

IBM DB2 Information Integrator



# レプリケーションとイベント・パブリッシング 入門

バージョン 8.2



IBM DB2 Information Integrator



# レプリケーションとイベント・パブリッシング 入門

バージョン 8.2

**ご注意！**

本書および本書で紹介する製品をご使用になる前に、57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。

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

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

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

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

(URL は、変更になる場合があります)

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

原典：	GC18-7567-00 IBM DB2 Information Integrator Introduction to Replication and Event Publishing Version 8.2
発行：	日本アイ・ビー・エム株式会社
担当：	ナショナル・ランゲージ・サポート

第1刷 2004.8

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

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

© Copyright International Business Machines Corporation 2004. All rights reserved.

© Copyright IBM Japan 2004

# 目次

本書について	v
本書の対象読者	v
本書で使用される表記上のきまり	v
本書で使用される用語	v
本書の追加情報の入手先	vi
<b>第 1 章 DB2 レプリケーションと発行の紹介</b>	<b>1</b>
DB2 レプリケーションと発行の紹介 — 概要	1
SQL レプリケーションの紹介	1
Q レプリケーションの紹介	2
イベント発行の紹介	3
DB2 レプリケーションおよび発行の管理インターフェース	4
DB2 レプリケーションおよび発行のユーティリティ	5
<b>第 2 章 SQL レプリケーション</b>	<b>7</b>
SQL レプリケーション — 概要	7
SQL レプリケーション環境のインフラストラクチャー	8
SQL レプリケーションでのソースの登録	9
SQL レプリケーションのサブスクリプション・セット	12
SQL レプリケーションでの DB2 UDB ソースからのデータのキャプチャー	17
SQL レプリケーションでの DB2 UDB ターゲットへのデータのアプライ	19
SQL レプリケーションでの非 DB2 ソースからのデータのキャプチャー	20
SQL レプリケーションでの非 DB2 ターゲットへのデータのアプライ	21
<b>第 3 章 Q レプリケーション</b>	<b>23</b>
Q レプリケーション — 概要	23
Q レプリケーション環境のインフラストラクチャー	24
Q レプリケーションのソースとターゲット	25
Q レプリケーションでのデータのキャプチャー	27
Q レプリケーションでのターゲットへのデータのアプライ	29
Q レプリケーションのレプリケーション・タイプ	30
<b>第 4 章 SQL レプリケーションと Q レプリケーションの比較</b>	<b>35</b>
SQL レプリケーションと Q レプリケーションの比較 — 概要	35

SQL レプリケーションと Q レプリケーションのインフラストラクチャーの比較	35
SQL レプリケーションと Q レプリケーションのソースとターゲットの比較	36
SQL レプリケーションと Q レプリケーションのデータのキャプチャーとアプライの比較	37
一般的なシナリオで使用すべきレプリケーション・ソリューション	39

<b>第 5 章 イベント発行</b>	<b>41</b>
イベント発行 — 概要	41
イベント発行環境のインフラストラクチャー	42
イベント発行のソース	43
イベント発行でのデータのキャプチャー	46

<b>第 6 章 SQL レプリケーション、Q レプリケーション、およびイベント発行の比較</b>	<b>49</b>
---	-----------

<b>第 7 章 Q レプリケーションと高可用性災害時リカバリー (HADR) の比較</b>	<b>51</b>
---	-----------

<b>第 8 章 イベント発行と DB2 UDB MQ ユーザー定義関数の比較</b>	<b>53</b>
---	-----------

<b>ユーザー補助</b>	<b>55</b>
キーボード入力およびナビゲーション	55
キーボード入力	55
キーボード・ナビゲーション	55
キーボード・フォーカス	55
利用可能なモニター	56
フォントの設定	56
色に依存しない	56
支援テクノロジーとの互換性	56
利用しやすい資料	56

<b>特記事項</b>	<b>57</b>
商標	59

<b>索引</b>	<b>61</b>
-----------	-----------

<b>IBM と連絡を取る</b>	<b>63</b>
製品情報	63
資料についてのコメント	63



---

## 本書について

本書は、SQL レプリケーション、Q レプリケーション、およびイベント発行について紹介します。SQL レプリケーション、Q レプリケーション、およびイベント発行と IBM® レプリケーション・ソリューションの他の製品を併用して、業務のニーズに合わせてレプリケーションおよび発行環境を調整することができます。

---

## 本書の対象読者

本書は、データベース管理者、レプリケーション・スペシャリスト、および SQL レプリケーション、Q レプリケーション、およびイベント発行環境のセットアップおよび保守の担当者を対象としています。本書は、読者が以下の点に精通していることを前提とします。

- 標準データベース用語。
- データベース設計、データベース管理、データベース・セキュリティー、サーバー接続、およびネットワークング。
- SQL レプリケーション、Q レプリケーション、およびイベント発行環境に関するオペレーティング・システム。
- 複製および発行するデータ。
- メッセージを受け取るアプリケーション (イベント発行のみ)。

---

## 本書で使用される表記上のきまり

本書では、以下の強調表示規則を使用しています。

- **太字体** は、コマンド、またはフィールド、フォルダー、アイコン、メニュー選択項目の名前のようなユーザー・インターフェース・コントロールやコマンドを表します。
- `モノスペース (Monospace)` は、示された通りに正確に入力するテキストの例を表します。
- *イタリック* は、値に置き換える変数を表します。また、本の表題や語の強調にも使用されます。

---

## 本書で使用される用語

本書では、データベース、接続、コピー、SQL、および LAN の概念における標準用語を使用しています。

特に指定しない限り、以下を意味するものとします。

**Linux** Linux は、すべての Linux プラットフォーム (Linux Intel、Linux z/Series など) の SQL レプリケーション、Q レプリケーション、およびイベント発行を指します。

**UNIX** UNIX は、すべての UNIX プラットフォーム (HP UX、Solaris オペレーティング環境、AIX など) の SQL レプリケーション、Q レプリケーション、およびイベント発行を指します。

#### Windows

Windows は、すべての Windows プラットフォーム (2000、XP など) の SQL レプリケーション、Q レプリケーション、およびイベント発行を指します。

**z/OS** z/OS は、z/OS および OS/390 の SQL レプリケーション、Q レプリケーション、およびイベント発行を指します。z/OS は、OS/390 の次世代オペレーティング・システムで、z/OS 上の UNIX System Services (USS) も含まれます。

---

## 本書の追加情報の入手先

このセクションでは、SQL レプリケーション、Q レプリケーション、およびイベント発行について、役に立つと思われるその他の情報源を示します。

目的	参照先
DB2 Information Integrator レプリケーション・ソリューション・ソリューションについての情報を得る	<a href="http://www.ibm.com/software/data/dpropr/">www.ibm.com/software/data/dpropr/</a>
SQL レプリケーション、Q レプリケーション、またはイベント発行のセットアップ方法および保守方法についての情報を得る	以下のいずれかです。 <ul style="list-style-type: none"><li>• <a href="http://publib.boulder.ibm.com/infocenter/db2help/">http://publib.boulder.ibm.com/infocenter/db2help/</a> の DB2 インフォメーション・センター</li><li>• 「<i>IBM DB2 Information Integrator SQL レプリケーション・ガイド</i>および<i>リファレンス</i>」</li><li>• 「<i>IBM DB2 Information Integrator レプリケーションとイベント・パブリッシング ガイド</i>および<i>リファレンス</i>」</li></ul>
DB2 Universal Database および DB2 Information Integrator 用の情報セットを表示する	DB2 インフォメーション・センター <a href="http://publib.boulder.ibm.com/infocenter/db2help/">http://publib.boulder.ibm.com/infocenter/db2help/</a> を参照してください。
技術支援リソースおよび顧客支援オプションについて知る	<a href="http://www.ibm.com/software/data/integration/db2ii/support.html">http://www.ibm.com/software/data/integration/db2ii/support.html</a>
WebSphere MQ についての情報を得る	「 <i>WebSphere MQ システム管理ガイド</i> 」を参照してください。 <a href="http://www-306.ibm.com/software/integration/mqfamily/library/manualsa/manuals/crosslatest.html">http://www-306.ibm.com/software/integration/mqfamily/library/manualsa/manuals/crosslatest.html</a> から PDF をダウンロードするか、HTML バージョンを表示してください。
DB2 Information Integrator フェデレーテッド・システムについての情報を得る	「 <i>DB2 Information Integrator フェデレーテッド・システム・ガイド</i> 」を参照してください。このマニュアルの情報は、DB2 インフォメーション・センター ( <a href="http://publib.boulder.ibm.com/infocenter/db2help/">http://publib.boulder.ibm.com/infocenter/db2help/</a> ) にも含まれています。



---

## 第 1 章 DB2 レプリケーションと発行の紹介

---

### DB2 レプリケーションと発行の紹介 — 概要

DB2® Universal Database バージョン 8.2 は、リレーショナル・データベースとの間のデータの複製で使用できる、SQL レプリケーション および Q レプリケーション という 2 つの異なるソリューションを提供しています。SQL レプリケーションでは、コミットされたソースへの変更が、ターゲット・システムに複製される前にリレーショナル表にステージングされます。Q レプリケーションでは、コミットされたソースへの変更がメッセージとして書き込まれ、WebSphere® MQ メッセージ・キューを介してターゲット・システムにトランスポートされます。

DB2 UDB バージョン 8.2 は、イベント発行 というソリューションも提供しています。これは、コミットされたソースへの変更を XML 形式のメッセージに変換し、メッセージ・ブローカーなどのアプリケーションにそれらのメッセージを発行するソリューションです。

以下のトピックでは、レプリケーションおよび発行ソリューションについて簡潔に紹介しています。また、これらのソリューション用の管理インターフェースや、レプリケーションおよび発行環境の保守を支援するユーティリティについても説明しています。

- SQL レプリケーションの紹介
- Q レプリケーションの紹介
- イベント・パブリッシングの紹介
- DB2 レプリケーションおよびパブリッシングの管理インターフェース
- DB2 レプリケーションおよびパブリッシングのユーティリティ

#### 関連概念:

- 4 ページの『DB2 レプリケーションおよび発行の管理インターフェース』
- 2 ページの『Q レプリケーションの紹介』
- 1 ページの『SQL レプリケーションの紹介』
- 5 ページの『DB2 レプリケーションおよび発行のユーティリティ』
- 3 ページの『イベント発行の紹介』

---

### SQL レプリケーションの紹介

SQL レプリケーションはソースの変更をキャプチャーし、コミットされたトランザクション・データをステージング表を使って保管します。続いて、変更がステージング表から読み取られ、対応するターゲット表に複製されます。ステージング表を使うと、データのキャプチャーとステージングを 1 回行うだけで、複数のターゲットに対し、さまざまなフォーマットで、さまざまな送信インターバルに従って送信ができます。

SQL レプリケーションは、複製データを必要とするさまざまな目的、たとえば、容量の軽減、データウェアハウスおよびデータマートへのフィード、および変更履歴の監査などに使用できます。

レプリケーションは、連続で行うことも、一定間隔で行うことも、1 回限りで行うこともできます。飛行機の予約をするアプリケーションのように、アプリケーションがリアルタイムに近いデータを必要とする場合は、連続のレプリケーションが便利です。次に、オフピーク時間に多数のデータを複製する場合は、一定間隔のレプリケーションが便利です。また、データベース・イベントによってレプリケーションを起動することもできます。

ソースとターゲットは、DB2<sup>®</sup> UDB データベースか、非 DB2 のリレーショナル・データベースのいずれかに置くことができます。ソースからの複製、また、ターゲットへの複製を行えるリレーショナル・データベース管理システムとして、Linux、UNIX<sup>®</sup>、Windows<sup>®</sup>、z/OS<sup>™</sup>、および iSeries<sup>™</sup> 上の DB2 UDB、Microsoft<sup>®</sup> SQL Server、Oracle、Sybase、および Teradata (ターゲットのみ) があります。非 DB2 リレーショナル・データベースとの間の複製を計画している場合、DB2 Information Integrator のフェデレーテッド・サーバー機能が必要になります。フェデレーテッド・サーバーについて詳しくは、「*DB2 Information Integrator フェデレーテッド・システム・ガイド*」を参照してください。

複製するデータに関しては非常に柔軟性があります。たとえば、すべての行および列を複製することを選択することも、そのサブセットのみを選択することもできます。行と列のサブセットを選択することにより、必要なデータのみをネットワークで転送することができます。

データを削除したり、集約したり、あるいは操作したりすることもできます。データを操作する場合、ソースで一元的にその操作を行ってその後操作データを配布することも、データの複製時に操作を行い、一部のターゲットは加工済みデータを取得し、他のターゲットは加工済みデータを取得しないようにすることもできます。

#### 関連概念:

- 1 ページの『DB2 レプリケーションと発行の紹介 — 概要』
- 7 ページの『SQL レプリケーション — 概要』
- 39 ページの『一般的なシナリオで使用すべきレプリケーション・ソリューション』
- 49 ページの『第 6 章 SQL レプリケーション、Q レプリケーション、およびイベント発行の比較』

---

## Q レプリケーションの紹介

Q レプリケーションは、多数のデータを非常に短い待ち時間で複製できるレプリケーション・ソリューションです。Q レプリケーションは、ソース表に対する変更をキャプチャーし、コミットされたトランザクション・データをメッセージに変換します。このデータはソースでコミットされた後すぐに送信され、Q レプリケーションによって読み取られます。データが表でステージングされることはありません。メッセージは WebSphere<sup>®</sup> MQ メッセージ・キューを介してターゲット・ロケーションに送信されます。メッセージはそのメッセージ・キューでキューから読み取ら

れ、トランザクション・データに再び変換されます。その後、トランザクションは、データの保全性を保持する高並列処理方式によりターゲット表に適用されません。

Q レプリケーションは、複製データを必要とするさまざまな目的、たとえば、フェイルオーバー、容量の軽減、地理的に分散したアプリケーション、ローリング・アップグレード中または他の計画停止中のデータ使用可能性などに使用できます。

Linux、UNIX<sup>®</sup>、Windows<sup>®</sup>、または z/OS<sup>™</sup> 上の DB2<sup>®</sup> UDB サーバーのリレーショナル表がソースとターゲットになることができます。これらいずれかのプラットフォーム上の DB2 UDB ソースから、同じプラットフォーム、またはサポートされる他のいずれかのプラットフォーム上の DB2 UDB ターゲットに複製することができます。データのトランスポートに WebSphere MQ メッセージ・キューを使用するには、ソースおよびターゲット・システム上に WebSphere MQ が必要です。

ソース表から列および行のサブセットを複製することができます。サブセットの選択はすべて、必要なデータのみをネットワークでトランスポートできるように、ソース・ロケーションで行われます。データ形式変更を実行する場合、複製データを独自のストアード・プロシージャに渡すことができます。

#### 関連概念:

- 1 ページの『DB2 レプリケーションと発行の紹介 — 概要』
- 23 ページの『Q レプリケーション — 概要』
- 39 ページの『一般的なシナリオで使用すべきレプリケーション・ソリューション』
- 49 ページの『第 6 章 SQL レプリケーション、Q レプリケーション、およびイベント発行の比較』

---

## イベント発行の紹介

イベント発行は、ソース表に対する変更をキャプチャーし、コミットされたトランザクション・データを Extensible Markup Language (XML) 形式のメッセージに変換します。それぞれのメッセージにはトランザクション全体を含めることもできますし、行レベルの変更のみを含めることもできます。これらのメッセージは WebSphere<sup>®</sup> MQ メッセージ・キューに置かれ、メッセージ・ブローカーまたは他のアプリケーションによって読み取られます。必要なデータのみを発行できるよう、ソース表からの列および行のサブセットを発行することが可能です。

イベント発行は、発行データを必要とするさまざまな目的、たとえば、中央情報ブローカーおよび Web アプリケーションへの送り込んだり、およびソース表に対する更新、挿入、または削除に基づいたアクションの起動などに使用できます。

Linux、UNIX<sup>®</sup>、Windows<sup>®</sup>、または z/OS<sup>™</sup> 上の DB2<sup>®</sup> UDB サーバーのリレーショナル表がソース表になることができます。データのトランスポートに WebSphere MQ メッセージ・キューを利用するには、ソース上およびメッセージを受信するシステム上で WebSphere MQ を使用可能にしておく必要があります。

イベント発行はさまざまな用途に威力を発揮します。価格と在庫の変更が潜在的バイヤーに対して発行されるシナリオを検討してみましょう。たとえば、食品卸売業者がバナナのような生鮮食品を世界市場から大量に調達し、それらを食料品店や流通業者に販売するとします。

バナナの価値は、倉庫に置かれる期間が長ければ長いほど下がります。卸売業者は、その潜在的バイヤーに、価格と在庫の変更データを通知したいと思います。そこで、イベント発行をセットアップすることができます。価格が変更されるたびに、潜在的バイヤーに XML メッセージを送信し、「価格変更イベント」を通知することができます。

それぞれのバイヤー（小売業者または流通業者）は最大限の利益を得たいと思っています。それらの業者は、価格、経過日数（悪くなるまでの時間）、および特定の数量を売りさばける期間に関するこれまでの経験と知識に基づいて、バナナを購入するタイミングを決定できます。購入が早すぎると、バナナに対して高い金額を支払うことになるため、最大限の利益を得ることができません。購入が遅すぎると、バナナが傷んでしまうため、これも最大限の利益を得ることになりません。適切なタイミングで購入することによって、最大限の利益を得ることができます。アプリケーションはイベント通知メッセージを受け取り、最大限の利益が得られる適切な時点で、購入オーダーを自動的に生成します。

#### 関連概念:

- 1 ページの『DB2 レプリケーションと発行の紹介 — 概要』
- 41 ページの『イベント発行 — 概要』
- 49 ページの『第 6 章 SQL レプリケーション、Q レプリケーション、およびイベント発行の比較』

---

## DB2 レプリケーションおよび発行の管理インターフェース

レプリケーション・センターは、レプリケーションおよび発行環境の定義、操作、およびモニターに使用できるグラフィカル・ユーザー・インターフェースです。これは DB2® Administration Client に付属し、Linux、UNIX®、および Windows® システム上で実行されます。レプリケーション・センターは、複数のシステムに渡るさまざまなプラットフォームのレプリケーション環境を管理するための、単一のインターフェースを提供しています。

以下に、レプリケーション・センターの実用性の高い機能のいくつかをリストします。

- レプリケーションおよび発行環境の基本的な構成方法を段階的に教えてくれるランチパッド。
- 単純なものから高度にカスタマイズされたものまで、レプリケーションおよび発行構成のセットアップを支援するウィザード。
- 独自の規則やストレージ要件に準拠したスキーマ、名前、および他の属性を持つレプリケーション・オブジェクトの作成を可能にするカスタマイズ可能プロファイル。
- レプリケーションおよび発行環境の実行内容を示すレポート。

必要があれば、コマンド行ツールの ASN コマンド行プロセッサ (ASNCLP) を使ってレプリケーションおよび発行環境を定義および変更することができます。その後、システム・コマンドを使ってレプリケーションおよび発行のプログラムを操作できます。

---

## DB2 レプリケーションおよび発行のユーティリティ

レプリケーションおよび発行テクノロジーに付属する 3 つのユーティリティは、レプリケーションおよび発行環境をモニターし、ソースとターゲットの同期を維持する上で役に立ちます。

