プロンプトで、以下のコマンドを指定されたディレクトリーで実行し、構成ツールを起動します。

```
¥<InfoSphere CDC Installation Directory>%bin%dmconfigurets
```
3. 変更するインスタンスが開始している場合は、「**Instances**」領域でそのインスタンスを選択し、「**Stop**」をクリックします。
4. 「**Instances**」領域でインスタンスを選択し、「**Edit**」をクリックします。

「**InfoSphere CDC Edit Instance**」ダイアログが開きます。
5. インスタンスの追加の際に指定した値を、このダイアログ・ボックスで変更できます。
6. 「**Apply**」をクリックして変更を保存し、「**Close**」をクリックします。

構成ツールがインスタンスを変更します。
7. 「**Instances**」領域で変更したインスタンスを選択し、「**Start**」をクリックしてインスタンスを開始します。

InfoSphere CDC のインスタンスを削除するには (Windows) 手順

1. InfoSphere CDC が開始している場合は、`dmshutdown` コマンドを使用して停止します。
2. コマンド・プロンプトで、以下のコマンドを指定されたディレクトリーで実行し、構成ツールを起動します。

```
¥<InfoSphere CDC Installation Directory>%bin%dmconfigurets
```
3. 削除するインスタンスが開始している場合は、「**Instances**」領域でそのインスタンスを選択し、「**Stop**」をクリックします。
4. 「**Instances**」領域でインスタンスを選択し、「**Delete**」をクリックします。
5. 「**Yes**」をクリックして、インスタンスを永続的に削除します。

InfoSphere CDC の構成 (UNIX および Linux)

InfoSphere CDC のインストール後に、インストール・プログラムが、構成ツールを起動します。構成ツールでは、ご使用の環境用に InfoSphere CDC を構成できます。レプリケーションを開始するには、InfoSphere CDC を構成する必要があります。

InfoSphere CDC インスタンスの構成 (UNIX および Linux)

InfoSphere CDC のインスタンスを追加、編集、および削除できます。InfoSphere CDC 構成ツールを使用して、インスタンスに対する作業を行います。

InfoSphere CDC の新しいインスタンスを追加するには (UNIX および Linux)

始める前に

インストール後に InfoSphere CDC の最初のインスタンスを構成する場合は、この手順のステップ 4 に進みます。

手順

1. コマンド・プロンプトで、以下のコマンドを指定されたディレクトリーで実行し、構成ツールを起動します。

```
<InfoSphere CDC Installation Directory>/bin/dmconfigurets
```
2. ウェルカム・メッセージで、**Enter** キーを押して続行します。
3. InfoSphere CDC の新しいインスタンスを追加するために、2 を入力して **Enter** キーを押します。
4. InfoSphere CDC インスタンスの名前を入力し、**Enter** キーを押します。インスタンス名はユニークである必要があります。
5. InfoSphere CDC が、Management Console およびその他のサーバーを実行しているクライアント・ワークステーションとの通信に使用するポート番号を入力します。InfoSphere CDC は、デフォルト・ポート番号の 11101 を表示します。**Enter** キーを押します。

注: このポート番号は、同じサーバーにインストールされている別のアプリケーションで使用することはできません。このポート番号は、Management Console のアクセス・マネージャー・パースペクティブでデータ・ストアのアクセス・パラメーターを指定するときに使用します。詳しくは、Management Console の資料を参照してください。

注: 同じノードに複数のインスタンスをインストールする場合、各インスタンスのポート番号はユニークである必要があります。

6. オートディスカバリー UDP ポート番号を入力するか、**Enter** キーを押して DISABLE のデフォルト値を使用します。このポート番号は、Access Server から送信されるオートディスカバリー・ブロードキャストで使用されます。オートディスカバリーについて詳しくは、Management Console の資料を参照してください。
7. InfoSphere CDC に割り振る RAM の最大量を入力します。構成する各インスタンスに少なくとも 64 MB を割り振る必要があります。デフォルトでは、32 ビットのインスタンスには 512 MB の RAM が割り振られ、64 ビットのインスタンスには 1024 MB の RAM が割り振られます。
8. 構成する solidDB サーバーの構成タイプを選択します。

オプション	説明
Single server	1 を入力し、 Enter キーを押します。
HA Configuration (HotStandby)	2 を入力し、 Enter キーを押します。

9. 使用する構成タイプに従ってホスト名とポート番号を入力します。

オプション	説明
Single server	<ol style="list-style-type: none"> 指定したサーバーのホスト名を入力し、Enter キーを押します。 指定したサーバーのポート番号を入力し、Enter キーを押します。デフォルトは 1964 です。
HA Configuration (HotStandby)	<ol style="list-style-type: none"> 指定した 1 次サーバーのホスト名を入力し、Enter キーを押します。 指定した 1 次サーバーのポート番号を入力し、Enter キーを押します。デフォルトは 1964 です。 指定した 2 次サーバーのホスト名を入力し、Enter キーを押します。 指定した 2 次サーバーのポート番号を入力し、Enter キーを押します。デフォルトは 1964 です。 <p>注: 1 次側と 2 次側は別のノードに配置されると想定されているため、1 次側と 2 次側のデフォルトのポート番号は同じです。例えば、評価の目的で、1 次サーバーと 2 次サーバーを同じノードに配置する場合には、両方のデフォルトのポート番号を同じにすることはできません。</p>

10. 必要に応じて、solidDB で共有メモリー・アクセス (SMA) を使用できるように選択します。

オプション	説明
Use default settings	n を入力し、 Enter キーを押します。
Enable SMA	y を入力し、 Enter キーを押します。

11. 必要に応じて、詳細パラメーター (JDBC パラメーター) を構成します。

オプション	説明
Use default settings	n を入力し、 Enter キーを押します。
Modify settings	<ol style="list-style-type: none"> y を入力し、Enter キーを押します。 <parameter>=<value>;<parameter>=<value>;... 構文を使用して、パラメーター設定を入力します。 <p>注: HA セットアップでは、パラメーター solid_tf_level は、デフォルトで 'CONNECTION' に設定されています。</p>

- 指定したデータベースのユーザー名を入力し、**Enter** キーを押します。
- 指定したデータベースのパスワードを入力し、**Enter** キーを押します。構成ツールが、データベースでスキーマを検索します。
- 使用するメタデータ・スキーマに対応する番号を入力し、**Enter** キーを押します。

15. データベースへのバルク挿入に使用するディレクトリーのパスを入力します。
Enter キーを押します。solidDB データベースと InfoSphere CDC の両方に、このディレクトリーに対する読み取り権限と書き込み権限が必要です。

注:

- InfoSphere CDC のインスタンスごとに、異なるディレクトリーを使用する必要があります。
 - このディレクトリーには、レプリケーション用のデータベース表が含まれることがあります。このディレクトリーへのユーザー・アクセス権限を決定するときは、このことを考慮してください。
16. 構成ツールによって InfoSphere CDC インスタンスが作成され、インスタンスの開始を促すプロンプトが出されます。y を入力して、インスタンスを開始します。

注: 構成によって既存のインスタンスのメタデータが上書きされようとした場合、構成ツールによってプロンプトが出されます。

InfoSphere CDC のインスタンスを編集するには (UNIX および Linux)

手順

1. InfoSphere CDC が開始している場合は、dmshutdown コマンドを使用して停止します。
2. 以下のコマンドを指定されたディレクトリーで実行し、構成ツールを起動します。

```
/<InfoSphere CDC Installation Directory>/bin/dmconfigurets
```
3. InfoSphere CDC のインストール済みインスタンスをリストするために、1 を入力して **Enter** キーを押します。変更するインスタンスの名前を記録します。
4. InfoSphere CDC のインスタンスを変更するために、3 を入力して **Enter** キーを押します。
5. 変更するインスタンスの名前を入力し、**Enter** キーを押します。

構成ツールを使用すると、インスタンスの追加の際に指定したいいくつかの値を編集できます。

6. 変更後、変更を適用してメインメニューに戻るには、5 を入力して **Enter** キーを押します。変更内容を破棄するには、6 を入力して **Enter** キーを押します。

InfoSphere CDC のインスタンスを削除するには (UNIX および Linux)

手順

1. InfoSphere CDC が開始している場合は、dmshutdown コマンドを使用して停止します。
2. 以下のコマンドを指定されたディレクトリーで実行し、構成ツールを起動します。

```
/<InfoSphere CDC Installation Directory>/bin/dmconfigurets
```
3. InfoSphere CDC のインストール済みインスタンスをリストするために、1 を入力して **Enter** キーを押します。削除するインスタンスの名前を記録します。

4. InfoSphere CDC のインスタンスを削除するために、4 を入力して **Enter** キーを押します。
5. 削除するインスタンスの名前を入力し、**Enter** キーを押します。

InfoSphere CDC の開始と停止

このセクションでは、InfoSphere CDC インスタンスの開始および停止の段階的な手順を説明します。

InfoSphere CDC の開始

サポートされている Windows サーバー上に InfoSphere CDC をインストールすると、初期構成後に手動で開始できます。InfoSphere CDC を開始すると、Windows でサービスが開始されます。サービスは、リブート後に自動的に開始します。

サポートされている Linux サーバー上に InfoSphere CDC をインストールすると、コマンドを実行して開始できます。インストール後に InfoSphere CDC を開始して、Management Console でこのインスタンス用のデータ・ストアを作成できます。

InfoSphere CDC を開始するには (Windows)

手順

1. コマンド・プロンプトで、以下のコマンドを指定されたディレクトリーで実行し、構成ツールを起動します。

```
¥<InfoSphere CDC Installation Directory>¥bin¥dmconfigurets
```

2. 「**Instances**」領域で開始するインスタンスを選択し、「**Start**」をクリックします。

構成ツールが InfoSphere CDC のインスタンスを開始します。

次のタスク

Windows の「サービス」ダイアログを使用して、InfoSphere CDC サービスの開始と停止を行うこともできます。

InfoSphere CDC を開始するには (UNIX および Linux)

手順

InfoSphere CDC を実行しているオペレーティング・システムに応じて、以下のいずれかの開始コマンドを実行します。

- dmts32 - I <instance_name>
- dmts64 - I <instance_name>

InfoSphere CDC の停止

InfoSphere CDC 構成ツールを使用して構成の設定を変更したい場合、InfoSphere CDC を停止することが必要な場合があります。

Windows では、InfoSphere CDC を停止すると Windows でサービスが停止します。サービスは、リブート後にもう一度自動的に開始します。

UNIX と Linux では、コマンドを実行して InfoSphere CDC を停止できます。保守のため、または InfoSphere CDC のアップグレードのためにサーバーやデータベースをオフラインにする前に、このコマンドを使用してください。

InfoSphere CDC を停止するには (Windows)

手順

1. 以下のコマンドを指定されたディレクトリで実行し、構成ツールを起動します。

```
<InfoSphere CDC Installation Directory>/bin/dmconfigurets
```

2. 「**Instances**」領域で停止するインスタンスを選択し、「**Stop**」をクリックします。

構成ツールが InfoSphere CDC のインスタンスを停止します。

次のタスク

Windows の「サービス」ダイアログを使用して、InfoSphere CDC サービスの開始と停止を行うこともできます。

InfoSphere CDC を停止するには (UNIX および Linux)

手順

1. Management Console ですべてのサブスクリプションのレプリケーションを終了します。サブスクリプションのレプリケーションの終了方法については、Management Console の資料を参照してください。
2. InfoSphere CDC の停止方法に応じて、以下のいずれかの停止コマンドを実行します。

オプション	説明
<code>dmshutdown -I <instance_name></code>	このコマンドを使用して、InfoSphere CDC を正常にシャットダウンします。 同じ Linux サーバーに、アクティブな InfoSphere CDC インストールが複数存在しており、すべてをシャットダウンしたい場合は、このコマンドを InfoSphere CDC の各インスタンスのインストール・ディレクトリから実行します。
<code>dmterminate -I <instance_name></code>	このコマンドを使用して、Linux サーバーで実行中のすべてのインスタンスについて、すべての InfoSphere CDC プロセスを強制終了します。dmshutdown コマンドを使用しても InfoSphere CDC のシャットダウンを完了できない場合は、このコマンドを使用してください。

Management Console での SQL ステートメントの使用可能化

InfoSphere CDC では、ターゲット表に対して表レベルのクリア操作またはリフレッシュ操作を適用した後、ユーザーによる SQL ステートメントの実行が可能となり

ます。SQL ステートメントは、Management Console の「**Additional SQL**」ダイアログ・ボックスに指定できます。デフォルトでは、セキュリティ上の理由で、この機能は InfoSphere CDC で使用不可に設定されています。この機能は、InfoSphere CDC をインストールしたデータベースに TS_SQL_EXECAUTH という表を作成することにより使用可能に設定できます。この表の構造は重要ではありません。ただし、この表は、InfoSphere CDC の構成中に、メタデータ表と同じスキーマを使用して作成する必要があります。Management Console での SQL ステートメントの指定について詳しくは、Management Console の資料で『リフレッシュ操作を制御するための SQL の指定』を参照してください。

Management Console で SQL ステートメントを使用可能にするには 手順

1. InfoSphere CDC 用に作成したデータベースをターゲット・サーバーで見つけます。InfoSphere CDC の使用方法に応じて、これは InfoSphere CDC にとっての複製先または複製元となるデータベースです。

注: インストール中に、InfoSphere CDC は、InfoSphere CDC プロセスに必要なメタデータ表をこのデータベース内に配置します。

2. SQL ステートメントの指定を使用可能にする場合は、データベース内に TS_SQL_EXECAUTH という名前の表を作成します。

注: 表の構造は任意ですが、InfoSphere CDC の構成時に指定したスキーマでこの表を作成しなければなりません。

InfoSphere CDC がサポートするデータ型

Management Console でレプリケーション用にソース列とターゲット列をマップする場合、どのデータ型に互換性があるかを認識しておく必要があります。レプリケーション用に表をマップする方法については、Management Console の資料の『表のマッピング』のセクションを参照してください。

サポートされているデータ型

このセクションでは、InfoSphere CDC が複製できるデータ型を示します。

- bigint
- binary
- blob
- char
- clob
- date
- decimal
- double precision
- float
- integer
- long varbinary

- long varchar
- nchar
- nclob
- numeric
- nvarchar
- real
- smallint
- time
- timestamp
- tinyint
- varbinary
- varchar

サポートされているデータ型の詳細については、「*IBM solidDB SQL ガイド*」を参照してください。

サポートされているマッピング

このセクションでは、サポートされているデータ型に対して Management Console でサポートされるマッピングを示します。

パブリッシュされるデータ型	サポートされるマッピング
bigint	任意の数値、バイナリー、または LOB データ型
binary	任意のバイナリーまたは LOB データ型
blob	任意のバイナリーまたは LOB データ型
char	任意の文字、可変長文字、CLOB、バイナリー、またはその他の LOB データ型
clob	任意の文字、可変長文字、CLOB、バイナリー、またはその他の LOB データ型
date	任意のデータ型
decimal	任意の数値、バイナリー、または LOB データ型
double precision	任意の数値、バイナリー、または LOB データ型
float	任意の数値、バイナリー、または LOB データ型
integer	任意の数値、バイナリー、または LOB データ型
long varbinary	任意のバイナリーまたは LOB データ型
long varchar	任意の文字、可変長文字、CLOB、バイナリー、またはその他の LOB データ型
nchar	任意の文字、可変長文字、CLOB、バイナリー、またはその他の LOB データ型

パブリッシュされるデータ型	サポートされるマッピング
nclob	任意の文字、可変長文字、CLOB、バイナリー、またはその他の LOB データ型
nvarchar	任意の文字、可変長文字、CLOB、バイナリー、またはその他の LOB データ型
numeric	任意の数値、バイナリー、または LOB データ型
real	任意の数値、バイナリー、または LOB データ型
smallint	任意の数値、バイナリー、または LOB データ型
tim	任意の時刻データ型
timestamp	任意の日付、時刻、またはタイム・スタンプ・データ型
tinyint	任意の数値、バイナリー、または LOB データ型
varbinary	任意のバイナリーまたは LOB データ型
varchar	任意の文字、可変長文字、CLOB、バイナリー、またはその他の LOB データ型

InfoSphere CDC メタデータ表

InfoSphere CDC は、現行のレプリケーション構成に関するデータを表す表集合を維持しています。いろいろな理由で、これらの表をバックアップしたい場合や、その他の同様な作業を行いたい場合があります。ただし、IBM 担当員から要求された場合を除いて、これらの表の内容を変更しないでください。

以下に、InfoSphere CDC が作成するメタデータ表の名前を示します。

- TS_AUTH

注: Management Console でアクセス・マネージャー のパースペクティブに追加したすべてのユーザーに、TS_AUTH メタデータ表に対する GRANT SELECT 特権を付与するようにしてください。Management Console で アクセス・マネージャー のパースペクティブにユーザーを追加する方法については、Management Console の資料を参照してください。

