

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

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 ガイドおよびリファレンス

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、553 ページの『特記事項』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： SC18-7568-00
IBM DB2 Information Integrator
Replication and Event Publishing
Guide and Reference
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	xi
本書の対象読者	xi
本書の表記規則	xi
本書で使用される用語	xii
本書の追加情報の入手先	xii
構文図の見方	xiii

第 1 部 Q レプリケーションおよびイベント発行の紹介 1

第 1 章 Q レプリケーションの紹介 3

Q レプリケーションの紹介 -- 概要	3
Q レプリケーション	3
単方向レプリケーション	4
双方向レプリケーション	4
ピアツーピア・レプリケーション	4
Q サブスクリプション	5
レプリケーション・キュー・マップ	6

第 2 章 イベント発行の紹介 9

イベント発行の紹介 -- 概要	9
イベント発行	9
イベント発行のオブジェクト	10
メッセージ内容の制御	11
XML 発行	12
発行キュー・マップ	13

第 3 章 複製および発行で使用するプログラムの紹介 15

複製および発行で使用するプログラムの紹介 -- 概要	15
Q キャプチャー・プログラム	15
変更のキャプチャー	16
データのフィルタリング	16
メッセージ・フォーマット	17
メッセージのコミット	17
パラメーターの変更	17
Q キャプチャーとの通信	17
再始動メッセージ	18
Q アプライ・プログラム	19
ブラウザー・スレッドとエージェント・スレッド	19
トランザクションの並列でのアプライ	20
パラメーターの変更	20
Q アプライ・プログラムと Q キャプチャー・プログラムのスキーマ	21

第 2 部 Q レプリケーションおよびイベント発行の準備 23

第 4 章 メモリーおよびストレージの要件の計画 25

Q レプリケーションおよびイベント発行のメモリーおよびストレージ要件の計画	25
Q レプリケーションおよびイベント発行のメモリー要件の計画	25
Q レプリケーションおよびイベント発行のメモリー要件の計画	25
Q キャプチャー・プログラムによって使用されるメモリー	26
Q アプライ・プログラムによって使用されるメモリー	29
レプリケーション・アラート・モニターによって使用されるメモリー	32
Q レプリケーションおよびイベント発行の LOB データ・タイプのためのメモリー	32
Q レプリケーションおよびイベント発行のストレージ要件の計画	32
Q レプリケーションおよびイベント発行のストレージ要件の計画	32
Q レプリケーションおよびイベント発行の DB2 UDB ログのストレージ要件	33
Q レプリケーションおよびイベント発行の診断ファイルのストレージ	34
Q キャプチャー・プログラムがメモリー限度を超える場合のストレージ要件	34
Q レプリケーションおよびイベント発行のトレースのストレージ要件	35

第 5 章 データ変換 37

Q レプリケーションおよびイベント発行のデータ変換 -- 概要	37
Q レプリケーションのデータ変換	37
イベント発行のデータ変換	38

第 6 章 ユーザー ID およびパスワードのセットアップ 39

ユーザー ID およびパスワードのセットアップ -- 概要	39
Q レプリケーションおよびイベント発行の許可要件	39
Q レプリケーションおよびイベント発行の許可要件 -- 概要	39
Q キャプチャー・プログラムの許可要件	39
Q アプライ・プログラムの許可要件	40
レプリケーション・アラート・モニターの許可要件	42
Q レプリケーションおよびイベント発行を管理するための許可要件	42
リモート・サーバー用のユーザー ID およびパスワードの保管	43

第 7 章 Q レプリケーションおよびイベント発行用の WebSphere MQ をセットアップする方法 47

Q レプリケーションおよびイベント発行用の WebSphere MQ のセットアップ -- 概要	47
Q レプリケーションおよびイベント発行に必要な WebSphere MQ オブジェクト	47
Q レプリケーションおよびイベント発行に必要な WebSphere MQ オブジェクト -- 概要	47
必須 WebSphere MQ オブジェクトの概観	48
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	50
単一方向レプリケーション (リモート) に必要な WebSphere MQ オブジェクト	52
イベント発行に必要な WebSphere MQ オブジェクト	53
双方向またはピアツーピア・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)	55
ピアツーピア・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	58
WebSphere MQ オブジェクトの設定要件	62
ソースでの WebSphere MQ オブジェクト	62
ターゲットでの WebSphere MQ オブジェクト	63
WebSphere MQ チャンネル・オブジェクト	65
WebSphere MQ メッセージ・サイズ	68
Q レプリケーションおよびイベント発行の WebSphere MQ のストレージ要件	69
Q アプライ・プログラムの予備キューのストレージ要件	70
WebSphere MQ オブジェクトの接続および許可に関する要件	70
ラージ・オブジェクト (LOB) 値のキュー項目数に関する考慮事項	72
Q レプリケーションおよびイベント発行でのキュー・マネージャーのクラスタリング	73

第 8 章 Q レプリケーションおよびイベント発行用のサーバーの構成 75

Q レプリケーションおよびイベント発行用のサーバーの構成 -- 概要	75
Q レプリケーションおよびイベント発行の接続要件	75
Q レプリケーションおよびイベント発行用のデータベースの構成 (Linux、UNIX、Windows)	76
Q レプリケーションおよびイベント発行用のデータベースの構成 (Linux、UNIX、Windows)	76
環境変数の設定 (Linux、UNIX、Windows)	77
Q キャプチャー・プログラムが動作するソース・データベースの構成 (Linux、UNIX、Windows)	78
Q アプライ・プログラムが動作するターゲット・データベースの構成 (Linux、UNIX、Windows)	80
オプション : プログラム・パッケージのバインド (Linux、UNIX、Windows)	81

Q レプリケーションおよびイベント発行用のデータベースの構成 (z/OS)	85
レプリケーション・センターのソフトウェア前提条件	86
Q キャプチャー・プログラムと Q アプライ・プログラムのコントロール表の作成	87

第 3 部 Q レプリケーションおよびイベント発行のセットアップ 89

第 9 章 ソースからターゲットへの (単一方向の) レプリケーションのセットアップ 91

ソースからターゲットへの (単一方向) レプリケーションのセットアップ -- 概要	91
単一方向レプリケーション	91
レプリケーション・キュー・マップと Q サブスクリプションのグループ化	93
Q キャプチャー・プログラムによる送信キューの処理方法	94
Q アプライ・プログラムによる受信キューの処理方法	94
類似した Q サブスクリプションのレプリケーション・キュー・マップによるグループ化の提案	94
レプリケーション・キュー・マップの作成	95
単一方向レプリケーションの Q サブスクリプションの作成	98
単一方向レプリケーションの Q サブスクリプションの作成 -- 概要	98
Q レプリケーションのターゲット・オブジェクト・プロファイル	99
単一方向レプリケーションの単一の Q サブスクリプションの作成	100
単一方向レプリケーションの複数の Q サブスクリプションの作成	103
Q サブスクリプションのソース列 (単一方向レプリケーション)	105
Q キャプチャー・プログラムがメッセージを送信する頻度 (単一方向レプリケーション)	106
行のフィルタリングのための検索条件 (単一方向レプリケーション)	108
ソース列からターゲット列にマップする方法 (単一方向レプリケーション)	111
ターゲットの索引列またはキー列 (単一方向レプリケーション)	112
ターゲット表での予期しない条件のオプション (単一方向レプリケーション)	114
Q レプリケーションのエラー・オプション	118
ターゲットに適用されるデータを操作するストアード・プロシージャの使用	121
Q レプリケーションのターゲットに適用されるデータを操作するストアード・プロシージャの使用 -- 概要	121
Q レプリケーションのソース・データを操作するストアード・プロシージャ	122

Q レプリケーションのソース・データを操作する ストアード・プロシージャの作成	124
Q レプリケーションの操作タイプを識別する ストアード・プロシージャ・パラメーター	126
各ソース列が抑制されたかどうかを識別する ストアード・プロシージャ・パラメーター	128
Q レプリケーションのトランザクションを識別 するストアード・プロシージャ・パラメーター	129
Q レプリケーションのソース列にマップする ストアード・プロシージャ・パラメーター	129

第 10 章 ソースからターゲットへのレプリケーションのセットアップ (複数方向) 133

ソースからターゲットへのレプリケーションのセッ トアップ (複数方向) -- 概要	133
双方向レプリケーション	133
双方向レプリケーションでのレプリケーション・ オブジェクト	134
双方向レプリケーションでの競合の検出	135
ピアツーピア・レプリケーション	136
2 つのサーバーが関係するピアツーピア・レプリ ケーション用のレプリケーション・オブジェクト	137
3 つのサーバーが関係するピアツーピア・レプリ ケーション用のレプリケーション・オブジェクト	139
競合解決および参照保全	140
双方向レプリケーションとピアツーピア・レプリケ ーション	141
双方向レプリケーションの機能に最適のシナリオ ピアツーピア・レプリケーションの機能に最適な シナリオ	142
高可用性シナリオ用の双方向レプリケーションと ピアツーピア・レプリケーション	142
双方向レプリケーションの Q サブスクリプション の作成	144
双方向レプリケーションの Q サブスクリプショ ンの作成	144
競合検出のオプション (双方向レプリケーショ ン)	147
ピアツーピア・レプリケーションの Q サブスクリ プションの作成	153
ピアツーピア・レプリケーションの Q サブスク リプションの作成 -- 概要	153
2 つのサーバーによるピアツーピア・レプリケー ションの Q サブスクリプションの作成	153
3 つ以上のサーバーによるピアツーピア・レプリ ケーションの Q サブスクリプションの作成	155
2 つのサーバーを使用した双方向およびピアツーピ ア・レプリケーションの開始	158
2 つのサーバーを使用した双方向およびピアツーピ ア・レプリケーションの停止	159
3 つ以上のサーバーを使用したピアツーピア・グル ープでのレプリケーションの開始	160
3 つ以上のサーバーを使用したピアツーピア・グル ープでのレプリケーションの停止	161

第 11 章 Q レプリケーションにターゲ ット表をロードするためのオプション . . . 163

Q レプリケーションのターゲット表のロードのオプ ション -- 概要	163
Q レプリケーションのターゲット表をロードする際 の推奨事項	164
Q レプリケーションの自動ロード・オプション	164
Q レプリケーションの自動ロード・オプション -- 概要	164
Q レプリケーションの自動ロード・オプション で使用するユーティリティ	165
z/OS プラットフォーム用の自動ロードに関する 考慮事項	166
Q レプリケーションの自動ロード・オプション でのニックネームの指定	167
Q レプリケーションの手動ロード・オプション	169
Q レプリケーションのロードなしオプション	172
さまざまなタイプの Q レプリケーションのロー ド・オプション	173

第 12 章 ソースからの発行 (イベント発 行) のセットアップ 179

ソースからの発行 (イベント発行) の設定 -- 概要	179
発行キュー・マップおよび XML 発行のグループ化	179
Q キャプチャー・プログラムによる送信キュー の処理方法	180
発行キュー・マップを使用した類似の XML 発 行のグループ化の提案	180
発行キュー・マップの作成	180
XML 発行の作成	182
XML 発行の作成 -- 概要	182
単一 XML 発行の作成	183
複数の XML 発行の作成	185
XML 発行のソース列	188
Q キャプチャー・プログラムが XML 発行用の メッセージを発行する時期	188
XML 発行の行をフィルター処理する検索条件	189
XML 発行のキー列	192
XML 発行用のメッセージに未変更の列を組み込 むためのオプション	193
XML 発行用のメッセージに変更前の値を組み込 むためのオプション	194

第 13 章 データ・タイプの複製および 発行に関する考慮事項 197

Q レプリケーションおよびイベント発行でのデー タ・タイプの複製および発行に関する考慮事項 -- 概要	197
Q レプリケーションおよびイベント発行での一般 的なデータ・タイプに関する考慮事項	197
Q レプリケーションおよびイベント発行でのラ ージ・オブジェクト (LOB) データ・タイプに関する 考慮事項	198

第 14 章 レプリケーション・センターからの SQL スクリプトおよび操作可能コマンドの実行	199
レプリケーション・センターからの SQL スクリプトおよび操作可能コマンドの実行 -- 概要	199
レプリケーション・センターが生成する SQL スクリプトおよび操作可能コマンド	199
レプリケーション・センターからの SQL スクリプトの実行および保管	200
レプリケーション・センターからのコマンドの実行および保管	201

第 4 部 Q レプリケーションおよびイベント発行環境の管理およびモニター **203**

第 15 章 Q レプリケーション環境の変更	205
Q レプリケーション環境の変更 -- 概要	205
Q サブスクリプションの属性の変更	205
既存の Q サブスクリプションへの列の追加	207
Q サブスクリプションでの許容可能な SQL 状態の変更	209
レプリケーション・キュー・マップの属性の変更	211
Q サブスクリプションの削除	213
レプリケーション・キュー・マップの削除	215
Q キャプチャー・スキーマまたは Q アプライ・スキーマのドロップ	216

第 16 章 イベント発行環境の変更	219
イベント発行環境の変更 -- 概要	219
XML 発行の属性の変更	219
既存の XML 発行への列の追加	221
発行キュー・マップの属性の変更	223
XML 発行の削除	225
発行キュー・マップの削除	226
Q キャプチャー・スキーマのドロップ	227

第 17 章 Q キャプチャー・プログラムの操作	229
Q キャプチャー・プログラムの操作 -- 概要	229
Q キャプチャー・プログラムの開始	229
Q キャプチャー・プログラムのコールド・スタートに関する考慮事項	232
Q キャプチャー・プログラムのパラメーター	233
Q キャプチャー・プログラムのパラメーター -- 概要	233
Q キャプチャー稼働パラメーターのデフォルト値	234
Q キャプチャー・パラメーターの説明	235
Q キャプチャー・パラメーターの変更	248
Q キャプチャー・パラメーターの変更 -- 概要	248
Q キャプチャー稼働パラメーターの変更方法	249

IBMQREP_CAPPARMS 表に保管された Q キャプチャー・パラメーターの変更	250
Q キャプチャー・プログラムの実行中に行うパラメーターの動的変更	251
Q サブスクリプションまたは XML 発行のアクティブ化	253
Q サブスクリプションまたは XML 発行の非アクティブ化	254
Q キャプチャー・プログラムの停止	256

第 18 章 Q アプライ・プログラムの操作 **259**

Q アプライ・プログラムの操作 -- 概要	259
Q アプライ・プログラムの開始	259
Q アプライ・プログラムのパラメーター	262
Q アプライ・プログラムのパラメーター -- 概要	262
Q アプライ稼働パラメーターのデフォルト値	262
Q アプライ・パラメーターの説明	263
Q アプライ・パラメーターの変更	271
Q アプライ・パラメーターの変更 -- 概要	271
IBMQREP_APPLYPARMS 表に保管された Q アプライ・パラメーターの変更	271
Q アプライ・プログラムの実行中のパラメーターの動的変更	272
受信キュー上のメッセージ処理の開始	273
受信キュー上のメッセージ処理の停止	275
Q アプライ・プログラムの停止	276

第 19 章 Q レプリケーションおよびイベント発行プログラムに関するレポートの表示 **279**

Q レプリケーションおよびイベント発行プログラムに関するレポートの表示 -- 概要	279
Q レプリケーションおよびイベント発行プログラムの状況のチェック	279
Q キャプチャー・プログラム、Q アプライ・プログラム、およびレプリケーション・アラート・モニター・プログラムのスレッド	281
Q レプリケーションおよびイベント発行プログラムの履歴およびパフォーマンス・データ	283
待ち時間	285
例外	288

第 20 章 レプリケーション・アラート・モニターによるレプリケーションのモニター **291**

レプリケーション・アラート・モニターによるレプリケーションのモニター -- 概要	291
レプリケーション・アラート・モニター	291
レプリケーション・アラート・モニターのアラート条件および通知	294
レプリケーション・アラート・モニターのアラート条件および通知 -- 概要	294
レプリケーション・アラート・モニターのアラート条件	294

レプリケーション・アラート条件の E メール通知	298
レプリケーション中のアラートを送信する	
ASNMAIL exit ルーチン	299
レプリケーション・アラート・モニターのセットアップ	300
レプリケーション・アラート・モニターのセットアップ	300
レプリケーション・アラート・モニターのコントロール表の作成	301
レプリケーション・アラート・モニターの連絡先情報の定義	302
レプリケーション・アラート・モニターの操作	303
レプリケーション・アラート・モニターの操作	303
レプリケーションまたは発行用のモニターの作成	304
レプリケーション・アラート・モニターのアラート条件の選択	305
モニターの開始	306
モニターの再初期化	307
レプリケーション・アラート・モニターの操作に使用するパラメーターのデフォルト値	307
レプリケーション・アラート・モニターの操作に使用するパラメーターの説明	308
レプリケーション・アラート・モニターのパラメーターの設定	311
モニターの停止	314