### レプリケーション・アラート・モニター

レプリケーション・アラート・モニターは、レプリケーションおよびイベント発行ソリューションの一部となっているプログラムの正常性を検査するユーティリティです。このユーティリティは、プログラムが終了したり、警告またはエラー・メッセージを発行したり、指定の値のしきい値に達したり、特定アクションを実行したりする状況をチェックします。レプリケーション・アラート・モニターに対し、監視対象とする状態を指定できます。対象状態のいずれかになると、レプリケーション・アラート・モニターは、その状態についての該当する連絡先として指定されたユーザーまたはユーザー・グループに E メール・メッセージを送信します。

たとえば、レプリケーション・プログラムが実行されていない場合、または予想最大メモリー使用量に達した場合に、レプリケーション・アラート・モニターが通知を行うようにすることができます。

### 同期調整ユーティリティ

たとえばターゲット表がアプリケーションによって変更されると、ソース表とターゲット表の同期が失われる場合があります。 `tdiff` および `trepair` ユーティリティは、Q レプリケーションおよび SQL レプリケーションのソース表とターゲット表の間の相違を検出し、修復することができます。これらを使うことによって、手動で表を比較し、ターゲットを再ロードしてソースと再同期する手間が省けます。

`tdiff` ユーティリティはソース表とターゲット表の間の相違のリレーショナル表を生成します。 `trepair` ユーティリティは `tdiff` ユーティリティの実行後に実行できます。このユーティリティは、その相違の表を使用して、ターゲット表で挿入、更新、および削除するべき行を判別し、ソース表と同期させます。

### 関連概念:

- 1 ページの『DB2 レプリケーションと発行の紹介 — 概要』



## 第 2 章 SQL レプリケーション

### SQL レプリケーション — 概要

以下のトピックでは、SQL レプリケーションに関する基本的な概念について説明しています。特定の詳細については、「*IBM DB2 Information Integrator SQL レプリケーション・ガイドおよびリファレンス*」を参照してください。

- SQL レプリケーション環境のインフラストラクチャー
- SQL レプリケーションでのソースの登録
- SQL レプリケーションのサブスクリプション・セット
- SQL レプリケーションでの DB2 UDB ソースからのデータのキャプチャー
- SQL レプリケーションでの DB2 UDB ターゲットへのデータの適用
- SQL レプリケーションでの非 DB2 ソースからのデータのキャプチャー
- SQL レプリケーションでの非 DB2 ターゲットへのデータの適用

以下の図は、SQL レプリケーションの単純な構成を示しています。上記のトピックではそれぞれ、この図の各セクションについて詳しく取り上げて説明しています。

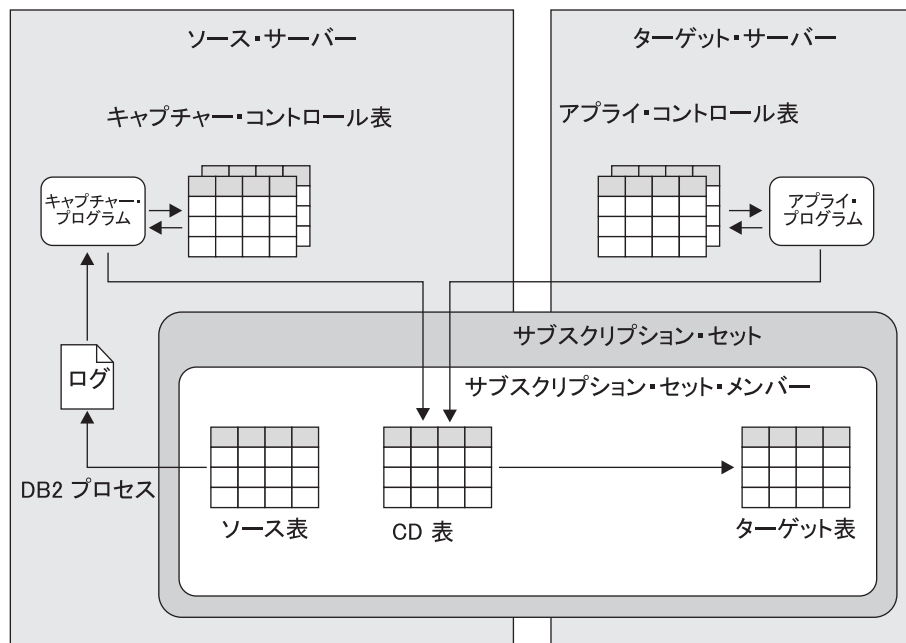


図 1. SQL レプリケーションの単純な構成

#### 関連概念:

- 21 ページの『SQL レプリケーションでの非 DB2 ターゲットへのデータのアプライ』
- 19 ページの『SQL レプリケーションでの DB2 UDB ターゲットへのデータのアプライ』



- 20 ページの『SQL レプリケーションでの非 DB2 ソースからのデータのキャプチャー』
- 17 ページの『SQL レプリケーションでの DB2 UDB ソースからのデータのキャプチャー』
- 8 ページの『SQL レプリケーション環境のインフラストラクチャー』
- 12 ページの『SQL レプリケーションのサブスクリプション・セット』
- 9 ページの『SQL レプリケーションでのソースの登録』

---

## SQL レプリケーション環境のインフラストラクチャー

SQL レプリケーションを使用すると、キャプチャーおよびアプライという 2 つのプログラムを使うことによって DB2<sup>®</sup> UDB ソースからターゲットにデータを複製することができます。キャプチャー・プログラムはソース・システム上で実行されます。キャプチャー・プログラムは変更されたソース・データに関する DB2 リカバリー・ログを読み取り、コミット済みの変更データをステージング表に保管します。アプライ・プログラムは通常、ターゲット・システム上で実行されます。アプライ・プログラムはステージング表からキャプチャーされたデータを検索し、そのデータをターゲットに送信します。どちらのプログラムも、タスクの実行に必要な情報を追跡したり、生成される情報 (プログラムの実行状態の調査に使用できる情報など) を保管したりするために DB2 表のセットを使用します。これらの表は、レプリケーション・ソースが何でターゲットが何かなどの設定を行う前にあらかじめ作成します。

キャプチャー・プログラムは、キャプチャー・コントロール表 という DB2 UDB 表のセットを使用します。これらの表には、レプリケーション・ソースと、DB2 リカバリー・ログのキャプチャー・プログラムの現在位置に関する情報が入ります。ほとんどの場合、キャプチャー・プログラムのコントロール表は、プログラムと関連付けられているソースと同じ DB2 UDB サーバー上に存在していなければなりません。

同じ DB2 UDB サーバー上で複数のキャプチャー・プログラムを実行することができます。各キャプチャー・プログラムは、独自のキャプチャー・コントロール表セットを使用します。キャプチャー・コントロール表のセットと関連付けられているスキーマは、それらのコントロール表を使用するキャプチャー・プログラムを識別します。このスキーマは、キャプチャー・スキーマ と呼ばれます。

非 DB2 リレーショナル・データベースにソースが存在する場合、そのソースに対する変更のキャプチャーにトリガーが使用されます。ただし、コントロール表のセットも引き続き必要です。

アプライ・プログラムは、アプライ・コントロール表 という DB2 UDB 表のセットを使用します。これらの表には、ターゲットと、それに対応するソースのロケーションに関する情報が入ります。アプライ・プログラムのコントロール表は通常、アプライ・プログラムが実行されるシステム上にあります。キャプチャー・プログラムとは異なり、同じコントロール表のセットを使用する複数のアプライ・プログラムを作成することができます。それぞれのアプライ・プログラムは、アプライ修飾子 と呼ばれる名前によって、これらのコントロール表の中で識別されます。



DB2 UDB によってサポートされる各オペレーティング・システムまたはプラットフォーム、および SQL レプリケーションによってサポートされるそれぞれの非 DB2 リレーショナル・データベース管理システムごとに、コントロール表によって使用される表スペースについての独自のデフォルト指定を設定することができます。

以下の図は、SQL レプリケーションの単純な構成のインフラストラクチャーを示しています。

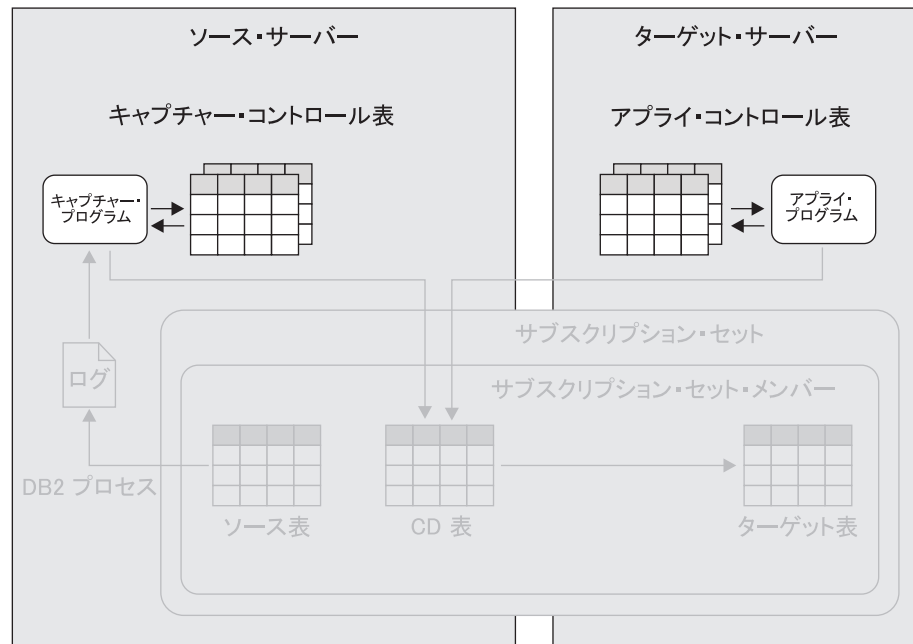


図2. SQL レプリケーションの単純な構成のインフラストラクチャー： ソース・サーバー上に、キャプチャー・コントロール表と呼ばれる DB2 リレーショナル表のセットを作成します。ソースに関する情報はこれらの表に書き込まれます。ソース・サーバー上で実行されるキャプチャー・プログラムは、この情報を使用して、予想されるキャプチャー・データを識別します。ターゲットに関する情報はアプライ・コントロール表に渡されます（この表は通常、ターゲット・サーバー上にあります）。アプライ・プログラム（これも通常はターゲット・サーバー上にある）は、この情報を使用して、予想されるデータの書き込み先ターゲットを識別します。

**関連概念:**

- 7 ページの『SQL レプリケーション — 概要』

## SQL レプリケーションでのソースの登録

レプリケーション・ソースを最終的にどのようなものにしたいかが分かっている場合、それらを登録することによって SQL レプリケーションに知らせます。ソースが DB2® UDB 表およびビュー、または非 DB2 リレーショナル・データベース上の表である場合に、それらを登録できます。

**注:** ソース表を非 DB2 リレーショナル・データベース上で登録する際は、SQL レプリケーションとともに DB2 Information Integrator を使用します。ソース・データベースをフェデレーテッド・データベースにマップし、それぞれのソース表ごとにニックネームを作成します。

表の複製には、3 つの方式のうちのいずれかを指定できます。最初の方式は、フル・リフレッシュ・レプリケーション と呼ばれる方式で、2 番目と 3 番目は、変更キャプチャー・レプリケーション ・タイプの方式です。

**ターゲット表をリフレッシュすることによって複製する。変更はキャプチャーしない。** アプライ・プログラムは後で指定できるインターバルで 1 つ以上の DB2 UDB ユーティリティーを呼び出し、登録済み表に対応するターゲットの内容を削除し、登録済み表から得られる内容をターゲットにロードします。表を登録する際に必要なのは、そのスキーマと名前のみです。ターゲットが小さい場合、またはそれほど頻繁に複製を行わない場合、この方法による複製を選択できます。このレプリケーション方式では、キャプチャー・プログラムまたはキャプチャー・トリガーは必要ありません。

**複製する 1 つ以上の列の値が変更される時には常に行をキャプチャーする。** この方式は最も一般的な SQL レプリケーションの使用法であり、この方式を使用すると、変更が頻繁に未登録の列のみに影響を与える場合に、複製されるデータの量が削減されます。たとえば、表には 100 個の列があり、それらの列うち 50 個をレプリケーション用に登録しているとします。SQL レプリケーションは、表の中の 50 個の登録済み列の 1 つに変更が加えられるときには常に、変更が加えられた行のためにそれら 50 個の列の値をキャプチャーします。行が変更されても、変更が他の 50 個の列に限定されている場合、レプリケーションのためにデータがキャプチャーされることはありません。

**表のいずれかの列の値が変更される時には常に行をキャプチャーする。** この方式は、表の中の変更がほとんど登録済み列のみの場合、または監査目的で情報が複製される場合によく選択されます。たとえば、表の中の変更された行のみを知りたいとします。表には 100 個の列があり、主キー列をレプリケーション用に登録しています。SQL レプリケーションは、表の中のいずれかの列に変更が加えられると、変更が加えられた行の主キー列の値をキャプチャーします。

いずれかの変更キャプチャー・レプリケーション・タイプで表を登録するときは、SQL レプリケーションに同じ情報を提供し、SQL レプリケーションは同じオブジェクトを作成します。

• **ソースが DB2 UDB 表の場合:**

SQL レプリケーションに、ソース表のスキーマと名前、およびそこから複製したい列を知らせます。選択する列は、登録済み 列と呼ばれます。また、それらの列にはソース表に対して行われた変更の結果生じる値が含まれるため、変更後イメージ 列とも呼ばれます。たとえば、登録済み表の 1 つの行の、登録済み列 A の値が 25 から 30 に変更される場合、30 が変更後イメージになります。

また、登録済み列の変更前イメージ の複製を選択することもできます。変更前イメージとは、ソース表に対して変更が行われる前に存在していた値のことです。上記の例では、25 が変更前イメージとなります。変更前イメージは、さまざまな

目的に使用できます。たとえば、キー値が変更されたときの、アプライ・プログラムによるターゲットの行の検索を可能にします。

SQL レプリケーションは、表に対してコミットされた変更を記録するための変更データ (CD) 表を作成します。ソース表で登録する列はすべて CD 表にも存在します。列の変更前イメージの複製を選択する場合、それらの変更前イメージを記録するために CD 表の中に列が作成されます。

- **ソースが非 DB2 UDB 表の場合:**

SQL レプリケーションに、ソース表のニックネームのスキーマと名前、およびソース表から複製したい列を知らせます。DB2 UDB 表を登録する場合と同じように、選択する列は登録済み列と呼ばれ、変更後イメージおよび変更前イメージを複製できます。SQL レプリケーションは、非 DB2 リレーショナル・データベースのソース表上にキャプチャー・トリガーと呼ばれる 3 つのトリガーを直接作成します。挿入用トリガー、更新用トリガー、および削除用トリガーの 3 つです。SQL レプリケーションは、ソース表のステージング表も作成します。ステージング表は整合変更データ (CCD) 表と呼ばれ、トリガーによってデータの追加が行われます。たとえば、ソース表で更新が実行される場合、表の更新トリガーは更新を、対応する CCD 表にレコードとして保管します。コミットされたトランザクション・データのみ CCD 表に保管されます。CCD 表のニックネームがフェデレーテッド・データベース上に作成されます。

それぞれのソース・サーバーで、CD 表および CCD 表に関する独自のデフォルト命名規則を設定できます。CD 表および CCD 表で使用される表スペースにも独自のデフォルトを設定できます。

ビューの登録について詳しくは、「*IBM DB2 Information Integrator SQL レプリケーション・ガイド*およびリファレンス」を参照してください。

以下の図は、ソース表の登録時に関係するオブジェクトを示しています。

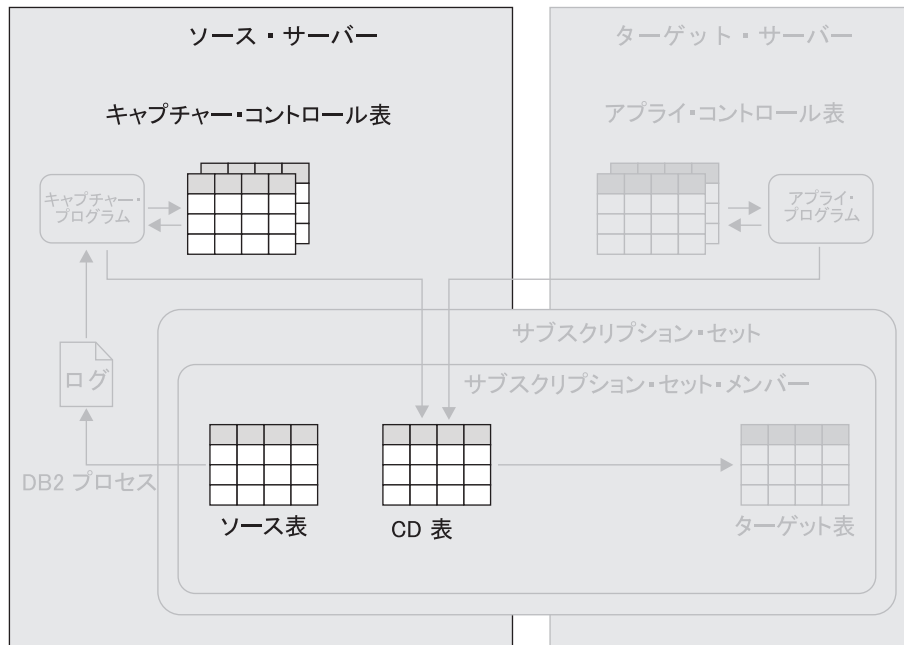


図3. ソース表の登録：レプリケーション用のソース表を登録する際、SQL レプリケーションは、登録情報をキャプチャー・コントロール表に保管します。SQL レプリケーションは、キャプチャー・プログラムがソース表に対する変更の記録に使用する CD 表も作成します。

**関連概念:**

- 7 ページの『SQL レプリケーション — 概要』

## SQL レプリケーションのサブスクリプション・セット

ソースを登録した後、ソースとターゲットの対を作るためのサブスクリプション・セットを作成します。ソースとターゲットの対はそれぞれ、それが作成されるサブスクリプション・セットのメンバーと呼ばれます。サブスクリプション・セットを使用して、1つのソース・サーバーから1つのターゲット・サーバーへのデータ(1つ以上のソースとターゲットの対)のレプリケーションをスケジュールに入れることができます。アプライ・プログラムは、このレプリケーションがソース・サーバー上のオリジナルのトランザクションと整合した形でターゲットにデータを適用するよう、レプリケーションを調整します。

作成する各サブスクリプション・セットごとに、以下の詳細を SQL レプリケーションに知らせます。

**サブスクリプション・セットの処理に使用するアプライ・プログラム**

サブスクリプション・セットを作成する際、アプライ修飾子を選択することにより、そのサブスクリプション・セットとアプライ・プログラムを関連付けます。各アプライ・プログラムは、複数のサブスクリプション・セットを処理できます。

**ソース表またはビューがある場所**

サブスクリプション・セットのメンバーのソースはすべて同じサーバーに存在します。

### ターゲット表がある場所

サブスクリプション・セットのメンバーのターゲットはすべて同じサーバーに存在します。

### ソースからターゲットにデータを複製する頻度

サブスクリプション・セットのデータの複製は、一定間隔で行うか連続で行うか、またはイベントの発生時に行うかを選択できます。

### インターバル・タイミング

(相対タイミングとも呼ばれる) この方式は、サブスクリプション・セットの最も単純なスケジューリング方式です。インターバルは概算値で、そのときのワークロードや使用可能なシステム・リソースによって決まります。このインターバルは、サブスクリプション・セット内のすべてのサブスクリプション・セット・メンバーに関係します。そのため、サブスクリプション・セット内のすべての表がそれぞれのインターバルで複製されます。