- TS_BOOKMARK
- TS_CONFAUD

InfoSphere CDC は、構成プロセスで指定したデータベースとスキーマ内にこれらの表を作成およびインストールします。

InfoSphere CDC がターゲット表に表レベルのリフレッシュ操作またはクリア操作を適用した後で SQL ステートメントを指定したい場合には、TS_SQL_EXECAUTH 表を作成し、維持する必要があります。

- TS_AGED_TABLES

このメタデータ表は、InfoSphere CDC for solidDB に固有のもので、この表には、solidDB フロントエンド内の表のエージング状況に関する情報が含まれています。

InfoSphere CDC のコマンド

このセクションでは、InfoSphere CDC で使用可能なコマンドについて説明します。これらのコマンドを使用して、レプリケーションの制御、レプリケーション用の表の管理、レプリケーションのモニター、およびその他のさまざまな作業を行うことができます。

InfoSphere CDC コマンドの使用

InfoSphere CDC コマンドは、コマンド行プロンプトで実行することも、バッチ・ファイルまたはシェル・スクリプトの一部として実行することもできます。コマンドは、InfoSphere CDC インストール・ディレクトリーの `bin` ディレクトリーにあります。コマンドを実行するには、このディレクトリーにナビゲートします。

注: コマンドに使用可能なフラグと各フラグの簡略説明をリストするには、コマンド・プロンプトでコマンドの名前と `-?` フラグを入力し、**Enter** キーを押します。例えば、`dmterminate -?` のように入力します。

コマンド・フォーマット

コマンドごとに、以下の項目の情報を記載しています。

- **構文:** コマンドの名前を示し、コマンド・パラメーターをリストします。
- **パラメーター:** コマンドの中の各パラメーターについて説明し、指定可能な値を示します。
- **結果:** コマンドが正常に終了した場合に、コマンドから返される値を示します。これらの値は、スクリプトを記述する上で役に立つことがあります。このセクションでは、コマンドの実行結果として画面に表示される情報があればそれについても説明します。
- **例:** コマンド実行の 1 つ以上の例を示します。

パラメーター・フォーマット

コマンド・パラメーターの定義での以下の規則に注意してください。

- 不等号括弧 (`<>`) は**必須**パラメーターを表します。
- 大括弧 (`[]`) は**オプション**・パラメーターを表します。パラメーターを省略した場合には、InfoSphere CDC はデフォルト値を使用します。
- 1 つ以上のパラメーターを区切る垂直バー (`|`) は、リストにあるパラメーターのうち 1 つのみ使用可能であることを表します。大括弧 `[]` に囲まれたパラメーターのリストに 1 つ以上の垂直バーがある場合、選択項目はリスト内のパラメーターに限られますが、パラメーターを何も指定しなくてもよいという意味になります。
- 省略符号 (`...`) は、パラメーターまたはオプションを複数回、繰り返すことができるということを意味します。

- 特に記載がなければ、コマンドはすべてのオペレーティング・システムに適用されます。

TSINSTANCE 環境変数の設定

コマンドを使用する前に、TSINSTANCE 環境変数を InfoSphere CDC インスタンスの名前に設定できます。

TSINSTANCE 環境変数を設定した後は、コマンドの実行時にインスタンス名を指定する必要がありません。

Windows プラットフォーム

コマンド・プロンプトで、以下のコマンドを実行します。

```
SET TSINSTANCE=<instance_name>
```

ここで、

- <instance_name> は、InfoSphere CDC インスタンスの名前です。

Linux プラットフォーム

以下のコマンドを実行します。

```
EXPORT TSINSTANCE=<instance_name>
```

ここで、

- <instance_name> は、InfoSphere CDC インスタンスの名前です。

レプリケーション・コマンドの制御

このセクションでは、InfoSphere CDC でレプリケーションを制御するコマンドについて説明します。

dmendreplication: レプリケーションの終了

このコマンドを使用して、指定したサブスクリプションのリフレッシュまたはミラーリングを終了します。このコマンドは、指定されたサブスクリプションのレプリケーションが正常に完了した後、終了します。

構文

```
dmendreplication -I <instance_name> [-c|-i] [--A|-s <subscription_names> ...> [-L <locale>]
```

パラメーター

-I <instance_name>

レプリケーションを終了する InfoSphere CDC インスタンスを指定します。あるいは、この値の代わりに TSINSTANCE 環境変数を指定することもできます。

- c InfoSphere CDC がレプリケーションを制御された方式で終了することを指定します。このオプションを指定すると、InfoSphere CDC は、進行中の操作をすべて完了し、保留になっている変更内容をターゲット表に適用します。c と i のオプションをいずれも省略した場合は、InfoSphere CDC はデフォルトで c を指定したものとみなします。

- i InfoSphere CDC がレプリケーションをすぐに終了することを指定します。このオプションは、現行の操作をすべて中断します。InfoSphere CDC は、保留になっている変更内容をターゲット表に適用しません。
- A InfoSphere CDC がすべてのサブスクリプションのレプリケーションを終了することを指定します。
- s <subscription_names>
InfoSphere CDC が指示されたサブスクリプションのレプリケーションを終了することを指定します。サブスクリプションを複数指定する場合は、リスト形式で指定してください。
- L <locale>
InfoSphere CDC インスタンスに使用するロケールの名前。デフォルトはマシンのロケールです。

結果

このコマンドは、成功した場合には 0 の値を返し、失敗した場合にはゼロ以外の値を返します。

例

```
dmendreplication -I myinstance -c -s Finance
```

InfoSphere CDC は、指定されたインスタンスの **Finance** サブスクリプションのレプリケーションを制御された方式で終了します。

dmrefresh: サブスクリプションのリフレッシュ

このコマンドを使用して、指定したサブスクリプションをリフレッシュします。サブスクリプションをリフレッシュすると、InfoSphere CDC は、ターゲット表とソース表を確実に同期します。一般的には、表のレプリケーション方式を「**Refresh**」に設定している場合に、ターゲット表をリフレッシュします。

ただし、レプリケーション方式が「**Mirror**」に設定されているターゲット表もリフレッシュすることが可能です。ミラーリング対象として構成されている表をリフレッシュする場合は、InfoSphere CDC は、ターゲット表がソース表と同期するようにターゲット表をリフレッシュし、ミラーリングの開始点としてジャーナルにログ位置を設定します。

このコマンドは、指定されたサブスクリプションのリフレッシュが正常に完了した後、終了します。このプログラムを実行中に強制終了すると、InfoSphere CDC は指定されたサブスクリプションのレプリケーションをすぐに終了します。

構文

```
dmrefresh -I <instance_name> [-a|-f] <-A|-s <subscription_names> ...> [-L <locale>]
```

パラメーター

-I <instance_name>

1 つ以上のサブスクリプションをリフレッシュする InfoSphere CDC インスタンスを指定します。あるいは、この値の代わりに TSINSTANCE 環境変数を指定することもできます。

- a InfoSphere CDC がサブスクリプション内のすべてのターゲット表をリフレッシュすることを指定します。
- f リフレッシュのフラグが立っているターゲット表のみを InfoSphere CDC がリフレッシュすることを指定します。-a と -f のオプションをいずれも省略した場合は、InfoSphere CDC はデフォルトで -f を指定したものとみなします。
- A InfoSphere CDC がすべてのサブスクリプションをリフレッシュすることを指定します。
- s <subscription_names>
InfoSphere CDC が指示されたサブスクリプションをリフレッシュすることを指定します。サブスクリプションを複数指定する場合は、リスト形式で指定してください。
- L <locale>
InfoSphere CDC インスタンスに使用するロケールの名前。デフォルトはマシンのロケールです。

結果

このコマンドは、成功した場合には 0 の値を返し、失敗した場合にはゼロ以外の値を返します。

例

```
dmrefresh -I new_instance -a -s Finance
```

InfoSphere CDC は、**Finance** サブスクリプション内のすべてのターゲット表をリフレッシュします。

dmstartmirror: ミラーリングの開始

このコマンドを使用して、指定したサブスクリプションのミラーリングを開始します。このコマンドは、レプリケーション方式が「**Mirror**」で、状況が「**Refresh**」または「**Active**」の任意の表のミラーリングを開始します。ミラーリングを開始する前に、レプリケーション方式が「**Mirror**」で、状況が「**Refresh**」のすべての表は、最初にサブスクリプションでリフレッシュされます。連続的なミラーリングを開始すると、このコマンドは指定したサブスクリプションのミラーリングを正常に開始した後、終了します。最終的な変更のミラーリングを開始すると、このコマンドは指定したサブスクリプションの最終的な変更のミラーリングを完了した後、終了します。

構文

```
dmstartmirror -I <instance_name> [-c|-n] <-A|-s <subscription_names> ...>
[-L <locale>]
```

パラメーター

- I <instance_name>
ミラーリングを開始する InfoSphere CDC インスタンスを指定します。あるいは、この値の代わりに TSINSTANCE 環境変数を指定することもできます。

- c InfoSphere CDC が連続的なミラーリングを開始することを指定します。c と n のオプションをいずれも省略した場合は、InfoSphere CDC はデフォルトで c を指定したものとみなします。
- n InfoSphere CDC が最終的な変更のミラーリングを開始することを指定します。
- A InfoSphere CDC がすべてのサブスクリプションのミラーリングを開始することを指定します。
- s <subscription_names>
InfoSphere CDC が指示されたサブスクリプションのミラーリングを開始することを指定します。サブスクリプションを複数指定する場合は、リスト形式で指定してください。
- L <locale>
InfoSphere CDC インスタンスに使用するロケールの名前。デフォルトはマシンのロケールです。

結果

このコマンドは、成功した場合には 0 の値を返し、失敗した場合にはゼロ以外の値を返します。

例

```
dmstartmirror -I myinstance c -s Finance
```

InfoSphere CDC は、**Finance** サブスクリプションの連続的なミラーリングを開始します。

データベース・トランザクション・ログ・コマンド

このセクションでは、データベース・トランザクションのログやブックマークの管理を支援するコマンドについて説明します。

dmdecodebookmark: ブックマーク詳細情報の表示

このコマンドを使用して、ブックマークの詳細情報を表示します。

構文

```
dmdecodebookmark -I <instance_name> (-b | -f) [-d] [-L <locale>]
```

パラメーター

-I <instance_name>

InfoSphere CDC インスタンスの名前。TSINSTANCE 環境変数を InfoSphere CDC インスタンスの名前に設定することができます。これが完了すると、コマンドを実行する際に、インスタンスを指定する必要がなくなります。

-b <bookmark>

16 進エンコード・ストリングのブックマーク。

-f <bookmark_file>

バイナリー・ファイルのブックマーク・ファイル。

[-d] <database_version>

指定したブックマークの生成元が InfoSphere CDC の旧バージョンの場合、ブックマークの生成元のデータベースとバージョン。

-L <locale>

InfoSphere CDC インスタンスに使用するロケールの名前。デフォルトはマシンのロケールです。

結果

このコマンドは、成功した場合には 0 の値を返し、失敗した場合にはゼロ以外の値を返します。

例

```
dmdecodebookmark -f bookmark.txt
```

InfoSphere CDC は bookmark.txt ファイルの情報を表示します。

dmsetbookmark: ブックマークの設定

このコマンドを使用して、サブスクリプションの収集ポイントを変更します。

構文

```
dmsetbookmark -I <instance_name> -s <subscription_name> (-b | -l | -f | -t) [-a] [-L <locale>]
```

パラメーター**-I <instance_name>**

InfoSphere CDC インスタンスの名前。 TSINSTANCE 環境変数を InfoSphere CDC インスタンスの名前に設定することができます。これが完了すると、コマンドを実行する際に、インスタンスを指定する必要がなくなります。

-s <subscription_name>

InfoSphere CDC がブックマークを設定するサブスクリプションの名前。

-b <bookmark>

データベース・ログにおける InfoSphere CDC のミラーリング再開位置を決めるブックマークを指定します。次のミラーリングのときに、InfoSphere CDC は指定の位置で収集します。ブックマークは、dmshowbookmark コマンドから取得される 16 進エンコード・ストリングです。

-l <bookmark>

新しい収集ポイントを示すブックマーク。ブックマークは、dmdecodebookmark コマンドから取得されるストリングです。詳しくは、111 ページの『dmdecodebookmark: ブックマーク詳細情報の表示』を参照してください。

-f <bookmark_file>

データベース・ログにおける InfoSphere CDC のミラーリング再開位置を決めるブックマークを格納しているバイナリー・ファイルを指定します。次のミラーリングのときに、InfoSphere CDC は指定の位置で収集します。ブックマーク・ファイルは、その位置を格納するバイナリー・ファイルです。

-t <datetime>

次回に InfoSphere CDC が指定日時の前後にログに記録されたエントリーで収集を再開することを指定します。

-a 新規の収集ポイント以降、サブスクリプション内のすべての表 (停止している表を除く) をアクティブに設定します。

-L <locale>

InfoSphere CDC インスタンスに使用するロケールの名前。デフォルトはマシンのロケールです。

結果

このコマンドは、成功した場合には 0 の値を返し、失敗した場合にはゼロ以外の値を返します。

例

```
dmsetbookmark -I MYINSTANCE -b 2FC5GJHKLKJL458K9K809IK9  
-s FINANCE
```

InfoSphere CDC は、指定されたインスタンスの **Finance** サブスクリプションにブックマーク位置を設定します。このコマンドは、データベース・ログ内の指示された位置でミラーリングが再開することを指定します。

dmshowbookmark: ブックマーク情報の表示

このコマンドを使用して、ターゲットにコミットされるサブスクリプションの最新のレプリケーション位置を表示します。

構文

```
dmshowbookmark -I <instance_name> -s <subscription_name> [-f] [-v] [-L <locale>]
```

パラメーター

-I <instance_name>

InfoSphere CDC インスタンスの名前。TSINSTANCE 環境変数を InfoSphere CDC インスタンスの名前に設定することができます。これが完了すると、コマンドを実行する際に、インスタンスを指定する必要がなくなります。

-s <subscription_name>

ブックマークを表示するサブスクリプションの名前を指定します。

[-f] <file_name>

ブックマーク出力のバイナリー・ファイルの名前を指定します。

[-v]

16 進エンコード・ストリングを含む、ブックマークに関する詳細情報を表示します。表示される情報の量は、ソース・エンジンのタイプとバージョンによって異なります。16 進エンコード・ストリングは常に表示されます。それは、dmdecodebookmark コマンドの表示内容のサブセットです。指定しない場合は、16 進エンコード・ストリングのみが表示されます。

-L <locale>

InfoSphere CDC インスタンスに使用するロケールの名前。デフォルトはマシンのロケールです。

結果

このコマンドは、成功した場合には 0 の値を返し、失敗した場合にはゼロ以外の値を返します。

例

```
dmshowbookmark -I myinstance -s master
```

InfoSphere CDC は、**master** サブスクリプションのブックマーク情報を表示します。

dmshowlogdependency: ログの従属関係の表示

このコマンドを使用して、InfoSphere CDC に使用され、レプリケーションに必要なデータベース・ログに関する情報を表示します。このコマンドを使用して、ログ保存ポリシーをインプリメントします。このコマンドを使用すると、以下の情報を表示することができます。

- 指定されたインスタンスに必要なデータベース・ログのリスト
- 指定されたインスタンスに現在欠落しているデータベース・ログのリスト
- 指定されたインスタンスのデータベース・ログの中にある最も古いオープン・トランザクション
- InfoSphere CDC の指定されたインスタンスが現在、ソースで読み取っているデータベース・ログ
- InfoSphere CDC の指定されたインスタンスが現在、ターゲットに適用しているサブスクリプションのデータベース・ログ

構文

```
dmshowlogdependency -I <instance_name> (-c | -i | -t | -l)  
(-s <subscription_name> | -A) [-v] [-L <locale>]
```

パラメーター

-I <instance_name>

InfoSphere CDC インスタンスの名前。 TSINSTANCE 環境変数を InfoSphere CDC インスタンスの名前に設定することができます。これが完了すると、コマンドを実行する際に、インスタンスを指定する必要がなくなります。

- c 指定されたインスタンスに現在欠落しており、レプリケーションに必要なアーカイブ済みデータベース・ログのリストをチェックして表示します。これらのログは、InfoSphere CDC を使用してレプリケーションを開始する前に、リストアップしておく必要があります。ログに欠落がなければ、このコマンドの出力に「OK」が表示されます。出力をスクリプトまたはバッチ・ファイルで解析することができます。
- i 指定されたインスタンスに必要なデータベース・ログの完全なリストを表示します。これらのログは、レプリケーションの開始に必要なログで、その中にはターゲットにまだ適用されていないデータが含まれています。
- t 指定された InfoSphere CDC インスタンスが現在読み取っているサブスクリプションの現行ターゲット・データベース・ログを表示します。これは、ターゲットが確認した現在位置を含むログです。

-I 指定された InfoSphere CDC インスタンスが現在読み取っている現行ソース・データベース・ログを表示します。これは、収集の現在位置を含むログです。