第 21 章 Q レプリケーションおよび発行環境の保守 315

Q レプリケーションおよびイベント発行環境の保守 -- 概要	315
Q レプリケーションおよびイベント発行のソース・システムの保守に関する考慮事項	315
Q レプリケーションおよびイベント発行のソース・システムの保守に関する考慮事項 -- 概要	315
Q レプリケーションおよびイベント発行環境でのソース表の保守	316
Q レプリケーションおよびイベント発行のログ・ファイルの保存	316
Q レプリケーションおよびイベント発行環境でのコンプレッション・ディクショナリーの管理に関する考慮事項 (z/OS)	321
Q レプリケーションおよびイベント発行でのコントロール表の保守	322
Q レプリケーションおよびイベント発行でのコントロール表の保守 -- 概要	322
Q レプリケーションおよびイベント発行でのコントロール表の整理	323
Q レプリケーションおよびイベント発行におけるコントロール表での RUNSTATS ユーティリティの使用に関する考慮事項	324
Q レプリケーションおよびイベント発行コントロール表の再編成	324
レプリケーション・プログラムが DB2 UDB サーバーに接続できない場合	325

Q レプリケーションおよびイベント発行でのターゲット表の保守	325
Q レプリケーションおよびイベント発行のパッケージおよびプランの再バインドに関する考慮事項	326

第 22 章 ソース表とターゲット表間の相違検出および修復 329

ソース表とターゲット表間の相違検出および修復	329
Tdiff: 表相違検出ユーティリティ	329
Trepair: 表修復ユーティリティ	332

第 23 章 システム・サービスを使用してレプリケーション・プログラムを操作する方法 335

システム・サービスを使用して Q レプリケーションおよびイベント発行プログラムを操作する方法 -- 概要	335
システム・サービスを使用したレプリケーション・プログラムの操作 (z/OS)	335
システム・サービスを使用した Q レプリケーションおよびイベント発行プログラムの操作 (z/OS) -- 概要	335
JCL を使用したレプリケーション・プログラムの実行	336
システム開始タスクを使用した Q レプリケーションおよびイベント発行プログラムの実行	339
自動リスタート・マネージャー (ARM) のリカバリー・システム	340
Q レプリケーションおよびイベント発行プログラムを再始動するために自動リスタート・マネージャー (ARM) をセットアップする方法	341
レプリケーション・サービスの管理によるレプリケーション・プログラムの操作 (Windows)	342
Windows Service Control Manager (SCM) を使用してレプリケーション・サービスを管理する方法 -- 概要	342
Q レプリケーションおよびイベント発行のレプリケーション・サービス (Windows)	342
レプリケーション・サービスの作成	344
レプリケーション・サービスの開始	345
レプリケーション・サービスの停止	345
レプリケーション・サービスのリストの表示	346
レプリケーション・サービスのドロップ	346
レプリケーション・プログラムのスケジューリング	347
レプリケーション・プログラムのスケジューリング -- 概要	347
レプリケーション・プログラムのスケジューリング (Linux、UNIX)	347
レプリケーション・プログラムのスケジューリング (Windows)	348
レプリケーション・プログラムのスケジューリング (z/OS)	349

第 5 部 Q レプリケーションおよびイベント発行の参照情報 351

第 24 章 命名規則およびガイドライン 353

Q レプリケーションおよびイベント発行の命名規則とガイドライン -- 概要	353
Q レプリケーションおよびイベント発行オブジェクトの命名規則	353
Q レプリケーションおよびイベント発行で小文字のオブジェクト名が処理される方法.	356

第 25 章 Q レプリケーションおよびイベント発行用のコマンド 359

Q レプリケーションおよびイベント発行用のシステム・コマンド -- 概要.	359
ロードマップ: Q レプリケーションおよびイベント発行のシステム・コマンド.	359
asnqcap: Q キャプチャー・プログラムの開始.	361
asnqccmd: 実行中の Q キャプチャー・プログラムの操作.	368
asnqapp: Q アプライ・プログラムの開始	372
asnqacmd: 実行中の Q アプライ・プログラムの操作	376
asnmon: レプリケーション・アラート・モニターの始動	379
asnmcmd: 実行中のレプリケーション・アラート・モニターの操作.	384
asnqanalyze: Q レプリケーション・アナライザーの操作	386
ansrcrt: レプリケーション・プログラムを開始する DB2 レプリケーション・サービスの作成	390
asnslst: DB2 レプリケーション・サービスのリスト	394
ansndrop: DB2 レプリケーション・サービスのドロップ	395
asnpwd: パスワード・ファイルの作成および保守	396
asntrc: レプリケーション・トレース機能の操作	400
asntdiff: ソース表とターゲット表とのデータの比較	408
asntrep: ソース表とターゲット表の間の違いの修復	410
asnqmfmt: Q レプリケーションとイベント発行のメッセージのフォーマットおよび表示.	412

第 26 章 Q レプリケーションおよびイベント発行のコントロール表 415

Q レプリケーションおよびイベント発行のコントロール表 -- 概要	415
コントロール表の概観	415
Q キャプチャー・サーバーのコントロール表のリスト	421
Q アプライ・サーバーのコントロール表のリスト	422
モニター・コントロール・サーバーの表のリスト	424
Q キャプチャー・コントロール表の詳細な構造	425
Q キャプチャー・コントロール表の詳細な構造 -- 概説	425
IBMQREP_ADMINMSG 表	425
IBMQREP_CAPENQ 表	426
IBMQREP_CAPMON 表	427
IBMQREP_CAPPARMS 表	428
IBMQREP_CAPQMON 表	432

IBMQREP_CAPTRACE 表	433
IBMQREP_SRCH_COND 表.	434
IBMQREP_SENDQUEUES 表	434
IBMQREP_SIGNAL 表	437
IBMQREP_SRC_COLS 表	441
IBMQREP_SUBS 表	442
Q アプライ・コントロール表の詳細な構造	448
Q アプライ・コントロール表の詳細な構造 -- 概説	448
IBMQREP_APPENQ 表	448
IBMQREP_APPLYMON 表	449
IBMQREP_APPLYPARMS 表	452
IBMQREP_APPLYTRACE 表	454
IBMQREP_DELTOMB 表	455
IBMQREP_DONEMSG 表	456
IBMQREP_EXCEPTIONS 表	457
IBMQREP_RECVQUEUES 表	459
IBMQREP_SAVERI 表	461
IBMQREP_SPILLEDROW 表	462
IBMQREP_SPILLQS 表	463
IBMQREP_TRG_COLS 表	464
IBMQREP_TARGETS 表.	465
モニター・コントロール表の詳細な構造	471
モニター・コントロール表の詳細な構造 -- 概説	471
IBMSNAP_ALERTS 表	472
IBMSNAP_CONDITIONS 表	474
IBMSNAP_CONTACTGRP 表	481
IBMSNAP_CONTACTS 表	482
IBMSNAP_GROUPS 表	483
IBMSNAP_MONENQ 表.	483
IBMSNAP_MONPARMS 表.	483
IBMSNAP_MONSERVERS 表	485
IBMSNAP_MONTRACE 表.	487
IBMSNAP_MONTRAIL 表	488
ピアツーピア・レプリケーションのバージョン管理用追加列の詳細な構造	489

第 27 章 イベント発行の XML メッセージの構造 491

イベント発行の XML メッセージの構造 -- 概要	491
XML メッセージ・タイプおよび要件	491
XML メッセージ・タイプおよび要件 -- 概要	491
XML メッセージ・タイプ	491
XML メッセージの技術要件	492
文字データで XML 区切り文字を処理する方法	493
Q キャプチャーからユーザー・アプリケーションへの XML メッセージの構造.	493
Q キャプチャーからユーザー・アプリケーションへの XML メッセージの構造 -- 概要	493
Q キャプチャーからユーザー・アプリケーションへの XML メッセージのリスト	494
msg: Q キャプチャーからユーザー・アプリケーションへの XML メッセージのルート・エレメント	495
トランザクション・メッセージ	497
行操作メッセージ	506

ラージ・オブジェクト (LOB) メッセージ	508
サブスクリプション非アクティブ化メッセージ	511
ロード完了受信メッセージ	512
エラー・レポート・メッセージ	513
ハートビート・メッセージ	514
サブスクリプション・スキーマ・メッセージ (subSchema)	515
列の追加メッセージ	520
ユーザー・アプリケーションから Q キャプチャー への XML メッセージの構造	521
ユーザー・アプリケーションから Q キャプチャー への XML メッセージの構造 -- 概説	521
ユーザー・アプリケーションから Q キャプチャー への XML メッセージのリスト	521
msg: ユーザー・アプリケーションから Q キャ プチャーへの XML メッセージのルート・エレ メント	522
送信キュー・メッセージの無効化	524
ロード完了メッセージ	524
サブスクリプション・メッセージのアクティブ化	525
サブスクリプション・メッセージの非アクティブ 化	526

**付録 A. Q レプリケーションおよびイベ
ント発行で使用するサンプル・プログラ
ム 529**

Q レプリケーションおよびイベント発行で使用する サンプル・プログラム -- 概要	529
サンプル・プログラム (Linux、UNIX、Windows)	529
Q レプリケーションおよびイベント発行で使 用するサンプル・プログラム (Linux、UNIX、Windows) -- 概要	529
Q レプリケーションおよびイベント発行をセッ トアップするためのサンプル (Linux、UNIX、Windows)	529

Q レプリケーション・プログラムを操作するた めのサンプル (Linux、UNIX、Windows)	530
サンプル・プログラム (z/OS)	531
Q レプリケーションおよびイベント発行で使 用するサンプル・プログラム (z/OS) -- 概要	531
Q レプリケーションおよびイベント発行をセッ トアップするためのサンプル (z/OS)	532
Q レプリケーションおよびイベント発行のサン プル JCL プログラム (z/OS)	532

**付録 B. ASNCLP: レプリケーションお
よびイベント発行を管理するコマンド行
インターフェース 535**

用語集 537	
用語集	537

ユーザー補助 551

キーボード入力およびナビゲーション	551
利用可能なモニター	551
フォントの設定	551
色の非依存	551
代替アラート・キュー	551
支援テクノロジーとの互換性	552
利用しやすい資料	552

特記事項 553

商標	555
--------------	-----

索引 557

IBM と連絡を取る 575

製品情報	575
資料についてのコメント	575

本書について

本書は、Q レプリケーションおよびイベント発行を使用する、データ・レプリケーションおよび発行環境の計画、セットアップ、保守、およびモニターの方法について説明します。Q レプリケーションおよびイベント発行を、IBM® レプリケーション・ソリューションの他の製品とともに使用して、ご使用の環境に適合するようにレプリケーションおよび発行環境を調整できます。本書には、「*IBM DB2 Information Integrator* レプリケーションとイベント・パブリッシング入門」で紹介されている、Q レプリケーションおよびイベント発行コンポーネントに関する手引きと参照情報が記載されています。

本書のほとんどのセクションは、サポートされているすべてのオペレーティング・システム環境に関するものです。たとえば、『Q アプライ・プログラムの開始』というセクションでは、DB2 Universal Database for Linux and UNIX (全 UNIX プラットフォーム対応)、DB2 Universal Database for Windows、または DB2 Universal Database for z/OS and OS/390 から Q アプライ・プログラムを開始する方法について説明しています。特定のオペレーティング・システムに関する情報は、それを明示しています。

本書の対象読者

本書は、データベース管理者、データ・レプリケーションの専門家、および Q レプリケーションまたは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。本書の読者は、以下に精通していることを前提としています。

- 標準的なデータベースの用語。
- 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。
- 「*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」で説明されている、Q レプリケーションおよびイベント発行に関する概念。
- Q レプリケーションまたはイベント発行環境に関するオペレーティング・システム環境。
- 複製または発行するデータ。
- イベント発行でメッセージを受け取るようにするアプリケーション。
- WebSphere MQ の概念およびオブジェクト。(本書では、WebSphere MQ の構成および使用に関する詳細については、他の情報源を参照しています。)

本書の表記規則

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、フィールド名、フォルダー名、アイコン、またはメニュー選択などの、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、示されたとお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。本書で使用されるレプリケーションおよび発行の概念はすべて、537 ページの『用語集』で定義されています。本書で使用されている主な概念は、「*IBM DB2 Information Integrator* レプリケーションとイベント・パブリッシング入門」で紹介され、説明されています。

特に指定しない限り、以下の意味は次のとおりです。

Linux Linux は、すべての Linux プラットフォーム (Linux Intel や Linux z/Series など) での Q レプリケーションおよびイベント発行について言及します。

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) での Q レプリケーションおよびイベント発行について言及します。

Windows

Windows は、Windows プラットフォーム (2000 や XP など) での Q レプリケーションおよびイベント発行について言及します。

z/OS z/OS は、z/OS および OS/390 での Q レプリケーションおよびイベント発行について言及します。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、DB2 レプリケーションについて、役に立つと思われるその他の情報を示します。

目的	参照先
DB2 Information Integrator レプリケーション・ソリューションについての情報を得る	DB2 DataPropagator については、 www.ibm.com/software/data/dpropr/
レプリケーションおよびイベント発行の基本的な概念について学習する	以下のいずれか <ul style="list-style-type: none"> • DB2 インフォメーション・センター (http://publib.boulder.ibm.com/infocenter/db2help/) • <i>IBM DB2 Information Integrator</i> レプリケーションとイベント・パブリッシング入門
SQL レプリケーションをセットアップおよび保守する方法について学習する	以下のいずれか <ul style="list-style-type: none"> • DB2 インフォメーション・センター (http://publib.boulder.ibm.com/infocenter/db2help/) • <i>IBM DB2 Information Integrator SQL</i> レプリケーション・ガイドおよびリファレンス
製品の最終変更についての情報を得る	CD-ROM 上のインストール情報、または製品と一緒にインストールされたりリリース情報。

目的	参照先
レプリケーション・アナライザーのコマンド行ツールを使用して、レプリケーション環境に関する HTML レポートを作成する方法について学習する	DB2 DataPropagator 製品資料については、 www.ibm.com/software/data/dpropr/library.html
ASNCLP コマンド行プロセッサおよびサポート文書をダウンロードする	http://www.ibm.com/software/data/dpropr/library.html
エラー・メッセージをデバッグする	以下のいずれかから入手した ASN メッセージ <ul style="list-style-type: none"> DB2 インフォメーション・センター (http://publib.boulder.ibm.com/infocenter/db2help/) DB2 UDB メッセージ・リファレンス 第 1 巻
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	www.ibm.com/software/data/integration/db2ii/
DB2 Universal Database および DB2 Information Integrator 用に設定された情報を表示する	DB2 インフォメーション・センター (http://publib.boulder.ibm.com/infocenter/db2help/) を参照

構文図の見方

本書で使用される構文図には、以下の規則が適用されます。



- 構文図は、左から右へ、上から下へ、線のパスに沿って読んでいきます。

記号  は、ステートメントの先頭を示します。

記号  は、ステートメント構文が次の行へ続くことを示します。



記号  は、ステートメントが前の行から続いていることを示します。

記号  は、ステートメントの終わりを示します。

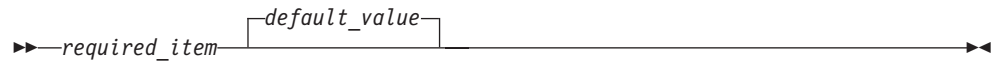
完全なステートメントでない構文ユニット図は、記号  で始まり、記号  で終わります。

- キーワード、その同義語、および予約パラメーターは、オペレーティング・システムによって異なりますが、英大文字または小文字のいずれかで示されます。これらの項目は、示された通りに正確に入力する必要があります。変数は、小文字のイタリック体で表示されます (たとえば、*column-name*)。これらは、ユーザー定義のパラメーターまたはサブオプションを表します。

コマンドを入力する際、区切りの句読点がなければ、パラメーターとキーワードは少なくとも 1 つスペースを入れて区切ります。

- 句読点記号 (スラッシュ、コンマ、ピリオド、括弧、引用符、等号など) や数を指定された通りに入力します。
- 脚注は、たとえば (1) のように、括弧内の数で示されます。
- 必須項目は、水平線 (メイン・パス) 上に表示されます。
 *required_item* 

- パラメーターのデフォルト値は、以下のように、パスの上に表示されます。

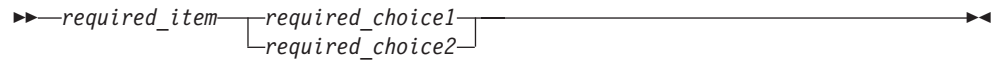


- 任意指定の項目は、メイン・パスの下に表示されます。

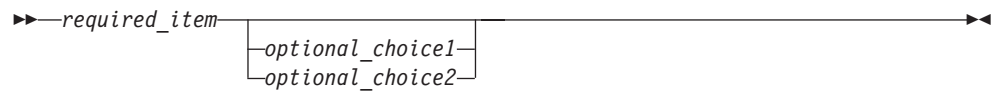


- 複数の項目を選択できる場合は、垂直に積み重なって (スタックに) 表示されま
す。

必ず 1 つ選択する 必要がある 場合は、スタックの項目の 1 つがメイン・パス
上に表示されています。



項目の選択が任意指定の場合は、スタック全体がメイン・パスの下に表示されま
す。



第 1 部 Q レプリケーションおよびイベント発行の紹介

本書の第 1 部は、以下の章で構成されています。

- 3 ページの『第 1 章 Q レプリケーションの紹介』では、単一方向および複数方向のレプリケーションを含む Q レプリケーションの処理方法の概要と、さまざまな構成の概要を記載しています。この章では、レプリケーション・キュー・マップ、Q サブスクリプション、およびターゲット・オブジェクト・プロファイルを含む、Q レプリケーションに関連したオブジェクトも説明しています。
- 9 ページの『第 2 章 イベント発行の紹介』では、イベント発行の処理方法の概要を記載しており、発行キュー・マップや XML 発行を含む、イベント発行に関連したオブジェクトを説明しています。
- 15 ページの『第 3 章 複製および発行で使用するプログラムの紹介』では、Q キャプチャー・プログラムと Q アプライ・プログラムを説明しています。Q キャプチャー・プログラムは、Q レプリケーションおよびイベント発行プログラムで、データをソース表からキャプチャーします。Q アプライ・プログラムは、Q レプリケーションのプログラムで、データをターゲット表へアプライしたり、データ操作のためにデータをストアード・プロシージャーに渡します。