### 連続レプリケーション

この方式を使用すると、アプライ・プログラムは、サブスクリプション・セット内のデータを、そのワークロードと使用可能リソースに応じて可能な限り複製します。

### イベント・タイミング

この方式を使用すると、アプリケーションまたはユーザーは、複製を開始するタイミングを決定できます。アプリケーションまたはユーザーは、イベント表と呼ばれるアプライ・コントロール表に行を挿入します。アプライ・プログラムがその行を認識すると、複製が開始します。

### データ・ブロッキングの使用

データ・ブロッキングを使用すると、変更になるデータをアプライ・プログラムが一度に何分間複製できるかを指定できます。指定する分数は、データ・ブロックのサイズを決定します。データ・ブロックのサイズより大きい変更データのバックログが存在する場合、アプライ・プログラムは、現行の変更データがすべて複製されるまでデータ・ブロックを複製することによって、サブスクリプション・セットを処理します。データ・ブロッキングの利点の 1 つとして、より小さいデータ集合を検索することにより、アプライ・プログラムはネットワーク負荷と検索されたデータに一時的に必要なスペースの両方を減少させます。もう 1 つの利点は、エラーが発生した場合にアプライ・プログラムが行うロールバックが、変更データ・セット全体ではなく、現行のデータ・ブロックだけで済むことです。

たとえば、SQL レプリケーションを使用して 24 時間ごとにデータウェアハウスの表にデータを追加するとします。アプライ・プログラムは、ソース表に対して行われた変更のセット全体を 1 つのブロックで 24 時間ごとに複製する代わりに、24 時間のデータ・ブロック全体が複製されるまで一度に 20 分の変更を複製できます。ブロックのサイズは、独自の要件および制約事項と一致するように調整できます。

### すべてのアプライ・データに 1 つの COMMIT を発行するか、または一時コミットを発行するか

アプライ・プログラムは、表モードまたはトランザクション・モードでデータをアプライできます。表モードでは、アプライ・プログラムはそれぞれの

表ごとに個別にフェッチした変更を処理し、データがすべてアプライされた後で単一コミットを発行します。トランザクション・モードでは、アプライ・プログラムはフェッチした変更を、そのソース・サーバー上の対応するトランザクションで変更が発生した順にすべてのターゲット表にアプライします。アプライ・プログラムはこれらのトランザクションをトランザクション境界でコミットします。各コミットの前に、アプライするトランザクションの数を指定します。

### SQL スクリプトまたはストアド・プロシージャを使用したサブスクリプション・セットのデータの変換

サブスクリプション・セットのソース表内のデータが対応するターゲットに複製される際、SQL スクリプトまたはストアド・プロシージャを使用してデータを変換することができます。

サブスクリプション・セット内に作成するそれぞれのソースとターゲットの対ごとに、新規ターゲット表を作成するか、既存のターゲット表を使用するかを指定します。新規ターゲット表を作成する場合、以下のタイプのいずれかを指定できます。

#### ユーザー・コピー

この読み取り専用のターゲット・タイプは、基本的なデータ・レプリケーションで最も一般的です。このターゲット表のタイプの構造は、ソース表と同じにすることができます。あるいは、ソース列のサブセットのみ同じにすることができます。変更前イメージまたは変更後イメージと計算済み列も定義可能です。このターゲット・タイプは、ターゲット表の各行を一意的に記述する複製列を必要とします。これらの列は、ユニーク索引か主キーのいずれかです。

#### ポイント・イン・タイム

この読み取り専用ターゲット表タイプの構造は、ユーザー・コピーのターゲット表の構造と類似しています。ソース表のデータがいつコミットされたかをキャプチャー・プログラムが示すことができるように、タイム・スタンプ列が追加されます。ターゲット表に変更がアプライされたポイント・イン・タイムをトラッキングする場合は、このターゲット・タイプを選択できます。

#### 集約

これは、ソース表の内容全体またはその変更データを合計する、読み取り専用のターゲット表です。ターゲット列は、SUM、COUNT、MIN、MAX、および AVG などの SQL 列関数から定義されます。これらの列には、実際のソース・データではなく、計算された SQL 関数の値が入ります。アプライ・プログラムが集約を実行した時間を示すタイム・スタンプも組み込まれます。集約ターゲット表には次の 2 つのタイプがあります。

- **基礎集約。** ソース表が複製されるたびにソース表の内容全体を合計します。たとえば、販売員または地域ごとの過去 1 年間の合計または平均販売数をトラッキングする場合、このターゲット・タイプを使用できます。
- **変更集約。** ソース表の最後の複製以降ソース表で発生した変更を合計します。たとえば、販売員、地域、または顧客ごとの月別販売合計をトラッキングする場合、このターゲット・タイプを使用できます。

#### CCD

この読み取り専用のターゲット・タイプには、ソースでコミットされた変更、およびソース表からの操作タイプ (挿入、更新、および削除) を判別するレプリケーション・コントロール表が含まれます。CCD 表にはさまざま



なタイプがあり、使用するタイプは目標とレプリケーション要件によって決まります。 CCD 表を使用すると、 CCD 表をどのように定義するかによって決まる多様な方法で、ソースの変更履歴をトラッキングできます。たとえば、データの変更が発生したときの変更前後の比較や、ソース表への更新を行ったユーザー ID をトラッキングできます。

また、複数層構成をセットアップすることもできます。この構成では、ターゲット CCD 表は他のターゲット表に対するソースとして機能します。ソース表は第 1 層、CCD は第 2 層、CCD にマップされるターゲットは第 3 層になります。複数層レプリケーション環境をセットアップする 1 つの理由は、第 3 層のターゲットに継続的なソースを提供することです。変更を層 1 から層 2 の CCD 表に収集できるため、層 3 のターゲットに複製される変更の数を削減できます。しかも、ソース・システムへのデータベース接続の多くを回避できるので、接続コストを第 2 層に移動できます。

## レプリカ

このターゲット・タイプは、Update-anywhere レプリケーションで使用されます。他のターゲット・タイプは読み取り専用ですが、このターゲット・タイプでは、ソース表 (マスター表として機能する) に再び複製できる書き込みが可能です。マスター表のデータに関して何らかの競合が発生する場合、レプリカから複製される変更はリジェクトされます。非 DB2 リレーショナル・データベースではレプリカ・ターゲット表は定義できません。

各ソースとターゲットの対で、以下も指定します。

- **ターゲットに複製するソース列。** ソースの登録時に列のサブセットを選択できますが、サブスクリプション・セット・メンバーの作成時にもさらにサブセットを選択できます。
- **ソースからの列をターゲットの列にマップする方法。** ターゲット・ロケーションにすでにターゲット表が存在する場合、SQL レプリケーションに、ソース表またはビューの登録済み列をターゲット表の列にマップする方法を知らせます。SQL レプリケーションにターゲット表の作成を実行させたい場合、登録済みソース列はターゲット表の列に自動的にマップされます。

どちらの場合も、対応するデータ・タイプとともにソース列をターゲット列にマップすることにより、またはソース列を計算済み列にマップすることにより、データを変換できます。

- **行のサブセットをターゲットに複製するための述部。** ソースから複製する行のサブセットを選択できます。
- **ターゲット表をロードする方式。** ほとんどの場合、レプリケーションを開始する前に、ターゲット表がソース表と同じになるように、ソース表からのデータはターゲット表にロードされます。SQL レプリケーションでは、自動ロードと手動ロードの 2 つのターゲット表のロード方法を使用できます。

自動ロードの場合、アプライ・プログラムに、1 つのユーティリティーかまたはユーティリティーの対を呼び出すように命令できます。アプライ・プログラムは、アプライ・プログラムが実行されるプラットフォームに応じて、LOAD ユーティリティーの LOAD FROM CURSOR オプション、EXPORT ユーティリティーと LOAD ユーティリティー、または EXPORT ユーティリティーと

IMPORT ユーティリティーを呼び出すことができます。ターゲット表に最も適切なオプションを選択するようアプライ・プログラムに命令することもできます。

ターゲット表を手動でロードすることにする場合、どれでも好きな方式を選択できます。ロードの実行後、ターゲット表の準備が整い、レプリケーションを開始できることをアプライ・プログラムに通知します。

サブスクリプション・セット・メンバーを作成する際、登録済みソース列のサブセットのみを複製するかどうか、またソースからの行のサブセットのみを複製するかどうかも決定できます。

それぞれのターゲット・サーバーで、ターゲット表に関する独自のデフォルト命名規則を設定できます。ターゲット表で使用する表スペースにも独自のデフォルトを設定できます。

以下の図は、SQL レプリケーションの単純な構成のサブスクリプション・セットを示しています。

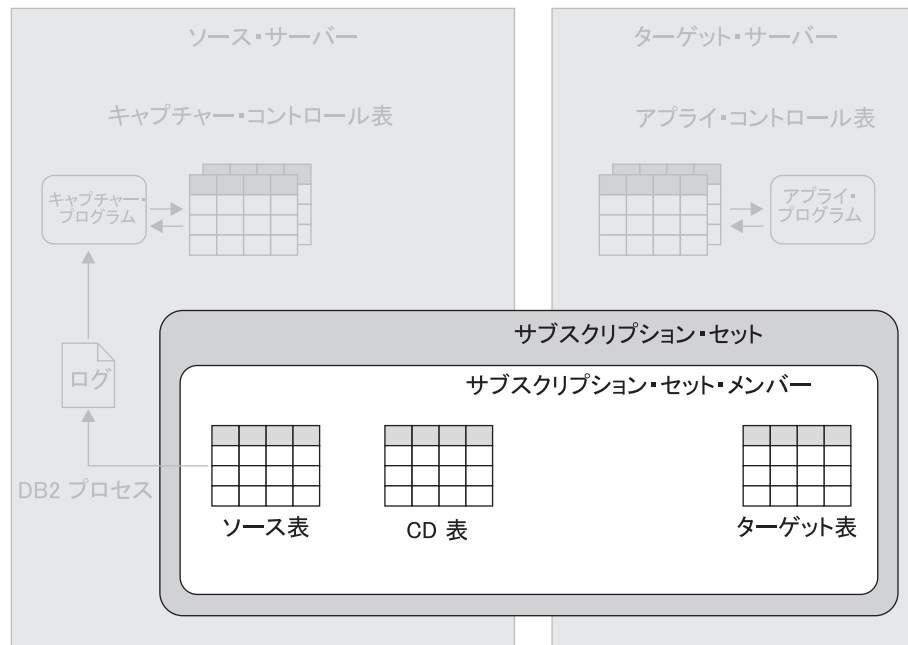


図4. サブスクリプション・セットの作成：サブスクリプション・セットの作成の一部として、ソースをターゲットにマップします。サブスクリプション・セットは、1つ以上のソースとターゲットの対 (サブスクリプション・セット・メンバーとも呼ばれる) をグループ化します。この図では、ソース表がターゲット表にマップされています。ターゲットに複製されるデータは、まずソースの CD 表でステージングされます。

#### 関連概念:

- 7 ページの『SQL レプリケーション — 概要』



## SQL レプリケーションでの DB2 UDB ソースからのデータのキャプチャー

DB2<sup>®</sup> UDB レプリケーションを登録し、サブスクリプション・セットを作成してソースとターゲットの対を作成した後、ソースに対する変更のキャプチャーを開始できます。これは、キャプチャー・プログラムと呼ばれるプログラムを使って行います。

キャプチャー・プログラムを開始し、そのプログラムが、ソースとターゲットが同期されていることを示す信号をアプライ・プログラムから受信した後、キャプチャー・プログラムは、対象のソース表の変更に関する DB2 ログを順次読み取ります。キャプチャー・プログラムは、いずれかのソース表に対する変更を読み取ると、メモリー内に保持されている対応するデータベース・トランザクションに変更を追加します。メモリー内のトランザクションは、潜在的に、ログ内の対応するトランザクションのサブセットになります。これには、ソース表に対する変更のみが入ります。キャプチャー・プログラムは、変更が行われるトランザクションに関する ROLLBACK または COMMIT ステートメントのいずれかを読み取るまで、メモリーに変更を収集します。キャプチャー・プログラムは、ROLLBACK ステートメントを読み取ると、ロールバック・トランザクションに関連した変更をメモリーから消去し、COMMIT を読み取ると、コミット済みトランザクションに関連した変更を保管します。

たとえば、表 A および表 B をレプリケーション・ソースとして登録したとします。これらそれぞれの表で、SQL レプリケーションは、登録プロセスの一部として CD 表を作成します。キャプチャー・プログラムを開始し、そのプログラムが、ターゲット表がソース表と同期されていることを示す信号をアプライ・プログラムから受信した後、キャプチャー・プログラムはそれらのソース表の変更に関する DB2 ログを読み取ります。アプリケーション 1 は、表 A に対して一連の変更を行います。変更はそれぞれ DB2 ログに記録されます。キャプチャー・プログラムはこれらの変更をメモリーに収集します。アプリケーション 1 は ROLLBACK ステートメントを発行します。キャプチャー・プログラムがこのステートメントを読み取ると、そのトランザクションに関連した変更をメモリーから消去します。

アプリケーション 2 は、表 B に対して一連の変更を行います。前の例と同様、キャプチャー・プログラムはこれらの変更をメモリーに収集します。その後、アプリケーション 2 は COMMIT ステートメントを発行します。キャプチャー・プログラムがこのステートメントを読み取ると、表 B に関するそれぞれの変更のコピーを CD 表に付加します。

少なくとも 1 つのキャプチャー・プログラムが、レプリケーション・ソースが存在する各サーバーで実行されていなければなりません。ただし、以下の理由により、1 つのサーバー上で複数のキャプチャー・プログラムを実行することもできます。

### 並列処理のパフォーマンスを向上させたい場合

トラフィックを並列処理するために、複数のキャプチャー・プログラムを使用することができます。複数のキャプチャー・プログラムを使用するとパフォーマンスが向上し、より高いスループットを達成できます。これは、大規模シスプレックスでも有効です。トレードオフは、ログ・リーダーが複数存在することに関連した CPU オーバーヘッドの増加です。複数のキャプチャー・プログラムを使用すると、DB2 UDB 接続もさらに必要になります。

### 異なるレプリケーション要件を満たす必要がある場合

ソースへの変更のフローを異なる使用に向けるために、複数のキャプチャー・プログラムを作成することができます。たとえば、多数のソース表を非常に短い待ち時間で複製する必要があり、現行のキャプチャー・プログラムがすでに多数のソース表のデータをキャプチャーしている場合、その表に対してのみ行われる変更をキャプチャーするために、別のキャプチャー・プログラムを使用することができます。

たとえば、ソース表の 1 つのセットを非常に短い待ち時間で複製する必要があり、日々のデータウェアハウスへのフィードを行うためにソース表の別のセットを複製する必要がある場合、パフォーマンスおよび調整要件は大きく異なる可能性があります。それぞれのソース表のセットに対して個別にキャプチャー・プログラムを使用し、それぞれのアプリケーションのニーズに合わせてキャプチャー・プログラムを調整することができます。

### 個別にコード化スキームを使用する必要がある場合

z/OS™ システムでは、UNICODE または EBCDIC コード化スキームを個別に処理するために、複数のキャプチャー・プログラムを使用できます。

以下の図は、SQL レプリケーションの単純な構成の、キャプチャー・プログラムによるデータのキャプチャーを示しています。

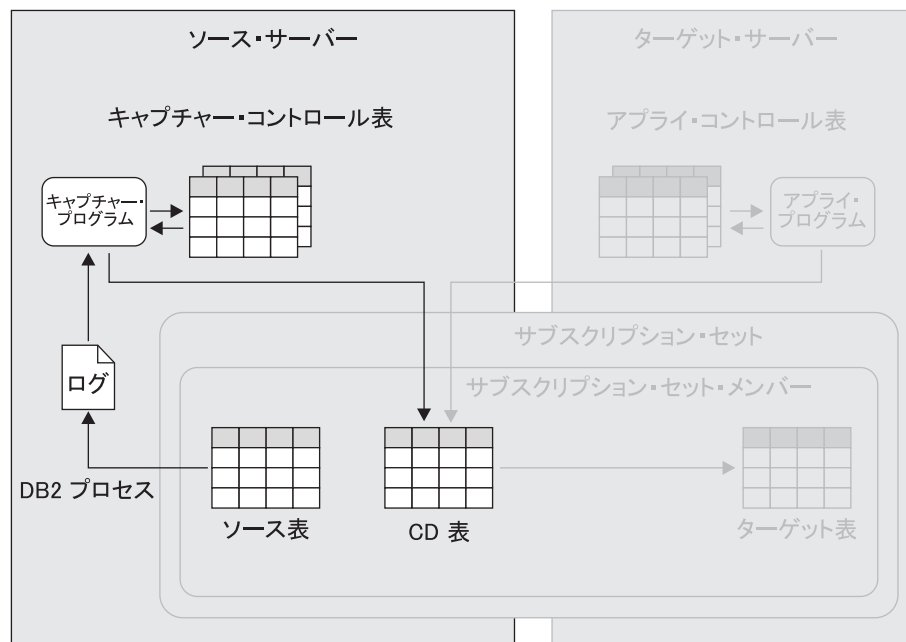


図5. DB2 UDB ソース表からのデータのキャプチャー：キャプチャー・プログラムはソースに対して加えられる変更についての DB2 リカバリー・ログを読み取り、それらの変更をリレーショナル表でステージングします。この図では、キャプチャー・プログラムはソース表に対する変更についての DB2 ログを読み取り、ソースの変更データ (CD) 表でコミット済みトランザクション・データをステージングしています。キャプチャー・プログラムはそのコントロール表を、その進行のモニターに使用できる情報によって継続して更新します。

### 関連概念:

- 7 ページの『SQL レプリケーション — 概要』

## SQL レプリケーションでの DB2 UDB ターゲットへのデータのアプライ

ソースからターゲットへのデータの複製を開始するには、アプライ・プログラムを開始します。その後、アプライ・プログラムは割り当てられたサブスクリプション・セットの処理を開始します。アプライ・プログラムは、サブスクリプション・セットの作成時に指定したスケジューリングまたはイベント基準に従って、すべてのアクティブ・サブスクリプション・セットを一度に 1 つずつ処理します。たとえば、サブスクリプション・セット Set\_One を 60 分ごとに複製することを選択した場合、アプライ・プログラムは可能な限り 60 分に近いインターバルでそのサブスクリプション・セットを複製します。60 分のインターバルが経過したときにアプライ・プログラムに完了すべき他の作業が残っている場合、Set\_One はアプライ・プログラムがその他の作業を完了した後すぐに複製されます。

以下のリストは、アプライ・プログラムによるサブスクリプション・セットの各メンバーの処理方法を示しています。

### 変更キャプチャー・レプリケーション用に登録したソースの場合

アプライ・プログラムが対応するサブスクリプション・セットを最初に処理する際、アプライ・プログラムはターゲットにソースの内容を取り込むことができます。アプライ・プログラムに、EXPORT ユーティリティや LOAD ユーティリティなどの 1 つ以上のユーティリティを呼び出して、ターゲットにデータを取り込むように命令できます。アプライ・プログラムが選択するユーティリティは、アプライ・プログラムが実行されているプラットフォームによって決まります。

その後アプライ・プログラムは、サブスクリプション・セットの作成時に指定した時間間隔で、あるいは毎回イベントが発生するたびに、アプライ・プログラムによる最後のチェック以降 CD 表に挿入された行に関して、ソースの CD 表を読み取ります。CD 表内の行は、それらが、対応するソースに対する削除、更新、または挿入のレコードかどうかを示します。