-s <subscription_name>

InfoSphere CDC が現在読み取っているターゲット・データベース・ログを表示するサブスクリプションの名前を指定します。ターゲット・データベース・ログを表示するには、このパラメーターを -t パラメーターと組み合わせて使用します。

-A すべてのサブスクリプションを指定します。

-v 詳細出力を指定します (これを指定しない場合には、出力はスクリプト用にフォーマットされます)。

-L <locale>

InfoSphere CDC インスタンスに使用するロケールの名前。デフォルトはマシンのロケールです。

結果

このコマンドは、成功した場合には 0 の値を返し、失敗した場合にはゼロ以外の値を返します。

例

```
dmshowlogdependency -I MyInstance
```

レプリケーション・コマンドに関する表の管理

このセクションでは、InfoSphere CDC で複製したい表の管理を支援するコマンドについて説明します。

dmdescribe: ソース表の記述

このコマンドを使用して、ソース表マッピングの変更内容をターゲットに送信します。

このコマンドは、指定されたサブスクリプションの記述が正常に完了した後、終了します。

構文

```
dmdescribe -I <instance_name> [-A|-s <subscription_names> ...] [-L <locale>]
```

パラメーター

-I <instance_name>

ソース表マッピングの変更内容をターゲットに送信する InfoSphere CDC インスタンスを指定します。あるいは、この値の代わりに TSINSTANCE 環境変数を指定することもできます。

-A すべてのサブスクリプションに対するソース表マッピングの変更内容を InfoSphere CDC がターゲットに送信することを指定します。

-s <subscription_names>

指示されたサブスクリプションに対するソース表マッピングの変更内容を

InfoSphere CDC がターゲットに送信することを指定します。サブスクリプションを複数指定する場合は、リスト形式で指定してください。

-L <locale>

InfoSphere CDC インスタンスに使用するロケールの名前。デフォルトはマシンのロケールです。

結果

このコマンドは、成功した場合には 0 の値を返し、失敗した場合にはゼロ以外の値を返します。

例

```
dmdescribe -I new_instance -s Finance
```

InfoSphere CDC は、指定されたインスタンスのターゲットに、**Finance** サブスクリプションのソース表マッピングの変更内容を送信します。

dmflagforrefresh: リフレッシュ対象にフラグを立てる

このコマンドを使用して、リフレッシュ対象としてソース表にフラグを立てます。リフレッシュ対象として表にフラグを立てると、その先のある時点でリフレッシュしたい表を選択することになります。サブスクリプションのレプリケーション方式として「**Refresh**」を選択した場合に、このプロシージャーを使用してください。

構文

```
dmflagforrefresh -I <instance_name> -s <subscription_names>  
<-A|-t <schema>.<table> ...> [-L <locale>]
```

パラメーター

-I <instance_name>

InfoSphere CDC インスタンスの名前を指定します。あるいは、この値の代わりに TSINSTANCE 環境変数を指定することもできます。

-s <subscription_names>

サブスクリプションの名前を指定します。サブスクリプションを複数指定する場合は、リスト形式で指定してください。

-A InfoSphere CDC がサブスクリプション内でリフレッシュ対象としてすべてのソース表にフラグを立てることを指定します。

-t <schema>.<table>

InfoSphere CDC がサブスクリプション内でリフレッシュ対象としてフラグを立てるソース表の名前を指定します。*schema.table* というフォーマットで表の名前を指定する必要があります。表を複数指定する場合は、リスト形式で指定してください。

-L <locale>

InfoSphere CDC インスタンスに使用するロケールの名前。デフォルトはマシンのロケールです。

結果

このコマンドは、成功した場合には 0 の値を返し、失敗した場合にはゼロ以外の値を返します。

例

```
dmflagforrefresh -I myinstance -s Finance -A
```

InfoSphere CDC は、指定されたインスタンスの **Finance** サブスクリプション内でリフレッシュ対象としてすべてのソース表にフラグを立てます。

dmmarktablecapturepoint: ソース表に表キャプチャー・ポイントのマークを付ける

このコマンドを使用して、ソース表に表キャプチャー・ポイントのマークを付け、その表をアクティブな状態にします。このコマンドの実行前に表を変更した場合は、その変更内容は複製されません。

変更データのストリーム内の既存の位置をオーバーライドするときは、ソース表に表キャプチャー・ポイントのマークを付けてください。これは、Management Console 以外のアプリケーション (例えばデータベース・プラットフォームのインポート機能またはエクスポート機能など) を使用してソース表とターゲット表を既に同期 (リフレッシュ) しており、ソースとターゲットが相互に同期する時点を知っているときに可能です。InfoSphere CDC は、変更データのストリーム内の現在位置からターゲット表に変更内容をミラーリングします。「Map Tables」ウィザードで表をマップした後に「Mirror (Change Data Capture)」を選択すると、InfoSphere CDC がこの位置を設定します。InfoSphere CDC が設定した位置をオーバーライドする場合は、Management Console で表キャプチャー・ポイントのマークを手動で付けることができます。サブスクリプションのミラーリングを開始する場合、データベースの変更内容をキャプチャーし、ターゲットに複製する時点として設定した位置を InfoSphere CDC が識別します。

構文

```
dmmarktablecapturepoint -I <instance_name> -s <subscription_names>  
<-A|-t <schema>.<table> ...> [-L <locale>]
```

パラメーター

-I <instance_name>

InfoSphere CDC インスタンスの名前を指定します。あるいは、この値の代わりに TSINSTANCE 環境変数を指定することもできます。

-s <subscription_names>

サブスクリプション名を指定します。サブスクリプションを複数指定する場合は、リスト形式で指定してください。

-A InfoSphere CDC がサブスクリプション内のすべてのソース表を対象に、変更データのストリーム内の既存の位置をオーバーライドすることを指定します。

-t <schema>.<table>

InfoSphere CDC が表キャプチャー・ポイントのマークを付けるサブスクリプシ

ョン内のソース表の名前を指定します。*schema.table* というフォーマットで表の名前を指定する必要があります。表を複数指定する場合は、リスト形式で指定してください。

-L <locale>

InfoSphere CDC インスタンスに使用するロケールの名前。デフォルトはマシンのロケールです。

結果

このコマンドは、成功した場合には 0 の値を返し、失敗した場合にはゼロ以外の値を返します。

例

```
dmmarktablecapturepoint -I myinstance -s Finance -A
```

InfoSphere CDC は、**Finance** サブスクリプション内のすべてのソース表を対象に、変更データのストリーム内の既存の位置をオーバーライドします。

```
dmmarktablecapturepoint -I myinstance -s Finance -t myschema.mytable
```

InfoSphere CDC は、**Finance** サブスクリプション内の指定された表をアクティブにします。

dmpark: 表の停止

このコマンドを使用して、ソース表を停止します。ソース表を停止することにより、サブスクリプション内のその表については、変更内容をキャプチャーしないことを InfoSphere CDC に指示します。表を停止した場合、InfoSphere CDC は、それ以降にソース表に変更があっても、その変更内容を複製しなくなるため、ソース表とターゲット表との間に不整合が発生する可能性があります。

注: ソース表を停止にできるようにするため、その表をターゲットにミラーリングしている場合には、サブスクリプションのレプリケーションを終了させる必要があります。詳しくは、108 ページの『dmendreplication: レプリケーションの終了』を参照してください。

構文

```
dmpark -I <instance_name> -s <subscription_names> [-A|-t <schema>.<table> ...> [-L <locale>]
```

パラメーター

-I <instance_name>

InfoSphere CDC インスタンスの名前を指定します。あるいは、この値の代わりに TSINSTANCE 環境変数を指定することもできます。

-s <subscription_names>

サブスクリプション名を指定します。サブスクリプションを複数指定する場合は、リスト形式で指定してください。

-A InfoSphere CDC がサブスクリプション内のすべてのソース表を停止することを指定します。

-t <schema>.<table>

InfoSphere CDC が停止するサブスクリプション内のソース表の名前を指定します。 *schema.table* というフォーマットで表の名前を指定する必要があります。表を複数指定する場合は、リスト形式で指定してください。

-L <locale>

InfoSphere CDC インスタンスに使用するロケールの名前。デフォルトはマシンのロケールです。

結果

このコマンドは、成功した場合には 0 の値を返し、失敗した場合にはゼロ以外の値を返します。

例

```
dmpark -I myinstance -s Finance -A
```

InfoSphere CDC は **Finance** サブスクリプションのすべてのソース表を停止します。

dmreadddtable: ソース表の定義の更新

このコマンドを使用して、InfoSphere CDC メタデータの中にあるソース表の定義を更新します。RDBMS を使用してソース表の定義を変更した後に、このコマンドを実行してください。

構文

```
dmreadddtable -I <instance_name> [-A|-t <schema>.<table> ...] [-L <locale>]
```

パラメーター

-I <instance_name>

InfoSphere CDC インスタンスの名前を指定します。あるいは、この値の代わりに TSINSTANCE 環境変数を指定することもできます。

-A レプリケーションで使用可能なすべてのソース表の定義を InfoSphere CDC が更新することを指定します。

-t <schema>.<table>

InfoSphere CDC が定義を更新するサブスクリプション内のソース表の名前を指定します。 *schema.table* というフォーマットで表の名前を指定する必要があります。表を複数指定する場合は、リスト形式で指定してください。

-L <locale>

InfoSphere CDC インスタンスに使用するロケールの名前。デフォルトはマシンのロケールです。

結果

このコマンドは、成功した場合には 0 の値を返し、失敗した場合にはゼロ以外の値を返します。

例

```
dmreadddtable -I new_instance -A
```

InfoSphere CDC は、レプリケーションで使用可能なすべてのソース表の定義を更新します。

dmreassigntable: ターゲット表の定義の更新

このコマンドを使用して、InfoSphere CDC メタデータの中にあるターゲット表の定義を更新します。RDBMS を使用してターゲット表の定義を変更した後に、このコマンドを実行してください。

構文

```
dmreassigntable -I <instance_name> -s <subscription_names>  
<-A|-t <schema>.<table> ...> [-L <locale>]
```

パラメーター

-I <instance_name>

InfoSphere CDC インスタンスの名前を指定します。あるいは、この値の代わりに TSINSTANCE 環境変数を指定することもできます。

-s <subscription_names>

表を含む InfoSphere CDC サブスクリプションを指定します。サブスクリプションを複数指定する場合は、リスト形式で指定してください。

-A InfoSphere CDC がサブスクリプション内のすべてのターゲット表の定義を更新することを指定します。

-t <schema>.<table>

InfoSphere CDC が定義を更新するサブスクリプション内のターゲット表の名前を指定します。schema.table というフォーマットで表の名前を指定する必要があります。表を複数指定する場合は、リスト形式で指定してください。

-L <locale>

InfoSphere CDC インスタンスに使用するロケールの名前。デフォルトはマシンのロケールです。

結果

このコマンドは、成功した場合には 0 の値を返し、失敗した場合にはゼロ以外の値を返します。

例

```
dmreassigntable -I new_instance -s Finance -A
```

InfoSphere CDC は **Finance** サブスクリプション内のすべてのターゲット表の定義を更新します。

dmsetreplicationmethod: レプリケーション方式の設定

このコマンドを使用して、サブスクリプション内の表のレプリケーション方式を変更します。このコマンドを実行すると、InfoSphere CDC は「**Active**」な表の状況を「**Refresh**」に変更します。

注: このコマンドを実行する前に、サブスクリプションのレプリケーションを終了する必要があります。

構文

```
dmsetreplicationmethod -I <instance_name> <-r|-m> -s <subscription_names>  
<-A|-t <schema>.<table> ...> [-L <locale>]
```

パラメーター

-I <instance_name>

InfoSphere CDC インスタンスの名前を指定します。あるいは、この値の代わりに `TSINSTANCE` 環境変数を指定することもできます。

-m 表がレプリケーション方式として「**Mirror (Change Data Capture)**」を使用することを指定します。

-r 表がレプリケーション方式として「**Refresh (Snapshot)**」を使用することを指定します。

-s <subscription_names>

サブスクリプションの名前を指定します。

-A サブスクリプション内のすべての表が、指示されたレプリケーション方式を使用することを指定します。

-t <schema>.<table>

指示されたレプリケーション方式を使用するサブスクリプション内のソース表の名前を指定します。`schema.table` というフォーマットで表の名前を指定する必要があります。表を複数指定する場合は、リスト形式で指定してください。

-L <locale>

InfoSphere CDC インスタンスに使用するロケールの名前。デフォルトはマシンのロケールです。

結果

このコマンドは、成功した場合には 0 の値を返し、失敗した場合にはゼロ以外の値を返します。

例

```
dmsetreplicationmethod -I myinstance -r -s Finance -A
```

指定された InfoSphere CDC インスタンスで、**Finance** サブスクリプション内のすべての表が、レプリケーション方式として「**Refresh**」を使用します。

```
dmsetreplicationmethod -I new_instance -m -s Finance -t acct.taxcodes
```

指定された InfoSphere CDC インスタンスで、**Finance** サブスクリプション内のソース表 `acct.taxcodes` が、レプリケーション方式として「**Mirror**」を使用します。

レプリケーション・コマンドのモニター

このセクションでは、InfoSphere CDC でのレプリケーションのモニターを支援するコマンドについて説明します。

dmclearevents: イベントのクリア

このコマンドを使用して、Management Console の「**Event Log**」ビューからイベントを削除します。

構文

```
dmclearerevents -I <instance_name> [-S|-T-|-B] <-A|-s <subscription_names> ...>
[-L <locale>]
```

パラメーター

-I <instance_name>

InfoSphere CDC インスタンスの名前を指定します。あるいは、この値の代わりに TSINSTANCE 環境変数を指定することもできます。

-S InfoSphere CDC がソースからイベントをクリアすることを指定します。

-T InfoSphere CDC がソースとターゲットの両方からイベントをクリアすることを指定します。S、T、および B オプションのいずれも指定しない場合は、InfoSphere CDC はデフォルトで B を指定したものとみなします。

-B InfoSphere CDC がログ位置を設定するサブスクリプション内のソース表の名前を指定します。schema.table というフォーマットで表の名前を指定する必要があります。表を複数指定する場合は、リスト形式で指定してください。

-A InfoSphere CDC がすべてのサブスクリプションのイベントをクリアすることを指定します。

-s <subscription_names>

InfoSphere CDC が指示されたサブスクリプションのイベントをクリアすることを指定します。サブスクリプションを複数指定する場合は、リスト形式で指定してください。

-L <locale>

InfoSphere CDC インスタンスに使用するロケールの名前。デフォルトはマシンのロケールです。

結果

このコマンドは、成功した場合には 0 の値を返し、失敗した場合にはゼロ以外の値を返します。

例

```
dmclearerevents -I myinstance -S -A
```

InfoSphere CDC は、指定されたインスタンスのサブスクリプションすべてのソースからイベントをクリアします。

```
dmclearerevents -I myinstance -T -s Finance Marketing
```

InfoSphere CDC は、指定されたインスタンスの **Finance** サブスクリプションと **Marketing** サブスクリプションのソースとターゲットの両方からイベントをクリアします。

dmgetsubscriptionstatus: サブスクリプションの状況の取得

このコマンドを使用して、サブスクリプションの現在の状態を示す情報をリトリートし、標準出力に結果を送信します。

構文

```
dmgetsubscriptionstatus -I <instance_name> [-p] <-A|-s <subscription_name> ...> [-L <locale>]
```

パラメーター

-I <instance_name>

InfoSphere CDC インスタンスの名前を指定します。あるいは、この値の代わりに TSINSTANCE 環境変数を指定することもできます。

-p InfoSphere CDC が状態情報を標準出力に送信することを指定します。

-A InfoSphere CDC がすべてのサブスクリプションの状態情報をリトリートすることを指定します。

-s <subscription_name>

状態情報をリトリートするサブスクリプションの名前を指定します。サブスクリプションを複数指定する場合は、リスト形式で指定してください。

-L <locale>

InfoSphere CDC インスタンスに使用するロケールの名前。デフォルトはマシンのロケールです。

結果

このコマンドは、以下のいずれかを返します。

- 0: 指定されたサブスクリプションの状態が「**Inactive**」になっている場合。
- 1: 指定されたサブスクリプションのいずれかの状態が「**Inactive**」以外になっている場合。
- 負の値: 状況情報のリトリート中にエラーが発生した場合。

例

```
dmgetsubscriptionstatus -I myinstance -p -A
```

InfoSphere CDC は、すべてのサブスクリプションの状態情報をリトリートし、結果を指定されたインスタンスの標準出力に送信します。

dmshowevents: InfoSphere CDC イベントの表示

このコマンドを使用して、InfoSphere CDC イベントを標準出力に表示します。InfoSphere CDC イベントを Management Console の「Event Log」ビューで表示する代わりに、このコマンドを使用することができます。

このコマンドの出力では、最新のイベントがリストの先頭に来るように発生順にイベントが表示されます。

構文

```
dmshowevents -I <instance_name> <-a|-s <subscription> ...> [-t <source_ID> ...|-s <subscription> ... -t <source_ID> ...> [-h] [-c max_msg] [-L <locale>]
```