第 1 章 Q レプリケーションの紹介

Q レプリケーションの紹介 -- 概要

以下のトピックでは、Q レプリケーションの主な概念について紹介します。

- 『Q レプリケーション』
- 5 ページの『Q サブスクリプション』
- 6 ページの『レプリケーション・キュー・マップ』

関連概念:

- 15 ページの『複製および発行で使用するプログラムの紹介 -- 概要』
- 9 ページの『イベント発行の紹介 -- 概要』

Q レプリケーション

Q レプリケーション は、大ボリュームで待ち時間が短いレプリケーションの解決策で、WebSphere® MQ メッセージ・キューを使用してソースとターゲットのデータベースやサブシステムの間でトランザクションを伝送します。Q キャプチャー・プログラムは、DB2® リカバリー・ログを読み取って、指定されたソース表に変更します。次にこのプログラムはトランザクションをメッセージとしてキュー上に送信し、このキューは Q アプライ・プログラムによって読み取られてターゲットに適用されます。

このタイプのレプリケーションには、次のようないくつかの利点があります。

最小の待ち時間

変更がソースでコミットされ、ログから読み取られるとすぐに送信されます。

大ボリューム・スループット

Q キャプチャー・プログラムは、ソースでの急速な変化に対応でき、マルチスレッド Q アプライ・プログラムは、通信チャンネルの速度に対応できます。

最小のネットワーク・トラフィック

メッセージが圧縮形式で送信され、さらにデータ送信オプションによって最少量のデータを送信できます。

非同期 メッセージ・キューの使用により、Q アプライ・プログラムが、ソースのデータベースやサブシステムに接続せずにトランザクションを受信できます。いずれかのレプリケーション・プログラムが停止した場合、メッセージはキュー上に残り、プログラムが使用可能になったときに処理されます。メッセージは永続的なため、システムまたはデバイスに障害が発生した場合でも、ソースとターゲットは同期された状態に保たれます。

Q レプリケーションでは、さまざまな構成が可能です。リモート・サーバー間や、単一のサーバー内で複製ができます。単一方向または複数方向で変更を複製できま

す。複数方向でのレプリケーションは、双方向レプリケーション (スタンバイ・システムまたはバックアップ・システムの管理に役立つ) またはピアツーピア・レプリケーション (実動システム上のデータの同期に役立つ) になります。

Q レプリケーションを使用するには、次のタイプのオブジェクトを作成します。

- データを送受信するための WebSphere MQ キューを識別するレプリケーション・キュー・マップ。
- 複製または発行する行および列などのオプション、およびターゲット表のロードのオプションを識別する、Q サブスクリプション。

以下のセクションでは、Q レプリケーションの 3 つのタイプを簡単に概説します。

- 『単一方向レプリケーション』
- 『双方向レプリケーション』
- 『ピアツーピア・レプリケーション』

単一方向レプリケーション

単一方向レプリケーション は、次の特性を持つ構成です。

- ソース表に加えられた変更内容が、WebSphere MQ キューを通してターゲット表に複製されるか、データ操作のためにストアード・プロシージャーに入力パラメーターとして渡されます。
- ターゲット表で加えられた変更内容は、ソース表に複製され戻されません。
- 通常、ターゲット表は、読み取り専用であるか、Q アプライ・プログラムのみにより更新されます。

双方向レプリケーション

双方向レプリケーション は、次の特性を持つ構成です。

- 2 つのサーバー上の表の間で複製が行われます。表のコピーの 1 つに加えられた変更が、その表の 2 番目のコピーに複製され、さらに 2 番目のコピーに行われた変更が最初のコピーに複製して戻されます。
- 一方のサーバー上の更新が、もう一方のサーバーに複製されます。
- どのサーバー上のアプリケーションでも、これらの表内の同じ行を同時に更新できます。ただし、複製された表の中の同一のデータが両方のサーバーによって同時に更新される可能性はほとんどあるいはまったくありません。同一の行が一度に 1 つのサーバーによって更新されるか、または、一方のサーバーが特定のデータ列のみを更新し、もう一方のサーバーはその他のデータ列を更新するかのどちらかです。
- 競合が発生した場合は、表のどちらのコピーを優先するか選択できる。

ピアツーピア・レプリケーション

ピアツーピア・レプリケーション (マルチマスター・レプリケーションとも呼ばれる) は、次の特性を持つ構成です。

- 複数のサーバー上の表の間でレプリケーションが行われます。
- いずれかのサーバー上の更新が、ピアツーピア構成に含まれる他のすべての関連サーバーに複製されます。

- いずれかのサーバー上のアプリケーションが、これらの表内の同一の行および列を同時に更新できます。
- すべてのサーバーは、データに対して等しい所有権を持つ対等の関係であり、どのサーバーも「マスター」や、データのソース所有者ではありません。

関連概念:

- 133 ページの『双方向レプリケーション』
- 136 ページの『ピアツーピア・レプリケーション』
- 19 ページの『Q アプライ・プログラム』
- 15 ページの『Q キャプチャー・プログラム』
- 3 ページの『Q レプリケーションの紹介 -- 概要』
- 5 ページの『Q サブスクリプション』
- 6 ページの『レプリケーション・キュー・マップ』
- 91 ページの『単一方向レプリケーション』

Q サブスクリプション

Q レプリケーションでは、データを単一のソース表から単一のターゲット表に複製したり、データ操作のためにストアード・プロシージャのパラメーターに渡したりする方法を、Q サブスクリプション と呼ばれるオブジェクトを作成して定義します。Q サブスクリプションは、ソース表からキャプチャーする変更、変更メッセージの送受信に使用するキュー、およびメッセージの処理方法を、レプリケーション・プログラムに指示します。図1は、Q サブスクリプションにおけるソース表とターゲット表の関係を表しています。

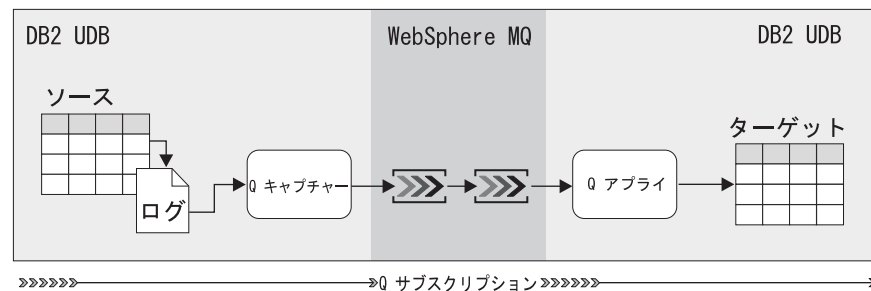


図1. Q サブスクリプション：ソース表の変更は、WebSphere MQ キューを通してターゲット表に複製されます。

Q サブスクリプションは、ソース対ターゲットのペアそれぞれに対して作成する必要があります。各 Q サブスクリプションは、以下の情報を識別する単一のオブジェクトです。

- 変更の複製元にするソース表
- 変更内容の複製先とするターゲット表またはストアード・プロシージャ
- 複製するソース表の列および行
- レプリケーション・キュー・マップ。これは、ソース・サーバーとターゲット・サーバーの間で情報を転送する WebSphere® MQ キューを指定する。

Q サブスクリプションの場合、以下の選択を指定できます。

- どの列を複製し、それらをどのようにターゲット表の列 (またはストアード・プロシージャのパラメーター) にマップするか
- 複製されるソース表の行を決定する検索条件
- Q アプライ・プログラムが、ソース・データベースの行とターゲット・データベースの行の間の競合を検出し、それに対処する方法

双方向レプリケーションまたはピアツーピア・レプリケーションの Q サブスクリプションでは、必ずソース表のすべての列が複製されます。これらのタイプのレプリケーションでは、さらに特殊な競合検出および競合処理が必要になります。

Q サブスクリプションを作成後に変更する場合は、それを再初期化して、Q キャプチャー・プログラムにそれらの変更を認識させる必要があります。

重要: Q サブスクリプションは XML 発行とは異なるオブジェクトです。Q サブスクリプションは、XML 発行で発行されるデータは複製しません。Q サブスクリプションは、データを複製するためのものであり、XML 発行は、データを発行するためのものです。

関連概念:

- 3 ページの『Q レプリケーション』
- 98 ページの『単一方向レプリケーションの Q サブスクリプションの作成 -- 概要』
- 12 ページの『XML 発行』

関連タスク:

- 205 ページの『Q サブスクリプションの属性の変更』
- 213 ページの『Q サブスクリプションの削除』

レプリケーション・キュー・マップ

Q レプリケーションにおいて、レプリケーション・キュー・マップ は、Q キャプチャー・プログラムと Q アプライ・プログラムがデータ転送と通信に使用する WebSphere® MQ キューを識別します。各レプリケーション・キュー・マップは、以下の各 WebSphere MQ キューのいずれかを識別します。

送信キュー

Q キャプチャー・プログラムが、ソース・データおよび通知メッセージを送信する WebSphere MQ キュー。レプリケーション・キュー・マップで識別される送信キューを使用して送信されるメッセージは、Q アプライ・プログラムが読み取るメッセージ・フォーマットである圧縮形式でなければなりません。

受信キュー

Q アプライ・プログラムが、ソース・トランザクションをターゲット表にアプライしたり、ストアード・プロシージャに渡したりする前の、それらのソース・トランザクションの受信先の WebSphere MQ キュー。

管理キュー

Q アプライ・プログラムは、この管理キューを使用して Q キャプチャー・プログラムに制御メッセージを送信します。

単一のレプリケーション・キュー・マップを、1 つ以上の Q サブスクリプションのデータの転送に使用できます。さらに、単一の Q キャプチャー・プログラムが、多数の送信キューにデータを書き込むことができ、単一の Q アプライ・プログラムが、多数の受信キューからデータを読み取ってアプライすることもできます。

WebSphere MQ キューを識別することに加えて、レプリケーション・キュー・マップは、送信キューおよび受信キューに関するレプリケーション属性も含んでいます。各レプリケーション・キュー・マップに対して、送信キューに関する以下の属性を指定します。

- 送信キューの WebSphere MQ 名。
- Q キャプチャー・プログラムがこの送信キューに書き込めるメッセージの最大サイズ。(この制限は、WebSphere MQ Series の最大メッセージ長とは別ですが、この制限は WebSphere MQ Series の最大メッセージ長以下にする必要があります。)
- WebSphere MQ キューでエラーが発生した場合の Q キャプチャー・プログラムの応答方法。
- 複製する変更がない場合に、Q キャプチャー・プログラムが実行していることを Q アプライ・プログラムに知らせるために、このキュー上のメッセージを送信する頻度。

受信キューに関する以下の属性が指定されます。

- 受信キューの WebSphere MQ 名。
- トランザクションをターゲット表にアプライするため、またはデータ操作のためにトランザクションをストアード・プロシージャに渡すために作成される各 Q アプライ・ブラウザ用のスレッドの数。
- Q アプライ・プログラムが、受信キューからのメッセージを処理するために使用できるメモリーの量。
- トランザクションを含むメッセージを送信している Q キャプチャー・プログラムを識別するスキーマ。
- 管理キュー (Q アプライ・プログラムが、Q キャプチャー・プログラムと通信する必要があるときに、メッセージを送信するキュー) として使用する WebSphere MQ メッセージ・キューの名前。

関連概念:

- 13 ページの『発行キュー・マップ』
- 3 ページの『Q レプリケーションの紹介 -- 概要』
- 93 ページの『レプリケーション・キュー・マップと Q サブスクリプションのグループ化』

関連タスク:

- 95 ページの『レプリケーション・キュー・マップの作成』

第 2 章 イベント発行の紹介

イベント発行の紹介 -- 概要

以下のトピックでは、イベント発行の主な概念を紹介しています。

- 『イベント発行』
- 12 ページの『XML 発行』
- 13 ページの『発行キュー・マップ』

関連概念:

- 15 ページの『複製および発行で使用するプログラムの紹介 -- 概要』
- 3 ページの『Q レプリケーションの紹介 -- 概要』

イベント発行

イベント発行 では、ソース表に対する変更は XML メッセージに変換され、選択したユーザー・アプリケーションに WebSphere® MQ キューを使用して送信されます。

以下の例では、イベント発行の使用を選択する理由を示しています。

Web サイトの更新

株価を記録したデータベースまたはサブシステムへの変更は、DB2® ログからキャプチャーされ、XML メッセージとしてアプリケーション・サーバー (WebSphere アプリケーション・サーバーなど) で実行される JSP (Java サーバー・ページ) アプリケーションに発行されます。次いでこの XML メッセージは、最新の株情報を表示する HTML ページの更新に使用されます。

中央のインテグレーション・ブローカーへのフィード

お客様が住所を更新すると、トランザクションは XML メッセージとして、中央の WebSphere Business Integration Message Broker に発行されます。このブローカーはその XML メッセージを、ビジネス、そのパートナー、およびサプライヤーが使用するさまざまなアプリケーションで理解できる形式に変換します。

ヒント: Q レプリケーションのセットアップおよびイベント発行のサンプル・プログラムには、Q キャプチャー・プログラムが発行する XML メッセージを利用する Web ベース・アプリケーションのサンプルがあります。このサンプルでは、ビジネス・シナリオでの XML 発行の使用方法を示しています。

イベント発行では、Q アプライ・プログラムは使用されず、Q キャプチャー・プログラムのみ使用されます。Q キャプチャー・プログラムは、XML 発行と呼ばれるオブジェクトを作成したとき、ユーザーが指定する変更をキャプチャーします。こ

のトランザクションまたは行レベルの変更は、次いでキューに送信されます。発行キュー・マップと呼ばれるオブジェクトを作成したときに使用するキューを指定します。

イベント発行によって、広範な利用目的がある XML フォーマットで発行されたトランザクションのデータを柔軟に使用することができます。Q アプライ・プログラムを使用してターゲットに変更を複製する場合は、イベント発行ではなく Q レプリケーションを使用してください。

イベント発行のオブジェクト

イベント発行では、以下のオブジェクトがサーバー間に存在します。

発行キュー・マップ

少なくとも 1 つの発行キュー・マップを作成し、各ソース・サーバーにある Q キャプチャー・プログラムからデータを転送しなければなりません。

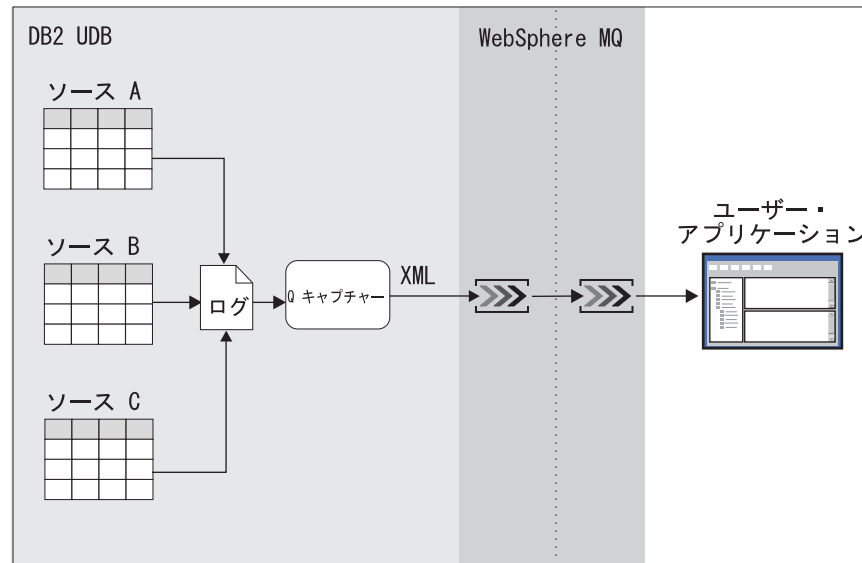
XML 発行

変更を発行したいソース表ごとに、少なくとも 1 つの XML 発行を作成しなければなりません。XML 発行は、発行したい行や列と同様に表を指定します。複数の XML 発行で同一のソース表を指定することができます。たとえば、EMPLOYEE 表からのアプリケーション A およびアプリケーション B への変更や、DEPARTMENT 表からのアプリケーション A への変更を発行する場合は、以下のように 2 つの発行キュー・マップのある 3 つの XML 発行を作成します。