アプライ・プログラムは CD 表からのデータを使用して、ターゲットで行を挿入、更新、および削除します。更新または削除される行の識別には、述部が使用されます。

### フル・リフレッシュ・レプリケーション用に登録したソースの場合

アプライ・プログラムは、指定のインターバルで、ターゲットにソースの内容を取り込みます。アプライ・プログラムに、EXPORT ユーティリティや LOAD ユーティリティなどの 1 つ以上のユーティリティを呼び出して、ターゲット表をリフレッシュするように命令できます。アプライ・プログラムが選択するユーティリティは、アプライ・プログラムが実行されているプラットフォームによって決まります。

以下の図は、SQL レプリケーションの単純な構成の、アプライ・プログラムによるデータのアプライを示しています。

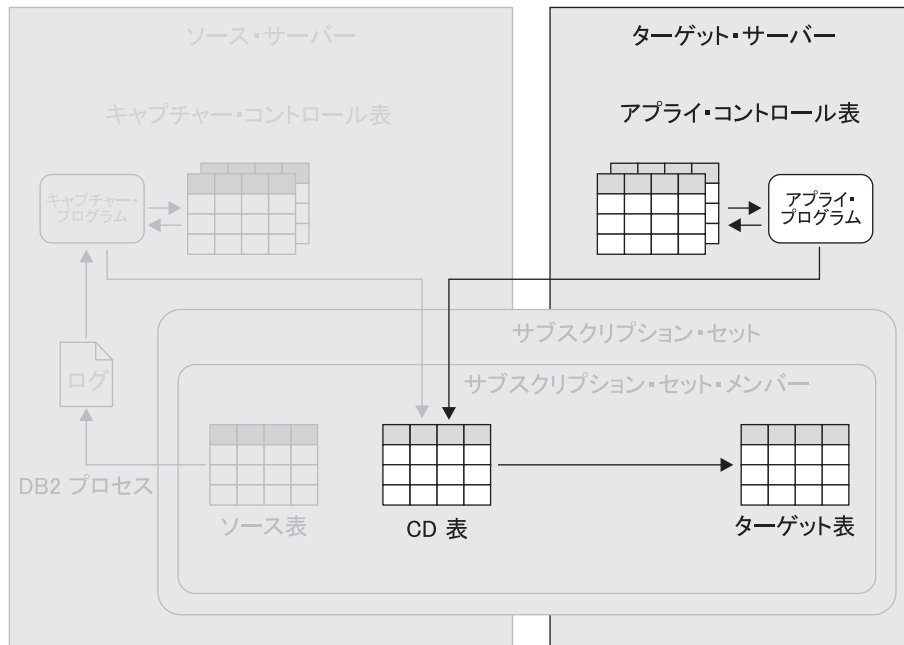


図6. DB2® UDB ターゲットへのデータのアプライ：アプライ・プログラムはステージング表からデータを読み取り、ターゲットに対して適切な変更を行います。この図では、アプライ・プログラムはソース表の CD 表からデータを読み取った後、ターゲット表に対して適切な変更を行っています。アプライ・プログラムはそのコントロール表を、その進行のモニターに使用できる情報によって継続して更新します。

#### 関連概念:

- 7 ページの『SQL レプリケーション — 概要』

## SQL レプリケーションでの非 DB2 ソースからのデータのキャプチャー

非 DB2 リレーショナル・データベース上の表に対する変更のキャプチャーでは、キャプチャー・プログラムは関与しません。その代わりに、表を登録する際に、SQL レプリケーションは表に関するキャプチャー・トリガーと呼ばれる 3 つの変更後トリガー (INSERT トリガー、DELETE トリガー、および UPDATE トリガー)、および整合変更データ (CCD) 表を作成します。この表はソース表に対して行われる変更のレコードを保管するので、CD 表と類似しています。また、SQL レプリケーションは、ソース表のニックネームが存在する同じ DB2® Information Integrator (II) フェデレーテッド・サーバー上に、対応する CCD 表のニックネームを作成します。アプライ・プログラムはこれらのニックネームを使用して CCD 表にアクセスします。

これらのトリガーのいずれかがソース表に対する変更によって活動化されると、トリガーは CCD 表に変更を記録します。アプリケーション・トランザクションがロールバックされる場合、ステージング表の新規レコードもロールバックされます。コミット済みトランザクションの一部である変更のみが複製されます。

また、SQL レプリケーションは、ソース・データベースとして使用される非 DB2 リレーショナル・データベース上に 4 番目のトリガーを作成します。これは、アプ

ライ・プログラムがターゲット表に変更を正常に複製した後、複製されたレコードを CCD 表から削除する整理トリガーです。

以下の図は、非 DB2 リレーショナル・データベース上のソース表に対する変更を DB2 UDB ターゲット表に複製する方法を示しています。

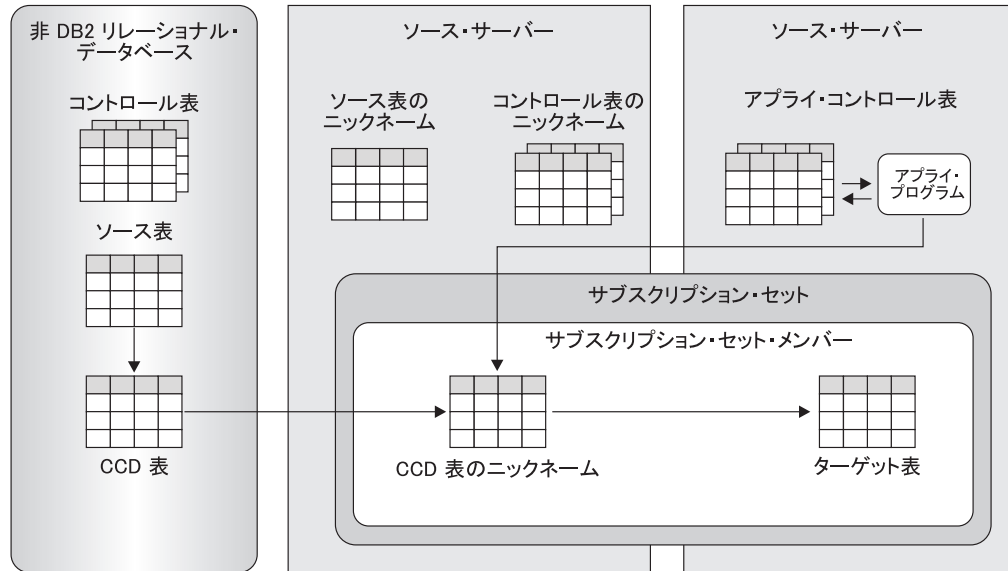


図7. 非 DB2 ソースから DB2 UDB ターゲットへのデータの複製：トリガーは、整合変更データ (CCD) と呼ばれるステージング表に、ソース表に対して行われた変更のレコードを取り込みます。アプライ・プログラムは、DB2 II フェデレーテッド・データベース上の表のニックネームを使用して CCD 表を読み取ります。その後、アプライ・プログラムは、ターゲット表に対して適切な変更を行います。

**関連概念:**

- 7 ページの『SQL レプリケーション — 概要』

## SQL レプリケーションでの非 DB2 ターゲットへのデータのアプライ

DB2® UDB ターゲットに複製する場合と同様、アプライ・プログラムは 2 つの方法で非 DB2 ターゲットへの複製を行えます。変更キャプチャー・レプリケーションを使用する方法と、フル・リフレッシュ・レプリケーションを使用する方法の 2 つです。

### 非 DB2 リレーショナル・データベースのターゲット用の変更キャプチャー・レプリケーション

ソースが DB2 サーバー上に存在する場合、アプライ・プログラムはソースの変更データ (CD) 表を読み取ります。その後、アプライ・プログラムは、フェデレーテッド・データベース上のターゲット表のニックネームにアクセスすることによって、ターゲットに対する変更をアプライします。

ソースが非 DB2 リレーショナル・データベース上に存在する場合、アプライ・プログラムは、CCD 表のニックネームにアクセスすることによって、ソースの CCD 表を読み取ります。その後、ターゲット表のニックネームにアクセスすることによって、ターゲット表に対する変更をアプライします。CCD 表のニックネームとターゲット表のニックネームは、同じ DB2 フェ

デレテッド・データベースに共存することもできますし、異なる DB2 フェデレテッド・データベースに存在することもできます。

### 非 DB2 リレーショナル・データベースのターゲット用のフル・リフレッシュ・レプリケーション

ソースが DB2 サーバー上に存在する場合、アプライ・プログラムはソースを直接読み取り、ターゲット表のニックネームにアクセスすることによってターゲット表の行を削除し、ターゲット表のニックネームにアクセスすることによってターゲット表にソース行を挿入します。

ソースが非 DB2 リレーショナル・データベース上に存在する場合、アプライ・プログラムは、ソースのニックネームにアクセスすることによって、ソースの行を読み取ります。その後、ニックネームにアクセスすることによってターゲットの行を削除し、ターゲット表のニックネームにアクセスすることによってターゲット表にソース行を挿入します。ソース表のニックネームとターゲット表のニックネームは、同じ DB2 フェデレテッド・データベースに共存することもできますし、異なる DB2 フェデレテッド・データベースに存在することもできます。

以下の図は、DB2 UDB ソースに対する変更を非 DB2 リレーショナル・データベース上の表に複製する方法を示しています。

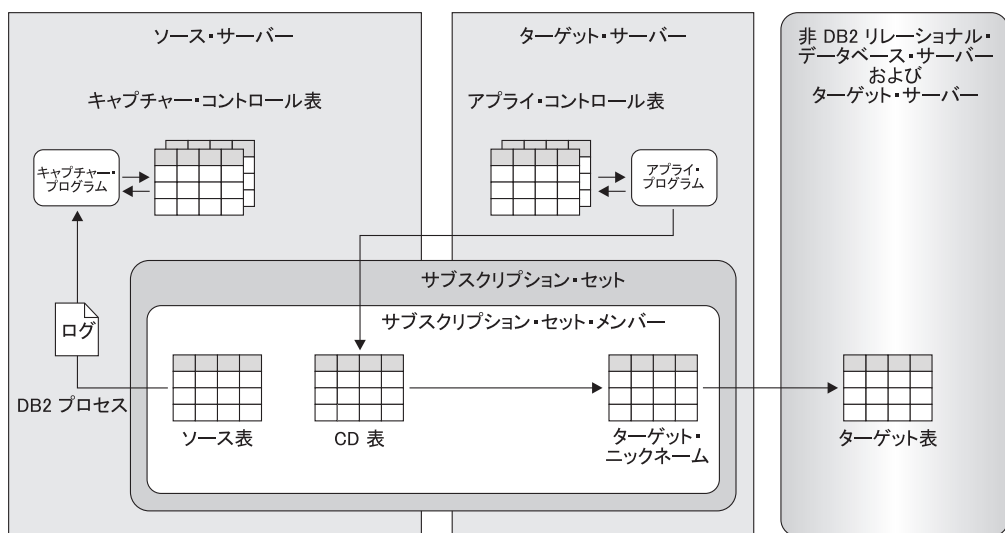


図8. DB2 UDB ソースから非 DB2 ターゲットへの複製：キャプチャー・プログラムは、CD 表に、対応するソース表に対して加えられる変更をキャプチャーします。アプライ・プログラムは CD 表のデータを読み取り、DB2 Information Integrator フェデレテッド・データベース上のターゲットのニックネームを使用してターゲット表に変更をアプライします。

#### 関連概念:

- 7 ページの『SQL レプリケーション — 概要』



## 第 3 章 Q レプリケーション

### Q レプリケーション — 概要

以下のトピックでは、Q レプリケーションに関する概念について説明しています。特定の詳細については、「*IBM DB2 Information Integrator レプリケーションとイベント・パブリッシング ガイド*およびリファレンス」を参照してください。

- Q レプリケーション環境のインフラストラクチャー
- Q レプリケーションのソースとターゲット
- Q レプリケーションでのデータのキャプチャー
- Q レプリケーションでのターゲットへのデータの適用
- Q レプリケーションのレプリケーション・タイプ

以下の図は、Q レプリケーションの単純な構成を示しています。上記のトピックではそれぞれ、この図の各セクションについて詳しく取り上げて説明しています。

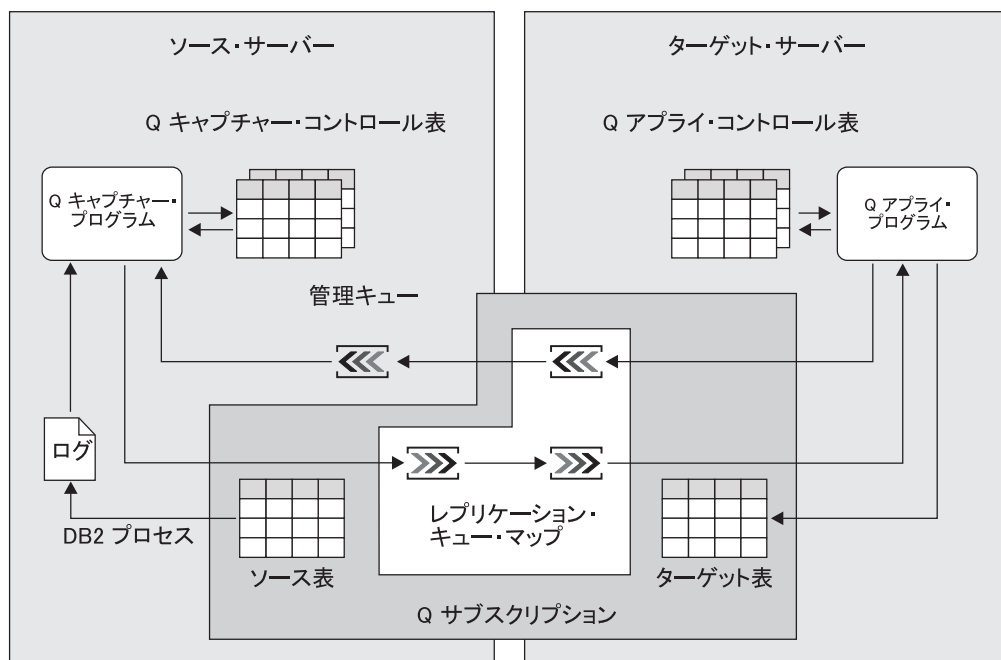


図9. Q レプリケーションの単純な構成

#### 関連概念:

- 29 ページの『Q レプリケーションでのターゲットへのデータのアプライ』
- 27 ページの『Q レプリケーションでのデータのキャプチャー』
- 24 ページの『Q レプリケーション環境のインフラストラクチャー』
- 25 ページの『Q レプリケーションのソースとターゲット』
- 30 ページの『Q レプリケーションのレプリケーション・タイプ』

## Q レプリケーション環境のインフラストラクチャー

Q レプリケーションを使用すると、Q キャプチャーおよび Q アプライという 2 つのプログラムを使うことによって DB2® UDB ソースからターゲットにコミット済みトランザクション・データを複製することができます。Q キャプチャー・プログラムはソース・システム上で実行されます。Q キャプチャー・プログラムは変更されたソース・データに関する DB2 リカバリー・ログを読み取り、WebSphere® MQ キューに変更を書き込みます。Q アプライ・プログラムは、ターゲット・システム上で実行されます。Q アプライ・プログラムは、キューからキャプチャーされた変更を検索し、ターゲットに変更を書き込みます。どちらのプログラムも、タスクの実行に必要な情報を追跡したり、生成される情報（プログラムの実行状態の調査に使用できる情報など）を保管したりするために DB2 表のセットを使用します。これらの表は、レプリケーション・ソース何でターゲットののが何かなどの設定を行う前にあらかじめ作成します。

Q キャプチャー・プログラムは、Q キャプチャー・コントロール表 というコントロール表のセットを使用します。これらの表には、レプリケーション・ソース、それに対応するターゲット、および Q キャプチャー・プログラムによって使用されている WebSphere MQ キュー・マネージャーに関する情報が入ります。また、これらの表には、Q キャプチャー・プログラムのパフォーマンスのチェックおよびモニターに使用できるデータ、たとえば DB2 リカバリー・ログ内の Q キャプチャー・プログラムの現行位置に関するデータも入ります。

同じ DB2 UDB サーバー上で複数の Q キャプチャー・プログラムを実行することができます。各 Q キャプチャー・プログラムは、独自のコントロール表セットを使用します。Q キャプチャー・コントロール表のセットと関連付けられているスキーマは、それらのコントロール表を使用する Q キャプチャー・プログラムを識別します。このスキーマは、Q キャプチャー・スキーマ と呼ばれます。

Q アプライ・プログラムは、Q アプライ・コントロール表 というコントロール表のセットを使用します。これらの表には、ターゲットに関する情報、それに対応するソースのロケーションに関する情報、および Q アプライ・プログラムによって使用されている WebSphere MQ キュー・マネージャーに関する情報が入ります。Q キャプチャー・コントロール表と同様、これらの表には、Q キャプチャー・プログラムのパフォーマンスのチェックおよびモニターに使用できるデータも入ります。

Q キャプチャー・プログラムと同様、同じ DB2 UDB サーバー上で複数の Q アプライ・プログラムを実行することができます。各 Q アプライ・プログラムは、独自のコントロール表セットを使用します。Q アプライ・コントロール表のセットと関連付けられているスキーマは、それらのコントロール表を使用する Q アプライ・プログラムを識別します。このスキーマは、Q アプライ・スキーマ と呼ばれます。

以下の図は、Q レプリケーションの単純な構成のインフラストラクチャーを示しています。



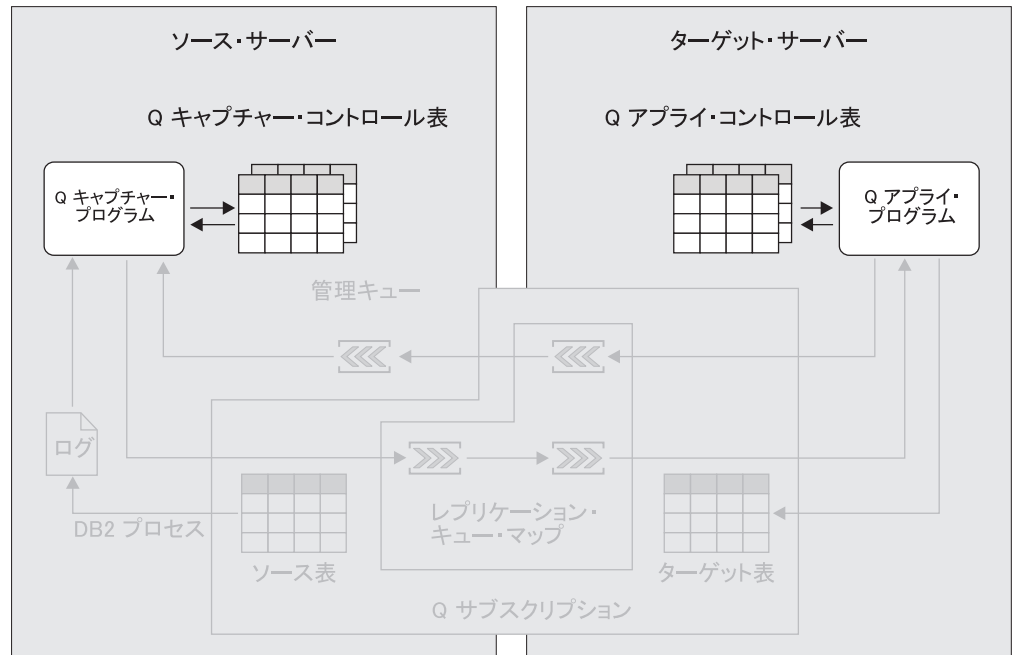


図 10. Q レプリケーションの単純な構成のインフラストラクチャー： ソース・サーバー上に、Q キャプチャー・コントロール表と呼ばれる DB2 リレーショナル表のセットを作成します。ソースとターゲットに関する情報はこれらの表に書き込まれます。ソース・サーバー上で実行される Q キャプチャー・プログラムは、この情報を使用して、予想されるキャプチャー・データを識別し、Q アプライ・プログラムに送信します。ソースとターゲットに関する情報は Q アプライ・コントロール表に渡されます (この表は、ターゲット・サーバー上にあります)。ターゲット・サーバー上で実行される Q アプライ・プログラムは、この情報を使用して、予想されるデータの書き込み先ターゲットを識別します。