または

```
dmshowevents -I <instance_name> <-a|-s <subscription>|-t <source_ID> ...> [-h] [-c max_msg] [-L <locale>]
```


パラメーター

-I <instance_name>

InfoSphere CDC インスタンスの名前を指定します。あるいは、この値の代わりに TSINSTANCE 環境変数を指定することもできます。

- a** InfoSphere CDC がすべてのサブスクリプションのイベントを表示することを指定します。

-s <subscription>

InfoSphere CDC がイベントを表示するソース・サブスクリプションの名前を指定します。サブスクリプションを複数指定する場合は、リスト形式で指定してください。

-t <source_ID>

InfoSphere CDC がイベントを表示するソース ID を指定します。ソース ID を複数指定する場合は、リスト形式で指定してください。

- h** InfoSphere CDC がイベントのリストの前にヘッダーを表示することを指定します。このオプションは、イベントごとに表示される情報の各項目を識別するのに役立ちます。

-c max_msg

InfoSphere CDC が表示するイベントの最大数を指定します。このパラメーターを省略した場合、またはイベントの総数よりも大きな値を指定した場合、InfoSphere CDC は、指定されたサブスクリプションまたはソース ID、またはその両方のイベントをすべて表示します。

- 最小設定値: 0。イベントは表示されません。
- 最大設定値: 2147483647

-L <locale>

InfoSphere CDC インスタンスに使用するロケールの名前。デフォルトはマシンのロケールです。

結果

このコマンドは、成功した場合には 0 の値を返し、失敗した場合にはゼロ以外の値を返します。

例

```
dmshoevents -I new_instance -s Finance
```

InfoSphere CDC は、指定されたインスタンスの **Finance** サブスクリプションのイベントをすべて表示します。

```
dmshoevents -I myinstance -a -h
```

InfoSphere CDC は、すべてのサブスクリプションのイベントをすべて表示します。指定されたインスタンスのイベントのリストの前にヘッダーが表示されます。

```
dmshoevents -I newinstance -s Finance -t Atlanta -s Marketing -h -c 20  
dmshoevents -I myinstance -s Finance Marketing -t Atlanta -h -c 20
```

InfoSphere CDC は、**Finance** サブスクリプションと **Marketing** サブスクリプション、および Atlanta ソース ID のイベントのうち最新の 20 件を表示します。指定されたインスタンスのイベントのリストの前にヘッダーが表示されます。

出力例

```
EVENTTIME|EVENTSOURCE|ORIGINATOR|EVENTID|SEVERITY|EVENTPROGRAM|EVENTTEXT
```

```
2006-04-21 17:23:08.817|T|ATLANTA|95|Information|class com.datamirror.ts.target.  
publication.c|Transformation Server Communications ending.
```

```
2006-04-21 17:23:08.614|T|ATLANTA|1538|Information|class com.datamirror.ts.target.  
publication.c|---Transformation Server for ATLANTA terminating normally.
```

```
2006-04-21 17:23:08.333|T|ATLANTA|1537|Information|class com.datamirror.ts.target.  
publication.c|Describe conversation with ATLANTA completed successfully.
```

```
2006-04-21 17:23:07.911|T|ATLANTA|1536|Information|class com.datamirror.ts.target.  
publication.c|Describe conversation started by ATLANTA.
```

```
2006-04-21 17:23:07.333|T|ATLANTA|1531|Information|class com.datamirror.ts.target.  
publication.c|Communication with ATLANTA successfully started on Data channel.
```

```
2006-04-21 17:23:06.973|T|ATLANTA|1534|Information|class com.datamirror.ts.engine.a  
|Code page conversation from the source database's code page 1252 to the target  
database's code page Cp1252 for ATLANTA will be performed by the Remote system
```

各レコードのフィールドは、垂直バー（|）で区切られます。これらのフィールドは、出力の先頭行で識別されます。*EVENTSOURCE* フィールドの *S* はソースを表し、*T* はターゲットを表します。

構成コマンドのエクスポートとインポート

このセクションでは、InfoSphere CDC グローバル構成のエクスポートとインポートを行うためのコマンドについて説明します。

dmexportconfiguration: InfoSphere CDC 構成のエクスポート

このコマンドを使用して、InfoSphere CDC のインスタンスのインストール時に設定した構成の詳細をエクスポートします。構成の詳細は、XML 構成ファイルに送信されます。このコマンドで作成した XML ファイルを InfoSphere CDC の別のインスタンスにインポートするために、dmimportconfiguration コマンドを使用することができます。

注: このコマンドは、Management Console で構成されたサブスクリプション固有の設定をエクスポートしません。Management Console 内でサブスクリプション固有の設定を XML ファイルにエクスポートすることができます。詳しくは、Management Console の資料を参照してください。

このコマンドは対話式であり、パスワードの入力を促すプロンプトが出されます。スクリプトの中でこのコマンドを使用することはできません。

構文

```
dmexportconfiguration <path_to_configuration_file> [-L <locale>]
```

パラメーター

<path_to_configuration_file>

エクスポートする XML 構成ファイルの相対パスまたは絶対パス。相対パスは、InfoSphere CDC のインストール・ディレクトリーを基準としたパスです。

-L <locale>

InfoSphere CDC インスタンスに使用するロケールの名前。デフォルトはマシンのロケールです。

結果

このコマンドは、成功した場合には 0 の値を返し、失敗した場合にはゼロ以外の値を返します。

例

```
dmexportconfiguration c:%configurations%configuration.xml
```

InfoSphere CDC は、指定された相対パスに XML ファイルをエクスポートします。

dmimportconfiguration: InfoSphere CDC 構成のインポート

このコマンドを使用して、dmexportconfiguration コマンドで作成した XML ファイルから InfoSphere CDC 構成設定をインポートします。

注: サイレント・インストールと連携してこのコマンドをスクリプトの中で使用することで、複数のシステムに InfoSphere CDC をデプロイできます。

構文

```
dmimportconfiguration <path_to_configuration_file> [-L <locale>]
```

パラメーター**<path_to_configuration_file>**

インポートする XML 構成ファイルの相対パスまたは絶対パス。相対パスは、InfoSphere CDC のインストール・ディレクトリーを基準としたパスです。

-L <locale>

InfoSphere CDC インスタンスに使用するロケールの名前。デフォルトはマシンのロケールです。

結果

このコマンドは、成功した場合には 0 の値を返し、失敗した場合にはゼロ以外の値を返します。

例

```
dmimportconfiguration c:%configurations%configuration.xml
```

InfoSphere CDC は、指定された相対パスから XML 構成ファイルをインポートします。

その他のコマンド

このセクションでは、InfoSphere CDC のバージョンの判別、通信の検査、シャットダウン、InfoSphere CDC の終了 (UNIX サーバーのみ)、システム・パラメーターの設定、メタデータのバックアップを実行する各種コマンドについて説明します。

dmbackupmd: メタデータのバックアップ

このコマンドを使用して、メタデータ・データベースのバックアップ・コピーを作成します。バックアップ・コピーは、InfoSphere CDC インストール・ディレクトリ内の instance/<instance_name>/conf/backup ディレクトリに作成されます。サブスクリプションの構成と表の状況を変更した後に、バックアップしてください。InfoSphere CDC が実行中でも、メタデータをバックアップすることができます。

構文

```
dmbackupmd -I <instance_name> [-L <locale>]
```

パラメーター

-I <instance_name>

InfoSphere CDC インスタンスの名前を指定します。あるいは、この値の代わりに TSINSTANCE 環境変数を指定することもできます。

-L <locale>

InfoSphere CDC インスタンスに使用するロケールの名前。デフォルトはマシンのロケールです。

結果

このコマンドは、成功した場合には 0 の値を返し、失敗した場合にはゼロ以外の値を返します。

dmconfigurets - Configure InfoSphere CDC

このコマンドを使用して、InfoSphere CDC 構成ツールを起動します。このツールを使用すると、インスタンスを作成して、InfoSphere CDC のインストール済み環境を構成できます。

構文

```
dmconfigurets [-L <locale>]
```

パラメーター

-L <locale>

InfoSphere CDC インスタンスに使用するロケールの名前。デフォルトはマシンのロケールです。

結果

このコマンドは、成功した場合には 0 の値を返し、失敗した場合にはゼロ以外の値を返します。

dmmdcommander

このコマンドは、内部使用専用です。

dmmdconsole

このコマンドは、内部使用専用です。

dmset: InfoSphere CDC システム・パラメーターの設定

このコマンドを使用して、InfoSphere CDC システム・パラメーターの表示または変更を行います。Management Console でシステム・パラメーターを変更することもできます。詳しくは、Management Console の資料を参照してください。

注: このコマンドを使用すれば、どのシステム・パラメーターでも設定することができます。ただし、表示されるシステム・パラメーターは、デフォルト以外の値に設定されているもののみです。

構文

```
dmset -I <instance_name> [<parameter_name>[=<parameter_value>]] [-L <locale>]
```

パラメーター

-I <instance_name>

InfoSphere CDC インスタンスの名前を指定します。あるいは、この値の代わりに TSINSTANCE 環境変数を指定することもできます。

<parameter_name>

InfoSphere CDC システム・パラメーターの名前を指定します。

<parameter_value>

システム・パラメーターに割り当てる値を指定します。

-L <locale>

InfoSphere CDC インスタンスに使用するロケールの名前。デフォルトはマシンのロケールです。

結果

このコマンドは、成功した場合には 0 の値を返し、失敗した場合にはゼロ以外の値を返します。

例

```
dmset -I myinstance
```

デフォルト以外の値に設定されているシステム・パラメーターをすべて表示します。

```
dmset -I myinstance global_unicode_as_char=false
```

global_unicode_as_char システム・パラメーターを false に設定します。

```
dmset -I myinstance global_unicode_as_char
```

指定されたパラメーターの現行値を表示します。

```
dmset -I myinstance stop_replication=
```

stop_replication システム・パラメーターを削除します。

dmsetaccessserverparams - Access Server のパラメーターの設定

このコマンドは、Access Server へのアクセス・データおよびログイン・データの定義に使用します。このコマンドは、リフレッシュ・ストアード・プロシージャーを使用する場合に必要です。

構文

```
dmsetaccessserverparams [-u <username>] [-p <password>] [-H <hostname>] [-P <port>]
```

パラメーター

- **-u <username>**—アクセス・マネージャー・ユーザー
- **-p <password>**—アクセス・マネージャー・ユーザーのパスワード
- **-H <hostname>**—Access Server 稼働時のワークステーションのホスト名 (システム名) または全 IP アドレス
- **-P <port>**—Access Server への接続に使用するユニークな TCP/IP 番号。Access Server のインストール時および Management Console へのログオン時に、このポート番号を指定します。デフォルト値は 10101 です。

dmsetaccessserverparams の初回使用時には、すべてのパラメーターを指定します。このパラメーター値を指定しない場合、以下のデフォルト値が設定されます。

- ユーザー - Admin
- パスワード - "" (ブランク)
- ホスト - localhost
- ポート - 10101

全部または一部のパラメーターの値の設定が完了したら、このコマンドをもう一度発行することで、これらの値を変更することができます。

結果

このコマンドは、成功した場合には 0 の値を返し、失敗した場合にはゼロ以外の値を返します。

例

```
dmsetaccessserverparams -u dba -p dba -H localhost -P 10101
```

ユーザー「dba」およびパスワード「dba」のアクセス・データとログイン・データを設定しています。

```
dmsetaccessserverparams -H newmachine
```

Access Server を稼働するワークステーションのホスト名を変更しています。

dmshowversion: InfoSphere CDC バージョンの表示

このコマンドを使用して、InfoSphere CDC のバージョンとビルド番号を表示します。IBM 担当員へのお問い合わせの際は、事前にこのコマンドを実行して、実行している InfoSphere CDC のバージョンとビルド番号をご提供ください。

構文

```
dmshowversion [-L <locale>]
```

パラメーター

-L <locale>

InfoSphere CDC インスタンスに使用するロケールの名前。デフォルトはマシンのロケールです。

結果

このコマンドは、成功した場合には 0 の値を返し、失敗した場合にはゼロ以外の値を返します。

dmshutdown: InfoSphere CDC のシャットダウン

このコマンドを使用して、InfoSphere CDC を正常にシャットダウンします。保守のためにサーバーやデータベースをオフラインにする前、または InfoSphere CDC を最新バージョンにアップグレードする前に、このコマンドを使用することができます。

このコマンドを実行する前に、Management Console ですべてのサブスクリプションのレプリケーションを終了して、確実にシャットダウンを完了するようにしてください。詳しくは、Management Console の資料を参照してください。

このコマンドが InfoSphere CDC のシャットダウンを完了できない場合には、dmterminate コマンドを使用して、強制的にシャットダウンを完了してください。

構文

```
dmshutdown -I <instance_name> [-L <locale>]
```

パラメーター

-I <instance_name>

InfoSphere CDC インスタンスの名前を指定します。あるいは、この値の代わりに TSINSTANCE 環境変数を指定することもできます。

-L <locale>

InfoSphere CDC インスタンスに使用するロケールの名前。デフォルトはマシンのロケールです。

結果

このコマンドは、成功した場合には 0 の値を返し、失敗した場合にはゼロ以外の値を返します。

dmterminate: InfoSphere CDC プロセスの強制終了

注: このコマンドは Windows ではサポートされていません。

このコマンドを使用して、dmshutdown コマンドではシャットダウンを完了できない UNIX サーバーまたは Linux サーバー上で実行中のインスタンスについて、すべて

の InfoSphere CDC プロセスを強制終了します。InfoSphere CDC は、このコマンドの実行に使用した UNIX アカウントで開始したプロセスだけを終了します。

保守のためにサーバーやデータベースをオフラインにする前、または InfoSphere CDC を最新バージョンにアップグレードする前に、このコマンドを使用することができます。

InfoSphere CDC を正常にシャットダウンするには、dmshutdown コマンドを使用してください。dmshutdown が InfoSphere CDC のシャットダウンを完了できない場合は、dmterminate を使用して、dmshutdown の実行後も残っているアクティブな InfoSphere CDC プロセスを強制終了してください。

構文

```
dmterminate [-L <locale>]
```

パラメーター

-L <locale>

InfoSphere CDC インスタンスに使用するロケールの名前。デフォルトはマシンのロケールです。

結果

このコマンドは、成功した場合には 0 の値を返し、失敗した場合にはゼロ以外の値を返します。

dmmts32: InfoSphere CDC の開始

このコマンドを使用して、InfoSphere CDC の 32 ビット・バージョンを開始します。

構文

```
dmmts32 -I <instance_name> [-L <locale>]
```

パラメーター

-I <instance_name>

開始する InfoSphere CDC インスタンスを指定します。

-L <locale>

InfoSphere CDC インスタンスに使用するロケールの名前。デフォルトはマシンのロケールです。

結果

このコマンドは、成功した場合には 0 の値を返し、失敗した場合にはゼロ以外の値を返します。

例

```
dmmts32 -I -I myinstance
```

InfoSphere CDC が指定されたインスタンスに対して開始します。

dmts64: InfoSphere CDC の開始

このコマンドを使用して、InfoSphere CDC の 64 ビット・バージョンを開始します。

構文

```
dmts64 -I <instance_name> [-L <locale>]
```

パラメーター

-I <instance_name>

開始する InfoSphere CDC インスタンスを指定します。

-L <locale>

InfoSphere CDC インスタンスに使用するロケールの名前。デフォルトはマシンのロケールです。

結果

このコマンドは、成功した場合には 0 の値を返し、失敗した場合にはゼロ以外の値を返します。

例

```
dmts64 -I myinstance
```

InfoSphere CDC が指定されたインスタンスに対して開始します。

InfoSphere CDC のユーザー出口

ユーザー出口を使用して、指定された表でデータベース・イベントが発生する前または発生した後に InfoSphere CDC で実行可能な、一連のアクションを定義することができます。ユーザー出口を使用すれば、ビジネス要件に合わせて環境をカスタマイズすることができます。

Java クラスまたはストアード・プロシージャのユーザー出口をコンパイルした後、Management Console でユーザー出口を構成することができます。ユーザー出口の構成については詳しくは、Management Console の資料の『ユーザー出口の構成』を参照してください。

InfoSphere CDC と一緒にインストールされる Javadoc (API) 情報には、InfoSphere CDC で使用可能な Java クラスのユーザー出口に関する詳細なクラス仕様とインターフェース仕様が記載されています。インターフェースごとに、サポートされる呼び出し可能なメソッドが識別されます。

ユーザー出口に関する Javadoc (API) の資料は、<system drive>:\%<installation directory>%docs%api ディレクトリにあります。ご使用のブラウザでヘルプを開くには、index.html をクリックします。