- 発行キュー・マップ A を使用する EMPLOYEE 表からアプリケーション A への XML 発行 EMPLOYEE0001。
- 発行キュー・マップ A を使用する DEPARTMENT 表からアプリケーション A への XML 発行 DEPARTMENT0001。
- 発行キュー・マップ B を使用する EMPLOYEE 表からアプリケーション B への XML 発行 EMPLOYEE0002。

アプリケーション A は、Q キャプチャー・プログラムが発行キュー・マップ A にある送信キューに送信したメッセージを読み取り、アプリケーション B は、Q キャプチャー・プログラムが発行キュー・マップ B にある送信キューに送信したメッセージを読み取ります。

11 ページの図 2 は、同じサーバーにある 3 つのソース表のイベント発行を示しています。3 つの XML 発行と少なくとも 1 つの発行キュー・マップが必要です。



>>>>>>→ ソース A からの XML 発行 >>>>>>→
 >>>>>>→ ソース B からの XML 発行 >>>>>>→
 >>>>>>→ ソース C からの XML 発行 >>>>>>→

図2. イベント発行：ソース表からの変更は、XML フォーマットで 1 つ以上の WebSphere MQ 送信キューに発行されるので、ユーザー・アプリケーションはこのメッセージを検索して使用することができます。

メッセージ内容の制御

Q キャプチャー・プログラムが送信する XML メッセージにソース表からの 1 行のみの変更だけを含めるか、それともトランザクション全体を含めるかを選択できます。発行キュー・マップの作成時にこの選択ができます。Q キャプチャー・プログラムは、この発行キュー・マップを指定するすべての XML 発行に対して選択したメッセージ・フォーマットを使用します。

たとえば、EMPLOYEE 表からアプリケーション A に発行されるメッセージに個別の行への変更を含めることや、DEPARTMENT 表からアプリケーション A に発行されるメッセージにトランザクション全体を含めることを指定できます。この場合、以下の構成を使用できます。

- 発行キュー・マップ A (発行キュー・マップ A はメッセージの行の個別の変更を指定している) を使用する EMPLOYEE 表からアプリケーション A への XML 発行 EMPLOYEE0001。
- 発行キュー・マップ C (発行キュー・マップ C はメッセージのトランザクション全体を指定している) を使用する DEPARTMENT 表からアプリケーション A への XML 発行 DEPARTMENT0001。

関連概念:

- 13 ページの『発行キュー・マップ』
- 15 ページの『Q キャプチャー・プログラム』
- 3 ページの『Q レプリケーション』
- 9 ページの『イベント発行の紹介 -- 概要』

- 12 ページの『XML 発行』

関連資料:

- 529 ページの『Q レプリケーションおよびイベント発行をセットアップするためのサンプル (Linux、UNIX、Windows)』

XML 発行

イベント発行により、XML 発行 と呼ばれるオブジェクトを作成し、XML 形式で発行された単一のソース表から WebSphere® MQ 送信キューへの変更方法を定義します。次いで、選択したユーザー・アプリケーションで、この XML メッセージを検索して使用できるようにします。XML 発行は、どの列または行が変更を発行したいソース表にあるかを Q キャプチャー・プログラムに通知します。Q キャプチャー・プログラムがキャプチャーするトランザクションまたは行レベルの変更は、XML メッセージとして送信され、選択したユーザー・アプリケーションでモニターできる送信キューに送信されます。図 3 では、XML 発行がソース表を WebSphere MQ 送信キューに接続する方法を示しています。

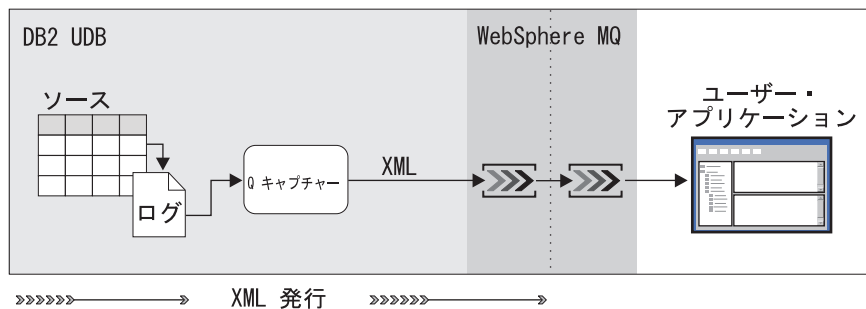


図 3. XML 発行：ソース表からの変更は、XML 形式で WebSphere MQ キューに発行されるので、ユーザー・アプリケーションがこのメッセージを検索して使用することができます。

XML 発行ごとに、以下の情報を指定します。

- 変更を発行したいソース表。
- ソース表から発行したい列および行。
- ソースの変更を発行する Q キャプチャー・プログラム。
- 使用する発行キュー・マップ。
- XML 発行が最初に作成されたときに、開始時に自動的にアクティブにするか、または Q キャプチャー・プログラムを再初期化する (デフォルトの動作) かどうか。また、非アクティブ状態で XML 発行の作成を選択し、Q キャプチャー・プログラムの XML 発行をアクティブにして変更のキャプチャーを開始することができる。

XML 発行を作成後に変更する場合は、再初期化して、Q キャプチャー・プログラムにこの変更を通知しなければなりません。

重要: XML 発行は、Q サブスクリプションとは別のオブジェクトです。Q サブスクリプションは、XML 発行で発行されるデータを複製しません。

関連概念:

- 182 ページの『XML 発行の作成 -- 概要』
- 5 ページの『Q サブスクリプション』
- 9 ページの『イベント発行』

関連タスク:

- 219 ページの『XML 発行の属性の変更』
- 225 ページの『XML 発行の削除』

発行キュー・マップ

イベント発行において、発行キュー・マップとは、Q キャプチャー・プログラムがデータを転送し、XML メッセージを受信するユーザー・アプリケーションと通信するために使用する、WebSphere® MQ キューのことです。各発行キュー・マップは、1 つの送信キュー (Q キャプチャー・プログラムがソース・データと通知メッセージを送信する WebSphere MQ キュー) を識別します。

イベント発行では、Q キャプチャー・プログラムが、XML フォーマットでソース・データを送信キューに発行します。Q アプライ・プログラムは、これらのメッセージを検索しません。WebSphere MQ や選択するアプリケーションは、送信キュー上の XML メッセージを処理するように構成できます。

単一の発行キュー・マップを、1 つ以上の XML 発行のデータの転送に使用できます。また、単一の Q キャプチャー・プログラムで、複数の発行キュー・マップを処理することもできます。

Q キャプチャー・プログラムの変更の送信先の WebSphere MQ キューを識別することに加え、発行キュー・マップは、送信キューに関する発行属性も含みます。各発行キュー・マップに対して、送信キューに関する以下の属性を指定します。

- 送信キューの WebSphere MQ 名。
- 送信キューが転送するメッセージの内容のタイプ。XML 発行に使用されるメッセージ・キューでは、低レベルの変更またはトランザクションを転送できます。
- Q キャプチャー・プログラムがこの送信キューに書き込めるメッセージの最大サイズ。(この制限は、WebSphere MQ Series の最大メッセージ長には依存しませんが、この制限は WebSphere MQ Series の最大メッセージ長以下にする必要があります。)
- WebSphere MQ キューでエラーが発生した場合の Q キャプチャー・プログラムの応答方法。
- 発行する変更がない場合に、Q キャプチャー・プログラムが、実行していることをユーザー・アプリケーションに知らせるために送信キューにメッセージを送信する頻度。

関連概念:

- 179 ページの『発行キュー・マップおよび XML 発行のグループ化』
- 6 ページの『レプリケーション・キュー・マップ』
- 9 ページの『イベント発行の紹介 -- 概要』

関連タスク:

- 180 ページの『発行キュー・マップの作成』

第 3 章 複製および発行で使用するプログラムの紹介

複製および発行で使用するプログラムの紹介 -- 概要

以下のトピックでは、Q レプリケーションおよびイベント発行を使用したデータの複製および発行に使用されるプログラムの概要を説明します。

- 『Q キャプチャー・プログラム』
- 19 ページの『Q アプライ・プログラム』
- 21 ページの『Q アプライ・プログラムと Q キャプチャー・プログラムのスキーマ』

関連概念:

- 3 ページの『Q レプリケーションの紹介 -- 概要』
- 9 ページの『イベント発行の紹介 -- 概要』

Q キャプチャー・プログラム

Q キャプチャー・プログラム は、ソース表で発生した変更を DB2[®] リカバリー・ログから読み取って変更点をメッセージにし、そのメッセージを、WebSphere[®] MQ キューを介して送信します。メッセージは、このキューで Q アプライ・プログラムまたはユーザー・アプリケーションによって処理されます。Q キャプチャー・プログラムでは、ソース表から発行または複製するデータを指定できます。

- どのソースの変更点を送信するかは、ソース表、またはさらにソース表の中の行と列を指定することによって制御できます。
- キュー上を流れるデータの速度と量は、パラメーターの中でもとりわけプログラムがメッセージをコミットする間隔を設定することによって制御できます。

複数の Q キャプチャー・プログラムが単一のソース・サーバーのデータを独自にキャプチャーするように定義できます。サーバー上の各 Q キャプチャー・プログラムは、固有のスキーマ名で識別されます。

さらに、Q キャプチャー・プログラムでは、データを圧縮形式で Q アプライ・プログラムに送信するか、または XML フォーマットでユーザー・アプリケーションに送信するかを選択できます。Q キャプチャー・プログラムの稼働パラメーターは、いくつかの方法で変更できますが、それには実行中に動的に行う方法も含まれます。以下のセクションでは、Q キャプチャー・プログラムについて詳しく説明します。

- 16 ページの『変更のキャプチャー』
- 16 ページの『データのフィルタリング』
- 17 ページの『メッセージ・フォーマット』
- 17 ページの『メッセージのコミット』
- 17 ページの『パラメーターの変更』
- 17 ページの『Q キャプチャーとの通信』

変更のキャプチャー

ソース表で行が変更されると、Q キャプチャー・プログラムは、その表がアクティブの Q サブスクリプションまたは XML 発行の一部かどうかをログ・レコードから読み取ります。一部であれば、Q キャプチャー・プログラムは、メモリー内の対応するデータベース・トランザクションに行を追加します。行の変更に、ラージ・オブジェクト (LOB) データを含む列が関係している場合、Q キャプチャー・プログラムは LOB データをソース表から送信キューへ直接コピーします。

検索条件を定義すると、Q キャプチャー・プログラムは、その検索条件を使用してメモリー内の各行を評価します。検索条件と合致する行は、それらが属するトランザクションがソース・データベースでコミットされるときにメッセージとして集められます。その後、Q キャプチャー・プログラムは、指定された送信キューにメッセージを書き込みます。

データのフィルタリング

Q キャプチャー・プログラムでは、検索条件で行をフィルタリングしたり、送信される列値を制限したり、削除の伝搬を抑制することによって、レプリケーションの処理速度を高め、ネットワーク・トラフィックを最小化できます。

検索条件

検索条件は、各 Q サブスクリプションまたは XML 発行に対して指定できます。検索条件は、制限付きの SQL WHERE 文節です。検索条件と合致しない行変更は、メッセージに含められません。

列オプション

XML 発行の場合は、次のオプションの中から選択することによって、メッセージに追加する列値を制限できます。(Q サブスクリプションの場合は、これらの列オプションは、他の選択項目に基づいて自動的に選択されます。)

- **すべての変更行:** デフォルトでは、Q キャプチャー・プログラムは、Q サブスクリプションまたは XML 発行の一部である列が変更された場合にのみ、行を送信します。いずれかの列が変更されたらプログラムが行を送信するようにすることもできます。
- **変更前の値:** デフォルトでは、Q キャプチャー・プログラムは、更新された非キー列の変更前の値を送信しません。プログラムが非キー列の変更前の値を送信するようにすることもできます。
- **変更済み列のみ:** デフォルトでは、Q キャプチャー・プログラムは、変更されたサブスクライブ列のみを送信します。プログラムが未変更のサブスクライブ列も送信するようにすることもできます。

削除の抑制

デフォルトでは、Q キャプチャー・プログラムは、ログから削除をキャプチャーして、それらを発行または複製します。しかし、削除を抑制することもできます。

メッセージ・フォーマット

Q キャプチャー・プログラムは、ソース表の変更を 2 つの異なるメッセージ・フォーマットに変換します。Q レプリケーションでは、Q アプライ・プログラムが読み取れる圧縮形式が使用されます。イベント発行では、XML フォーマットのメッセージが使用されます。XML メッセージの場合のみ、メッセージに単一行操作のみを含めるか、トランザクション内のすべての行操作を含めるかを選択できます。

メッセージのコミット

Q キャプチャー・プログラムは、1 つ以上の送信キューにメッセージを書き込むと、キュー・マネージャーに対してコミット・コールを発行し、送信キュー上のメッセージを Q アプライ・プログラムやユーザー・アプリケーションに使用可能にするように指示します。Q キャプチャー・プログラムがメッセージをコミットする頻度はユーザーが決定します。各コミットでグループ化されるすべての DB2 トランザクションが、1 つの WebSphere MQ トランザクションと見なされます。通常、各 WebSphere MQ トランザクションには、いくつかの DB2 トランザクションが含まれます。

コミットの間隔は、COMMIT_INTERVAL パラメーターの値を変更することによって調整できます。コミット間隔を短くすれば、エンドツーエンド待ち時間 (トランザクションがソースでコミットされてから、ターゲット・サーバーでコミットされるまでの遅延時間) を短くできます。

パラメーターの変更

Q キャプチャーの稼働パラメーターは、プログラムがトランザクションの作成用に割り当てるメモリー量、開始時に実行するアクション、コントロール表から古いデータを削除する頻度、およびその他の動作を制御します。これらのパラメーターは、次の 3 つの方法で変更できます。

- Q キャプチャー・プログラムがそのパラメーター値を読み取るコントロール表を更新する。
- プログラムを開始するときに、保管されている値を一時的にオーバーライドする。
- プログラムの実行中に、パラメーターを動的に変更する。

最後の 2 つの方法のいずれかを使用する場合、変更内容は、Q キャプチャー・プログラムが実行している間だけ有効です。停止して再始動すると、値を再度オーバーライドしない限り、コントロール表に保管されている値が使用されます。

Q キャプチャーとの通信

Q キャプチャー・プログラムは、コマンド、SQL シグナル、および XML と圧縮のメッセージに応答します。

プログラムの以下の動作は、レプリケーション・センターまたはシステム・コマンドを使用して制御できます。

- Q キャプチャー・プログラムを開始し、オプションで始動パラメーターを変更する
- Q キャプチャー・プログラムの実行中に、パラメーター値を変更する

- Q キャプチャー・プログラムを再初期化する
- 送信キューを再初期化する
- Q キャプチャー・プログラムを停止する

レプリケーション・センターは、SQL シグナルを発行して、以下の要求を Q キャプチャー・プログラムに伝達します (これらのタスクは、手動で SQL シグナルを挿入して実行することもできます)。

- Q キャプチャー・プログラムによる Q サブスクリプションまたは XML 発行のアクティブ化または非アクティブ化の要求
- ターゲット表がロードされたことのレポート
- Q キャプチャー・プログラムへの Q サブスクリプションまたは XML 発行の再初期化の指示

以下のタスクは、手動で SQL シグナルを挿入して実行できます。

- アクティブの単一方向の Q サブスクリプションまたは XML 発行に列を追加する
- 送信キューに対して定義されているエラー・アクションを実行する
- Q キャプチャー・プログラムを停止する
- トランザクションを無視する

Q アプライ・プログラムとユーザー・アプリケーションは、それぞれ圧縮メッセージと XML のメッセージを送信することによって、Q キャプチャー・プログラムと通信します。

再始動メッセージ

Q キャプチャー・プログラムは、ローカル WebSphere MQ キューを使用して再始動情報を保管します。再始動キューには、Q キャプチャー・プログラムの再始動時に、Q キャプチャー・プログラムに DB2 リカバリー・ログ内の読み取り開始場所を示す単一のメッセージが含まれます。

Q キャプチャー・プログラムは、そのコミット間隔に達するごとに、その再始動情報を更新する必要があるかどうかチェックします。その必要がある場合、Q キャプチャー・プログラムは、再始動キューのメッセージを、関連する再始動情報 (これには特に、再始動時にログ・レコードの処理を開始する必要があるログ内の最古のポイントが含まれる) を含む新しいメッセージで置き換えます。コールド・スタート・オプションを使用する場合、Q キャプチャー・プログラムは、ログ内の最新のポイントからログ・レコードの処理を開始することをプログラムに指示するメッセージで再始動メッセージを置き換えます。

関連概念:

- 15 ページの『複製および発行で使用するプログラムの紹介 -- 概要』
- 229 ページの『Q キャプチャー・プログラムの操作 -- 概要』
- 248 ページの『Q キャプチャー・パラメーターの変更 -- 概要』
- 32 ページの『Q レプリケーションおよびイベント発行の LOB データ・タイプのためのメモリー』
- 19 ページの『Q アプライ・プログラム』

- 3 ページの『Q レプリケーションの紹介 -- 概要』
- 21 ページの『Q アプライ・プログラムと Q キャプチャー・プログラムのスキーマ』
- 9 ページの『イベント発行の紹介 -- 概要』

Q アプライ・プログラム

Q アプライ・プログラム は、WebSphere® MQ キューからメッセージを取り込み、そのメッセージに含まれるトランザクションを再ビルドして、そのトランザクションをターゲットの表またはストアド・プロシージャにアプライします。Q アプライ・プログラムは、トランザクションを複数のターゲットに同時にアプライすることによって、複数のソースでの急速な変更について行くように設計されています。プログラムは、関係するターゲット表の間の参照保全を保ちながら、並行して変更をアプライできます。複数の Q アプライ・プログラムが、それぞれ独自に単一のターゲット・サーバーにデータをアプライするように定義できます。サーバー上の各 Q アプライ・プログラムは、固有のスキーマ名で識別されます。

Q アプライ・プログラムに、ターゲットにアプライ済みの変更をトラッキングさせるには、各ターゲット表で行を固有にする何らかのメカニズムが必要です。この固有性は、ターゲットの 1 次キー、ユニーク制約、またはユニーク索引によって付与することができます。

プログラムでトランザクションを並行してアプライするかどうかを指定し、いくつかの方法でこの操作動作を変更し、プログラムで競合を検出および処理する方法を指定できます。

以下のセクションでは、Q アプライ・プログラムについてさらに詳しく説明します。

- 『ブラウザー・スレッドとエージェント・スレッド』
- 20 ページの『トランザクションの並列でのアプライ』
- 20 ページの『パラメーターの変更』

ブラウザー・スレッドとエージェント・スレッド

Q アプライ・プログラムは、複数の受信キューからのトランザクションを処理できます。Q アプライ・プログラムは、各受信キューに対して、ブラウザー・スレッドと呼ばれる単一のスレッドを立ち上げます。

ブラウザー・スレッドは、受信キューからトランザクション・メッセージを取り込み、トランザクション間の従属関係を記録します。ブラウザー・スレッドは、すでにアプライ済みのトランザクションもトラッキングします。従属トランザクション間の参照保全を維持するには、受信キューを識別する各レプリケーション・キュー・マップが単一のブラウザー・スレッドによって処理されなければならない。従属表が関係する Q サブスクリプションが同一のレプリケーション・キュー・マップを使用していなければなりません。したがって、2 つの Q アプリケーション・プログラムが同一のレプリケーション・キュー・マップからトランザクションを取り込むことはできません。

各ブラウザー・スレッドは、1 つ以上のエージェント・スレッドを立ち上げます。エージェント・スレッドは、ブラウザー・スレッドからトランザクション・メッセージを取り込み、すべての行の変更に対する SQL ステートメントを再ビルドして、その変更をターゲットに適用し、トランザクションに対してコミット・ステートメントを発行します。

トランザクションの並列でのアプライ

WebSphere MQ では、トランザクション・メッセージは、必ずソースでコミットされた順序で受信キューに到着します。デフォルトでは、Q アプライ・プログラムは、複数のエージェント・スレッドを使用して、並列でトランザクションをアプライします。多数の変更点がソース・サーバーから複製される場合は、このような並列処理が重要になります。Q アプライ・プログラムでトランザクションを厳密な到着順でアプライする場合は、エージェント・スレッドの数を 1 に設定できます。

行は必ずしもソースでの変更の発生順序でアプライされるとは限らないので、トランザクションを並列でアプライする場合、Q アプライ・プログラムは特別な予防措置をとります。

たとえば、ソースの DEPARTMENTS 表に行を挿入することによって、新しい部門が作成されるとします。その後、ソースの EMPLOYEES 表に行が挿入されることによって、その新しい部門に従業員が追加されます。ソースの参照保全制約のため、EMPLOYEES 表に挿入される行には、DEPARTMENTS 表の中に一致するレコードがなければなりません。複製されるこれらの 2 つの表のコピーの間には、同じ親子従属関係が存在します。

これらのソース・トランザクションを含むメッセージが受信キューに到着すると、ブラウザー・スレッドは、これらの 2 つのトランザクションの間の従属関係を検出します。Q アプライ・プログラムが複数のエージェントを使用して並列でトランザクションをアプライしている場合、ブラウザー・スレッドは、一致する部門の行がアプライされるまで、ターゲット表への従業員レコードの挿入を遅らせます。その間、エージェント・スレッドは、並列で引き続き他のトランザクションをアプライします。

パラメーターの変更

Q アプライ稼働パラメーターでは、プログラムがパフォーマンス・データを保管または整理する頻度、プログラムが診断ログを保管する場所、およびデッドロックやロック・タイムアウトの後にプログラムがターゲットへの変更のアプライを再試行する頻度を設定できます。これらのパラメーターは、次の 3 つの方法で変更できます。

- Q アプライ・プログラムがそのパラメーター値を読み取るコントロール表を更新する。
- プログラムを開始するときに、保管されている値を一時的にオーバーライドする。
- プログラムの実行中に、パラメーターを動的に変更する。

最後の 2 つの方法のいずれかを使用する場合、変更は、Q アプライ・プログラムが実行している間だけ存続します。停止して再始動すると、値を再度オーバーライドしない限り、コントロール表に保管されている値が使用されます。

アプライ・エージェントの数とメモリー制限という 2 つの重要な稼働パラメーターは、Q アプライ・プログラムが処理する各受信キューに対するその動作を決定します。これらのパラメーターの値は、レプリケーション・キュー・マップを作成する際に指定できます。Q アプライ・プログラムが受信キューからメッセージを取り込んでいる最中でも、Q アプライ・エージェントの数とその受信キューのメモリー制限は、Q アプライ・プログラムを停止せずに変更できます。

関連概念:

- 15 ページの『複製および発行で使用するプログラムの紹介 -- 概要』
- 259 ページの『Q アプライ・プログラムの操作 -- 概要』
- 271 ページの『Q アプライ・パラメーターの変更 -- 概要』
- 15 ページの『Q キャプチャー・プログラム』
- 3 ページの『Q レプリケーションの紹介 -- 概要』
- 21 ページの『Q アプライ・プログラムと Q キャプチャー・プログラムのスキーマ』
- 9 ページの『イベント発行の紹介 -- 概要』

Q アプライ・プログラムと Q キャプチャー・プログラムのスキーマ

Q キャプチャーまたは Q アプライのスキーマは、Q キャプチャーまたは Q アプライ・プログラムによって使用されるコントロール表のセットに対して与えられる名前です。Q レプリケーションは、スキーマを使用して、特定のコントロール表のセットを使用する Q キャプチャーまたは Q アプライ・プログラムを識別します。たとえば、スキーマ SCHEMA_GREEN に Q キャプチャー・コントロール表のセットを作成する場合、そのコントロール表のセットを使用する Q キャプチャー・プログラムは、スキーマによって識別されます。

サーバーがレプリケーション・センターのオブジェクト・ツリーに表示される前に、少なくとも 1 つの Q キャプチャーまたは Q アプライ・スキーマが含まれていなければなりません。しかし、サーバー上で複数のスキーマを作成することができます。以下のような理由から、複数の Q キャプチャー・スキーマの作成が必要になることがあります。

並列処理が増加する場合

トラフィックを並列処理するには、複数の Q キャプチャー・プログラムを使用します。独立した Q キャプチャー・プログラムとログ・リーダーを使用して、表からの行の水平サブセット、表、または表のグループを複製するように複数の Q キャプチャー・スキーマをセットアップできます。プログラムはそれぞれ別個の送信キューのセット、または別個のキュー・マネージャーを使用して、変更メッセージの転送を加速することができます。この構成によって、たとえば大規模なソース表を使って、または大型のシスプレックス上で、パフォーマンスを向上させ、より高いスループットを実現することができます。トレードオフは、実行中の複数のログ・リーダーの追加 CPU コストと、必要となる DB2® UDB との追加接続です。

各レプリケーションのそれぞれ異なる要件を満たす場合

異なるワークロード特性を処理してソース変更のフローをさまざまな用途に向けるために、複数の Q キャプチャー・スキーマを作成します。独立した

Q キャプチャー・スキーマを使用することによって、Q キャプチャー稼働パラメーターをそれぞれのワークロードごとに個別に調整することができます。たとえば、異なるスキーマを持つ 2 つのキャプチャー・プログラムが同じソース表から読み取りを行えるとします。1 つは XML メッセージ形式で変更を発行するもので、もう 1 つは Q アプライ・プログラムへの変更を複製するものです。複製待ち時間の少ない Q キャプチャー・プログラムの場合、コミット・インターバルを 500 ミリ秒に設定できます。そして発行用の Q キャプチャー・プログラムの場合、コミット・インターバルを 5 秒に設定できます。XML 発行と Q サブスクリプションの両方を処理する、1 つの Q キャプチャー・スキーマによって識別されるキャプチャー・プログラムを 1 つ定義することにより、この同じシナリオを実現することができます。ただしこの場合、Q キャプチャーが使用する稼働パラメーターは同じになります。

単一のサーバーに複数の Q アプライ・スキーマを作成しても、ほとんどの場合、1 つのサーバーに 1 つの Q アプライ・プログラムがあれば十分です。大量のトランザクションが多数の表に複製されるレプリケーション構成の場合でも同様です。

第 2 部 Q レプリケーションおよびイベント発行の準備

本書の第 2 部は、以下の章で構成されています。

- 25 ページの『第 4 章 メモリーおよびストレージの要件の計画』では、Q レプリケーションとイベント発行に必要なメモリーとストレージの量について説明しており、適切なメモリーとストレージの量を割り振るための計画に役立てることができます。
- 37 ページの『第 5 章 データ変換』では、ユーザーの構成におけるコード・ページの計画方法を説明しています。
- 39 ページの『第 6 章 ユーザー ID およびパスワードのセットアップ』では、ユーザー ID およびパスワードのセットアップ方法と、さまざまなプログラムの権限要件を説明しています。
- 47 ページの『第 7 章 Q レプリケーションおよびイベント発行用の WebSphere MQ をセットアップする方法』では、WebSphere MQ 環境のセットアップ方法と、WebSphere MQ が必要とするオブジェクトを説明しています。
- 75 ページの『第 8 章 Q レプリケーションおよびイベント発行用のサーバーの構成』では、サーバーのセットアップ方法を説明し、サーバーが相互に接続し、必要な DB2 UDB オブジェクトを作成できるようにします。

第 4 章 メモリーおよびストレージの要件の計画

Q レプリケーションおよびイベント発行のメモリーおよびストレージ要件の計画

Q レプリケーションおよび発行環境では、トランザクションをキャプチャー、適用、およびモニターするために一定量のメモリーが必要です。必要なメモリー量は多くの要因によって決まりますが、一部の要因は調整可能です。ソース変更の複製または発行を始める前に、プログラムがトランザクションを予備ファイルに書き込むことがないよう、十分なメモリーを確保してください。また、プログラムが予備ファイルに書き込まなければならなくなる場合に備えて、予備データを収容するための十分なストレージを確保してください。

手順:

次のようにして、必要なメモリー量およびストレージ量を判別します。

- Q レプリケーションおよびイベント発行に必要なメモリー量を計画する。詳細については、『Q レプリケーションおよびイベント発行のメモリー要件の計画』を参照してください。
- Q レプリケーションおよびイベント発行に必要なストレージ・スペースを計画する。詳細については、32 ページの『Q レプリケーションおよびイベント発行のストレージ要件の計画』を参照してください。

Q レプリケーションおよびイベント発行のメモリー要件の計画

Q レプリケーションおよびイベント発行のメモリー要件の計画

メモリーは、システム管理リソースです。一般に、プロセス (z/OS でのジョブ) には、プロセスが使用できる最大メモリー量を決定するメモリー割り当て量があります。Q レプリケーションまたはイベント発行については、このような割り当て量を設定しないことをお勧めしますが、設定する場合には、割り当て量を大きくしてください。プログラムは、プログラムに必要な分だけのメモリーしか使用しません。レプリケーション・プログラムでメモリーが不足した場合で、ストレージを利用できない場合には、プログラムは停止します。プログラムには、プログラムが使用するメモリー量を制御できるメモリー限度パラメーターがあります。

Q キャプチャー・プログラムおよび Q アプライ・プログラムがメモリーを使用する方法を理解することによって、これらのプログラムが使用中の環境でデータを複製または発行するために使用するメモリー量を決定できます。基本的に、処理するトランザクションの規模が大きい場合、多くの並行トランザクションがある場合、Q キャプチャー・プログラムがトランザクションをコミットする頻度を低く構成している場合は、多くのメモリーが使用されます。これらのプログラムは、LOB および BLOB 値を処理するために必要なメモリー量も管理します。データの複製または発

行を開始した後で、トランザクションが予備ファイルに書き出されることを避けるために、プログラムが使用するメモリー量を調整する必要があります。

レプリケーション・アラート・モニターも、少量のメモリーを必要とします。

推奨: プログラムが大量のメモリーを使用する可能性がほとんどない場合でも、予想ワークロードで必要とされる最大メモリーに合わせて計画してください。

手順:

メモリー要件を判別するには、以下の事柄を計画する必要があります。

- Q キャプチャー・プログラムが使用する予想メモリー量。詳細については、『Q キャプチャー・プログラムによって使用されるメモリー』を参照してください。
- Q アプライ・プログラムが使用する予想メモリー量。詳細については、29 ページの『Q アプライ・プログラムによって使用されるメモリー』を参照してください。
- レプリケーション・アラート・モニターが使用する予想メモリー量。詳細については、32 ページの『レプリケーション・アラート・モニターによって使用されるメモリー』を参照してください。
- 複製または発行したい LOB 列が使用する予想メモリー量。詳細については、32 ページの『Q レプリケーションおよびイベント発行の LOB データ・タイプのためのメモリー』を参照してください。

関連タスク:

- 32 ページの『Q レプリケーションおよびイベント発行のストレージ要件の計画』

Q キャプチャー・プログラムによって使用されるメモリー

Q キャプチャー・プログラムは、ソースについての情報を保管したり、DB2® UDB ログからトランザクションを再構成したりするために、メモリーを使用します。

アクティブ・ソースについての情報を保管するためのメモリー

Q キャプチャー・プログラムは、複製または発行したいソース・データなどの、アクティブ・ソースについての情報を保管するためにメモリーを使用します。プログラムを開始するとき、または Q キャプチャー・プログラムの実行中に Q サブスクリプションまたは XML 発行をアクティブ化するとき、Q キャプチャー・プログラムはコントロール表からデータを検索してメモリーに保管します。各 Q サブスクリプションまたは XML 発行は、最大で 1000 バイトのメモリーを使用します。

トランザクションを再構成するためのメモリー

Q キャプチャー・プログラムは、DB2 UDB ログを読み取る場合、関連したコミット・レコードまたはアポート・レコードを読み取るまで個々のトランザクション・レコードをメモリーに保管します。アポートされたトランザクションに関連したデータはメモリーから消去され、コミット・レコードに関連したデータは、レプリケーションまたは発行キュー・マップで指定した WebSphere® MQ キューに書き込まれます。コミットされたトランザクシ

ンは、Q キャプチャー・プログラムが作業をコミットするまでメモリー内に留まります。Q キャプチャー・プログラムは、コミット・インターバルに達すると作業をコミットします。

他の目的のためのメモリー

Q キャプチャー・プログラムがログ・レコードを読み取る場合は、Q キャプチャー・プログラムはメモリー・バッファーを使用します。Linux、UNIX[®] および Windows[®] オペレーティング・システムの場合は、バッファーのデフォルトのサイズは 200 1-KB (K バイト) ページです。z/OS[™] オペレーティング・システムの場合は、デフォルトのサイズは 66 1-KB ページで、メモリーは ECSA (拡張共通サービス域) ストレージです。

Q キャプチャー・プログラムによって使用されるメモリーのほとんどは、トランザクションを再構成するために使用されますが、そのアクティビティーに必要なメモリー量は、以下の要因によって決まります。

コミット・インターバル

Q キャプチャー・プログラムは、DB2 UDB によってコミットされたトランザクションを、Q キャプチャー・プログラムが送信キューにコミットできるようにするまでメモリーに保管します。 **commit_interval** パラメーターを使用して、Q キャプチャー・プログラムがトランザクションを送信キューにコミットする頻度を指定します。通常は、**commit_interval** の値を小さくすると、Q キャプチャー・プログラムが使用するメモリーは少なくなり、Q キャプチャー・プログラムによってトランザクションが頻繁にコミットされるため、トランザクションが予備ファイルに書き込まれる可能性は低くなります。 **commit_interval** の値を大きくすると、Q キャプチャー・プログラムがトランザクションをコミットする頻度は低くなります。