**関連概念:**

- 23 ページの『Q レプリケーション — 概要』

## Q レプリケーションのソースとターゲット

ソース表とターゲットの対は、Q サブスクリプション を使って作成します。Q サブスクリプションを作成する際、以下の基本属性を指定します。

**ソース・サーバーとソース表**

ソース・サーバーは、少なくとも 1 セットの Q キャプチャー・コントロール表を含む DB2® UDB サーバーです。ソース表には、ソース・サーバー上に存在する DB2 UDB リレーショナル表を使用できます。

**ターゲット・サーバーとターゲット表またはストアド・プロシージャ**

ターゲット・サーバーは、少なくとも 1 セットの Q アプライ・コントロール表を含む DB2 UDB サーバーです。ターゲット表には、ターゲット・サーバー上に存在する DB2 UDB リレーショナル表を使用できます。あるいは、Q サブスクリプションの作成時に Q レプリケーションに表の定義を命令することもできます。

複製データを変換したい場合は、ターゲットとしてストアド・プロシージャを使用できます。

## ターゲット表またはストアド・プロシージャーに複製するソース列。

ソースから複製する列のサブセットを選択できます。

## ストアド・プロシージャーのターゲット列またはパラメーターにソース列をマップする方法

ターゲット・ロケーションにすでにターゲット表が存在する場合、Q レプリケーションに、ソース表内で選択した列とターゲット表内の列の対応関係を知らせます。Q レプリケーションにターゲット表の作成を実行させたい場合、選択ソース列はターゲット表の列に自動的にマップされます。

ストアド・プロシージャーに複製している場合は、Q レプリケーションに、ソース表内で選択した列とストアド・プロシージャー内のパラメーターの対応関係を知らせます。

## 行のサブセットをターゲット表に複製するための述部。

ソースから複製する行のサブセットを選択することもできます。

## ターゲット表をロードする方法

ほとんどの場合、レプリケーションを開始する前に、ターゲット表がソース表と同じになるように、ソース表からのデータはターゲット表にロードされます。Q レプリケーションでは、Q サブスクリプションを開始した後、自動ロードと手動ロードの 2 つのターゲット表のロード方法を使用できます。

自動ロードでは、Q アプライ・プログラムはターゲット表のロードを管理します。Q アプライ・プログラムは、ロードを実行するために、1 つのユーティリティーまたはユーティリティーの対を呼び出します。Q アプライ・プログラムが実行されるプラットフォームに応じて、LOAD ユーティリティーの LOAD FROM CURSOR オプション、EXPORT ユーティリティーと LOAD ユーティリティー、または EXPORT ユーティリティーと IMPORT ユーティリティーを呼び出すことを Q アプライ・プログラムに命令できます。また、これらのオプションのうち最も Q サブスクリプションに適しているオプションを Q アプライ・プログラムが独自に決定するように命令することもできます。

手動ロードでは、ターゲット表のロードを処理した後、ロードの完了時に複製プログラムに信号を送ります。

Q サブスクリプションの開始時に、Q キャプチャー・プログラムおよび Q アプライ・プログラムによるロード・プロセスの調整を行わずに即時にレプリケーションを開始することを指定することもできます。レプリケーション環境に多数の新規表を追加した場合にもこれを行えます。この場合、休止期間 (たとえば週末など) を選択し、ソース表を静止し、ターゲット表を手動でロードし、その後 Q キャプチャー・プログラムおよび Q アプライ・プログラムを開始します。

## メッセージのトランスポートで使用する WebSphere® MQ メッセージ・キュー

Q キャプチャー・プログラムは、送信キューと呼ばれるキューにメッセージを挿入します。Q アプライ・プログラムは、受信キューと呼ばれるキューのメッセージを受け取ります。(この名前は Q レプリケーションでのみ使用され、WebSphere MQ では使用されません。) Q サブスクリプシ

ンのメッセージについて使用する送信キューと受信キューを Q レプリケーションに知らせるために、レプリケーション・キュー・マップを選択します。

以下の図は、Q レプリケーションの単純な構成の Q サブスクリプションの一部を示しています。

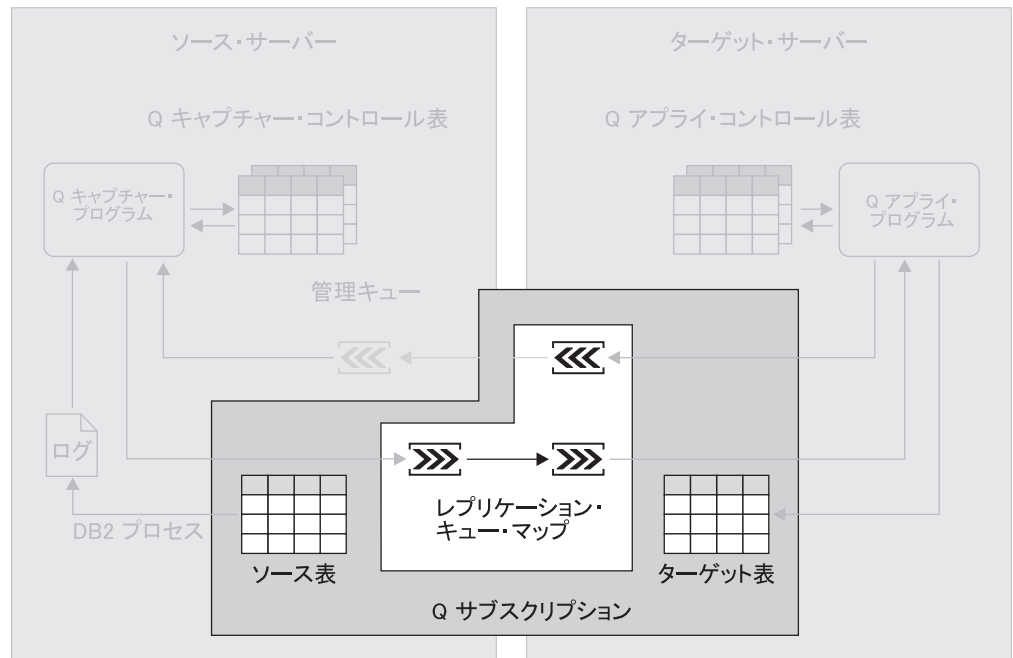


図 11. Q サブスクリプションの作成：Q サブスクリプションを作成することによって、ソースをターゲットにマップします。1 つの Q サブスクリプションは、1 つのソース表を 1 つのターゲット表にマップします。Q サブスクリプションのデータは、送信キューと受信キューを介して複製されます。これらのキューはいずれもレプリケーション・キュー・マップの一部です。

#### 関連概念:

- 23 ページの『Q レプリケーション — 概要』

## Q レプリケーションでのデータのキャプチャー

Q キャプチャー・プログラムは、DB2® ログを順次読み取り、ソース表に対して変更を加えます。Q キャプチャー・プログラムは、いずれかのソース表に対する変更を読み取ると、メモリー内に保持されている対応するデータベース・トランザクションに変更を追加します。そのため、メモリー内のトランザクションは、潜在的に、ログ内の対応するトランザクションのサブセットになります。これには、Q サブスクリプションのソース表に対する変更のみが入るためです。Q キャプチャー・プログラムがトランザクションの COMMIT ステートメントを読み取る際、トランザクションをメッセージに変換し、そのメッセージを送信キューに挿入します。

たとえば、同じソース・サーバー上に、2 つの異なるソース表に関する 2 つの Q サブスクリプション、QSUB1 および QSUB2 を作成します。どちらの Q サブスクリプションも、同じレプリケーション・キュー・マップを使用します。Q キャプ

チャー・プログラムは、両方の Q サブスクリプションのソース表への変更を含むデータベース・トランザクションの COMMIT を読み取ります。Q キャプチャー・プログラムは変更をメッセージに変換し、レプリケーション・キュー・マップの一部である送信キューにメッセージを書き込みます。

同じソース・サーバー上で複数の Q キャプチャー・プログラムを実行できます。1つの Q キャプチャー・プログラムでも多数のソースに対する変更をキャプチャーし、それらの変更を多数のターゲット・サーバーに送信することはできますが、複数の Q キャプチャー・プログラムを使用の方が都合のよい場合もあります。たとえば、複数の Q キャプチャー・プログラムを使用すると、トラフィックを並列処理することができます。複数の Q キャプチャー・プログラムを使用するとパフォーマンスが向上し、より高いスループットを達成できます。これは、大規模シスプレックスでも有効です。トレードオフは、ログ・リーダーが複数存在することに関連した CPU オーバーヘッドの増加です。複数の Q キャプチャー・プログラムを使用すると、DB2 UDB 接続もさらに必要になります。

以下の図は、Q レプリケーションの単純な構成の、Q キャプチャー・プログラムによるデータのキャプチャーを示しています。

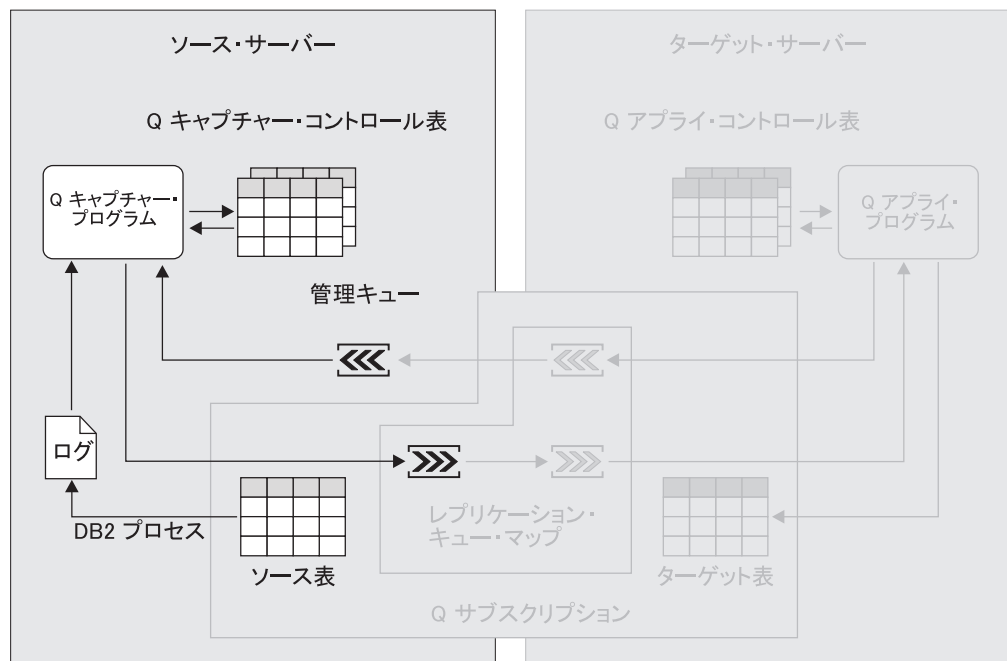


図 12. Q レプリケーションでのデータのキャプチャー： Q キャプチャー・プログラムはソースに対して加えられる変更についての DB2 リカバリー・ログを読み取り、コミット済みトランザクション・データをメッセージに変換し、そのメッセージは送信キューに挿入されます。Q キャプチャー・プログラムはそのコントロール表を、その進行のモニターに使用できる情報によって継続して更新します。

**関連概念:**

- 23 ページの『Q レプリケーション — 概要』

---

## Q レプリケーションでのターゲットへのデータのアプリ

Q アプリ・プログラムは、Q サブスクリプションに関するコミット済みトランザクション・データを含むメッセージを読み取ります。これらのメッセージは、ターゲット・サーバーの受信キューに入ります。Q アプリ・プログラムはそのメッセージを SQL に変換し、関連するターゲット表にトランザクションをアプリします。Q アプリ・プログラムはマルチスレッド・プログラムであるため、トランザクションが相互依存していない場合にはいつでも、それら複数のトランザクションを並行してアプリできます。トランザクションが相互依存している場合、Q アプリ・プログラムは、ソース・サーバーでコミットされた順にそれらのトランザクションをアプリします。

Q アプリ・プログラムは、多数の Q サブスクリプションに関する受信メッセージを 1 つの受信キューで受け取ることができ、トランザクションを高速でアプリできます。ほとんどの場合、単一のソース・サーバー/ターゲット・サーバー間では単一のレプリケーション・キュー・マップを使用でき、それほど大きな待ち時間は経験することはありません。

Q キャプチャー・プログラムと Q アプリ・プログラムの 1 つの対を、複数のレプリケーション・キュー・マップを使用して構成することができます。個々の表セットを更新する複数のアプリケーションがソース・サーバーで実行されている場合、それぞれの表セットでの並列送信とデータのアプリが可能となるように、複数のレプリケーション・キュー・マップの定義を検討する必要があります。Q アプリ・プログラムは、各受信キューごとにマルチスレッド・プロセスを作成します。

ほとんどの場合 1 つのサーバーにつき 1 つのアプリ・プログラムで十分です。多数の表に複製されている多数のトランザクションを持つレプリケーション構成の場合でも同様です。1 つの Q アプリ・プログラムが 1 つ以上の受信キューを処理するように構成できます。たとえば、1 つ以上の受信キューのデータを他の受信キューのデータとは別に処理する必要がある場合にも、サーバー上で複数の Q アプリ・プログラムを実行できます。

以下の図は、Q レプリケーションの単純な構成の、Q アプリ・プログラムによるデータのアプリを示しています。

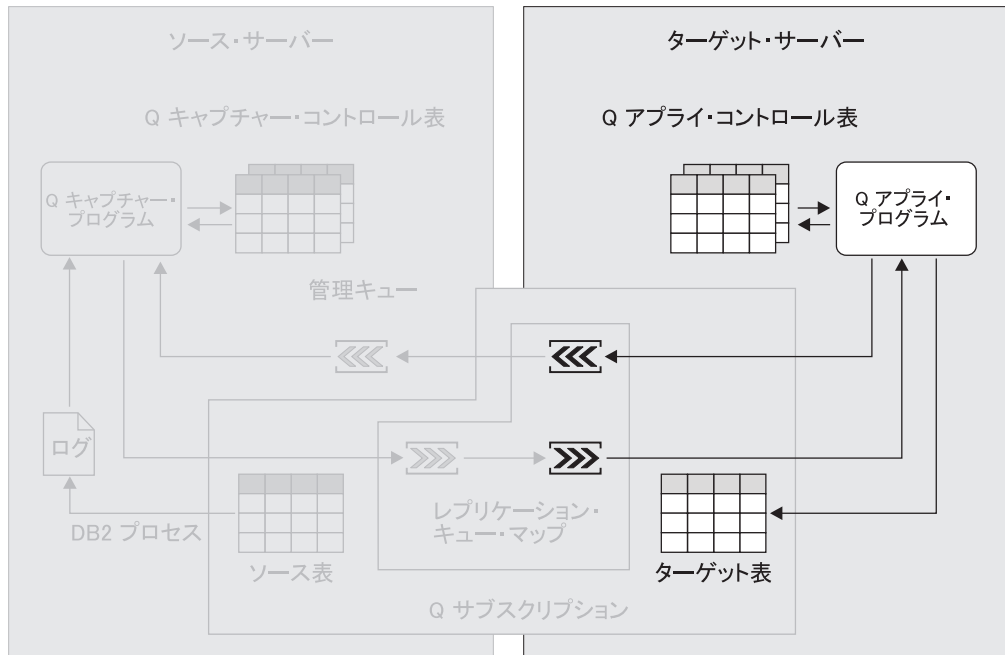


図 13. Q レプリケーションでのデータのアプライ：Q アプライ・プログラムは受信キューからメッセージを読み取り、それを SQL に変換し、データをターゲット表にアプライします。Q アプライ・プログラムはそのコントロール表を、その進行のモニターに使用できる情報によって継続して更新します。

**関連概念:**

- 23 ページの『Q レプリケーション — 概要』

## Q レプリケーションのレプリケーション・タイプ

Q レプリケーションでは、3 つの異なるレプリケーション・タイプを構成できます。

### 単一方向レプリケーション

単一方向レプリケーションでは、ソースに対して発生する変更がターゲットに複製されます。ターゲット表は通常、読み取り専用アプリケーションによってのみ使用されます。ソースからの行と列すべてを複製することも、列と行のサブセットを複製することを選択することもできます。ターゲットからソースへは複製できません。

1 つのソースを 1 つのターゲットに複製するか、1 つのソースを複数のターゲットに複製するか、複数のソースを 1 つのターゲットに複製するか、または複数のソースを複数のターゲットに複製することができます。ソースとそれに対応するターゲットごとに 1 つの Q サブスクリプションがあります。Q サブスクリプションのターゲットが論理的に関連している場合、その Q サブスクリプションは同じレプリケーション・キュー・マップを使用する必要があります。このように Q サブスクリプションをグループ化することによって、ターゲットに適用されるデータがソース・サーバー上のオリジナルのトランザクションと確実に整合するようにします。

それぞれのソース・サーバーで少なくとも 1 セットの Q キャプチャー・コントロール表を作成します。そして、それぞれのターゲット・サーバーで少なくとも 1 セットの Q アプライ・コントロール表を作成します。

単一方向レプリケーションの Q サブスクリプションでは、Q アプライ・プログラムを使用してストアード・プロシージャを呼び出して、ソース・データを入力パラメーターとしてストアード・プロシージャに渡すことができます。ソース列は、ターゲット列にマップする代わりに、ストアード・プロシージャ内のパラメーターにマップします。ソース列を直接ストアード・プロシージャのパラメーターにマップすることによって、着信データを構文解析せずに済み、クリーンで単純なプログラミング・モデルを持つことができます。Q アプライ・プログラムは行操作が行われるたびに、行を表に挿入する代わりにストアード・プロシージャを呼び出します。ストアード・プロシージャには、ソース・データを最終目的地に送る役割があります。

## 双方向レプリケーション

双方向レプリケーションの場合、レプリケーションは 2 つのサーバー上の表の間で行われます。レプリケーションに関係する一方のサーバー上の表の構造は、対応する他方のサーバー上の表の構造と同じになります。対応する 2 つの表の列の数、列の名前は同じになり、データ・タイプも互換性を持ちます。ただし、表のスキーマと名前は異なる場合があります。行のサブセットは複製できません。一方のサーバー上の表に加えられる変更は、対応する他方のサーバー上の表に複製されます。

一方のサーバー上のアプリケーションが表に対して変更を加えると、他方のサーバー上のアプリケーションも対応する表に対して同時に変更を加えます。対応する表の間で複製されるデータで競合が発生する場合、2 つの表のうち優先される表を選択できます。競合は、古い値と現行値を比較することによって検出できます。この検出方法はデータ競合を見つけれない場合もありますが、他の方法よりもオーバーヘッドは少なく済みます。