サンプルのユーザー出口が InfoSphere CDC と共に提供されています。これらのサンプルを環境に合わせて拡張または変更することができます。

表レベルおよび行レベルの操作のためのストアード・プロシージャ ー・ユーザー出口

ストアード・プロシージャは、データベース内に物理的に格納されたプログラム(すなわちプロシージャ)です。ストアード・プロシージャの利点は、ユーザー要求に応じて実行される場合に、データベース・エンジンによって直接実行されるということです。データベース・エンジンは通常、独立したデータベース・サーバー上で動作し、一般的にデータベース要求の処理が高速です。

ユーザー出口プログラムを作成してコンパイルした後、Management Console の「User Exits」タブで、どのユーザー出口ポイント(行レベル操作の前後、または表レベル操作の前後)でユーザー出口を実行するか指定できます。

ストアード・プロシージャ・ユーザー出口の定義

InfoSphere CDC でストアード・プロシージャを定義する場合、以下の点を検討してください。

- 多重定義されたストアード・プロシージャはサポートされません。
- ストアード・プロシージャには少なくとも 2 つのパラメーターが存在し、以下の順序で先頭の 2 つに定義する必要があります。
 - result。整数出力パラメーターで、イベント・ログに任意のエラー・コードを返すのに使用します。
 - returnMsg。文字出力パラメーターで、ログに記録するエラー・メッセージを返すのに使用します。

ストアード・プロシージャ・ユーザー出口のデータベース接続

ストアード・プロシージャ・ユーザー出口プログラムと InfoSphere CDC は、データベースに接続するデフォルト方式と同じ共有接続を使用します。この設定により、デフォルトで、InfoSphere CDC が表に対して行った変更が、ストアード・プロシージャ・ユーザー出口プログラムから確認できるようになります。

ストアード・プロシージャ・ユーザー出口でのデータのリトリート

ストアード・プロシージャにシステム・パラメーターを渡すことにより、ソース表からデータをリトリートできます。以下のタイプのデータをリトリートできます。

- システム値のリトリート (s\$)。s\$ 接頭部をストアード・プロシージャに渡すと、ストアード・プロシージャでソース・データベースのシステム値が使用可能になります。例えば、s\$entry は、InfoSphere CDC がユーザー出口を実行したエントリー・ポイントを識別します。
- ジャーナル管理フィールドのリトリート (j\$)。j\$ 接頭部をストアード・プロシージャに渡すと、ストアード・プロシージャでソース・データベースのジャーナル管理フィールドが使用可能になります。例えば、j\$USER は、ソース表で更新を行った人のユーザー ID を識別します。これは、ソース表で行われた表レベルまたは行レベルの操作を、ストアード・プロシージャを使用して監査する場合に役に立ちます。

- **データ値のリトリート**。ストアード・プロシージャーに渡す接頭部に応じて、ソース・データベースからデータをリトリートして、ストアード・プロシージャーで使用可能にすることができます。例えば、b\$ を使用して、ソース列の更新前イメージをリトリートすることができます。

これらの各値は、ユーザーが作成したストアード・プロシージャー・ユーザー出口に対する入力パラメーターとして使用できます。データのリトリートに使用するフォーマットは、使用している製品によって多少異なります。

- InfoSphere CDC では、フォーマットは `<x>${value}` です。

ここで、`<x>` は接頭部を表しており、`<value>` はリトリートされる値の名前を表しています。

s\$ 接頭部を使用したシステム値のリトリート

この接頭部は、システム値のリトリートに使用されます。以下の表にこれらの値を示し、簡単に説明します。

接頭部と値	データ型	説明
s\$entry	NUMBER	<p>ストアード・プロシージャが実行されたエントリー・ポイントを示します。以下のエントリー・ポイントからストアード・プロシージャを実行できます。</p> <ul style="list-style-type: none"> • 1: InfoSphere CDC が表クリア (切り捨て) 操作の前にストアード・プロシージャを実行したことを示します。 • 2: InfoSphere CDC が表クリア (切り捨て) 操作の後にストアード・プロシージャを実行したことを示します。 • 3: InfoSphere CDC が行挿入操作の前にストアード・プロシージャを実行したことを示します。 • 4: InfoSphere CDC が行挿入操作の後にストアード・プロシージャを実行したことを示します。 • 5: InfoSphere CDC が行更新操作の前にストアード・プロシージャを実行したことを示します。 • 6: InfoSphere CDC が行更新操作の後にストアード・プロシージャを実行したことを示します。 • 7: InfoSphere CDC が行削除操作の前にストアード・プロシージャを実行したことを示します。 • 8: InfoSphere CDC が行削除操作の後にストアード・プロシージャを実行したことを示します。 • 9: InfoSphere CDC が表リフレッシュ操作の前にストアード・プロシージャを実行したことを示します。 • 10: InfoSphere CDC が表リフレッシュ操作の後にストアード・プロシージャを実行したことを示します。

接頭部と値	データ型	説明
s\$srcSysId	VARCHAR	ソース・データの場所をユニークに識別します。
s\$srcTabId	VARCHAR	複製データをターゲットに送信するソース・データベース内のソース表の名前を表します。
s\$tgtTabId	VARCHAR	ソースから複製データを受信するターゲット・データベース内のターゲット表の名前を表します。

j\$ 接頭部を使用したジャーナル管理フィールドのリトリブ

この接頭部は、ソース・システム上で行われた操作に関する情報のリトリブに使用されます。InfoSphere CDC で **jb\$** を使用して、同じ情報をリトリブできます。

以下に、使用可能な値をリストします。

接頭部と値	データ型	説明
j\$CCID	VARCHAR	挿入、更新、または削除の操作を含むトランザクションを識別します。
j\$CODE	VARCHAR	ジャーナル・エントリーまたはログ・エントリーのタイプを識別します。リフレッシュ操作では「U」、ミラーリングでは「R」が使用されます。
j\$CTRR または j\$CNTRRN	VARCHAR	ジャーナル・エントリーまたはログ・エントリーを記録したソース表の相対レコード番号を識別します。 注: CTRR または CNTRRN には、リフレッシュを構成する挿入エントリーに対してストアード・プロシージャを実行する場合に、意味のある情報が含まれます。
j\$ENTT または j\$ENTTYP	VARCHAR	ソース・システム上で操作タイプを識別するジャーナル・コードまたはログ・コードを生成します。
j\$JRN または j\$JOURNAL	VARCHAR	InfoSphere CDC が挿入、更新、または削除の操作を読み取るジャーナルまたはログの名前。

接頭部と値	データ型	説明
j\$JOB	VARCHAR	ソース・システム上で挿入、更新、または削除を行ったジョブの名前を識別します。
j\$MBR または j\$MEMBER	VARCHAR	ソース表の名前またはその別名を識別します。
j\$NBR または j\$JOBNO	VARCHAR	挿入、更新、または削除の操作を行っているソース表上のプログラムのプロセス ID を識別します。
j\$PGM または j\$PROGRAM	VARCHAR	挿入、更新、または削除の操作を行ったソース・システム上のプログラムの名前を識別します。
j\$SEQN または j\$SEQNO	VARCHAR	ジャーナルまたはログ内の挿入、更新、または削除の操作のシーケンス番号を識別します。
j\$SYNM または j\$SYSTEM	VARCHAR	ソース・システムのホスト名を識別します。
j\$USER	VARCHAR	ソース上で挿入、更新、または削除の操作を行ったデータベース・ユーザーの名前を識別します。
j\$USPF	VARCHAR	ソース上で挿入、更新、または削除の操作を行ったオペレーティング・システム・ユーザーの名前を識別します。
j\$TSTP または j\$TIMESTAMP	VARCHAR	ソース上で挿入、更新、または削除の操作、またはリフレッシュを行った日時を識別します。マイクロ秒の精度をサポートする環境では、このジャーナル管理フィールドの日時フォーマットは YYYY-MM-DD-HH:MM:SS.UUUUUU です。その他の場合、InfoSphere CDC は、マイクロ秒の要素 UUUUUU をゼロに設定するか、またはまったく組み込みません。

b\$, a\$, k\$, および d\$ 接頭部を使用したデータ値のリトリブ

データのリトリブには、4 つの接頭部が使用されます。

接頭部	モード	説明
b\$<source column name>	入力	<p>ソース列内のデータの更新前イメージのリトリブに使用します。更新前イメージは、いずれのトランスフォーマーションも適用される前の、ソース表列からのオリジナル・イメージです。</p> <p>例えば、ソース表に対して、以下の UPDATE を行ったとします。</p> <pre>UPDATE source_table set MYCOLUMN = 2 where MYCOLUMN = 1;</pre> <p>これで、この SQL ステートメントを実行する前に MYCOLUMN が 1 であったすべての行が 2 に設定されます。</p> <p>ストアード・プロシージャを定義し、そのストアード・プロシージャで MYCOLUMN の更新前イメージをリトリブする場合には、以下のように指定します。</p> <pre>b\$MYCOLUMN;</pre> <p>これで、値 1 が返されます。</p>

接頭部	モード	説明
a\$<source column name>	入力	<p>ソース列内のデータの更新後イメージのリトリブに使用します。更新後イメージは、ソース表列からの、変換されたデータです。例えば、派生式によって変換されたデータです。</p> <p>例えば、ソース表に対して、以下の UPDATE を行ったとします。</p> <pre>UPDATE source_table set MYCOLUMN = 2 where MYCOLUMN = 1;</pre> <p>これで、この SQL ステートメントを実行する前に MYCOLUMN が 1 であったすべての行が 2 に設定されます。</p> <p>ストアード・プロシージャを定義し、そのストアード・プロシージャで MYCOLUMN の更新後イメージをリトリブする場合には、以下のように指定します。</p> <pre>a\$MYCOLUMN;</pre> <p>これで、値 2 が返されます。</p>
k\$<target key column name>	入力	<p>変更が必要な行を検索するために、ターゲット表へのアクセスに使用します。</p> <p>注: キー列は、監査には使用できません。</p>
d\$<target column name>	入出力	<p>トランスフォーメーション後のデータ値のリトリブに使用します。このデータ値は、ターゲット・データベース内の表の更新に使用されます。ストアード・プロシージャでは、これらの値のみを変更できます。</p>

ストアード・プロシージャ・ユーザー出口の例

以下のコード・スニペットは、ストアード・プロシージャ・ユーザー出口の例です。

コード	コメント
<pre> create or replace procedure PROD.AUDIT_STPROC (result OUT INT, returnMsg OUT CHAR, s\$entry IN NUMBER, s\$srcSysId IN CHAR, s\$srcTabId IN CHAR, s\$tgtTabId IN CHAR, j\$ENTT IN CHAR, a\$IDNO IN NUMBER, a\$PRICE IN NUMBER, a\$DESC IN CHAR, a\$LONGDESC IN CHAR, a\$TRANSDATE IN DATE, d\$IDNO IN NUMBER, d\$PRICE IN NUMBER, d\$DESC IN CHAR, d\$LONGDESC IN CHAR, d\$TRANSDATE IN DATE) </pre>	<p>宣言してストアード・プロシージャに渡すパラメーターは、有効なデータ型でなければなりません。</p> <p>以下のパラメーターは必須のもので、ストアード・プロシージャ内で宣言しておく必要があります。</p> <p>result: ストアード・プロシージャが成功したことを示す「0」の値か、エラーを示す整数を返します。</p> <p>returnMsg: イベント・ログにエラー・メッセージを返します。</p> <p>このストアード・プロシージャでは、以下のパラメーターが宣言されています。</p> <ul style="list-style-type: none"> • s\$entry: ストアード・プロシージャが呼び出されたエントリー・ポイントをリトリートします。この例では、InfoSphere CDC が各エントリー・ポイントでユーザー出口を呼び出します。 • s\$srcSysId: ソース・データの場所をリトリートします。 • s\$srcSysId: ソース・データの場所をリトリートします。 • s\$srcTabId: ソース表の名前をリトリートします。 • s\$srcTabId: ソース表の名前をリトリートします。 • s\$tgtTabId: ターゲット表の名前をリトリートします。 • s\$tgtTabId: ターゲット表の名前をリトリートします。 • j\$ENTT: ソース表で行われた操作のタイプを示すジャーナル・コードをリトリートします。 • j\$ENTT: ソース表で行われた操作のタイプを示すジャーナル・コードをリトリートします。 • a\$: IDNO、PRICE、DESC、LONGDESC、および TRANSDATE のソース列の更新後イメージをリトリートします。 • a\$: IDNO、PRICE、DESC、LONGDESC、および TRANSDATE のソース列の更新後イメージをリトリートします。 • d\$: IDNO、PRICE、DESC、LONGDESC、および TRANSDATE のターゲット列のトランスフォーム済みデータをリトリートします。 • d\$: IDNO、PRICE、DESC、LONGDESC、および TRANSDATE のターゲット列のトランスフォーム済みデータをリトリートします。
<pre> IS ENTRYPOINT VARCHAR(50); BEGIN CASE s\$entry WHEN 16 THEN ENTRYPOINT := 'User Exit program called Before Insert'; WHEN 1048576 THEN ENTRYPOINT := 'User Exit program called After Insert'; WHEN 64 THEN ENTRYPOINT := 'User Exit program called Before Update'; WHEN 4194304 THEN ENTRYPOINT := 'User Exit program called After Update'; END CASE; </pre>	<p>このストアード・プロシージャ・ユーザー出口は、これらのエントリー・ポイントから呼び出すことができます。</p>

コード	コメント
<pre>insert into PROD.AUDIT_TABLE1 values (s\$entry, s\$srcSysId, s\$srcTabId, s\$tgtTabId, j\$ENTT, a\$IDNO, a\$PRICE, a\$DESC, a\$LONGDESC, a\$TRANSDATE, d\$IDNO, d\$PRICE, d\$DESC, d\$LONGDESC, d\$TRANSDATE, ENTRYPOINT);</pre>	<p>このストアード・プロシージャ・ユーザー出口は、これらの値を、<i>PROD.AUDIT_TABLE1</i> に挿入します。</p>
<pre>result := 1; returnMsg := 'OK!'; END AUDIT_STPROC;</pre>	<p>このストアード・プロシージャ・ユーザー出口は成功しました。</p> <p>注: ストアード・プロシージャが '0' を返した場合は、メッセージがイベント・ログに生成されます。</p>

InfoSphere CDC のサンプル・ユーザー出口

InfoSphere CDC には、ユーザーが自分の環境に合わせて拡張または変更できるサンプル・ユーザー出口が用意されています。サンプルは、InfoSphere CDC インストール・ディレクトリーの `samples` ディレクトリーに置かれている `samples.jar` に入っています。Java ファイルには、以下のサンプルが含まれています。

- **ArchiveLogPathUserExitSample.java** — Oracle アーカイブ・ログ・ファイルへの絶対パス (ファイル名と拡張子を含む) を返します。このサンプルは、`com.datamirror.ts.target.publication.userexit.sample` にあります。
- **DEUserExitSample.java** — `%USERFUNC` 列関数を使用する式の中で使用されます。これはユーザーが提供した (式の中の) パラメーターの合計を計算し、その合計に 1 を加えた値を返します。このサンプルは、`com.datamirror.ts.derivedexpressionmanager` にあります。
- **SPUserExitSample.java** — ソースから着信するイメージを使用して、ストアード・プロシージャを呼び出します。このサンプルは、`com.datamirror.ts.target.publication.userexit.sample` にあります。
- **UserExitSample.java** — レプリケーション・イベントにサブスクライブして、発生したイベントの詳細をリトリブします。このサンプルは、`com.datamirror.ts.target.publication.userexit.sample` にあります。
- **UserExitSample1.java** — ターゲット上の表に挿入された新しい行を記録し、それらの行をテキスト・ファイルに保管します。ユーザーは、テキスト・ファイル名をパラメーターで指定します。このサンプルは、`com.datamirror.ts.target.publication.userexit.sample` にあります。
- **PopWindow**。このサンプルは、ダイアログ・ボックスを開いて、通知情報を表示します。X-Windows などの GUI エミュレーターが存在する場合、このサンプルは、UNIX および Linux のインストールで動作します。

以下の点に注意してください。

- サンプル・ユーザー出口を変更せずに実行するには、Management Console で、コンパイルしたユーザー出口への絶対パスを指定する必要があります。例えば、`com.datamirror.ts.target.publication.userexit.sample.UserExitSample` です。
- コンパイルしたサンプル・ユーザー出口は `ts.jar` ファイルに入っており、このファイルは、InfoSphere CDC インストール・ディレクトリーの `lib` ディレクト

リーにあります。ts.jar ファイル内のコンパイルされたユーザー出口には、*.class 拡張子が付くことに注意してください。

- サンプル・ユーザー出口を変更したい場合には、ソース・コードに変更を行った後、そのユーザー出口をコンパイルする必要があります。
- ユーザー出口クラスは、ユーザーのクラスパスにも含まれている必要があります。

Management Console で Java クラスまたはストアード・プロシージャー・ユーザー出口を指定する方法については、Management Console の資料を参照してください。

サンプル・ユーザー出口をコンパイルするには (Windows)