複製または発行されるトランザクションのサイズ

Q キャプチャー・プログラムが必要とするメモリー量は、トランザクションの有効範囲と更新されるレコードのサイズに比例します。トランザクションの規模が非常に大きく、Q キャプチャー・プログラムでメモリー不足が発生した場合、Q キャプチャー・プログラムはトランザクションを予備ファイルに書き込みます。コミットの頻度が低い長期実行トランザクションは、Q レプリケーションのメモリーおよびストレージ要件に影響を与える可能性があります。そのようなトランザクションでは、コミットの頻度をさらに高くすることを考慮してください。

同時実行されるトランザクションの数

Q キャプチャー・プログラムが同時に処理するトランザクションの数が多い場合、または Q キャプチャー・プログラムがインターリーブド・トランザクションを処理する場合は、Q キャプチャー・プログラムはより多くの情報をメモリーまたはディスクに保管する必要があります。

Q キャプチャー・プログラムがトランザクションをバッファーに入れるために使用するメモリー量を制御するには、Q キャプチャー・プログラムの開始時または実行中に、**memory_limit** パラメーターを設定します。デフォルトでは、このパラメーターは 32 MB に設定されています。 **memory_limit** の値を大きくすればするほど、Q キャプチャー・プログラムがトランザクションを予備ファイルに書き込む可能性は低くなります。

表1を参考にして、各Qキャプチャー・プログラムのメモリー限度を判別してください。これらの推定値をガイドとして使用することにより、ワークロード用の十分なメモリーをQキャプチャー・プログラムで確保でき、プログラムがトランザクションを予備ファイルに書き込まないようにすることができます。この表では、次の公式を使用して、**commit_interval**ごとのトランザクションの数を計算しています。

$$\text{txr}/1000 * \text{ci} = \text{txci}$$

ここで、

txr トランザクション率 (1秒当たりのトランザクションの数)

ci Commit_interval

txci **commit_interval** 当たりのトランザクションの数

commit_interval 当たりのトランザクションの数は、次の公式で使用してメモリー限度を算出します。

$$\text{txci} * \text{mtxs} = \text{ml}$$

ここで、

txci **commit_interval** 当たりのトランザクションの数

mtxs トランザクションの最大サイズ

ml バイト単位のメモリー限度

表1. Qキャプチャー・プログラムの概算メモリー所要量

トランザクション率 (txr)	トランザクションの最大サイズ (mtxs)	ミリ秒単位の commit_interval (ci)	commit_interval 当たりのトランザクションの数 (txci)	バイト単位の memory_limit (ml)
730	1 340	1 000	730	978 200
1 033	12 000	3 000	3 099	37 188 000
5 000	2 340	500	2 500	5 850 000
12 000	7 000	5000	60 000	420 000 000

推奨: Qキャプチャー・プログラムにはメモリー割り当て量 (z/OSの場合は領域サイズ)を設定しないことをお勧めします。しかし、設定する必要がある場合は、29ページの表2を参考にしてください。この表は、次の一般公式を使用してメモリー割り当て量を導き出す方法の例を示しています。

$$\text{ml} + (\text{mas} * 1000) + 2000000 = \text{q}$$

ここで、

ml バイト単位のメモリー限度

mas アクティブ・サブスクリプションの最大数

q バイト単位のメモリー割り当て量

表 2. Q キャプチャー・プログラムの概算メモリー割り当て量

バイト単位のメモリー限度 (m1)	アクティブ・サブスクリプション の最大数 (mas)	バイト単位のメモリー 割り当て量 (q)
37 188 000	500	39 688 000
420 000 000	300	422 300 000

データの複製または発行を開始した後で、以下のいずれかの方法を使用して、Q キャプチャー・プログラムがトランザクションに使用しているメモリー量をチェックし、この情報を使用して **memory_limit** 値を調整してください。

- レプリケーション・センターで、「Q キャプチャー・スループット」ウィンドウの「メモリー使用量」オプションを選択する。このウィンドウをオープンするには、チェックしたい Q キャプチャー・プログラムが含まれている Q キャプチャー・サーバーを右マウス・ボタンでクリックして、「レポート」→「Q キャプチャー・スループット」を選択します。詳細については、オンライン・ヘルプを参照してください。
- トランザクションをログから再構成するために Q キャプチャー・プログラムが使用しているメモリー量を確認するには、IBMQREP_CAPMON 表の CURRENT_MEMORY 列の値をチェックします。

また、Q キャプチャー・プログラムがトランザクションを予備ファイルに書き込んでいるかどうかを確認するには、IBMQREP_CAPMON 表の TRANS_SPILLED 列の値をチェックします。

Q キャプチャー・プログラムに十分なメモリーがない場合は、Q キャプチャー・プログラムは予備ファイルに書き込もうとします。ストレージが十分にない場合は、Q キャプチャー・プログラムは終了します。

関連概念:

- 248 ページの『Q キャプチャー・パラメーターの変更 -- 概要』
- 29 ページの『Q アプライ・プログラムによって使用されるメモリー』

関連タスク:

- 25 ページの『Q レプリケーションおよびイベント発行のメモリー要件の計画』
- 32 ページの『Q レプリケーションおよびイベント発行のストレージ要件の計画』

Q アプライ・プログラムによって使用されるメモリー

Q アプライ・プログラムは、ターゲットについての情報を保管したり、受信キューから検索したソースのコミットされたトランザクションを保管したりするために、メモリーを使用します。

アクティブ・ターゲットについての情報を保管するためのメモリー

Q アプライ・プログラムは、ソース・データの複製先にしたいターゲットなどの、アクティブ・ターゲットについての情報を保管するためにメモリーを使用します。プログラムを開始するとき、または Q アプライ・プログラムの実行中に Q サブスクリプションをアクティブ化するとき、Q アプ

イ・プログラムはこのデータを検索して保管します。各 Q サブスクリプションは、最大で 1000 バイトのメモリーを使用します。

ソース・トランザクションをターゲットにアプライするためのメモリー

Q アプライ・プログラムは、受信キューから検索したソース・トランザクションを保管するためにメモリーを使用し、Q アプライ・プログラムは、トランザクションをターゲットにアプライしている間は、ソース・データをメモリーに保持します。Q アプライ・プログラムが使用するメモリーの量は、受信するトランザクションのサイズ、受信するトランザクションの数、およびトランザクション間のトランザクション従属関係に比例します。Q アプライ・プログラムが使用するメモリーの量は、Q アプライ・プログラムがトランザクションをターゲットにアプライする率に反比例します。

Q アプライ・プログラムによって使用されるメモリーのほとんどは、トランザクションをターゲットにアプライするために使用されますが、そのアクティビティーに必要なメモリー量は、トランザクションのサイズによって決まります。Q アプライ・プログラムが必要とするメモリー量は、トランザクションの範囲と更新されるレコードのサイズに比例します。トランザクションの規模が非常に大きく、Q アプライ・プログラムでメモリー不足が発生する場合は、Q アプライ・プログラムはトランザクションをメモリーにキャッシングしなくなります。並列処理は制限されるため、処理中のトランザクションのサイズによっては、Q アプライ・プログラムがトランザクションを順次直列的に処理しなければならない場合があります。コミットの頻度が低い長期実行トランザクションは、Q レプリケーションのメモリーおよびストレージ要件に影響を与える可能性があります。そのようなトランザクションでは、コミットの頻度をさらに高くすることを考慮してください。

Q アプライ・プログラムの受信キューごとに **memory_limit** パラメーターを設定できます。**memory_limit** 値は、Q アプライ・プログラムの開始時または実行中に設定します。この値は、Q アプライ・プログラムが特定の受信キューから受信するトランザクションをターゲットにアプライするまで、Q アプライ・プログラムがそれらのトランザクションをバッファに入れるために使用できるメモリーの量を制御します。

表 3 を参考にして、Q アプライ・プログラムの受信キューごとに必要な **memory_limit** 値を決定してください。すべてのエージェントがトランザクションを処理するようにメモリー限度を設定するのが最適です。この表では、次の一般公式を使用しています。

$$(mtxs + 44000) * (a + 1) = ml$$

ここで、

mtxs トランザクションの最大サイズ (バイト単位)

a エージェントの数

ml 受信キューのメモリー限度 (バイト単位)

表 3. 受信キューごとの Q アプライ・プログラムの概算メモリー所要量

トランザクションの最大サイズ (mtxs)	エージェントの数 (a)	Memory_limit (ml)
1 340	128	5 848 860

表3. 受信キューごとの Q アプライ・プログラムの概算メモリー所要量 (続き)

トランザクションの 最大サイズ (mtxs)	エージェントの数 (a)	Memory_limit (m1)
12 000	16	952 000

すべてのエージェントを使用中にできるようにメモリー限度を設定してください。エージェントの数を増やす場合は、メモリー限度をチェックして、すべてのエージェントを使用中としておくために必要な追加のトランザクションを入れる余裕があることを確認してください。

推奨: Q アプライ・プログラムにはメモリー割り当て量を設定しないことをお勧めします。しかし、設定する必要がある場合は、受信キューごとに適切なメモリー限度を判別した後で、表4を参考にしてください。この表では、次の一般公式を使用しています。

$$sm1 + (mas * 1000) + 500000 = q$$

sm1 メモリー限度値の合計

mas アクティブ・サブスクリプションの最大数

q バイト単位のメモリー割り当て量

表4. Q アプライ・プログラムの概算メモリー割り当て量

すべての受信キューの memory_limit 値の合計 (sm1)	アクティブ・ サブスクリプションの 最大数 (mas)	メモリー割り当て量 (q)
5 848 860 + 952 000 = 6 800 860	100	7 400 860

データの複製または発行を開始した後で、各 Q アプライ・ブラウザーが使用しているメモリー量を確認するには、IBMQREP_APPLYMON 表の CURRENT_MEMORY 列の値をチェックします。この情報を使用して、Q アプライ・プログラム・キューごとの **memory_limit** 値を調整します。

Q アプライ・プログラムでメモリーが不足し、ストレージが十分でない場合は、診断情報を保管できません。Q アプライ・プログラムは稼働し続けます。

関連概念:

- 271 ページの『Q アプライ・パラメーターの変更 -- 概要』
- 26 ページの『Q キャプチャー・プログラムによって使用されるメモリー』
- 19 ページの『Q アプライ・プログラム』

関連タスク:

- 25 ページの『Q レプリケーションおよびイベント発行のメモリー要件の計画』
- 32 ページの『Q レプリケーションおよびイベント発行のストレージ要件の計画』

レプリケーション・アラート・モニターによって使用されるメモリ

レプリケーション・アラート・モニターによって使用されるメモリは、ほとんどありません。

推奨: レプリケーション・アラート・モニターにはメモリ割り当て量を設定しないことをお勧めします。ただし、設定する必要がある場合は、3 MB に設定してください。

関連タスク:

- 25 ページの『Q レプリケーションおよびイベント発行のメモリ要件の計画』

Q レプリケーションおよびイベント発行の LOB データ・タイプのためのメモリ

Q キャプチャー・プログラムは、ラージ・オブジェクト (LOB) データ・タイプを複製または発行するときに使用するメモリの量を管理します。LOB 値のサイズが Q キャプチャー・プログラムの LOB メッセージ・バッファー・サイズを超える場合は、LOB メッセージは複数の小さいメッセージに分割されます。LOB メッセージの最大バッファー・サイズは、送信キューに定義されたメッセージの最大長です。(MAX_MESSAGE_SIZE 属性の値は、キュー・マネージャーに定義された WebSphere® MQ のメッセージの最大長 (MAXMSGL) 属性より大きい値ではありません。)

たとえば、WebSphere MQ のメッセージの最大長 (MAXMSGL) が 100 MB 以上であるとし、MAX_MESSAGE_SIZE を 20 MB に設定すると、100 MB の各メッセージは、長さが 20 MB の 5 つのメッセージに分割されます。

多くの LOB 値または BLOB 値を複製または発行する場合は、十分なメモリとストレージを割り振り、適切なキュー項目数を設定します。

関連概念:

- 72 ページの『ラージ・オブジェクト (LOB) 値のキュー項目数に関する考慮事項』
- 198 ページの『Q レプリケーションおよびイベント発行でのラージ・オブジェクト (LOB) データ・タイプに関する考慮事項』
- 68 ページの『WebSphere MQ メッセージ・サイズ』

関連タスク:

- 25 ページの『Q レプリケーションおよびイベント発行のメモリ要件の計画』

Q レプリケーションおよびイベント発行のストレージ要件の計画

Q レプリケーションおよびイベント発行のストレージ要件の計画

ソース変更の複製または発行を始める前に、Q レプリケーションおよびイベント発行に必要なストレージを計画する必要があります。

手順:

ストレージ要件を判別するには、以下の事柄を計画する必要があります。

- 『Q レプリケーションおよびイベント発行の DB2 UDB ログのストレージ要件』
- 34 ページの『Q レプリケーションおよびイベント発行の診断ファイルのストレージ』
- 34 ページの『Q キャプチャー・プログラムがメモリー限度を超える場合のストレージ要件』
- 35 ページの『Q レプリケーションおよびイベント発行のトレースのストレージ要件』

関連概念:

- 69 ページの『Q レプリケーションおよびイベント発行の WebSphere MQ のストレージ要件』

関連タスク:

- 25 ページの『Q レプリケーションおよびイベント発行のメモリー要件の計画』

Q レプリケーションおよびイベント発行の DB2 UDB ログのストレージ要件

Q レプリケーションでは、ソース・サーバーおよびターゲット・サーバーで発生する変更の DB2[®] UDB ログのためのストレージが必要です。

ソース・サーバーでの DB2 UDB ログ要件

ソース・サーバーでは、ソース表で発生する変更のために十分なログ・スペースを割り振る必要があります。各ソース表で DATA CAPTURE CHANGES キーワードがオンになっているため、DB2 UDB は、UPDATE ステートメントごとに行全体のイメージのレコードを保持します。

また、コントロール表のための少量の追加スペースが必要です。SQL レプリケーション (キャプチャーされたデータをステージング表にステージングする) とは異なり、Q レプリケーション・ログ要件は、Q キャプチャー・プログラムが作動するために必要なコントロール表で発生する変更に必要なスペースだけに制限されます。Q キャプチャー・プログラムが作動するために必要なストレージの量はごくわずかです。

ターゲット・サーバーでの DB2 UDB ログ要件

ターゲット・サーバーでは、ターゲット表で発生する変更のために十分なログ・スペースを割り振る必要があります。また、トランザクションおよび統計をトラッキングするには、Q アプライ・プログラムのコントロール表で発生する変更のための追加スペースも必要です。この情報に必要なストレージの量はごくわずかです。

関連タスク:

- 32 ページの『Q レプリケーションおよびイベント発行のストレージ要件の計画』

Q レプリケーションおよびイベント発行の診断ファイルのストレージ

Q キャプチャー・プログラム、Q アプライ・プログラム、およびレプリケーション・アラート・モニターは、診断ログ・ファイルを使用して、プログラムのアクティビティに関する情報（プログラムの開始時刻と停止時刻、およびその他の情報メッセージまたはエラー・メッセージなど）を保管します。デフォルトでは、各プログラムは再始動後もログ・ファイルにメッセージを付加します。

これらのログ・ファイルを含むディレクトリーにはファイルを保管するのに十分なスペースが確実にあるようにしてください。これらのファイルのロケーションは、Q キャプチャー・プログラム、Q アプライ・プログラム、およびレプリケーション・アラート・モニターのそれぞれの開始時にパラメーター **capture_path**、**apply_path**、および **monitor_path** に設定された値によって決まります。

ストレージに関して不安がある場合、プログラム・ログに付加するのではなく、プログラム・ログを再利用するオプションがあります。 **logreuse** パラメーターを使用して、プログラムがログを再利用することを指定できます。こうすると、プログラムが開始するたびに、ログが削除されて再作成されます。

関連概念:

- 248 ページの『Q キャプチャー・パラメーターの変更 -- 概要』

関連タスク:

- 32 ページの『Q レプリケーションおよびイベント発行のストレージ要件の計画』

Q キャプチャー・プログラムがメモリー限度を超える場合のストレージ要件

Q キャプチャー・プログラムがメモリー限度を超えると、Q キャプチャー・プログラムはデータを一時的に保管するために予備ファイルを使用します。これらの予備ファイルのストレージ要件を計画する必要があります。各トランザクションには、それぞれの予備ファイルがあります。Q キャプチャー・プログラムは、最大のトランザクションをファイルに書き込みます。しかし、最大のトランザクションが、メモリー限度を超えたトランザクションであるとは限りません。

Q キャプチャー・プログラムの予備ファイルのサイズは、以下の要因によって決まります。

メモリー限度

Q キャプチャー・プログラムが使用できるメモリーが多ければ多いほど、予備ファイルに書き出される可能性は低くなります。 **memory_limit** パラメーターを使用して、Q キャプチャー・プログラムがトランザクションをバッファーに入れるために使用できるメモリー量を指定します。