対応する 2 つの表にはすべて 2 つの Q サブスクリプションがあります。たとえば、サーバー Red 上の Table\_One とサーバー Blue 上の Table\_Two の間で複製を行っている場合、次の 2 つの Q サブスクリプションが存在します。

- Table\_One から Table\_Two へ複製する Q サブスクリプション
- Table\_Two から Table\_One へ複製する Q サブスクリプション

Q サブスクリプションを作成する際、最初のロードを自動実行したい場合、開始データを入れる表を選択することができます。この例では、Table\_One に開始データを入れることを指定する場合、Table\_Two は Table\_One からのデータとともにロードされます。

Q キャプチャー・プログラムと Q アプライ・プログラムは両方のサーバーで実行されます。各サーバーの Q キャプチャー・プログラムと Q アプライ・プログラムのスキーマは同じになります。たとえば:

- 一方のサーバー上の Q キャプチャー・プログラムのスキーマが Green の場合、そのサーバー上の Q アプライ・プログラムのスキーマも Green になります。



- 他方のサーバー上の Q キャプチャー・プログラムのスキーマが Yellow の場合、そのサーバー上の Q アプライ・プログラムのスキーマも Yellow になります。

### 対等レプリケーション

対等レプリケーションの場合、レプリケーションは 2 つ以上のサーバー上の表の間で行われます。レプリケーションに関係する一方のサーバー上の表の構造は、対応する他のサーバー上の表の構造と同じになります。対応するすべての表の列の数、列の名前は同じになり、データ・タイプも互換性を持ちます。ただし、表のスキーマと名前は異なる場合があります。行のサブセットは複製できません。いずれかのサーバー上の表に加えられる変更は、対応する他のサーバー上の表に複製されます。

一方のサーバー上のアプリケーションが表に対して変更を加えると、他のサーバー上のアプリケーションも対応する表に対して同時に変更を加えます。競合は、Q サブスクリプションの作成時に表に追加するバージョン列およびトリガーによって検出され、解決されます。対等レプリケーションを使用すると、収束が可能です。これは、複製表に対する変更が停止され、すべての変更が伝搬される場合、対応する表も同じになることを意味します。

対応する各組の表にはそれぞれ 2 つの Q サブスクリプションがあります。たとえば、サーバー Red 上の Table\_One、サーバー Blue 上の Table\_Two、およびサーバー Green 上の Table\_Three の間で複製を行っている場合、次の 3 組の表の対応が存在します。

- Table\_One と Table\_Two
- Table\_Two と Table\_Three
- Table\_One と Table\_Three

これらの組にはそれぞれ 2 つの Q サブスクリプションが存在します。たとえば、Table\_One と Table\_Two の間には次の 2 つの Q サブスクリプションが存在します。

- Table\_One から Table\_Two へ複製する Q サブスクリプション
- Table\_Two から Table\_One へ複製する Q サブスクリプション

Q キャプチャー・プログラムと Q アプライ・プログラムはそれぞれのサーバーで実行されます。各サーバーの Q キャプチャー・プログラムと Q アプライ・プログラムのスキーマは同じになります。たとえば、2 つのサーバー間で複製を行っている場合、次のようになります。

- 一方のサーバー上の Q キャプチャー・プログラムのスキーマが Blue の場合、同じサーバー上の Q アプライ・プログラムのスキーマも Blue になります。
- 他方のサーバー上の Q キャプチャー・プログラムのスキーマが Red の場合、同じサーバー上の Q アプライ・プログラムのスキーマも Red になります。

### 関連概念:

- 「*IBM DB2 Information Integrator レプリケーションとイベント・パブリッシングガイド*およびリファレンス」の『単方向レプリケーション (リモート) で必要な WebSphere MQ オブジェクト』



- 「*IBM DB2 Information Integrator レプリケーションとイベント・パブリッシングガイドおよびリファレンス*」の『双方向または対等レプリケーションに必要な WebSphere MQ オブジェクト (2 つのリモート・サーバー)』
- 「*IBM DB2 Information Integrator レプリケーションとイベント・パブリッシングガイドおよびリファレンス*」の『同一のシステムでの単一方向レプリケーションに必要な WebSphere MQ オブジェクト』
- 「*IBM DB2 Information Integrator レプリケーションとイベント・パブリッシングガイドおよびリファレンス*」の『対等レプリケーション (3 つ以上のリモート・サーバー) に必要な WebSphere MQ オブジェクト』



---

## 第 4 章 SQL レプリケーションと Q レプリケーションの比較

---

### SQL レプリケーションと Q レプリケーションの比較 — 概要

以下のトピックでは、SQL レプリケーションと Q レプリケーションの間の類似点および違いについて説明しています。

- SQL レプリケーションと Q レプリケーションのインフラストラクチャーの比較
- SQL レプリケーションと Q レプリケーションのソースとターゲットの比較
- SQL レプリケーションと Q レプリケーションのデータの取り込みおよび適用の比較
- 一般的なシナリオで使用すべきレプリケーション・ソリューション

#### 関連概念:

- 35 ページの『SQL レプリケーションと Q レプリケーションのインフラストラクチャーの比較』
- 37 ページの『SQL レプリケーションと Q レプリケーションのデータのキャプチャーとアプライの比較』
- 36 ページの『SQL レプリケーションと Q レプリケーションのソースとターゲットの比較』
- 39 ページの『一般的なシナリオで使用すべきレプリケーション・ソリューション』

---

### SQL レプリケーションと Q レプリケーションのインフラストラクチャーの比較

表 1 は、SQL レプリケーションと Q レプリケーションのインフラストラクチャーを比較しています。

表 1. SQL レプリケーションと Q レプリケーションにおけるインフラストラクチャーの比較

比較のポイント	SQL レプリケーション	Q レプリケーション
キャプチャー・プログラム用のコントロール表のロケーション	キャプチャー・プログラム用のコントロール表は、キャプチャー・プログラムが実行されている DB2 <sup>®</sup> UDB サーバー上に作成する必要があります。ほとんどの場合、このサーバーは、プログラムと関連したソースが存在する DB2 UDB サーバーと同じサーバーになります。	Q キャプチャー・プログラム用のコントロール表は、Q キャプチャー・プログラムが実行されている DB2 UDB サーバー上に作成する必要があります。このサーバーは、そのプログラムと関連したソース表が存在する DB2 UDB サーバーと同じサーバーになります。
DB2 UDB サーバーごとに可能なキャプチャー・プログラムの数	複数可能。それぞれが独自のコントロール表セットを持ち、それらのコントロール表のスキーマによって識別されます。	複数可能。それぞれが独自のコントロール表セットを持ち、それらのコントロール表のスキーマによって識別されます。

表1. SQL レプリケーションと Q レプリケーションにおけるインフラストラクチャーの比較 (続き)

比較のポイント	SQL レプリケーション	Q レプリケーション
アプライ・プログラム用のコントロール表のロケーション	アプライ・プログラム用のコントロール表は、ソース・サーバーとターゲット・サーバーに接続可能な任意のサーバー上に作成できます。一般に、コントロール表は、アプライ・プログラムが実行されるサーバー上に存在していなければなりません。	Q アプライ・プログラム用のコントロール表は、そのプログラムに関連したターゲット表が存在する DB2 UDB サーバー上に作成する必要があります。
DB2 UDB サーバーごとに可能なアプライ・プログラムの数	複数可能。それぞれが 1 つのコントロール表セットを共用し、アプライ修飾子と呼ばれるストリングによって識別されます。	複数可能。それぞれが独自のコントロール表セットを持ち、それらのコントロール表のスキーマによって識別されます。

**関連概念:**

- 35 ページの『SQL レプリケーションと Q レプリケーションの比較 — 概要』

## SQL レプリケーションと Q レプリケーションのソースとターゲットの比較

表2 は、SQL レプリケーションおよび Q レプリケーションでのソースとターゲットの扱われ方を比較しています。

表2. SQL レプリケーションと Q レプリケーションのソースとターゲットの比較

比較のポイント	SQL レプリケーション	Q レプリケーション
ソースとターゲットのプラットフォームフォーム	ソースとターゲットは、次のリレーショナル・データベース管理システム上に存在できます。 <ul style="list-style-type: none"> <li>• DB2<sup>®</sup> UDB for Linux、UNIX<sup>®</sup>、および Windows<sup>®</sup></li> <li>• DB2 UDB for z/OS<sup>™</sup></li> <li>• DB2 UDB for iSeries<sup>™</sup></li> <li>• Informix<sup>®</sup></li> <li>• Microsoft<sup>®</sup> SQL Server</li> <li>• Oracle</li> <li>• Sybase</li> <li>• Teradata (ターゲットのみ)</li> </ul>	ソースとターゲットは、次のリレーショナル・データベース管理システム上に存在できます。 <ul style="list-style-type: none"> <li>• DB2 UDB for Linux、UNIX、および Windows</li> <li>• DB2 UDB for z/OS</li> </ul>
ターゲットになることができるデータベース・オブジェクト	DB2 UDB 表およびビュー、および非リレーショナル・データベース上の表。	DB2 UDB 表およびストアド・プロシージャ。

表2. SQL レプリケーションと Q レプリケーションのソースとターゲットの比較 (続き)

比較のポイント	SQL レプリケーション	Q レプリケーション
ソースとターゲットのペ ア 化	ソースを登録します。この情報はキャプチャー・コントロール表に保管されます。その後、1 つ以上のサブスクリプション・セット・メンバーを作成し、この登録済みソースをターゲットにマップします。これらのサブスクリプション・セット・メンバーに関する情報は、アプライ・コントロール表に保管されます。	Q サブスクリプションを作成し、ソースを単一のターゲットにマップします。ソースの登録は不要です。各ターゲットごとに 1 つの Q サブスクリプションを作成することによって、ソースを複数のターゲットに複製することができます。Q サブスクリプションに関する情報は、Q キャプチャー・コントロール表および Q アプライ・コントロール表に保管されます。
ソースとターゲットの対の グループ化	ソースとターゲットの対をサブスクリプション・セットにグループ化します。ソースとターゲットの対はそれぞれ、サブスクリプション・セット・メンバーと呼ばれます。	Q サブスクリプションをレプリケーション・キュー・マップごとにグループ化できます。サブスクリプション・セット・オブジェクトは Q レプリケーションに存在しません。
ソースの列および行のサブ セット選択の可否	可能。	可能。
データのトランスフォーム	<p>ストアード・プロシージャか、またはアプライ・プログラムによって実行される SQL ステートメントによってデータをトランスフォームできます。ソース・サーバー上では、ストアード・プロシージャまたは SQL ステートメントはステージングされたデータをトランスフォームできます。ターゲット・サーバー上では、データをターゲットにトランスフォームできます。</p> <p>ストアード・プロシージャ・インターフェースはトランスフォームを制限します。アプライ・プログラムはストアード・プロシージャにパラメーターを渡さず、これを、サブスクリプション・セット・メンバーごとではなく、サブスクリプション・セット全体に対して一度実行します。</p> <p>キャプチャー・プログラムがステージング表に書き込むデータのトランスフォームにトリガーを使うこともできます。</p>	<p>ストアード・プロシージャをターゲットとして使用することによって、データをトランスフォームできます。Q アプライ・プログラムはストアード・プロシージャを呼び出し、変更データをストアード・プロシージャにパラメーターとして渡します。ストアード・プロシージャを使用するこの方法を使うと、データのトランスフォームにおける汎用性がかなり広がります。</p>

**関連概念:**

- 35 ページの『SQL レプリケーションと Q レプリケーションの比較 — 概要』

## SQL レプリケーションと Q レプリケーションのデータのキャプチャーとアプライの比較

38 ページの表 3 は、SQL レプリケーションと Q レプリケーションのデータのキャプチャーとアプライを比較しています。

表 3. SQL レプリケーションと Q レプリケーションのデータのキャプチャーとアプライの比較

比較のポイント	SQL レプリケーション	Q レプリケーション
キャプチャー・プログラムのロケーション	キャプチャー・プログラムは、そのコントロール表が存在する DB2 <sup>®</sup> UDB サーバー上で実行されます。ほとんどの場合、ソース表も同じサーバー上に存在します。	Q キャプチャー・プログラムは、そのコントロール表が存在する DB2 UDB サーバー上で実行されます。ソース表も同じサーバー上に存在します。
DB2 変更データのキャプチャー方法	キャプチャー・プログラムは DB2 リカバリー・ログを読み取り、コミット済みトランザクション・データをステージング表に保管します。	Q キャプチャー・プログラムは DB2 リカバリー・ログを読み取り、コミット済みトランザクション・データをメッセージに変換します。
DB2 UDB 変更データのトランスポート方法	アプライ・プログラムはステージング表からデータをフェッチし、DB2 SQL を使用してそのデータをターゲットにアプライします。	Q キャプチャー・プログラムはデータをメッセージとしてキューに挿入します。WebSphere <sup>®</sup> MQ はメッセージをターゲット・システムに移動します。Q アプライ・プログラムはこれらのメッセージをキューから取り出し、DB2 SQL を使用してターゲットにデータをアプライします。
アプライ・プログラムのロケーション	アプライ・プログラムはネットワーク内の任意の DB2 UDB サーバー上で実行できます。ただし、ソース、ターゲット、およびアプライ・コントロール表を含むサーバーに接続可能でなければなりません。	Q アプライ・プログラムはターゲット・サーバー上で実行されます。
DB2 UDB ターゲットにデータをアプライする方法	<p>アプライ・プログラムは、表モードまたはトランザクション・モードでサブスクリプション・セットを処理できます。</p> <p>表モードでは、アプライ・プログラムはそれぞれの表ごとに個別にフェッチした変更を処理し、データがすべてアプライされた後で単一コミットを発行します。</p> <p>トランザクション・モードでは、アプライ・プログラムはフェッチした変更を、そのソース・サーバー上の対応するトランザクションで変更が発生した順にすべてのターゲット表にアプライします。アプライ・プログラムはこれらのトランザクションをトランザクション境界でコミットします。各コミットの前に、アプライするトランザクションの数を指定します。</p>	Q アプライ・プログラムは、ターゲット表に從属する変更を含まないときにはいつでも、トランザクションの同時アプライが可能です。Q アプライ・プログラムは双方向および多方向レプリケーションを可能にするために、競合の処理と解決をさまざまな方法で行います。

#### 関連概念:

- 35 ページの『SQL レプリケーションと Q レプリケーションの比較 — 概要』

## 一般的なシナリオで使用すべきレプリケーション・ソリューション

表 4 では、SQL レプリケーションおよび Q レプリケーションで可能な最も一般的なシナリオについて説明し、それぞれのシナリオで使用するレプリケーション・テクノロジーを推奨しています。この推奨は一般的なもので、必ずしもすべてのニーズに適用されない場合もあります。

表 4. 一般的なレプリケーション・シナリオに対して推奨されるレプリケーション・ソリューション

構成	構成の説明	推奨されるレプリケーション・ソリューション
データ統合	多数のソースから中央リポジトリにデータを複製できます。データ統合構成は、データウェアハウスまたはオペレーショナル・データ・ストアのビルドでよく使用されます。	このシナリオでは SQL レプリケーションまたは Q レプリケーションのいずれかを問題なく使用できます。
データ配布	ソースから、分散ネットワークの任意の場所に存在する 1 つ以上のターゲットにデータを複製できます。アプリケーションは、ネットワークまたは中央サーバーに過負荷をかけないように、ローカル・ターゲット表を使用します。各サーバーは、データのサブセットへのアクセスのみが必要な場合、あるいはそのアクセスのみ許容されている場合があります。各サイトが必要なデータのみを参照するように、列および行のサブセットの複製を選択できます。	多数のターゲットが存在する場合、SQL レプリケーションのほうが適していると思われます。SQL レプリケーションはデータを一度ステージングし、そのデータを多数のターゲットに複製できるからです。  それ以外の場合は、このシナリオでは SQL レプリケーションまたは Q レプリケーションのいずれかを問題なく使用できます。
Update-anywhere レプリケーション	1 つのマスター・データ・ソースと 1 つ以上のソースのレプリカとの間でデータを複製することができます。マスター・ソースに対して加えられる変更はレプリカに伝搬され、レプリカに対して加えられる変更もまたマスター・ソースに伝搬されます。マスター・ソースから送信されるデータとレプリカから送信されるデータの間で競合が発生するときは、常にマスター・ソースからのデータが優先されます。	複数のレプリカを持つ標準的なハブ・スポーク構成では、SQL レプリケーションが必須です。そのような構成では、1 つのマスター・データ・ソースは複数のレプリカに複製されます。  マスターとレプリカがそれぞれ 1 つのシナリオでは、Q レプリケーションをお勧めします。
ホット・スタンバイ	実動サーバーからバックアップ・サーバーにデータを複製できます。実動サーバーがダウンすると、アプリケーションはバックアップ・サーバー上のデータの使用にただちに切り替わります。実動サーバーが再びオンラインになると、バックアップ・サーバーでコミットされた変更を実動サーバーに複製し、アプリケーションは再び実動サーバー上のデータを使用できるようになります。	Q レプリケーションをお勧めします。この構成では、アプリケーションのニーズに応じて、双方向レプリケーションか対等レプリケーションのいずれかを使用できます。



表 4. 一般的なレプリケーション・シナリオに対して推奨されるレプリケーション・ソリューション (続き)

構成	構成の説明	推奨されるレプリケーション・ソリューション
対等レプリケーション	<p>2 つ以上のサーバー上の表コピーの間でデータを複製できます。1 つの表コピーに対してコミットされた変更は、対応する他のすべての表コピーに複製されます。このタイプのレプリケーションは、地理的に分散したアプリケーションをサポートできます。</p> <p>たとえば、オンライン・ビジネスにおいて、3 つのサーバーが 3 つの大陸にわたって存在するとします。それぞれのサーバーには同じ製品情報とオーダー情報があります。通常、顧客は地元の大陸のサーバーの製品データにアクセスし、オーダー情報を入力します。しかし、1 つの大陸のサーバーがダウンすると、それらの顧客は別の大陸のサーバーに転送されます。その後、地元の企業と配送業者がそのオーダー情報にアクセスできるよう、その情報は顧客の大陸のサーバーに再び複製されます。</p>	<p>競合を検出して解決する機能が高いため、Q レプリケーションをお勧めします。イベント発行対等レプリケーションは、レプリケーション・センターを使って簡単に Q レプリケーションでセットアップできます。</p>
照会のオフロード	<p>OLTP サーバーが過負荷状態になるとき、別のサーバーへの照会ワークロードをオフロードすることにより、そのサーバー上の OLTP アプリケーションのパフォーマンスを向上させることができます。</p> <p>たとえば、ケーブル・テレビ会社に、毎日顧客のチケットを処理する多数の技術者がいるとします。そのチケットのトラッキングに使用される表は、1 日のうちの多数の更新、挿入、および削除を受け取ります。これらの表は、同時に他のビジネスの側面を処理する実動システム上に存在しています。会社は関係のある表を別のシステムに複製し、そのシステム上でレポート作成ツールを実行することができます。</p>	<p>このシナリオでは SQL レプリケーションまたは Q レプリケーションのいずれかを問題なく使用できます。</p>

**関連概念:**

- 35 ページの『SQL レプリケーションと Q レプリケーションの比較 — 概要』

## 第 5 章 イベント発行

### イベント発行 — 概要

以下のトピックでは、イベント発行に関する概念について説明しています。特定の詳細については、「*Replication and Transaction Publishing Guide and Reference*」を参照してください。