手順

1. InfoSphere CDC を停止します。
2. samples.jar ファイルを、InfoSphere CDC インストール・フォルダー内の lib フォルダーに unzip します。jar ファイルの unzip 時には、フォルダー構造を必ず維持してください。

jar ファイルの unzip 後、以下のようなフォルダー構造になります。

```
<InfoSphere CDC installation folder>%lib%com%datamirror%ts%target%publication%userexit%sample
```

3. サンプル・ユーザー出口を変更します。
4. 変更済みのユーザー出口をコンパイルします。例えば、UserExitSample.java をコンパイルする場合は、コマンド・ウィンドウを開き、lib フォルダーにナビゲートして以下のコマンドを実行します。

```
javac -classpath ts.jar;. com%datamirror%ts%target%publication%userexit%sample%UserExitSample.java
```

このコマンドの実行が成功すると、画面には何も出力されません。

注: このコマンドを実行するには、システムに Java JDK が必要です。

5. コマンドの実行に成功したら、以下のディレクトリーにナビゲートして、UserExitSample.class ファイルを作成したことを確認してください。

```
<InfoSphere CDC installation directory>%lib%com%datamirror%ts%target%publication%userexit%sample
```

6. InfoSphere CDC を開始します。
7. ユーザー出口を構成するための最終ステップとして、Management Console で UserExitSample の絶対パスを指定します。例えば、以下のように指定します。
com.datamirror.ts.target.publication.userexit.sample.UserExitSample

注: .class 拡張子は指定しないでください。

次のタスク

Management Console での Java クラス・ユーザー出口の指定方法については、Management Console の資料を参照してください。

注: 実稼働環境でサンプル・ユーザー出口を使用する場合は、デプロイ前にサンプルをテストする必要があります。IBM は、変更またはカスタマイズされたユーザー出口クラスによって生じた不利な結果に対して責任を負いません。

サンプル・ユーザー出口をコンパイルするには (UNIX および Linux) 手順

1. InfoSphere CDC を停止します。
2. samples.jar ファイルを、InfoSphere CDC インストール・ディレクトリー内の lib ディレクトリーに unzip します。jar ファイルの unzip 時は、ディレクトリー構造を必ず維持してください。

jar ファイルの unzip 後、以下のようなディレクトリー構造になります。

```
<InfoSphere CDC installation directory>/lib/com/datamirror/ts/target  
/publication/userexit/sample
```

3. サンプル・ユーザー出口を変更します。
4. 変更済みのユーザー出口をコンパイルします。例えば、UserExitSample.java をコンパイルする場合は、コマンド・ウィンドウを開き、lib ディレクトリーにナビゲートして以下のコマンドを実行します。

```
javac -classpath ts.jar:. com/datamirror/ts/target/publication/userexit/sample  
/UserExitSample.java
```

このコマンドの実行が成功すると、画面には何も出力されません。

注: このコマンドを実行するには、システムに Java JDK が必要です。

5. コマンドの実行に成功したら、以下のディレクトリーにナビゲートして、UserExitSample.class ファイルを作成したことを確認してください。

```
<InfoSphere CDC installation directory>/lib/com/datamirror/ts/target  
/publication/userexit/sample
```

6. InfoSphere CDC を開始します。
7. ユーザー出口を構成するための最終ステップとして、Management Console で UserExitSample の絶対パスを指定します。例えば、以下のように指定します。
com.datamirror.ts.target.publication.userexit.sample.UserExitSample

注: .class 拡張子は指定しないでください。

次のタスク

Management Console での Java クラス・ユーザー出口の指定方法について詳しくは、Management Console の資料を参照してください。

注: 実稼働環境でサンプル・ユーザー出口を使用する場合は、デプロイ前にサンプルをテストする必要があります。IBM は、変更またはカスタマイズされたユーザー出口クラスによって生じた不利な結果に対して責任を負いません。

競合解決監査表

InfoSphere CDC は、ソース表とターゲット表の競合を解決するときに、解決に関する情報を TS_CONFAUD 表に記録します。InfoSphere CDC は、InfoSphere CDC の構成時に指定したターゲット・メタデータの場所にこの表を作成します。

このセクションでは、以下の内容を説明します。

競合解決監査表の構造

TS_CONFAUD 表を使用して、競合解決がターゲット表に与えた影響を追跡できます。例えば、AFTERIMG 列を照会して、ターゲット表が変更された時点を確認できます。次に、BEFOREIMG 列と AFTERIMG 列の内容を調べて、ターゲット表のデータの基になった、ソース表での変更内容を確認できます。これは、競合解決戦略で問題を識別するのに役立ちます。

競合検出と解決は、Management Console で構成します。詳しくは、Management Console の資料を参照してください。

以下に、TS_CONFAUD 表の構造を示します。

列	説明
CNFTIME	ターゲット上で競合が検出された日時。
SRCTIME	競合データがソース表に適用された時刻。
SRCSYSID	サブスクリプションのソース ID。
SRCSHEMA	ソース表のスキーマ名またはライブラリー名。
SRCNAME	ソース表の名前。
SRCMEMBER	このフィールドはブランクです。
TGTSCHEMA	ターゲット表のスキーマまたはライブラリー。
TGTNAME	ターゲット表の名前。
OPTYPE	競合の原因となったソース上での行レベルの操作。以下のいずれかの値を取ります。 <ul style="list-style-type: none">• 1: ソース表に行が挿入されました。• 2: ソース表で行が更新されました。• 3: ソース表から行が削除されました。
CNFTYPE	検出された競合のタイプ。以下のいずれかの値を取ります。 <ul style="list-style-type: none">• 1: ソース表に行が挿入されました。その行のキーは、既にターゲット表に存在します。• 2: ソース表で行が更新または削除されました。その行のキーは、ターゲット表に存在しません。• 3: ソース表で行が更新または削除されました。ソース表とターゲット表のイメージが一致しません。• 4: 予期しない競合が検出されました。

列	説明
RESMTD	<p>競合解決方式が使用されました。以下のいずれかの値を取ります。</p> <ul style="list-style-type: none"> • 1: ソースが優先 • 2: ターゲットが優先 • 3: 最大値が優先 • 4: 最小値が優先 • 5: ユーザー出口 <p>解決方式が None の場合、この表には行が挿入されません。これらの方式については、InfoSphere CDC の資料を参照してください。</p>
CNFRES	<p>競合が解決されたかどうかを示します。以下のいずれかの値を取ります。</p> <ul style="list-style-type: none"> • Y: 競合が解決されました。 • N: 競合が解決されませんでした。
BEFOREIMG	<p>変更前のソース表内の行の表記。この列のフォーマットについて詳しくは、146 ページの『行イメージ・フォーマット』を参照してください。</p>
BEFORETRNC	<p>BEFOREIMG に格納された更新前イメージが切り捨てられたかどうかを示します。以下のいずれかの値を取ります。</p> <ul style="list-style-type: none"> • Y: 値が切り捨てられました。 • N: 値が切り捨てられませんでした。
AFTERIMG	<p>変更後のソース表内の行の表記。この列のフォーマットについて詳しくは、146 ページの『行イメージ・フォーマット』を参照してください。</p>
AFTERTRNC	<p>AFTERIMG に格納された更新後イメージが切り捨てられたかどうかを示します。以下のいずれかの値を取ります。</p> <ul style="list-style-type: none"> • Y: 値が切り捨てられました。 • N: 値が切り捨てられませんでした。
TGTIMG	<p>レプリケーションが行われる前のターゲット表内の行の表記。この列のフォーマットについて詳しくは、146 ページの『行イメージ・フォーマット』を参照してください。</p>
TGTRNC	<p>TGTIMG に格納されたイメージが切り捨てられたかどうかを示します。以下のいずれかの値を取ります。</p> <ul style="list-style-type: none"> • Y: 値が切り捨てられました。 • N: 値が切り捨てられませんでした。
WINIMG	<p>競合解決が行われた後のターゲット表内の最終行の表記。この列のフォーマットについて詳しくは、146 ページの『行イメージ・フォーマット』を参照してください。</p>

列	説明
WINTRNC	WINIMG に格納されたイメージが切り捨てられたかどうかを示します。以下のいずれかの値を取ります。 <ul style="list-style-type: none"> • Y: 値が切り捨てられました。 • N: 値が切り捨てられませんでした。

行イメージ・フォーマット

監査表内の BEFOREIMG、AFTERIMG、TGTIMG、および WINIMG 列は、ソース表またはターゲット表内の行の表記を示しています。

これらの列内のイメージは、ターゲット・メタデータ・データベース上の VARCHAR データの最大長で制限されます。これらのイメージには、raw、バイナリー、および LOB 列内のデータを除いて、行内のすべての値が含まれます。各列からのデータは、以下のフォーマットで表示されます。

(length:value)

上記のフォーマットで、*value* は列内のデータ、および *length* はデータの表記に使用される文字の数です。イメージでは、数値データは文字ストリングとして表示され、NULL 値は (ヌル) として表示されます。

行イメージは、ソース表および競合解決監査表内の列順序と一致します。これらのイメージは、ターゲット・メタデータ・データベース内の VARCHAR データの最大長より長い場合には切り捨てられる可能性があります。表のキー列は、表内の最初の列でない場合には切り捨てられる可能性があります。

切り捨てられたイメージ

行イメージは、VARCHAR 列の最大長よりも長いと切り捨てられます。監査表に、各イメージ列が切り捨てられたかどうかを示す列があります。例えば、WINTRNC が Y の場合、WINIMG の値は切り捨てられています。切り捨てられた列のフォーマットは、以下のようになります。

(-length:value)

上記のフォーマットで、*value* は切り捨てられた値であり、*length* は切り捨てられたストリング内の文字数です。

監査対象外のデータ型

監査表では、そのイメージ内に以下のデータ型の列は含まれません。

- IMAGE
- NTEXT
- TEXT

ソース表またはターゲット表にこれらのデータ型の行が含まれる場合、イメージは、これらの行を単に見過ごします。バイナリー・データは、16 進数にエンコードされた文字としてイメージ内に現れます。イメージは、サポートされない列からは、どのような情報も格納しません。

Management Console 管理ガイドの付録

このセクションでは、solidDB に固有のユーザー出口およびシステム・パラメーターに関する情報を提供します。このセクションは、「InfoSphere Change Data Capture Management Console 管理ガイド」の付録です。

ユーザー出口の構成

ユーザー出口を使用して、指定された表でデータベース・イベントが発生する前または発生した後に InfoSphere CDC で実行可能な、一連のアクションを定義できます。InfoSphere CDC を使用するとき、行レベルの操作または表レベルの操作としてデータベース・イベントを定義します。行レベルの操作には、挿入、更新、削除があります。表レベルの操作には、リフレッシュ、切り捨て操作があります。例えば、InfoSphere CDC が特定のターゲット表に削除操作を複製した後でアラートを送信する、行レベルのユーザー出口プログラムを構成できます。

ユーザー出口は、「Before User Exit」または「After User Exit」としてグループ化できます。

- **Before User Exit:** InfoSphere CDC が行レベルまたは表レベルの操作をターゲット表に複製する前に、実行されます。
- **After User Exit:** InfoSphere CDC が行レベルまたは表レベルの操作をターゲット表に複製した後に、実行されます。

以下のリストで、行レベルまたは表レベルの操作の前または後のユーザー出口プログラムを作成する共通のシナリオを示します。

- InfoSphere CDC が行レベルの操作をターゲット表に複製するタイミングをカスタマイズする。例えば、特定の基準 (オリジナルの請求書の日付など) に基づいて挿入、更新、または削除の操作が行われるように、これらの操作のロジックを開発できます。InfoSphere CDC は、オリジナルの請求書の日付 (2004 年 1 月、2004 年 2 月、2006 年 11 月など) に基づいて、ユーザー出口を実行し、行レベルの操作 (挿入、更新、または削除) を適切なターゲット表に適用できます。
- デフォルトの行レベルまたは表レベルの操作を無効にして、カスタム操作を実行するユーザー出口プログラムを起動することで置き換える。例えば、表レベルの切り捨て操作に応じて、ターゲット表で永久的な削除ではなく一時的な削除を実行するユーザー出口を作成できます。

InfoSphere CDC for solidDB のユーザー出口の構成

InfoSphere CDC for solidDB では、Java クラス・ユーザー出口を構成できます。

Java クラス・ユーザー出口のメソッド名は、事前定義済みです。つまり、ユーザー出口プログラムを有効または無効にすることのみが可能で、InfoSphere CDC for solidDB で提供される UserExitIF インターフェース・クラスをインプリメントするユーザー出口を Java で構成する必要があります。

Java クラスのユーザー出口を構成するには:
手順

1. 「**Configuration**」 → 「**Subscriptions**」をクリックします。
2. サブスクリプションを選択します。
3. 「**Table Mappings**」ビューをクリックし、表マッピングを選択します。

4. 「**Edit Mapping Details**」を右クリックし、選択します。
5. 「**User Exits**」タブをクリックします。
6. 「**User Exit Type**」リストから「**Java Class**」を選択します。
7. 「**Class Name**」ボックスに、**UserExitIF** インターフェースをインプリメントする Java クラス・ユーザー出口の名前を入力します。

例えば、UserExitIF インターフェースをインポート済みであれば、関数でこのインターフェースをインプリメントするユーザー出口プログラム・クラスの定義は、`public class UE1 implements UserExitIF` のようになります。

「**Class Name**」ボックスには、以下を入力する必要があります。

オプション	説明
UE1	スタンドアロン・クラスの場合。
<Java package>.UE1	クラスが Java パッケージに含まれている場合 (com.datamirror.interface.UE1 など)。

ユーザー出口プログラムをコンパイルすることで生成されるファイルは、CLASSPATH 環境変数で参照されるライブラリーまたはフォルダーに置く必要があります。

8. 「**Parameter**」ボックスに、ユーザー出口プログラムで使用可能にするパラメーターを入力します。

初期化プロセスで `getParameter()` メソッドを呼び出して、ユーザー出口プログラム・クラスのパラメーターにアクセスできます。パラメーターの指定に関する規則はありません。このボックスに入力する値は、フリー・フォームです。パラメーター値のストリングの長さは、255 文字を超えることはできません。

9. 以下の操作 (複数可) のほかに InfoSphere CDC で呼び出すユーザー出口プログラムの名前を入力します。

オプション	説明
挿入前	挿入操作を複製する前に、InfoSphere CDC がユーザー出口を実行します。
挿入後	挿入操作を複製した後に、InfoSphere CDC がユーザー出口を実行します。
更新前	更新操作を複製する前に、InfoSphere CDC がユーザー出口を実行します。
更新後	更新操作を複製した後に、InfoSphere CDC がユーザー出口を実行します。
削除前	削除操作を複製する前に、InfoSphere CDC がユーザー出口を実行します。
削除後	削除操作を複製した後に、InfoSphere CDC がユーザー出口を実行します。
リフレッシュ前	リフレッシュ操作を複製する前に、InfoSphere CDC がユーザー出口を実行します。

オプション	説明
リフレッシュ後	リフレッシュ操作を複製した後に、InfoSphere CDC がユーザー出口を実行します。
切り捨て前	切り捨て操作を複製する前に、InfoSphere CDC がユーザー出口を実行します。
切り捨て後	切り捨て操作を複製した後に、InfoSphere CDC がユーザー出口を実行します。

10. 「Apply」をクリックします。

InfoSphere CDC for solidDB のシステム・パラメーター

システム・パラメーターを使用して、InfoSphere CDC の動作を制御できます。レプリケーション環境で特定の構成が必要な場合は、システム・パラメーターを使用して、InfoSphere CDC のデフォルト操作の動作を変更できます。デフォルトのシステム・パラメーター設定は、ほとんどのインストール済み環境に適しています。InfoSphere CDC の構成を理解するまでは、これらのデフォルト設定を維持してください。

InfoSphere CDC には、ソース・データ・ストアおよびターゲット・データ・ストアの動作を制御するシステム・パラメーターがあります。

注:

- アクティブ・レプリケーション時にシステム・パラメーターを変更する場合は、変更を有効にするために、InfoSphere CDC を停止し、再開してください。
- InfoSphere CDC の高位バージョンにアップグレードするときに、システム・パラメーターの既存の設定はすべて維持されます。

一般製品システム・パラメーター

一般製品システム・パラメーターを使用して、InfoSphere CDC の基本機能およびインストール時に指定した情報を制御できます。

retrieve_credentials:

このシステム・パラメーターは、InfoSphere CDC が solidDB Universal Cache で SQL パススルーを目的としてバックエンド・ログイン・データのフェッチを試行するかどうかを定義します。

このパラメーターは、以下のいずれかに設定します。

- true — InfoSphere CDC が、solidDB Universal Cache で SQL パススルーを目的としたバックエンド・ログイン・データのフェッチを試行することを示します。
- false — InfoSphere CDC が solidDB Universal Cache で SQL パススルーを目的としたバックエンド・ログイン・データのフェッチを試行しないことを示します。バックエンド・データ・サーバーが DB2 for iSeries、または DB2 for z/OS である場合、「false」に設定する必要があります。