複製または発行されるトランザクションのサイズ

Q キャプチャー・プログラムが必要とするメモリー量は、トランザクションの有効範囲と更新されるレコードのサイズに比例します。トランザクションの規模が非常に大きく、Q キャプチャー・プログラムでメモリー不足が

発生した場合、Q キャプチャー・プログラムはトランザクションを予備ファイルに書き込みます。コミットの頻度が低い長期実行トランザクションは、Q レプリケーションのメモリーおよびストレージ要件を増大させる可能性があります。そのようなトランザクションでは、コミットの頻度をさらに高くすることを考慮してください。

同時実行されるトランザクションの数

Q キャプチャー・プログラムが同時に処理するトランザクションの数が多い場合、または Q キャプチャー・プログラムがインターリーブド・トランザクションを処理する場合は、Q キャプチャー・プログラムは複数のトランザクションを予備ファイルに書き出す必要があります。

コミット・インターバル

commit_interval パラメーターを使用して、Q キャプチャー・プログラムがトランザクションを送信キューにコミットする頻度を指定します。Q キャプチャー・プログラムは、DB2[®] UDB によってコミットされたトランザクションを、Q キャプチャー・プログラムが送信キューにコミットできるようになるまでメモリーに保管します。**commit_interval** パラメーターを使用して、Q キャプチャー・プログラムがこのようなトランザクションをコミットする頻度を制御できます。通常、**commit_interval** の値を小さくすると、Q キャプチャー・プログラムが使用するメモリーは少なくなり、Q キャプチャー・プログラムによってトランザクションが頻繁にコミットされるため、トランザクションが予備ファイルに書き込まれる可能性は低くなります。**commit_interval** の値を大きくすると、Q キャプチャー・プログラムがトランザクションをコミットする頻度は低くなります。

Linux、UNIX[®]、Windows[®]: 予備ファイルは常にディスク上にあります。1 トランザクション当たり 1 つのファイルが **capture_path** ディレクトリーに作成されます。

z/OS[™]: CAPSPILL DD カードが指定されていない場合は、予備ファイルはデフォルトで VIO に書き込まれます。UNIT=SYSDA または UNIT=VIO を指定することによって、予備ファイルをそれぞれ DASD または VIO に書き出すことができます。

関連概念:

- 26 ページの『Q キャプチャー・プログラムによって使用されるメモリー』

関連タスク:

- 32 ページの『Q レプリケーションおよびイベント発行のストレージ要件の計画』

Q レプリケーションおよびイベント発行のトレースのストレージ要件

トレース機能が使用するログ・ファイルのストレージ要件を計画する必要があります。トレース機能は、Q キャプチャー・プログラム、Q アプライ・プログラム、およびレプリケーション・アラート・モニターのフローをログに記録します。このログ情報を IBM[®] ソフトウェア・サポートに提供して、トラブルシューティングに役立てることができます。

トレース・ファイルのサイズを見積もる場合は、考慮すべき以下のいくつかの要因があります。

- トレースの実行時間
- トレースに指定した詳細レベル
- トレース中に発生するトランザクションのボリューム

関連タスク:

- 32 ページの『Q レプリケーションおよびイベント発行のストレージ要件の計画』

関連資料:

- 400 ページの『asntrc: レプリケーション・トレース機能の操作』

第 5 章 データ変換

Q レプリケーションおよびイベント発行のデータ変換 -- 概要

Q レプリケーションおよびイベント発行では、データはさまざまなサーバーやプログラム間を受け渡しされ、データの変換が必要になることがあります。以下のトピックでは、Q レプリケーションおよびイベント発行がデータ変換を処理する方法を説明します。

- 38 ページの『イベント発行のデータ変換』
- 38 ページの『イベント発行のデータ変換』

Q レプリケーションのデータ変換

Q レプリケーションでは、データはさまざまなサーバーやプログラム間を受け渡しされ、データの変換が必要になることがあります。たとえば、プログラムに異なるコード・ページがある場合、またはプラットフォームやプロセッサが数値を異なる方法で処理する場合などです。Q レプリケーションでは、ソース・サーバーとターゲット・サーバーが異なるコード・ページにある場合でも、データは必要に応じて列レベルで変換されます。エンディアンおよび浮動小数点表記変換が処理されません。

推奨: 可能なら、以下のものについては、一致するコード・ページを使用することによってデータ変換を避けてください。

- Q キャプチャー・プログラム
- Q アプライ・プログラム
- ソース・サーバー
- ターゲット・サーバー

ソース・サーバーおよびターゲット・サーバーが異なるコード・ページを使用する必要がある場合、Q キャプチャー・プログラムとソース・サーバーに一致するコード・ページを使用してから、Q アプライ・プログラムとターゲット・サーバーに一致するコード・ページを使用してください。

Q キャプチャーおよび Q アプライ・プログラムが管理メッセージを受け渡しすると、これらのメッセージのデータ・フォーマットは変換が必要な場合があります。Q サブスクリプションを一緒に処理する Q キャプチャーおよび Q アプライ・プログラムが、互換性のあるコード・ページを持つことを確認する必要があります。

ソース表からのデータをターゲット表またはストアド・プロシージャに複製すると、Q キャプチャー・プログラムはソース表のフォーマットで WebSphere® MQ を介してデータを送信します。必要であれば、ターゲットでソース・データが変換されます。

Q アプライ・プログラムは、受信キューからソース・データを受け取ると、トランザクション中の各列、およびメッセージ中の他のすべてのデータを独自のコード・

ページに変換します。ターゲット・サーバーは、データが Q アプライ・プログラムのコード・ページ、および浮動小数点表記であることを予期しています。

推奨 (z/OS)[™]: zSeries[®] は、浮動小数点値を、Intel ベースの CPU とは異なる方法で表現するので、それによってデータの一部が失われる可能性があります。浮動小数点列をキーとして使用することは避けてください。

制限: Q キャプチャーおよび Q アプライ・プログラムのコード・ページは、UTF-16 にすることはできません。

異なるコード・ページを使用するサーバー間でデータを複製する予定である場合は、「管理ガイド: プランニング」をチェックして、使用するコード・ページに互換性があるかどうかを判別してください。

関連概念:

- 38 ページの『イベント発行のデータ変換』

関連資料:

- 「管理ガイド: プランニング」の『サポートされているテリトリー・コードおよびコード・ページ』

イベント発行のデータ変換

イベント発行では、データはソース・サーバーのコード・ページから、標準 XML エンコード・スキームである UTF-8 (コード・ページ 1208) として XML メッセージに変換されます。

ユーザー・アプリケーションが管理メッセージ (たとえば、サブスクリプションの非アクティブにされたメッセージ) を XML 形式で Q キャプチャー・プログラムに送信する場合、XML パーサーは、メッセージを Q キャプチャー・プログラムのコード・ページに変換します。

関連概念:

- 37 ページの『Q レプリケーションのデータ変換』

第 6 章 ユーザー ID およびパスワードのセットアップ

ユーザー ID およびパスワードのセットアップ -- 概要

Q レプリケーションおよびイベント発行プログラムを使用するには、分散システム上で DB2[®] UDB サーバーにアクセスするためのユーザー ID およびパスワードをセットアップする必要があります。

以下のトピックでは、必要な権限と特権、およびレプリケーションおよび発行プログラムが共用できる、暗号化されたパスワード・ファイルに必要なユーザー ID とパスワードを保管する方法を説明します。

- 『Q レプリケーションおよびイベント発行の許可要件 -- 概要』
- 43 ページの『リモート・サーバー用のユーザー ID およびパスワードの保管』

Q レプリケーションおよびイベント発行の許可要件

Q レプリケーションおよびイベント発行の許可要件 -- 概要

Q レプリケーションおよびイベント発行プログラムを実行するユーザー ID には、サーバーへの接続、テーブルへのアクセスまたは更新、および他の操作を実行するための権限が必要です。権限および特権は、次のトピックで説明されています。

- 『Q キャプチャー・プログラムの許可要件』
- 40 ページの『Q アプライ・プログラムの許可要件』
- 42 ページの『レプリケーション・アラート・モニターの許可要件』
- 42 ページの『Q レプリケーションおよびイベント発行を管理するための許可要件』

関連概念:

- 39 ページの『ユーザー ID およびパスワードのセットアップ -- 概要』

Q キャプチャー・プログラムの許可要件

Q キャプチャー・プログラムを実行するすべてのユーザー ID には、DB2[®] システム・カタログへのアクセス、すべての Q キャプチャー・コントロール表へのアクセスおよび更新、DB2 ログの読み取り、および Q キャプチャー・プログラム・パッケージの実行のための権限が必要です。以下のリストは、DB2 UDB 要件と、プラットフォーム別のオペレーティング・システム要件を要約しています。

Linux、UNIX[®]、および Windows[®] の場合の要件

Q キャプチャー・プログラムを実行するユーザー ID は、以下の権限および特権を持っている必要があります。

- DBADM または SYSADM 権限。

- **capture_path** パラメーターで指定されているディレクトリーに対する WRITE 特権。 Q キャプチャー・プログラムは、このディレクトリーで診断ファイルを作成します。

超並列処理 (MPP) 構成では、ユーザー ID はデータベース・パーティションへの接続、およびパスワード・ファイルの読み取りが可能でなければなりません。

z/OS™ の場合の要件

Q キャプチャー・プログラムを実行するユーザー ID には、以下を満たしている必要があります。

- UNIX システム・サービス (USS) へのアクセスに登録されている。
- APF 許可がある。
- z/OS UNIX または OS/390® UNIX (OMVS セグメントを持っていない) を使用するように定義されている。 OMVS セグメントとは、OMVS ログオン情報を含む RACF® プロファイルの一部分です。
- /tmp ディレクトリーか、TMPDIR 環境変数によって指定されたディレクトリーのいずれかに、読み取りおよび書き込みアクセス権がある。
- SYSADM 権限、または以下の特定の特権がある。
 - Q キャプチャー・サーバー上のすべてのコントロール表に対する SELECT、UPDATE、INSERT、および DELETE 特権。
 - DB2 カタログに対する SELECT (SYSIBM.SYSTABLES、SYSIBM.SYSCOLUMNS、SYSIBM.SYSTABCONST、SYSIBM.SYSKEYCOLUSE、SYSIBM.SYSINDEXES、および SYSIBM.SYSKEYS)
 - TRACE
 - MONITOR1 および MONITOR2
 - Q キャプチャー・プログラム・パッケージに対する EXECUTE
- **capture_path** パラメーター (USS) または上位修飾子 (z/OS) で指定されているディレクトリーに対する WRITE 特権がある。

関連概念:

- 39 ページの『Q レプリケーションおよびイベント発行の許可要件 -- 概要』

関連タスク:

- 43 ページの『リモート・サーバー用のユーザー ID およびパスワードの保管』

関連資料:

- 235 ページの『Q キャプチャー・パラメーターの説明』

Q アプライ・プログラムの許可要件

Q アプライ・プログラムを実行するすべてのユーザー ID には、DB2® システム・カタログへのアクセス、ターゲットへのアクセスおよび更新、Q アプライ・コントロール表へのアクセスおよび更新、Q アプライ・プログラム・パッケージの実行、および Q アプライ・パスワード・ファイルの読み取りのための権限が必要です。以下のリストは、DB2 UDB 要件と、プラットフォーム別のオペレーティング・システム要件を要約しています。

Linux、UNIX[®]、Windows[®] の場合の要件

Q アプライ・プログラムを実行するユーザー ID は、以下の権限および特権を持っている必要があります。

- DBADM または SYSADM 権限。
- Q アプライ・プログラムがターゲット表のロードに使用される場合、ソース表の SELECT 特権。
- **apply_path** パラメーターで指定されているディレクトリーに対する WRITE 特権。 Q アプライ・プログラムは、このディレクトリーで診断ファイルを作成します。

Q アプライ・プログラムが LOAD ユーティリティーの CURSOR からの LOAD オプションを使用してターゲット表をロードする場合、Q アプライ・サーバーはフェデレーテッド・サーバーでなければならず、Q アプライ・サーバー上にニックネーム、サーバー定義、およびユーザー・マッピングを作成する必要があります。ユーザー・マッピングで提供されるユーザー ID には、フェデレーテッド Q アプライ・サーバー上でニックネームから読み取る特権、およびソース表から読み取る特権がなければなりません。

z/OS[™] の場合の要件

Q アプライ・プログラムを実行するユーザー ID には、以下が当てはまる必要があります。

- UNIX システム・サービス (USS) へのアクセスに登録されている。
- Q アプライ・プログラムが自動再始動マネージャー (ARM) を使用するか、または SVCDUMP を作成するための APF 許可がある。
- z/OS UNIX または OS/390[®] UNIX (OMVS セグメントを持っていない) を使用するように定義されている。 OMVS セグメントとは、OMVS ログオン情報を含む RACF[®] プロファイルの一部分です。
- /tmp ディレクトリーか、TMPDIR 環境変数によって指定されたディレクトリーのいずれかに、読み取りおよび書き込みアクセス権がある。
- SYSADM 権限、または以下の特定の特権がある。
 - Q アプライ・サーバー上のすべてのコントロール表に対する SELECT、UPDATE、INSERT、および DELETE 特権。
 - Q アプライ・プログラムがターゲット表のロードに使用される場合、ソース表の SELECT。
 - DB2 カタログの SELECT (SYSIBM.SYSRELS、SYSIBM.SYSTABLES および SYSIBM.SYSDUMMY1)
 - Q アプライ・プログラム・パッケージに対する EXECUTE
- **apply_path** パラメーター (USS) または上位修飾子 (z/OS) で指定されているディレクトリーに対する WRITE 特権がある。

関連概念:

- 39 ページの『Q レプリケーションおよびイベント発行の許可要件 -- 概要』
- 165 ページの『Q レプリケーションの自動ロード・オプションで使用するユーティリティー』

関連資料:

- 263 ページの『Q アプライ・パラメーターの説明』

レプリケーション・アラート・モニターの許可要件

レプリケーション・アラート・モニターを実行するすべてのユーザー ID には、モニターしたい Q キャプチャー・サーバーまたは Q アプライ・サーバーにアクセスする権限が必要です。またユーザー ID には、モニター・コントロール・サーバー上のモニター・コントロール表へのアクセス権も必要です。

モニターを実行するユーザー ID は、以下の権限および特権を持っている必要があります。

- モニター CONTROL 表に対する SELECT、UPDATE、INSERT、および DELETE 特権
- モニターするサーバー上の Q キャプチャー・コントロール表および Q アプライ・コントロール表に対する SELECT 権限
- BINDADD 権限 (モニター・パッケージに対して自動バインド機能を使用する場合にのみ必要)
- モニター・プログラム・パッケージに対する EXECUTE 特権
- レプリケーション・アラート・モニターが診断ファイルを保管する **monitor_path** ディレクトリーに対する WRITE 特権
- レプリケーション・アラート・モニターによって使用されるパスワード・ファイルへの読み取りアクセス (Linux、UNIX[®]、および Windows[®] のみ)

関連概念:

- 39 ページの『Q レプリケーションおよびイベント発行の許可要件 -- 概要』
- 291 ページの『レプリケーション・アラート・モニター』

Q レプリケーションおよびイベント発行を管理するための許可要件

Q レプリケーションおよびイベント発行を管理するため、管理タスクに従って、レプリケーション・センター、システム・コマンド、または SQL を含む、いくつかの管理インターフェースから選択することができます。

レプリケーション構成に関与するすべてのデータベースに対するユーザー ID を少なくとも 1 つは持っている必要があります、さらにそのユーザー ID には、Q キャプチャー・サーバー、Q アプライ・サーバー、およびモニター・コントロール・サーバーでさまざまな管理用タスクを実行する権限が必要です。これらのタスクには以下のもものが含まれます。

- コントロール表の作成およびアクセス
- DB2[®] システム・カタログへのアクセス
- プランおよびパッケージのバインディング
- ターゲット表および表スペース、Q サブスクリプションおよび XML 発行、および他のオブジェクトを作成するための生成済み SQL の実行

レプリケーション・センターを使用して管理を単純化するには、「パスワードと接続の管理 (Manage Passwords and Connectivity)」ウィンドウを使用して、サーバーまたはシステムのユーザー ID を保管し、保管した ID を変更し、接続をテストします。このウィンドウをオープンするには、「レプリケーション・センター」フォル

ダーを右マウス・ボタンでクリックし、「レプリケーション・センター用パスワードの管理 (Manage Passwords for Replication Center)」を選択します。詳細については、オンライン・ヘルプを参照してください。

Q レプリケーションおよびイベント発行を管理するユーザー ID は、以下の権限および特権を持っている必要があります。

- Q キャプチャー・サーバー、Q アプライ・サーバー、およびモニター・コントロール・サーバーの CONNECT 特権
- Q キャプチャー・サーバー、Q アプライ・サーバー、およびモニター・コントロール・サーバーでコントロール表を作成するために必要なすべての表、表スペース、および索引特権
- Q キャプチャー・サーバー、Q アプライ・サーバー、およびモニター・コントロール・サーバー上のすべてのコントロール表用の SELECT、UPDATE、INSERT および DELETE 特権
- DB2 システム・カタログ用の SELECT 特権
- Q アプライ・サーバーでターゲットを作成するために必要なすべての表、表スペース、および索引特権
- Q キャプチャー・サーバー、Q アプライ・サーバー、およびモニター・コントロール・サーバーを含めた、レプリケーションまたは発行に関係する各 DB2 データベースでプランをバインドする特権
- 共有ライブラリーを使用したストアド・プロシージャの作成およびストアド・プロシージャの呼び出しの特権 (Linux、UNIX[®]、および Windows[®] のみ)