- イベント・パブリッシング環境のインフラストラクチャー
- イベント・パブリッシングのソース
- イベント・パブリッシングでのデータのキャプチャー

以下の図は、イベント発行の単純な構成を示しています。上記のトピックではそれぞれ、この図の各セクションについて詳しく取り上げて説明しています。

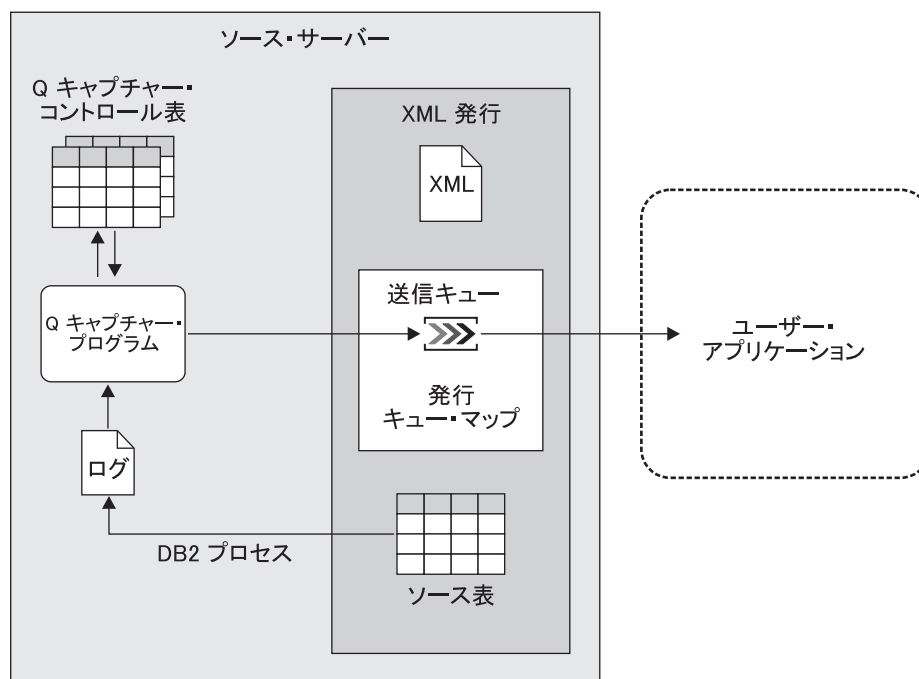


図 14. イベント発行の単純な構成

#### 関連概念:

- 46 ページの『イベント発行でのデータのキャプチャー』
- 42 ページの『イベント発行環境のインフラストラクチャー』
- 43 ページの『イベント発行のソース』
- 「*IBM DB2 Information Integrator レプリケーションとイベント・パブリッシングガイドおよびリファレンス*」の『イベント発行に必要な WebSphere MQ オブジェクト』

---

## イベント発行環境のインフラストラクチャー

イベント発行を使用すると、DB2<sup>®</sup> UDB 表からのコミット済みトランザクション・データまたは行レベルのデータを、メッセージとして WebSphere<sup>®</sup> MQ キューに発行することができます。メッセージは、ユーザー・アプリケーションによって直接読み取り、解釈することもできますし、WebSphere Business Integration Message Broker または DB2 MQ リスナー・デーモンなどのメッセージ・ブローカーによって最初に解釈することもできます。

イベント発行は、Q キャプチャー・プログラム と呼ばれるプログラムを使用することによりデータをキャプチャーします。これは Q レプリケーションで使用されるのと同じ Q キャプチャー・プログラムです。実際、1 つの Q キャプチャー・プログラムを Q レプリケーションとイベント発行で同時に使用することが可能です。Q キャプチャー・プログラムは、そのタスクの実行に必要な情報 (ソースは何か、そのソースから何を発行するか、に関する情報など)、および独自に生成する情報 (実行の状況に関する情報など) を保管するために、Q キャプチャー・コントロール表と呼ばれるコントロール表のセットを使用します。

同じ DB2 UDB サーバー上で複数の Q キャプチャー・プログラムを実行することができます。各 Q キャプチャー・プログラムは、独自のコントロール表セットを使用します。Q キャプチャー・コントロール表のセットと関連付けられているスキーマは、それらのコントロール表を使用する Q キャプチャー・プログラムを識別します。このスキーマは、Q キャプチャー・スキーマ と呼ばれます。

以下の図は、イベント発行の単純な構成のインフラストラクチャーを示しています。

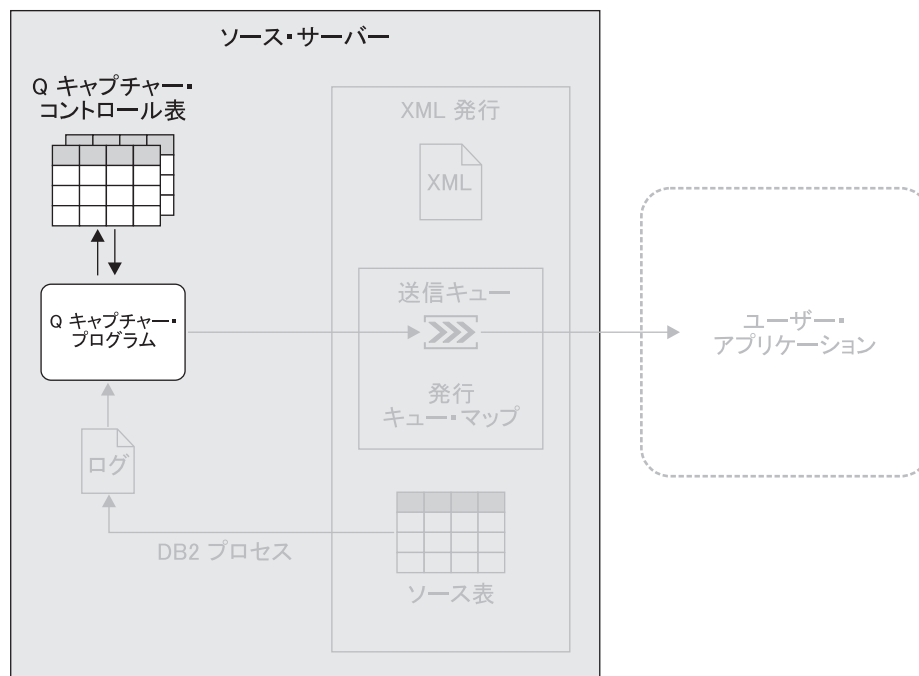


図 15. イベント発行の単純な構成のインフラストラクチャー：ソース・サーバー上に、Q キャプチャー・コントロール表と呼ばれる DB2 リレーショナル表のセットを作成します。ソースに関する情報はこれらの表に書き込まれます。ソース・サーバー上で実行される Q キャプチャー・プログラムは、この情報を使用して、キャプチャー・データ、および発行データを識別します。

#### 関連概念:

- 41 ページの『イベント発行 — 概要』

## イベント発行のソース

イベント発行を使用して XML 発行 と呼ばれるオブジェクトを作成し、単一のソース表からの変更を XML 形式で WebSphere® MQ メッセージ・キューに発行する方法を定義します。その後、アプリケーションを使ってそれらの XML メッセージを検索し、使用することができます。

XML 発行を作成する際、次の属性を指定します。

#### データの発行元のソース・サーバーとソース表

ソース・サーバーは、少なくとも 1 セットの Q キャプチャー・コントロール表を含む DB2® UDB サーバーです。ソース表には、ソース・サーバー上に存在する DB2 UDB リレーショナル表を使用できます。

#### 発行データを含む列

ソースから発行する列のサブセットを選択できます。

#### 行のサブセットを発行するための述部

ソースから発行する行のサブセットを選択できます。

#### 使用する発行キュー・マップ

Q キャプチャー・プログラムは、イベント発行では 送信キュー と呼ばれる WebSphere MQ オブジェクトにデータを書き込みます。送信キューは、

ユーザー・アプリケーションによるデータの読み取りが可能な別のキューにデータを渡します。イベント発行では、送信キューと、発行キュー・マップと呼ばれるオブジェクトにデータ・メッセージを送信するための他のオプションを手軽にグループ化することができます。それぞれの XML 発行について、使用発行キュー・マップを選択します。複数の XML 発行が同じ発行キュー・マップを使用できます。

選択された発行キュー・マップは、Q キャプチャー・プログラムが送信する各メッセージに、トランザクションで変更されるすべてのソース行についてのコミット済みデータを入れるか、それとも単一のソース行のみについてのコミット済みデータを入れるかを決定します。

たとえば、アプリケーションが購入オーダーを処理する場合には、トランザクションに関係するすべてのソース行についてのコミット済みデータを各メッセージに含めることとなります。この場合、請求、配送、および在庫トラッキング用のシステムにおいて、1 つのメッセージで処理を開始できるよう、製品タイプ、価格、配送先住所、請求情報などのデータの集合をまとめて発行する必要があります。

また、株価情報を提供する Web アプリケーションにストリーミング情報を提供する場合には、トランザクションに関係する 1 つのソース行についてのコミット済みデータを各メッセージに含めることとなります。この場合、株価の変更は他のどのデータとも関連しません。

#### 他の非キー列の変更とともに未変更の非キー列の値を発行するかどうか

デフォルトでは、選択した 1 つ以上の列が更新されると更新が発行されますが、未変更の列の値は変更されません。たとえば、表 T1 から列 A1 (主キー)、A2、および A3 を発行するとします。トランザクションが列 A2 の値を更新した後コミットする場合、Q キャプチャー・プログラムは、列 A2 の新しい値、およびキー列 A1 の値のみを含むメッセージを発行します。

非キー列の更新値および未変更値をメッセージに含めることを選択できます。列 A2 が更新される場合、キー列 A1 の値に加えて、Q キャプチャー・プログラムは列 A2 の新しい値、および列 A3 の未変更値を含むメッセージを発行します。

各列の値を常に予期するようにアプリケーションを作成する方が簡単な場合、このオプションを選択すると便利です。また、表に対する変更を監査するアプリケーションを作成し、そのアプリケーションが各行の完全なスナップショットを必要とする場合にもこのオプションを選択すると便利です。

#### 更新済み非キー列のデータに関する新しい値と古い値を含めるか、それとも新しい値のみを含めるか

選択した 1 つ以上の列に更新が発生すると、それらの更新を含むメッセージはデフォルトで新しい値のみを提供します。たとえば、表 T2 から列 B1 (主キー) および B2 を発行するとします。トランザクションが列 B2 を 5 から 6 に更新すると、その更新を含むメッセージは値 6 のみを提供します。

メッセージが新しい値とともに古い値も提供することを選択することもできます。この場合、列 B2 の更新を含むメッセージは値 5 と値 6 の両方を提供します。

発行された変更を受け取るアプリケーションが古い値と新しい値の両方を必要とする場合に、古い値の送信を選択します。たとえば、アプリケーションが情報を集約し、古い値と新しい値の相違を調べている場合があります。また、Web アプリケーションに対して重要な価格変更を発行し、そのアプリケーションが古い価格を必要とする場合、新しい価格とともに古い価格を送ることによってアプリケーション古い価格を探す手間が省けます。

**重要:** XML 発行は Q サブスクリプションの一部ではありません。Q サブスクリプションは、Q アプライ・プログラムへの情報の複製にのみ使用されます。XML 発行は、ユーザー・アプリケーションに XML 形式でデータを発行する場合にのみ使用されます。

以下の図は、イベント発行の単純な構成の XML 発行を示しています。

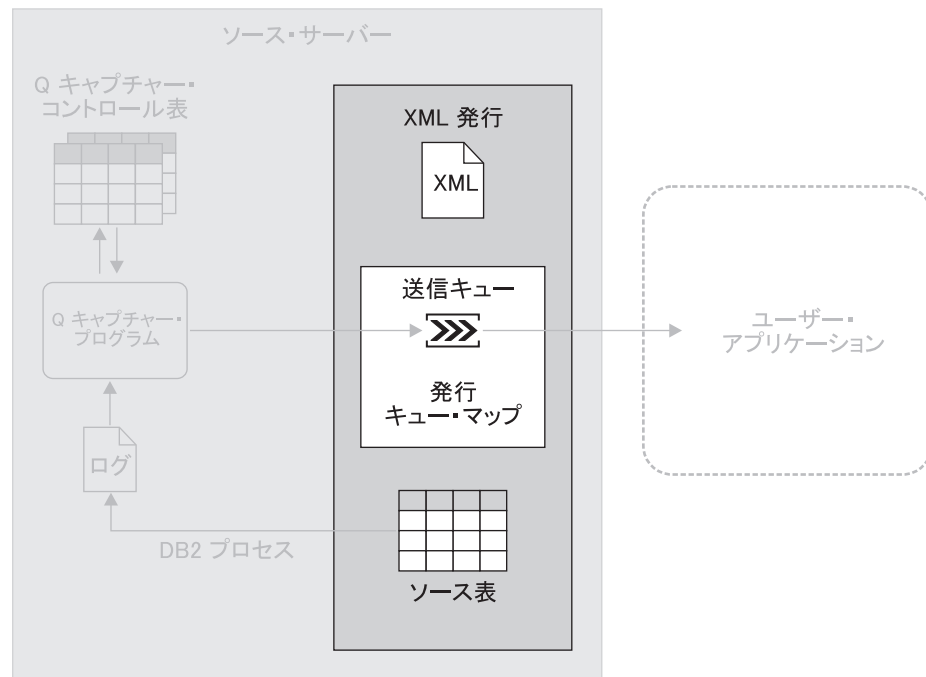


図 16. XML 発行：XML 発行は、Q キャプチャー・プログラムに、ソース表のどの変更をキャプチャーするかを知らせます。また、Q キャプチャー・プログラムに、XML メッセージのトランスポートに使用する送信キューも知らせます。

**関連概念:**

- 41 ページの『イベント発行 — 概要』

## イベント発行でのデータのキャプチャー

Q キャプチャー・プログラムは、DB2<sup>®</sup> ログを順次読み取り、ソース表に対して変更を加えます。Q キャプチャー・プログラムは、いずれかのソース表に対する変更を読み取ると、メモリー内に保持されている対応するデータベース・トランザクションに変更を追加します。そのため、メモリー内のトランザクションは潜在的に、ログ内の対応するトランザクションのサブセットになります。これには、ソース表に対する変更のみが入ります。Q キャプチャー・プログラムがトランザクションの COMMIT ステートメントを読み取ると、行レベルのメッセージを送信するかトランザクション・レベルのメッセージを送信するかに応じて、またトランザクションに関係するソース表によって使用される送信キューの数に応じて、トランザクションを 1 つ以上のメッセージに変換します。その後、Q キャプチャー・プログラムは、対応する送信キューにメッセージを挿入します。

たとえば、PUB1、PUB2、および PUB3 という 3 つの XML 発行を作成するとします。PUB1 と PUB2 は同じ発行キュー・マップ (SENDQ1 を使用する) を使用し、これらの XML 発行を使用して、コミット済みの行レベルの変更を含むメッセージを発行します。PUB3 については別の発行キュー・マップ (SENDQ2 を使用する) を選択し、この XML 発行を使用して、コミット済みのトランザクション・レベルの変更を含むメッセージを発行します。発行キュー・マップは行レベルのメッセージかトランザクション・レベルのメッセージのどちらかを送信できますが、両方のタイプのメッセージを送信することはできません。

Q キャプチャー・プログラムは、3 つすべての XML 発行のソース表への変更を含むデータベース・トランザクションの COMMIT を読み取ります。Q キャプチャー・プログラムは、PUB1 と PUB2 のソース表を含むトランザクションの一部を、SENDQ1 に書き込む行レベルのメッセージに変換します。また、Q キャプチャー・プログラムは、PUB3 のソース表を含むトランザクションの一部を、SENDQ2 に書き込む行レベルのメッセージに変換します。

同じソース・サーバー上で複数の Q キャプチャー・プログラムを実行できます。1 つの Q キャプチャー・プログラムでも多数のソースに対する変更をキャプチャーし、それらの変更を多数のユーザー・アプリケーションに送信することはできますが、複数の Q キャプチャー・プログラムを実行する方が都合のよい場合もあります。たとえば、複数の Q キャプチャー・プログラムを使用するとパフォーマンスが向上し、より高いスループットを達成できます。これは、大規模シスプレックスでも有効です。トレードオフは、ログ・リーダーが複数存在することに関連した CPU オーバーヘッドの増加です。複数の Q キャプチャー・プログラムを使用すると、DB2 UDB 接続もさらに必要になります。

以下の図は、イベント発行の単純な構成の、Q キャプチャー・プログラムによるデータのキャプチャーを示しています。



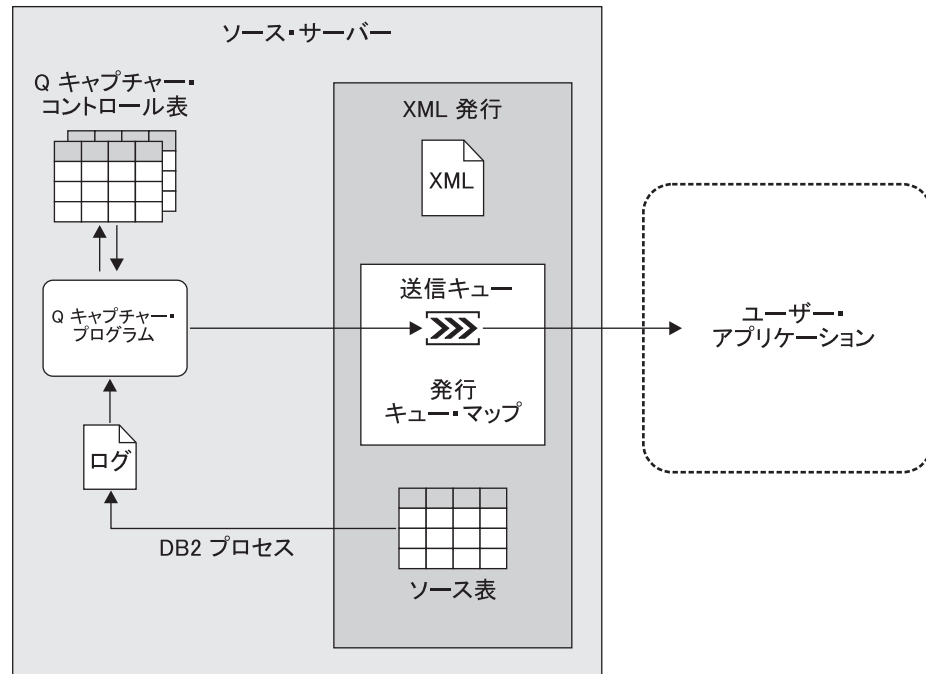


図 17. イベント発行でのデータのキャプチャー： Q キャプチャー・プログラムはソースに対して加えられる変更についての DB2 リカバリー・ログを読み取り、コミット済みトランザクション・データを XML メッセージに変換し、 Q キャプチャー・プログラムはそのメッセージを送信キューに挿入します。送信キューは、XML データを読み取ることでできる任意の数のアプリケーションにデータをトランスポートします。 Q キャプチャー・プログラムはそのコントロール表を、その進行のモニターに使用できる情報によって継続して更新します。