適用先: ソース

デフォルト設定: true

通知システム・パラメーター

通知システム・パラメーターを使用して、特定のイベントに対して「**Event Log**」で InfoSphere CDC メッセージを生成するかどうかを制御できます。

global_shutdown_after_no_heartbeat_response_minutes:

このシステム・パラメーターを使用して、サブスクリプションのアクティブな InfoSphere CDC 処理が停止するまでに、通信が非アクティブである期間を分単位で指定します。許容範囲外の値が指定された場合、デフォルト設定が使用されます。

適用先: ソース

デフォルト設定: 15 分

最小設定値: 3 分

最大設定値: 999 分

global_conversion_not_possible_warning:

このシステム・パラメーターを使用して、以下の状況で、InfoSphere CDC が Management Console の「**Event Log**」に警告を生成するかどうかを制御します。

- 特定のデータ値で、データ変換ができない。
- 範囲外である変換後のデータ型が検出された。

このパラメーターは、以下のいずれかに設定します。

true: 特定のデータ値でデータ変換ができない、または範囲外である変換後のデータ型が検出された場合に、「**Event Log**」に警告を生成します。

false: 特定のデータ値でデータ変換ができない、または範囲外である変換後のデータ型が検出された場合に、「**Event Log**」に警告を生成しません。

適用先: ターゲット

デフォルト設定: False

スループット最大化システム・パラメーター

InfoSphere CDC システム・パラメーターを使用して、ミラーリング時に、ターゲット・データベースの作業負荷を大幅に削減できます。InfoSphere CDC アプライ・プロセスは、ターゲットのトランザクションをグループ化して、作業負荷を削減します。ターゲット・データベース上のどのコミットも、ソース上のコミットに対応します。ただし、ソースで実行されたすべてのコミットが実行されるわけではありません。例えば、ソースが、それぞれに 1 つの操作が含まれる 3 つの小さなトランザクションを実行する場合、ターゲットは 3 つのすべての操作を単一トランザクションの一部としてコミットできます。このシステム・パラメーターのグループ化を使用して、ターゲット・データベースに必要なリソースを大幅に削減できます。デフォルト設定はほとんどのデータベースに適していますが、ターゲット・システムのリソースが限定されていて、待ち時間の増加を許容できる場合は、この設定を調整できます。

mirror_commit_after_max_transactions:

このシステム・パラメーターは、コミットの前にグループ化するトランザクションの最大数を指定します。通常、ターゲット・データベースに発行されるコミットは、ソースで実行されているアプリケーションが発行したコミットに対応します。このシステム・パラメーターを使用して、ターゲット・データベースに発行される頻度を制御することで、コミットを管理できます。このアプローチを使用すると、データベースへの頻繁なコミットのオーバーヘッドを削減できます。

適用先: ターゲット

デフォルト設定: 10

最小設定値: 1

mirror_commit_after_max_seconds:

このシステム・パラメーターは、ターゲット・データベースに小さなトランザクションをコミットするまでの時間を秒単位で指定します。通常、ターゲット・データベースに発行されるコミットは、ソースで実行されているアプリケーションが発行したコミットに対応します。このシステム・パラメーターを使用して、ターゲット・データベースに発行される頻度を制御することで、コミットを管理できます。このアプローチを使用すると、データベースへの頻繁なコミットのオーバーヘッドを削減できます。

適用先: ターゲット

デフォルト設定: 1 秒

最小設定値: 1

mirror_commit_after_max_operations:

このシステム・パラメーターは、コミットを発行する前にターゲット・データベースに適用する必要がある操作の数を指定します。通常、ターゲット・データベースに発行されるコミットは、ソースで実行されているアプリケーションが発行したコミットに対応します。このシステム・パラメーターを使用して、ターゲット・データベースに発行される頻度を制御することで、コミットを管理できます。このアプローチを使用すると、データベースへの頻繁なコミットのオーバーヘッドを削減できます。

適用先: ターゲット

デフォルト設定: 1000

最小設定値: 1

mirror_commit_on_transaction_boundary:

このシステム・パラメーターは、InfoSphere CDC がターゲット・データベースで行うコミットが、常にソース・データベースで発生したコミットに対応するかどうかを示します。ソース・データベースのコミットメント制御を無視すると、InfoSphere CDC で、大きなトランザクションの部分的な結果を表示できるようになります。

このパラメーターは、以下のいずれかに設定します。

- **true:** ソース・データベースのコミットメント制御を無視しません。コミットされたトランザクションのレコードだけがターゲットにミラーリングされます。この設定では、コミットされたトランザクションだけをターゲットに送信することにより、真のトランザクション整合性が提供されます。
- **false:** ソース・データベースのコミットメント制御を無視します。この値は、トランザクション処理のコミットメント制御を無効にします。ミラーリング時に、トランザクション整合性の維持を試みません。

適用先: ターゲット

デフォルト設定: true

refresh_commit_after_max_operations:

このシステム・パラメーターは、リフレッシュ時に各トランザクションを構成する行の数を識別します。リフレッシュ時のターゲット・データベースのワークロードを削減するために、InfoSphere CDC は、リフレッシュを単一の大きなトランザクションとして実行するのではなく、定期的にターゲット・データベースに変更をコミットします。

適用先: ターゲット

デフォルト設定: 1000

最小設定値: 1

エンコード・システム・パラメーター

システム・パラメーターによっては、定義されている Unicode 列のデータを処理するデフォルト方式を設定し、データベースにデフォルトの文字エンコードを設定できます。

global_unicode_as_char:

このシステム・パラメーターは、定義されている Unicode 列のデータを処理するデフォルト方式を示します。サーバー上の InfoSphere CDC インストールごとに、このシステム・パラメーターは、Unicode 列のデータを処理するシステム・デフォルト方式を定義します。Unicode 列がシステム・デフォルトに設定されている場合、このシステム・パラメーターで定義されているとおりに、現行のシステム・デフォルト方式が使用されます。

このパラメーターは、以下のいずれかに設定します。

- **true:** InfoSphere CDC は、Unicode 列のすべてのデータを 1 バイト文字として処理します。この設定は、Unicode 列に 1 バイト文字データが含まれているときに使用します。
- **false:** InfoSphere CDC は、Unicode 列のすべてのデータを連続したビット・ストリームとして処理します。この設定は、Unicode 列に 1 バイト以外の文字データが含まれているときに使用します。このシステム・パラメーターを false に設定すると、InfoSphere CDC は、以前の InfoSphere CDC リリースと同じように 1 バイト以外の文字データを処理します。

注: このパラメーターを `false` に設定することは、複製された Unicode 列の 1 バイト以外の文字データがターゲットで正しく表示されることを保証するわけではありません。複製された 1 バイト以外の文字データについては、Unicode 列のデータが正しく表示されるように、ユーザー出口プログラムまたはその他のカスタマイズの適用が必要になることがあります。ユーザー出口プログラムについては詳しくは、ご使用のプラットフォームの「*InfoSphere CDC* エンド・ユーザー向け資料」を参照してください。

適用先: ソース

デフォルト設定: `false`

ディスク・リソース・システム・パラメーター

システム・パラメーターには、InfoSphere CDC のメモリー使用法を制御するものがあります。パフォーマンスを向上させるために、InfoSphere CDC Java 仮想マシンにデフォルト値である 512 MB より大きな値を割り振ることができる場合は、増大したメモリーを使用するようにディスク・リソース・システム・パラメーターを調整できます。

mirror_memory_txqueue_total_mb:

このシステム・パラメーターは、ソースでデータのステージングに使用するメモリーの容量を制御します。パフォーマンスを最適化するために、このシステム・パラメーターは、ソース・データベースに今後存在するコミットされていないデータの最大量を保持するのに十分な大きさの値にする必要があります。

適用先: ソース

デフォルト設定: 15 メガバイト

mirror_memory_txqueue_each_mb:

このシステム・パラメーターは、ソースでデータのステージングに使用するメモリーの容量を制御します。パフォーマンスを最適化するために、このシステム・パラメーターは、ソースで発生する最大のトランザクションのデータを保持するのに十分な大きさの値にする必要があります。

適用先: ソース

デフォルト設定: 3 メガバイト

global_memory_lob_cache_mb:

このシステム・パラメーターは、ターゲットで LOB 値のステージングに使用するメモリーの容量を制御します。パフォーマンスを最適化するために、この値は、複製される最大の LOB 値のデータ全体を保持するのに十分な大きさの値にする必要があります。

デフォルト設定: 2 メガバイト

適用先: ターゲット

mirror_queue_for_buffers_between_cdc_threads_operations:

このシステム・パラメーターは、マルチ・プロセッサを利用する InfoSphere CDC のログの収集機能を制御します。ほとんどの状況で、デフォルト設定を受け入れることができます。拡張が非常に容易な環境では、この値を増やすことができます。

適用先: ソース

デフォルト設定: 100 項目

最小設定値: 100 項目

アプライ・プロセス・システム・パラメーター

システム・パラメーターには、InfoSphere CDC が行、列、データ、およびエラー処理を適用する方法を調整するものがあります。

mirror_end_on_error:

このシステム・パラメーターを使用して、ターゲット・データベースでアプライ・エラーが発生した後で、ミラーリングを終了するかどうかを示します。

このパラメーターは、以下のいずれかに設定します。

- **true:** ターゲット・データベースのアプライ・エラーの後、ミラーリングを終了します。
- **false:** ターゲット・データベースのアプライ・エラーの後、ミラーリングを終了しません。

適用先: ターゲット

デフォルト設定: true

refresh_end_on_error:

このシステム・パラメーターを使用して、アプライ・エラーが発生した後で、リフレッシュを終了するかどうかを示します。

このパラメーターは、以下のいずれかに設定します。

- **true:** アプライ・エラーが発生した後で、リフレッシュを終了します。
- **false:** アプライ・エラーが発生した後で、リフレッシュを終了しません。

適用先: ターゲット

デフォルト設定: true

refresh_with_referential_integrity:

このシステム・パラメーターを使用して、すべてのターゲット表から削除されたデータを再び追加する前に、リフレッシュするかどうかを示します。これは、リフレッシュする表に参照整合性制約がある場合に非常に便利です。

このパラメーターは、以下のいずれかに設定します。

- **true** — InfoSphere CDC は最初に、指定したリフレッシュ順序とは逆にすべてのデータを削除することを示します。リフレッシュ順序を指定する場合、一般的に、参照する子表の前に親表を表示する必要があります。

- false - InfoSphere CDC は、最初に表からすべてのデータを削除せずに、指定された順序で表をリフレッシュすることを示します。

適用先: ソース

デフォルト設定: false

solid_fast_refresh_apply_pipes:

このシステム・パラメーターを使用して、高速リフレッシュのパフォーマンスを向上させます。高速リフレッシュ機能により、バックエンド・データ・サーバーから solidDB フロントエンドへの大容量のデータの複製に必要な時間が削減されます。

このパラメーターの値を、システム内のプロセッサ (コア) の数に設定します。

適用先: ターゲット

デフォルト設定—2

関連資料

『solid_fast_refresh_on』

solid_fast_refresh_on:

このシステム・パラメーターを使用して、高速リフレッシュ機能を制御します。高速リフレッシュ機能により、バックエンド・データ・サーバーから solidDB フロントエンドへの大容量のデータの複製に必要な時間が削減されます。

このパラメーターは、以下のいずれかに設定します。

- true — 高速リフレッシュが有効であることを示します。
- false — 高速リフレッシュが無効であることを示します。

適用先: ターゲット

デフォルト設定: false

関連資料

『solid_fast_refresh_apply_pipes』

付録 A. ログ・リーダーのパラメーター

ログ・リーダーのパラメーターは、クライアント・サイドの `solid.ini` 構成ファイルの `[LogReader]` セクションで指定します。

表 22. ログ・リーダーのパラメーター

[LogReader]	説明	ファクトリー値	アクセス・モード
LogReaderEnabled	<p>このパラメーターを使用すると、ログ・リーダー機能の有効/無効を切り替えることができます。</p> <p>solidDB Universal Cache および InfoSphere CDC レプリケーションを使用する構成では、このパラメーターを Yes に設定する必要があります。</p>	なし	RO (読み取り専用)
MaxLogSize	<p>このパラメーターは、ディスク・ベースのトランザクション・ログの保護部分のサイズを定義します。</p> <p>例えば、バックアップの後で、ログ・ファイルを削除した場合、少なくとも指定したサイズのログ・データは保持されます。ログの保護部分により、レプリケーションが長い間アクティブでなかったときに、障害が発生した後で、可能なキャッチアップを簡単に実施できるようになります。</p> <p>ログ・ファイルを削除しないと、実際のログ・サイズが MaxLogSize の値を超える場合があります。プロパゲーターのログ位置が既存ログ内である限り、キャッチアップは可能です。</p> <p>最小値は 5 (5 MB) です。これよりも小さいログ・サイズを定義しようとすると、値は自動的に 5 MB に変更されます。指定可能な最大ログ・サイズは、事実上、制限がありません。</p> <p>単位: メガバイト。</p>	10240	RW

表 22. ログ・リーダーのパラメーター (続き)

[LogReader]	説明	ファクトリー値	アクセス・モード
MaxSpace	<p>このパラメーターは、スローダウンの前にバッファーに入れられるログ・レコードの最大数を定義します。</p> <p>ログ・レコードは、インメモリー・ログ・リーダー・バッファーに入れられます。ログ・レコードのサイズは、(バイナリーの) 行サイズに追加メタデータ・オーバーヘッドの数バイトを加算したものになります。</p> <p>バッファーがいっぱいになると、solidDB では、スループット・スロットルが適用され、ログ・リーダー・バッファーに空きができるまで、操作がブロックされます。</p> <p>スロットル処理は、ログの読み取りがアクティブの場合にのみ実行されます。ログ・リーダーのアクティビティが存在しない場合、solidDB は処理を続行し、少なくとも定義された MaxLogSize の限度に到達するまで、ログ・ファイルは保持されます (上記を参照)。</p>	100000	RW
MaxMemLogSize	<p>ロギングが有効でない (Logging.LogEnabled = No) の場合における、メモリー内のログ・リーダーのログ・ファイルの最大サイズ。最大サイズに到達すると、ログ・リーダーによるキャッチアップが実施できなくなる可能性があります。</p> <p>単位: メガバイト。</p>	1 MB	RW
サイレント	<p>「Yes」に設定した場合、Log Reader アクティビティは solmsg.out に出力されません。</p> <p>可能な値は、「Yes」と「No」です。</p>	No	RW/Startup

付録 B. SQL パススルー・パラメーター

SQL パススルーのパラメーターは、クライアント・サイドの `solid.ini` 構成ファイルの `[Passthrough]` セクションに指定されています。

表 23. SQL パススルー・パラメーター

[Passthrough]	説明	ファクトリー値	アクセス・モード
ComplexNumNonindexedConstr	<p>このパラメーターは、複合ステートメント内の非索引 WHERE 節制約の最小数を指定します。</p> <p>ステートメントに含まれる非索引制約 (WHERE 節制約が索引で解決しないか、索引が存在しないか、あるいは 옵ティマイザーが異なる索引を制約に選択するかのいずれかのタイプ) の数とその最小数より少ない場合、そのステートメントは複合ステートメントではなく、バックエンドにパススルーされません。</p> <p>値 0 (ゼロ) は、複合ステートメントであるかどうかの推定時に、その非索引制約の数が使用されないことを意味します。</p>	0	RW
ComplexNumOrderedRows	<p>このパラメーターは、複合ステートメントでソートする必要のある行の最小推定数を指定します。</p> <p>ステートメントに含まれるソート可能な行の数がその推定数よりも少ない場合、そのステートメントは複合ステートメントではなく、バックエンドにパススルーされません。</p> <p>値 0 (ゼロ) は、複合ステートメントであるかどうかの推定時に、そのソート可能な行の数が使用されないことを意味します。</p>	0	RW
ComplexNumTables	<p>このパラメーターは、複合ステートメント内の表の最小数を指定します。</p> <p>ステートメントの表の数が、このパラメーターで指定された表の数より少ない場合、このステートメントは複合ステートメントではなく、バックエンドにパススルーされません。</p> <p>値 0 (ゼロ) は、複合ステートメントであるかどうかの推定時に、その表の数が使用されないことを意味します。</p>	0	RW

表 23. SQL パススルー・パラメーター (続き)