**関連概念:**

- 41 ページの『イベント発行 — 概要』



## 第 6 章 SQL レプリケーション、Q レプリケーション、およびイベント発行の比較

表 5 は、SQL レプリケーション、Q レプリケーション、およびイベント発行の主な機能を比較しています。

表 5. SQL レプリケーション、Q レプリケーション、およびイベント発行の主な機能の比較

比較のポイント	SQL レプリケーション	Q レプリケーション	イベント発行
用途	複数。たとえば、容量の解放、データウェアハウスおよびデータマートへのデータ提供、変更履歴の監査など。	複数。たとえば、フェイルオーバー、容量の解放、地理的に分散したアプリケーションのサポート、ローリング・アップグレードまたは計画停止に関するデータ使用可能性など。	複数。たとえば、中央情報ブローカーおよび Web アプリケーションへのデータ提供、およびソース表に対する更新、挿入、または削除に基づいたアクションの起動など。
データの複製または発行方法	コミットされたトランザクション・データは、キャプチャー・プログラムによってステージング表にキャプチャーされ、保管されます。アプライ・プログラムはステージング表から情報を読み取り、ターゲット表にトランザクションをアプライします。	コミットされたトランザクション・データは、キャプチャー・プログラムによってキャプチャーされ、WebSphere <sup>®</sup> MQ メッセージ・キューに挿入されます。アプライ・プログラムはメッセージ・キューから情報を読み取り、ターゲット表にトランザクションをアプライします。	コミットされたトランザクション・データは、キャプチャー・プログラムによってキャプチャーされ、WebSphere MQ メッセージ・キューに挿入されます。
ソース	(DB2 <sup>®</sup> UDB またはサポートされる他のリレーショナル・データベース管理システムの) 表、ビュー	DB2 UDB の表とストアド・プロシージャ	DB2 UDB の表とストアド・プロシージャ
ターゲット	(DB2 UDB またはサポートされる他のリレーショナル・データベース管理システムの) 表、ビュー	DB2 UDB の表とストアド・プロシージャ	ユーザー・アプリケーション
フィルタリングの可能性	可能。列および行のフィルタリング。	可能。列および行のフィルタリング。	可能。列および行のフィルタリング。
フィルタリングの実行場所	ソース、ターゲット、あるいはその両方。	ソース	ソース

表 5. SQL レプリケーション、Q レプリケーション、およびイベント発行の主な機能の比較 (続き)

比較のポイント	SQL レプリケーション	Q レプリケーション	イベント発行
データ・トランスフォーメーションの可能性	可能。ターゲット表でのデータ・クレンジング、データ集約、および計算済み列。一度にデータを操作し、その後多数のターゲットに複製することも、選択されたターゲット向けに操作したデータを、複製することも可能です。	ユーザーが開発したストアード・プロシージャを使用することによって可能。	可能。ユーザー・アプリケーションによって、必要とされるすべてのデータ・トランスフォーメーションが実行されます。
ターゲット表の自動ロードでサポートされるユーティリティー	LOAD ユーティリティーの EXPORT/IMPORT、EXPORT/LOAD、LOAD FROM CURSOR オプション。ターゲットのプラットフォームによって異なります。	LOAD ユーティリティーの EXPORT/IMPORT、EXPORT/LOAD、LOAD FROM CURSOR オプション。ターゲットのプラットフォームによって異なります。	ターゲット表がないため自動ロードはありません。
サポートされるオペレーティング・システム	Linux、UNIX <sup>®</sup> 、Windows <sup>®</sup> 、z/OS <sup>™</sup> 、iSeries <sup>™</sup>	Linux、UNIX、Windows、z/OS	Linux、UNIX、Windows、z/OS
サポートされるリレーショナル・データベース管理システム	DB2 UDB、Oracle、Sybase、Teradata、Microsoft <sup>®</sup> SQL Server	DB2 UDB	DB2 UDB
管理インターフェース	レプリケーション・センターまたはコマンド行	レプリケーション・センターまたはコマンド行	レプリケーション・センターまたはコマンド行

**関連概念:**

- 7 ページの『SQL レプリケーション — 概要』
- 23 ページの『Q レプリケーション — 概要』
- 41 ページの『イベント発行 — 概要』

## 第 7 章 Q レプリケーションと高可用性災害時リカバリー (HADR) の比較

DB2® 高可用性災害時リカバリー (HADR) は、サイト全体の障害からのリカバリーを支援し、サイトの部分的な障害に対して超高速フェイルオーバーを要求するアプリケーションをサポートする高可用性テクノロジーを提供します。HADR はローリング・アップグレードおよび他の計画停止にも使用できます。HADR は、(プライマリーと呼ばれる) データベースのソース・コピーから (スタンバイと呼ばれる) データベースのターゲット・コピーに、データベース全体に関するデータベース・ログ・レコードを発行します。スタンバイ・データベースはアプリケーションからのアクセスを受けません。スタンバイはリストアまたは分割ミラーを使用して初期化されます。これは別のシステムに添付でき、さまざまな用途のある、同一で独立したディスク・ボリューム・コピーです。

HADR を開始すると、スタンバイがメモリー内のプライマリーのログ・セットに追いつくまで、ログ・レコードを検索し、それらのレコードをスタンバイ上で再生します。その後、プライマリーはローカル・ディスクにログ・データを書き込み、それらをスタンバイに送信します。プライマリー上で生成され、スタンバイ・データベースに対して発行されるログ・データをロールフォワードすることによって、スタンバイ・データベースへの更新が発生します。

サイト全体の障害からデータを保護し、サイトの部分的な障害、およびローリング・アップグレードや他の計画停止の場合のアプリケーションのフェイルオーバーをサポートするために、Q レプリケーションを使用することもできます。表 6 は、HADR と、HADR と同じ目的で使用される場合の Q レプリケーションの違いを要約しています。

表 6. HADR と Q レプリケーションの比較

比較のポイント	高可用性災害時リカバリー	Q レプリケーション
セットアップ手順の有効範囲	DB2 UDB データベース全体。	DB2 UDB データベースまたはサブシステム内の表。
スタンバイへのデータの伝搬方法	ログ操作がスタンバイ・データベースに対して発行され、順方向リカバリーを介して継続的に再生されます。	DB2 リカバリー・ログからコミット済みトランザクション・データがキャプチャーされ、ターゲット表に適用されます。
同期レプリケーションの可能性	可能	不可
非同期レプリケーションの可能性	可能	可能
クライアント・アプリケーションのスタンバイへの自動転送	可能	可能
サポートされるオペレーティング・システム	Linux、UNIX®、および Windows®。	Linux、UNIX、Windows、z/OS™。
アプリケーションはスタンバイからデータを読み取ることができるか	不可	可能
アプリケーションはスタンバイにデータを書き込むことができるか	不可	可能

表 6. HADR と Q レプリケーションの比較 (続き)

比較のポイント	高可用性災害時リカバリー	Q レプリケーション
SQL データ操作言語 (DML) の他に SQL データ定義言語 (DDL) も自動的に複製されるか	複製される。	複製されない。
スタンバイを実行できるハードウェア、オペレーティング・システム、および DB2 UDB のバージョン	ハードウェア、オペレーティング・システム、および DB2 UDB のバージョンは、ソース・データベースが存在するものと同じでなければなりません。	ハードウェア、オペレーティング・システム、および DB2 UDB のバージョンは、ソース・データベースが存在するものと同じである必要はありません。
パフォーマンスをモニターするツールはあるか	ある。	ある。
ネットワーク圧縮と暗号化は組み込まれているか	組み込まれていない。	組み込まれている。
ソース・データベースおよびスタンバイ・データベースをパーティション化できるか	不可	ある。

HADR について詳しくは、「データ・リカバリーと高可用性 ガイドおよびリファレンス」を参照してください。

---

## 第 8 章 イベント発行と DB2 UDB MQ ユーザー定義関数の比較

イベント発行を使用してセットアップできるシナリオと類似したシナリオで、DB2® Universal Database に付属する WebSphere® MQ 関数をユーザー定義関数として使用することができます。これらの関数を使用すると、ユーザーは SQL を使用して DB2 UDB から WebSphere MQ メッセージ・キューにアクセスすることができます。DB2 UDB によってサポートされるプログラム言語によって作成されたアプリケーションも、さまざまなシナリオでこれらの関数を使用できます。たとえば、次のようなシナリオです。

### データ収集

アプリケーションは 1 つ以上のソースからメッセージ形式で情報を受け取ります。どのアプリケーションも情報源になれます。アプリケーションはキューからデータを受け取り、追加処理のためにデータベース表にデータを保管します。

### ワークロード分散

アプリケーションは、同じアプリケーションの複数のインスタンスによって共有されるキューに作業要求を送ります。アプリケーション・インスタンスの作業の実行準備が整うと、作業要求を含むメッセージをキューの先頭から受け取ります。アプリケーションの複数インスタンスは、プールされた要求の単一キューによって表されるワークロードを共有できます。

### アプリケーション・シグナル

複数のプロセスが共同作業するような状況の場合、メッセージを使用してその作業を調整することができます。これらのメッセージには作業を実行するためのコマンドまたは要求が含まれる場合があります。

イベント発行は、イベントを生成するアプリケーションを変更せずにそのイベントをメッセージとして発行できるという点で、DB2 WebSphere MQ 関数とは異なります。DB2 WebSphere MQ 関数を使用する場合、それら进行处理するためにアプリケーションをコーディングする必要があります。

DB2 WebSphere MQ 関数について詳しくは、「*IBM DB2 Information Integrator アプリケーション開発者向けガイド*」を参照してください。

### 関連概念:

- 「*IBM DB2 Information Integrator アプリケーション開発者向けガイド*」の『DB2 内で WebSphere MQ 機能を使用する方法』





---

## ユーザー補助

アクセス支援機能は、身体に障害のある（身体動作が制限されている、視力が弱いなど）ユーザーがソフトウェア製品を十分活用できるように支援します。DB2®バージョン 8 製品に備わっている主なアクセス支援機能は、以下のとおりです。

- すべての DB2 機能は、マウスの代わりにキーボードを使ってナビゲーションできます。詳細については、『キーボード入力およびナビゲーション』を参照してください。
- DB2 ユーザー・インターフェースのフォント・サイズおよび色をカスタマイズすることができます。詳細については、56 ページの『利用可能なモニター』を参照してください。
- DB2 製品は、Java™ Accessibility API を使用するアクセス支援アプリケーションをサポートします。詳細については、56 ページの『支援テクノロジーとの互換性』を参照してください。
- DB2 資料は、アクセスしやすい形式で提供されています。詳細については、56 ページの『利用しやすい資料』を参照してください。

---

## キーボード入力およびナビゲーション

### キーボード入力

キーボードだけを使用して DB2 ツールを操作できます。マウスを使って実行できる操作は、キーまたはキーの組み合わせによっても実行できます。標準のオペレーティング・システム・キー・ストロークを使用して、標準のオペレーティング・システム操作を実行できます。

キーまたはキーの組み合わせによって操作を実行する方法の詳細については、キーボード・ショートカットおよびアクセラレーター: Common GUI helpを参照してください。

### キーボード・ナビゲーション

キーまたはキーの組み合わせを使用して、DB2 ツールのユーザー・インターフェースをナビゲートできます。

キーまたはキーの組み合わせによって DB2 ツールをナビゲートする方法の詳細については、キーボード・ショートカットおよびアクセラレーター: Common GUI helpを参照してください。

### キーボード・フォーカス

UNIX® オペレーティング・システムでは、アクティブ・ウィンドウの中で、キー・ストロークによって操作できる領域が強調表示されます。

---

## 利用可能なモニター

DB2 ツールには、視力の弱いユーザー、その他の視力障害をもつユーザーのためにアクセシビリティを向上させる機能が備わっています。これらのアクセシビリティ拡張機能には、フォント・プロパティのカスタマイズを可能にする機能も含まれています。

### フォントの設定

「ツール設定」ノートブックを使用して、メニューおよびダイアログ・ウィンドウに使用されるテキストの色、サイズ、およびフォントを選択できます。

フォント設定に関する詳細情報は、メニューおよびテキストのフォントを変更する: Common GUI helpを参照してください。

### 色に依存しない

本製品のすべての機能を使用するために、ユーザーは必ずしも色を識別する必要はありません。

---

## 支援テクノロジーとの互換性

DB2 ツールのインターフェースは、Java Accessibility API をサポートします。これによって、スクリーン・リーダーその他の支援テクノロジーを DB2 製品で利用できるようになります。

---

## 利用しやすい資料

DB2 形式は、ほとんどの Web ブラウザーで表示可能な XHTML 1.0 形式で提供されています。XHTML により、ご使用のブラウザーに設定されている表示設定に従って資料を表示できます。さらに、スクリーン・リーダーや他の支援テクノロジーを使用することもできます。

シンタックス・ダイアグラムはドット 10 進形式で提供されます。この形式は、スクリーン・リーダーを使用してオンライン・ドキュメンテーションにアクセスする場合にのみ使用できます。

---

## 特記事項

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

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

〒106-0032  
東京都港区六本木 3-2-31  
IBM World Trade Asia Corporation  
Licensing

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

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

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

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

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

IBM Corporation  
J46A/G4  
555 Bailey Avenue  
San Jose, CA 95141-1003  
U.S.A.

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができませんが、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をお願いします。

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した創作物には、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

以下は、IBM Corporation の商標です。

IBM  
DB2  
Informix  
iSeries  
UNIX  
WebSphere  
Windows  
z/OS

以下は、それぞれ各社の商標または登録商標です。

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。

Java は、Sun Inc. の米国およびその他の国における商標です。

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。





# 索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

## [ア行]

- アプライ修飾子
  - 定義 8
- アプライ・コントロール表
  - 定義 8
  - ロケーション 35
- アプライ・プログラム
  - 定義 8
  - ロケーション 37
- イベント発行
  - インフラストラクチャー 42
  - 紹介 3
  - ソース 43
  - データのキャプチャー 46
  - DB2 UDB MQ ユーザー定義関数との比較 53
- イベント・タイミング 12
- インターバル・タイミング 12
- インフラストラクチャー
  - イベント発行 42
  - Q レプリケーション 24
  - SQL レプリケーション 8

## [カ行]

- キャプチャー・コントロール表
  - 定義 8
  - ロケーション 35
- キャプチャー・スキーマ
  - 定義 8
- キャプチャー・トリガー 20
- キャプチャー・プログラム
  - 定義 8
  - ロケーション 37
- 高可用性災害時リカバリー (HADR)
  - Q レプリケーションとの比較 51

## [サ行]

- サブスクリプション・セット
  - 定義 12
- ソースとターゲットの対
  - Q レプリケーション 25
  - SQL レプリケーション 12

- ソース表
  - イベント発行 43
  - Q レプリケーション 27
  - SQL レプリケーション 9
  - 双方向レプリケーション 30

## [タ行]

- ターゲット表
  - Q レプリケーション 25
  - SQL レプリケーション 12
- ターゲット表のロード
  - Q レプリケーション 25
  - SQL レプリケーション 19
- 対等レプリケーション 30
- 単一方向レプリケーション 30
- データのアプライ
  - Q レプリケーション 29
  - SQL レプリケーション
    - 非 DB2 ターゲット 21
    - DB2 UDB ターゲット 19
- データのキャプチャー
  - イベント発行 46
  - Q レプリケーション 27
  - SQL レプリケーション
    - 非 DB2 ソース 20
    - DB2 UDB ソース 17
- トリガー
  - キャプチャー 20

## [ハ行]

- 発行キュー・マップ 43
- フル・リフレッシュ・レプリケーション 9
- 変更キャプチャー・レプリケーション 9
- ポイント・イン・タイム・ターゲット 12

## [マ行]

- マッピング
  - ソースからターゲット
    - Q レプリケーション 25
    - SQL レプリケーション 12

## [ヤ行]

- ユーザー・コピー・ターゲット 12

## [ラ行]

- レプリカ・ターゲット 12
- レプリケーション
  - 双方向 30
  - 対等 30
  - 単一方向 30
  - フル・リフレッシュ 9
  - 変更キャプチャー 9
- レプリケーションと発行のソリューション比較 49
- レプリケーションと発行のソリューションの比較 49
- レプリケーション・アラート・モニター 5
- レプリケーション・キュー・マップ 25
- レプリケーション・センター 4

## A

- ASNCLP プログラム 4

## C

- CCD ターゲット 12
- CCD 表 20, 21
- CD 表 9, 19, 20

## D

- DB2 UDB MQ ユーザー定義関数 53

## Q

- Q アプライ・コントロール表
  - 定義 24
  - ロケーション 35
- Q アプライ・スキーマ
  - 定義 24
- Q アプライ・プログラム
  - 定義 24
  - ロケーション 37
- Q キャプチャー・コントロール表
  - 定義
    - イベント発行 42
    - Q レプリケーション 24
  - ロケーション
    - イベント発行 42
    - Q レプリケーション 35

- Q キャプチャー・スキーマ
  - 定義
  - イベント発行 42
  - Q レプリケーション 24
- Q キャプチャー・プログラム
  - 定義
  - イベント発行 42
  - Q レプリケーション 24
  - ロケーション
    - イベント発行 42
    - Q レプリケーション 37
- Q サブスクリプション
  - 定義 25
- Q レプリケーション
  - インフラストラクチャー 24
  - 紹介 2
  - ソースとターゲットの対 25
  - データのアプリ 29
  - データのキャプチャー 27

## S

- SQL レプリケーション
  - インフラストラクチャー 8
  - 紹介 1
  - ソースとターゲットの対 12
  - ソースの登録 9
  - データのアプリ
    - 非 DB2 ターゲット 21
    - DB2 UDB ターゲット 19
  - データのキャプチャー
    - 非 DB2 ソース 20
    - DB2 UDB ソース 17
  - フル・リフレッシュ・レプリケーション 9
  - 変更キャプチャー・レプリケーション 9
- SQL レプリケーションでのソースの登録 9
- SQL レプリケーションと Q レプリケーション
  - 比較
    - インフラストラクチャー 35
    - ソースとターゲット 36
    - データのアプリ 37
    - データのキャプチャー 37

## T

- tdiff 5
- trepair 5

## X

- XML 発行 43

---

## IBM と連絡を取る

技術上の問題がある場合は、お客様サポートにご連絡ください。

---

### 製品情報

DB2 Information Integrator についての情報は、電話または Web から入手することができます。

米国にお住まいの場合は、以下のいずれかの番号にお問い合わせください。

- 製品の注文または一般情報の入手: 1-800-IBM-CALL (1-800-426-2255)
- 資料の注文: 1-800-879-2755

Web 上で [www.ibm.com/software/data/integration/db2ii/support.html](http://www.ibm.com/software/data/integration/db2ii/support.html) にアクセスします。このサイトには、以下の最新情報が記載されています。

- テクニカル・ライブラリー
- 資料の注文
- クライアント・ダウンロード
- ニュースグループ
- フィックスパック
- ニュース
- Web リソースへのリンク

---

### 資料についてのコメント

お客様のフィードバックは IBM が良質な情報を提供する助けになります。この資料や他の DB2 Information Integrator の資料についてのコメントをお送りください。コメントの送付には、以下のいずれかの方法を利用することができます。

- [www.ibm.com/software/data/rcf](http://www.ibm.com/software/data/rcf) で、オンラインの読者コメント・フォームを使用して送信する。
- E メールで [comments@us.ibm.com](mailto:comments@us.ibm.com) にコメントを送信する。お送りいただく情報には、製品の名前、製品のバージョン番号、および資料の名前と部品番号 (該当する場合) を含めてください。特定の本文についてコメントする場合は、本文の位置 (たとえば、タイトル、表の番号、またはページ番号) を含めてください。







Printed in Japan

GC88-9895-00



日本アイ・ビー・エム株式会社  
〒106-8711 東京都港区六本木3-2-12