[Passthrough]	説明	ファクトリー値	アクセス・モード
ErrorMapFileName	<p>バックエンドのネイティブ・エラー・コードを solidDB のエラー・コードにマッピングするためのファイル・パスとファイル名を指定します。</p> <pre><file_path><file_name></pre> <p>例えば、以下のように指定します。</p> <pre>[Passthrough] ErrorMapFileName=myfiles/db2tosoliderrors.txt</pre> <p>ErrorMapFileName が定義されていない場合、またはエラーがマップされない場合、ネイティブ・バックエンド・エラー・コードは solidDB エラー 13456 (Passthrough backend error: SQLState=<value>, NativeError=<back-end error identifier>, MessageText=<back-end error description>) へマップされます。</p> <p>マッピング・ファイル内の項目のフォーマットは、以下のとおりです。</p> <pre><backend_error> <solidDB error> ; rest of the line is comment</pre> <p>solid.ini 構成ファイルの場合のように、コメントを追加するためにセミコロンが使用されます。</p> <p>例:</p> <pre>; this file maps DB2 native errors to solidDB native errors -207 13015 ; column not found -407 13110 ; NULL not allowed for non NULL column ; end of errormappings</pre> <p>その他のマッピング・ファイルの例については、solidDB インストール・ディレクトリーの samples/sqlpassthrough ディレクトリーを参照してください。</p>	ファクトリー値なし。	RW/Startup
Force32bitODBCHandles	<p>Force32bitODBCHandles パラメーターは、バックエンド・データ・サーバーが DB2 for Linux、UNIX、および Windows であり、IBM Data Server Driver for CLI and ODBC が直接リンクとともに使用されている場合に、64 ビット環境が必要です。</p> <p>「yes」に設定すると、solidDB サーバーは ODBC ハンドルを、64 ビット・プラットフォームのネイティブである 64 ビット void ポインタではなく、32 ビット整数として扱います。</p>	no	RW/Startup
IgnoreOnDisabled	<p>IgnoreOnDisabled パラメーターは、アプリケーション・プログラムがパススルーが無効であることを認識する方法を定義します。この値が「yes」の場合は、パススルーに関連するすべてのステートメント (SET PASSTHROUGH ...) が無視されます。この値が「no」の場合は、これらのステートメントを実行しようとして、エラーが返されます。</p> <p>指定できる値は、「yes」と「no」です。</p>	yes	R/W

表 23. SQL パススルー・パラメーター (続き)

[Passthrough]	説明	ファクトリー値	アクセス・モード
PassthroughEnabled	<p>PassthroughEnabled パラメーターは、SQL パススルーの有効/無効を定義します。</p> <ul style="list-style-type: none"> パススルーが有効であるのに、これを初期化できない (例えば、ドライバが検出されない) 場合、ステートメントをバックエンドに渡そうとするたびに、エラーが返されます。 バックエンド・サーバーが、制御された方法でシャットダウンされると、PassthroughEnabled パラメーターの値は動的に「no」に設定することができます。その後、アプリケーションに公開される動作は、IgnoreOnDisabled パラメーターを使用して定義されます。 <p>指定できる値は、「yes」と「no」です。</p>	no	RW/Startup
RemoteServerDriverPath	<p>RemoteServerDriverPath パラメーターは、solidDB がリンクされるバックエンド・データ・サーバー固有の ODBC ドライバに関する、ドライバ・マネージャーのパスまたはドライバのパスを指定します。</p>		RW/Startup
RemoteServerDSN	<p>RemoteServerDSN パラメーターは、solidDB がリンクされるバックエンド・データ・サーバー固有の ODBC ドライバに関する、データ・ソース名 (ドライバ・マネージャーを使用する場合) または接続ストリングを指定します。</p> <p>接続ストリングは、ServerNam のように、ODBC 呼び出し SQLConnect() のフォーマットで指定する必要があります。</p>		RW/Startup
SqlPassthroughRead	<p>SqlPassthroughRead パラメーターは、solidDB サーバーからバックエンドに読み取りステートメントを渡す方法を定義します。</p> <p>指定できる値は、「None」、「Conditional」、および「Force」です。</p>	None	R/W
SqlPassthroughWrite	<p>SqlPassthroughWrite パラメーターは、solidDB サーバーからバックエンドに書き込みステートメントを渡す方法を定義します。</p> <p>指定できる値は、「None」、「Conditional」、および「Force」です。</p>	None	R/W

付録 C. SQL パススルーでの ODBC データ型のサポート

SQL パススルーは、solidDB がサポートするすべての標準 SQL 標準データ型をサポートします。

バックエンド・データ・サーバーに固有の専有データ型は、サポートされません。

サポートされているデータ型

表 24. サポートされているデータ型

SQL タイプ ID [1]	標準的な SQL データ型 [2]	標準的なタイプの説明
SQL_CHAR	CHAR(n)	固定長 n の文字ストリング。
SQL_VARCHAR	VARCHAR(n)	最大長 n の可変長文字ストリング
SQL_LONGVARCHAR	LONG VARCHAR	可変長文字データ。 最大長は、データ・ソースに応じて異なります。[9]
SQL_WCHAR	WCHAR(n)	固定長 n の Unicode 文字ストリング。
SQL_WVARCHAR	VARWCHAR(n)	最大長 n の Unicode 可変長文字ストリング。
SQL_WLONGVARCHAR	LONGWVARCHAR	Unicode 可変長文字データ。 最大長は、データ・ソースに応じて異なります。
SQL_DECIMAL	DECIMAL(p,s)	最小精度が p で位取りが s である符号付きの厳密な数値。 最大精度は、ドライバーで定義されます。 $1 \leq p \leq 15, s \leq p[4]$
SQL_NUMERIC	NUMERIC(p,s)	精度が p で位取りが s である符号付きの厳密な数値。 $1 \leq p \leq 15, s \leq p$ [4]
SQL_SMALLINT	SMALLINT	精度が 5 で位取りが 0 である厳密な数値 (符号あり: $-32,768 \leq n \leq 32,767$, 符号なし: $0 \leq n \leq 65,535$)。[3]
SQL_INTEGER	INTEGER	精度が 10 で位取りが 0 である厳密な数値 (符号あり: $-2[31] \leq n \leq 2[31] - 1$, 符号なし: $0 \leq n \leq 2[32] - 1$)。[3]
SQL_REAL	REAL	2 進数精度が 24 である符号付きの概算数値 (ゼロ、または $10[-38]$ から $10[38]$ までの絶対値)。
SQL_FLOAT	FLOAT(p)	最小の 2 進数精度が p である符号付きの概算数値。 最大精度は、ドライバーで定義されます。 [5]

表 24. サポートされているデータ型 (続き)

SQL タイプ ID [1]	標準的な SQL データ型 [2]	標準的なタイプの説明
SQL_DOUBLE	DOUBLE PRECISION	2 進数精度が 53 である符号付きの概算数値 (ゼロ、または 10[-308] から 10[308] までの絶対値)。
SQL_BIT==> SQL_INTEGER	BIT	シングル・ビット 2 進データ。 [8]
SQL_TINYINT	TINYINT	精度が 3 で位取りが 0 である厳密な数値 (符号あり: -128 <= n <= 127、符号なし: 0 <= n <= 255)。 [3]
SQL_BIGINT	BIGINT	精度が 19 (符号ありの場合) または 20 (符号なしの場合) で、位取りが 0 である厳密な数値 (符号あり: -2[63] <= n <= 2[63] - 1、符号なし: 0 <= n <= 2[64] - 1)。 [3]、 [9]
SQL_BINARY	BINARY(n)	固定長 <i>n</i> の 2 進データ。 [9]
SQL_VARBINARY	VARBINARY(n)	最大長 <i>n</i> の可変長 2 進データ。 最大長は、ユーザーが設定します。 [9]
SQL_LONGVARBINARY	LONG VARBINARY	可変長 2 進データ。 最大長は、データ・ソースに応じて異なります。 [9]
SQL_TYPE_DATE[6]	DATE	年、月、および日のフィールド。グレゴリオ・カレンダーのルールに準拠します。 詳しくは、Microsoft の「 <i>ODBC プログラマーズ・リファレンス (ODBC Programmer's Reference)</i> 」の『グレゴリオ・カレンダーの制約事項 (Constraints of the Gregorian Calendar)』を参照してください。
SQL_TYPE_TIME[6]	TIME(p)	時、分、および秒のフィールド。それぞれの有効値は、時が 00 から 23、分が 00 から 59、および秒が 00 から 61 です。 精度 <i>p</i> は、秒の精度を示します。
SQL_TYPE_TIMESTAMP[6]	TIMESTAMP(p)	年、月、日、時、分、および秒のフィールド。有効値は、DATE および TIME データ型で定義します。

変換されるデータ型

表 25. 変換されるデータ型

SQL タイプ ID [1]	標準的な SQL データ型 [2]	標準的なタイプの説明
SQL_BIT==> SQL_INTEGER	BIT	シングル・ビット 2 進データ。 [8]

サポートされていない SQL 標準データ型

表 26. サポートされていない SQL 標準データ型

SQL タイプ ID [1]	標準的な SQL データ型 [2]	標準的なタイプの説明
SQL_TYPE_UTCDATETIME	UTCDATETIME	年、月、日、時、分、秒、UTC 時、および UTC 分のフィールド。UTC 時と UTC 分のフィールドは、10 分の 1 マイクロ秒精度です。

表 26. サポートされていない SQL 標準データ型 (続き)

SQL タイプ ID [1]	標準的な SQL データ型 [2]	標準的なタイプの説明
SQL_TYPE_UTCTIME	UTCTIME	時、分、秒、UTC 時、および UTC 分のフィールド。UTC 時と UTC 分のフィールドは、10 分の 1 マイクロ秒精度です。
SQL_INTERVAL_MONTH[7]	INTERVAL MONTH(p)	2 つの日付間の月数。p は間隔の主要精度です。
SQL_INTERVAL_YEAR[7]	INTERVAL YEAR(p)	2 つの日付間の年数。p は間隔の主要精度です。
SQL_INTERVAL_YEAR_TO_MONTH[7]	INTERVAL YEAR(p) TO MONTH	2 つの日付間の年と月の数。p は間隔の主要精度です。
SQL_INTERVAL_DAY[7]	INTERVAL DAY(p)	2 つの日付間の日数。p は間隔の主要精度です。
SQL_INTERVAL_HOUR[7]	INTERVAL HOUR(p)	2 つの日付/時刻間の時間数。p は間隔の主要精度です。
SQL_INTERVAL_MINUTE[7]	INTERVAL MINUTE(p)	2 つの日付/時刻間の分数。p は間隔の主要精度です。
SQL_INTERVAL_SECOND[7]	INTERVAL SECOND(p,q)	2 つの日付/時刻間の秒数。p は間隔の主要精度で、q は間隔の秒精度です。
SQL_INTERVAL_DAY_TO_HOUR[7]	INTERVAL DAY(p) TO HOUR	2 つの日付/時刻間の日/時間の数。p は間隔の主要精度です。
SQL_INTERVAL_DAY_TO_MINUTE[7]	INTERVAL DAY(p) TO MINUTE	2 つの日付/時刻間の日/時間/分の数。p は間隔の主要精度です。
SQL_INTERVAL_DAY_TO_SECOND[7]	INTERVAL DAY(p) TO SECOND(q)	2 つの日付/時刻間の日/時間/分/秒の数。p は間隔の主要精度で、q は間隔の秒精度です。
SQL_INTERVAL_HOUR_TO_MINUTE[7]	INTERVAL HOUR(p) TO MINUTE	2 つの日付/時刻間の時間/分の数。p は間隔の主要精度です。
SQL_INTERVAL_HOUR_TO_SECOND[7]	INTERVAL HOUR(p) TO SECOND(q)	2 つの日付/時刻間の時間/分/秒の数。p は間隔の主要精度で、q は間隔の秒精度です。
SQL_INTERVAL_MINUTE_TO_SECOND[7]	INTERVAL MINUTE(p) TO SECOND(q)	2 つの日付/時刻間の分/秒の数。p は間隔の主要精度で、q は間隔の秒精度です。
SQL_GUID	GUID	固定長の GUID。

[1] これは SQLGetTypeInfo の呼び出しによって、DATA_TYPE 列に返される値です。

[2] これは SQLGetTypeInfo の呼び出しによって、NAME および CREATE PARAMS 列に返される値です。NAME 列は名称 (CHAR など) を返すのに対して、CREATE PARAMS 列は、精度、位取り、長さなどの作成パラメーターをコンマ区切りのリストで返します。

[3] アプリケーションは SQLGetTypeInfo または SQLColAttribute を使用して、結果セット内の特定のデータ型または特定の列が符号なしかどうかを判別します。

[4] SQL_DECIMAL および SQL_NUMERIC データ型の相違点は、その精度のみです。DECIMAL(p,s) の精度は、インプリメンテーションで定義される p 以上の 10 進数精度であり、NUMERIC(p,s) の精度は厳密に p です。

[5] インプリメンテーションに応じて、SQL_FLOAT の精度は 24 または 53 のいずれかになります。精度が 24 の場合は、SQL_FLOAT データ型は SQL_REAL と同じであり、53 の場合は、SQL_DOUBLE と同じです。

[6] ODBC 3.x では、SQL の日付、時刻、およびタイム・スタンプのデータ型は、それぞれ SQL_TYPE_DATE、SQL_TYPE_TIME、および SQL_TYPE_TIMESTAMP です。ODBC 2.x では、これらのデータ型は、SQL_DATE、SQL_TIME、および SQL_TIMESTAMP です。

[7] 間隔の SQL データ型の詳細については、Microsoft の「*ODBC プログラマーズ・リファレンス (ODBC Programmer's Reference)*」の『間隔のデータ型 (Interval Data Types)』を参照してください。

[8] SQL_BIT データ型の特性は、SQL-92 の BIT 型とは異なります。

[9] このデータ型には、SQL-92 の対応するデータ型がありません。

付録 D. バックエンド ODBC ドライバー接続ストリング用のフォーマット規則 (RemoteServerDSN パラメーター)

SQL パススルーのセットアップで、バックエンド ODBC ドライバーの接続ストリングは、`solid.ini` 構成ファイルの `[Passthrough]` セクションにある

RemoteServerDSN パラメーターを使用して定義されます。接続ストリングのフォーマットは、ODBC ドライバーによって異なります。

一般的な規則

- セミコロン (;) を含む場合、接続ストリングは二重引用符で囲んで指定し、最初の等号と二重引用符の間にスペースを入れてはなりません。

例えば、以下のようにします。

```
[Passthrough]
RemoteServerDSN="Driver={IBM DB2 ODBC DRIVER};Database=my_ids;Hostname=9.212.253.10;Port=9088;protocol=TCPIP;"
```

- 接続ストリングにバックエンド・データベース用のユーザー名とパスワードを入れる必要がある場合、`%s` をプレースホルダーとして使用し、ユーザー名とパスワードを表示すべき場所にマークを付けることができます。 `%s` は、接続時に、ユーザー名とパスワードが `SYS_SERVER` システム表から読み取られるように指示します。

例えば、ポート `9088` でデータベース `my_ids` を使用する IDS ODBC ドライバーの場合、以下ようになります。

```
RemoteServerDSN=my_ids:Port=9088;%s,%s
```

接続時に、ユーザー名用の `%s` とパスワード用の `%s` が、`SYS_SERVER` システム表に保管されているユーザー名 `Admin` とパスワード `pwd123` に置き換えられます。

```
RemoteServerDSN=my_ids:Port=9088;Admin,pwd123
```

特記事項

Copyright © Solid® Information Technology Ltd. 1993, 2010.

All rights reserved.

Solid Information Technology Ltd. または International Business Machines Corporation の書面による明示的な許可がある場合を除き、本製品のいかなる部分も、いかなる方法においても使用することはできません。

本製品は、米国特許 6144941、 7136912、 6970876、 7139775、 6978396、 7266702、 7406489、 7502796、 および 7587429 により保護されています。

本製品は、米国輸出規制品目分類番号 ECCN=5D992b に指定されています。

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

本書に記載の製品、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。

〒242-8502

神奈川県大和市下鶴間1623番14号

日本アイ・ビー・エム株式会社

法務・知的財産

知的財産権ライセンス渉外

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。

IBM Canada Limited
Office of the Lab Director
8200 Warden Avenue
Markham, Ontario
L6G 1C7
CANADA

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認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者にお願いします。

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

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名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。

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。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生的創作物にも、次のように、著作権表示を入れていただく必要があります。

© (お客様の会社名) (西暦年)。このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。

© Copyright IBM Corp. _年を入れる_。 All rights reserved.

商標

IBM、IBM ロゴ、ibm.com[®]、Solid、solidDB、InfoSphere、DB2、Informix、および WebSphere[®] は、International Business Machines Corporation の米国およびその他の国における商標です。これらおよび他の IBM 商標に、この情報の最初に現れる個所で商標表示 (® または ™) が付されている場合、これらの表示は、この情報が公開された時点で、米国において、IBM が所有する登録商標またはコモン・ロー上の商標であることを示しています。このような商標は、その他の国においても登録商標またはコモン・ロー上の商標である可能性があります。現時点での IBM の商標リストについては、「Copyright and trademark information」(www.ibm.com/legal/copytrade.shtml) をご覧ください。

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です。

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

Microsoft および Windows は、Microsoft 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。

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。



Printed in Japan

SC88-8163-01



日本アイ・ビー・エム株式会社
〒103-8510 東京都中央区日本橋箱崎町19-21