関連概念:

- 81 ページの『オプション : プログラム・パッケージのバインド (Linux、UNIX、Windows) -- 概要』
- 39 ページの『Q レプリケーションおよびイベント発行の許可要件 -- 概要』

関連タスク:

- 87 ページの『Q キャプチャー・プログラムと Q アプライ・プログラムのコントロール表の作成』
- 199 ページの『レプリケーション・センターからの SQL スクリプトおよび操作可能コマンドの実行 -- 概要』

リモート・サーバー用のユーザー ID およびパスワードの保管

Linux、UNIX、および Windows オペレーティング・システムでは、以下の Q レプリケーションおよびイベント発行プログラムで、リモート・サーバーに接続するためのユーザー ID およびパスワードを保管するためのパスワード・ファイルを作成する必要があります。

- Q アプライ・プログラムでは、EXPORT ユーティリティーを使用してターゲットをロードする Q サブスクリプションに対する Q キャプチャー・サーバーに接続するためのパスワード・ファイルが必要です。
- DB2 UDB Enterprise Server Edition で実行する Q キャプチャー・プログラムには、複数のパーティション・データベースに接続するためのパスワード・ファイルが必要です。

- レプリケーション・アラート・モニターには、モニターしたい Q キャプチャー・サーバーまたは Q アプライ・サーバーに接続するためのパスワード・ファイルが必要です。
- Q レプリケーション・アナライザーには、分析したい Q キャプチャー・サーバーまたは Q アプライ・サーバーに接続するためのパスワード・ファイルが必要です。

手順:

リモート・サーバーに対するユーザー ID とパスワードを保管するには、**asnpwd** コマンドを使用して Linux、UNIX、および Windows で実行している Q レプリケーションおよびイベント発行プログラム用に暗号化パスワード・ファイルを作成します。パスワード・ファイルは、以下のパラメーターにより設定されるパスに保管される必要があります。

Q アプライ・プログラム (Q Apply program)

apply_path

Q キャプチャー・プログラム

capture_path

レプリケーション・アラート・モニター

monitor_path

Q レプリケーション・アナライザー

アナライザーは、プログラムが呼び出されたディレクトリーで、または **password_filepath** パラメーターが指定する場所からパスワード・ファイルを検索します。

Q アプライ・プログラム、レプリケーション・アラート・モニター、およびレプリケーション・アナライザーが同じシステム上で実行している場合、同じパスワード・ファイルを共用できます。3つのプログラムにパスワード・ファイルを共用させる場合は、3つのプログラムすべてに同じパスおよびファイル名を指定するか、またはシンボリック・リンクを使って異なるディレクトリーにある同じパスワード・ファイルを共用させます。

注: レプリケーション・センターは、**asnpwd** コマンドで作成されるパスワード・ファイルを使用してリモート・サーバーに接続することはありません。初めてレプリケーション・センターがデータベースまたはサブシステムにアクセスする必要があると、ユーザー ID およびパスワードの入力を求めるプロンプトが出されます。これは、将来の使用のために保管されます。「パスワードと接続の管理 (Manage Passwords and Connectivity)」ウィンドウを使用して、サーバーまたはシステムのユーザー ID を保管し、保管した ID を変更し、接続をテストすることができます。このウィンドウをオープンするには、「レプリケーション・センター」フォルダーを右マウス・ボタンでクリックし、「レプリケーション・センター用パスワードの管理 (Manage Passwords for Replication Center)」を選択します。詳細については、オンライン・ヘルプを参照してください。

関連概念:

- 39 ページの『ユーザー ID およびパスワードのセットアップ -- 概要』
- 165 ページの『Q レプリケーションの自動ロード・オプションで使用するユーティリティー』

関連資料:

- 386 ページの『asnqanalyze: Q レプリケーション・アナライザーの操作』

第 7 章 Q レプリケーションおよびイベント発行用の WebSphere MQ をセットアップする方法

Q レプリケーションおよびイベント発行用の WebSphere MQ のセットアップ -- 概要

Q レプリケーションおよびイベント発行は WebSphere® MQ (以前の MQ Series) を使用して、トランザクション・データを送信し、他のメッセージを交換します。

このセクションでは、Q キャプチャー・プログラムおよび Q アプライ・プログラムの操作に必要な WebSphere MQ オブジェクトについて説明し、それぞれのレプリケーションまたは発行の構成に必要なオブジェクトのチェックリストを提供し、キュー、キュー・マネージャー、およびチャネル・オブジェクトに必要な設定について説明します。また、WebSphere MQ のメモリー、ストレージ、および許可に関する要件についても説明します。

このセクションは、次のトピックで構成されます。

- 『Q レプリケーションおよびイベント発行に必要な WebSphere MQ オブジェクト -- 概要』
- 68 ページの『WebSphere MQ メッセージ・サイズ』
- 69 ページの『Q レプリケーションおよびイベント発行の WebSphere MQ のストレージ要件』
- 70 ページの『Q アプライ・プログラムの予備キューのストレージ要件』
- 70 ページの『WebSphere MQ オブジェクトの接続および許可に関する要件』
- 62 ページの『WebSphere MQ オブジェクトの設定要件』
- 72 ページの『ラージ・オブジェクト (LOB) 値のキュー項目数に関する考慮事項』
- 73 ページの『Q レプリケーションおよびイベント発行でのキュー・マネージャーのクラスタリング』

Q レプリケーションおよびイベント発行に必要な WebSphere MQ オブジェクト

Q レプリケーションおよびイベント発行に必要な WebSphere MQ オブジェクト -- 概要

実行するレプリケーションまたは発行のタイプに応じて、さまざまな WebSphere® MQ オブジェクトが必要になります。

以下のトピックでは、Q レプリケーションおよびイベント発行で使用される WebSphere MQ オブジェクトについて説明します。また、さまざまなレプリケーシ

ョンおよび発行のシナリオで必要なオブジェクトのチェックリストも含まれています。それにはオブジェクトの設定要件が記載されています。

- 『必須 WebSphere MQ オブジェクトの概観』
- 50 ページの『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』
- 52 ページの『単一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』
- 53 ページの『イベント発行に必要な WebSphere MQ オブジェクト』
- 55 ページの『双方向またはピアツーピア・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)』
- 58 ページの『ピアツーピア・レプリケーション (3 つ以上のリモート・サーバー) で必要な WebSphere MQ オブジェクト』

WebSphere MQ オブジェクトの作成に関する詳細については、ご使用のプラットフォームの「*WebSphere MQ* システム管理ガイド」、*「WebSphere MQ 相互通信」*、および「*WebSphere MQ Script (MQSC)* コマンド・リファレンス」を参照してください。

関連概念:

- 47 ページの『Q レプリケーションおよびイベント発行用の WebSphere MQ のセットアップ -- 概要』

必須 WebSphere MQ オブジェクトの概観

以下に、Q キャプチャー・プログラムおよび Q アプライ・プログラムに必要な WebSphere® MQ オブジェクトのクイック・サマリーと、それらのプログラムに対してオブジェクトを定義する方法の要約を示します。

キュー・マネージャー

Q キャプチャー・プログラム、Q アプライ・プログラム、およびユーザー・アプリケーションのキューを管理するプログラム。各システムに 1 つのキュー・マネージャーが必要です。

送信キュー

Q キャプチャー・プログラムから Q アプライ・プログラムまたはユーザー・アプリケーションにデータ・メッセージを送信するキュー。リモート構成では、これはターゲット・システム上の受信キューのソース・システムのローカル定義です。各送信キューは、1 つの Q キャプチャー・プログラムのみによって使用されます。

受信キュー

Q キャプチャー・プログラムから Q アプライ・プログラムまたはユーザー・アプリケーションへのデータおよび通知メッセージを受信するキュー。これは、ターゲット・システム上のローカル・キューです。

管理キュー (Q キャプチャー用)

Q アプライ・プログラムまたはユーザー・アプリケーションから Q キャプチャー・プログラムへの制御メッセージを受信するキュー。これは、ソース・システム上のローカル・キューです。各管理キューは、1 つの Q キャプチャー・プログラムのみによって読み取られます。

管理キュー (Q アプライ用)

Q アプライ・プログラムまたはユーザー・アプリケーションから Q キャプチャー・プログラムに制御メッセージを送信するキュー。リモート構成では、これはソース・システム上の Q キャプチャー管理キューのターゲット・システムのローカル定義です。

再始動キュー

Q キャプチャー・プログラムに、再始動後の DB2® リカバリー・ログの中の読み取りを開始する場所を指示する単一のメッセージを保持するキュー。これは、ソース・システム上のローカル・キューです。それぞれの Q キャプチャー・プログラムは、独自の再始動キューを持っていない限りありません。

予備キュー

ターゲット表のロード時に Q キャプチャー・プログラムからのトランザクション・メッセージを保持する、ターゲット・システムで定義されているモデル・キュー。Q アプライ・プログラムは、モデル・キュー定義に基づいてロード・プロセス中にこれらの動的キューを作成し、次いで削除します。予備キューの名前は IBMQREP.SPILL.MODELQ にする必要があります。

Q レプリケーションおよびイベント発行用の WebSphere MQ オブジェクトを定義する方法

Q レプリケーションまたはイベント発行に必要な WebSphere MQ オブジェクトを作成した後、Q キャプチャー・プログラムと Q アプライ・プログラムに、オブジェクトに関して指示する必要があります。

これらのオブジェクトは、コントロール表の作成時、およびレプリケーション・キュー・マップまたは発行キュー・マップの作成時に定義します。

コントロール表の作成時

Q キャプチャー・コントロール表の場合、Q キャプチャー・プログラムが実行されるシステム上のキュー・マネージャーの名前、およびローカル管理キューとローカル再始動キューの名前を指定します。Q アプライ・コントロール表の場合、Q アプライ・プログラムが実行されるシステム上のキュー・マネージャーの名前を指定します。

レプリケーション・キュー・マップの作成時

Q キャプチャー・プログラムが実行されるシステム上の送信キューの名前、および Q アプライ・プログラムが実行されるシステム上の受信キューと管理キューの名前を指定します。

発行キュー・マップの作成時

Q キャプチャー・プログラムが実行されるシステム上の送信キューの名前を指定します。

Q レプリケーションまたはイベント発行プログラムについて、伝送キューやチャンネルなどの WebSphere MQ チャンネル・オブジェクトを定義する必要はありません。これらのオブジェクトはソース・キュー・マネージャーおよびターゲット・キュー・マネージャー内で定義するだけで済みます。

関連概念:

- 62 ページの『WebSphere MQ オブジェクトの設定要件』

- 13 ページの『発行キュー・マップ』
- 6 ページの『レプリケーション・キュー・マップ』
- 47 ページの『Q レプリケーションおよびイベント発行用の WebSphere MQ のセットアップ -- 概要』

関連タスク:

- 87 ページの『Q キャプチャー・プログラムと Q アプライ・プログラムのコントロール表の作成』
- 180 ページの『発行キュー・マップの作成』
- 95 ページの『レプリケーション・キュー・マップの作成』

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

Q キャプチャー・プログラムが同一のシステムの Q アプライ・プログラムにデータを複製する場合、必要なキュー・マネージャーは 1 つだけです。送信キューおよび受信キューで同一のローカル・キューを使用でき、2 つのプログラムが 1 つのローカル管理キューを共用できます。リモート・キュー定義、伝送キュー、またはチャネルは必要ありません。

以下のチェックリストでは、同一のシステムでの単一方向 Q レプリケーションまたはイベント発行に必要な WebSphere[®] MQ オブジェクトが示されています。

- ___ Q キャプチャー・プログラムと Q アプライ・プログラムの両方で使用される、1 つのキュー・マネージャー。
- ___ 送信キューと受信キューの両方として機能する、1 つのローカル・キュー。
- ___ Q キャプチャー・プログラムと Q アプライ・プログラムの両方で管理キューとして機能する、1 つのローカル・キュー。
- ___ 再始動キューとして機能する、1 つのローカル・キュー。

51 ページの図 4 では、同一のシステムでの単一方向 Q レプリケーションに必要な WebSphere MQ オブジェクトを示しています。

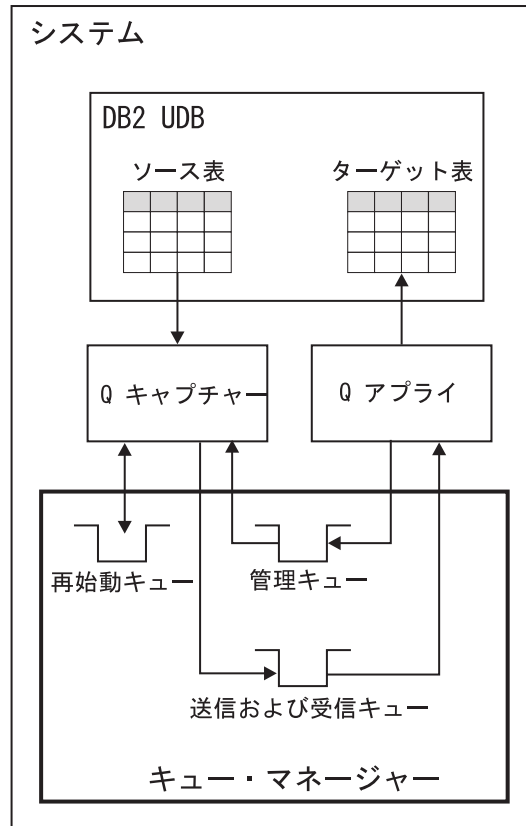


図4. 同一のシステムでの単一方向 Q レプリケーションに必要な WebSphere MQ オブジェクト： Q キャプチャー・プログラムおよび Q アプライ・プログラムが同一のシステム上で実行されている場合、必要なキュー・マネージャーは 1 つだけです。一方のローカル・キュー・サーバーは、送信キューと受信キューの両方として機能し、もう一方のローカル・キュー・サーバーは、Q キャプチャー・プログラムと Q アプライ・プログラムの両方で管理キューとして機能します。

同一のシステムで複製する Q キャプチャー・プログラムと Q アプライ・プログラムの両方で使用されるコントロール表を作成する場合、コントロール表の両方のセットで同じキュー・マネージャーを指定します。レプリケーション・キュー・マップを作成する場合、送信キューおよび受信キューで同一のローカル・キューを指定できます。Q キャプチャー・コントロール表を作成する際に指定する同一の管理キューを、レプリケーション・キュー・マップを作成する際に Q アプライ管理キューとして指定することもできます。

関連概念:

- 47 ページの『Q レプリケーションおよびイベント発行に必要な WebSphere MQ オブジェクト -- 概要』
- 62 ページの『WebSphere MQ オブジェクトの設定要件』
- 91 ページの『ソースからターゲットへの (単一方向) レプリケーションのセットアップ -- 概要』
- 48 ページの『必須 WebSphere MQ オブジェクトの概観』

単一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

リモート・サーバー間の単一方向 Q レプリケーションまたはイベント発行では、Q キャプチャー・プログラムおよび Q アプライ・プログラム用のキュー・マネージャーおよびキューが必要です。サーバーが分散しているため、ネットワーク経由のトランザクションの伝送および通信のための伝送キューおよびチャネルも必要です。

以下のチェックリストでは、2 つのリモート・サーバー間の単一方向レプリケーションに必要なオブジェクトが示されています。

ソース・システム上の非チャネル・オブジェクト

- ___ キュー・マネージャー。
- ___ 送信キューとして機能するリモート・キュー (このキューはターゲット・システム上の受信キューを指します)。
- ___ 管理キューとして機能するローカル・キュー。
- ___ 再始動キューとして機能するローカル・キュー。

ターゲット・システム上の非チャネル・オブジェクト

- ___ キュー・マネージャー。
- ___ 受信キューとして機能するローカル・キュー。
- ___ 管理キューとして機能するリモート・キュー (このキューはソース・システム上の管理キューを指します)。
- ___ Q アプライ・プログラムが作成し、ターゲット表のロード中に使用する、任意の一時ローカル予備キュー用のモデル・キュー定義。

ソースからターゲットへのチャネル

- ___ ソース・キュー・マネージャー内で定義される送信側チャネル。
- ___ 関連するローカル伝送キュー。
- ___ ターゲット・キュー・マネージャー内で定義される、対応する受信側チャネル。

ターゲットからソースへのチャネル

- ___ ターゲット・キュー・マネージャー内で定義される送信側チャネル。
- ___ 関連するローカル伝送キュー。
- ___ ソース・キュー・マネージャー内で定義される、対応する受信側チャネル。

53 ページの図 5 では、リモート・サーバー間の単一方向 Q レプリケーションに必要な WebSphere® MQ オブジェクトが示されています。

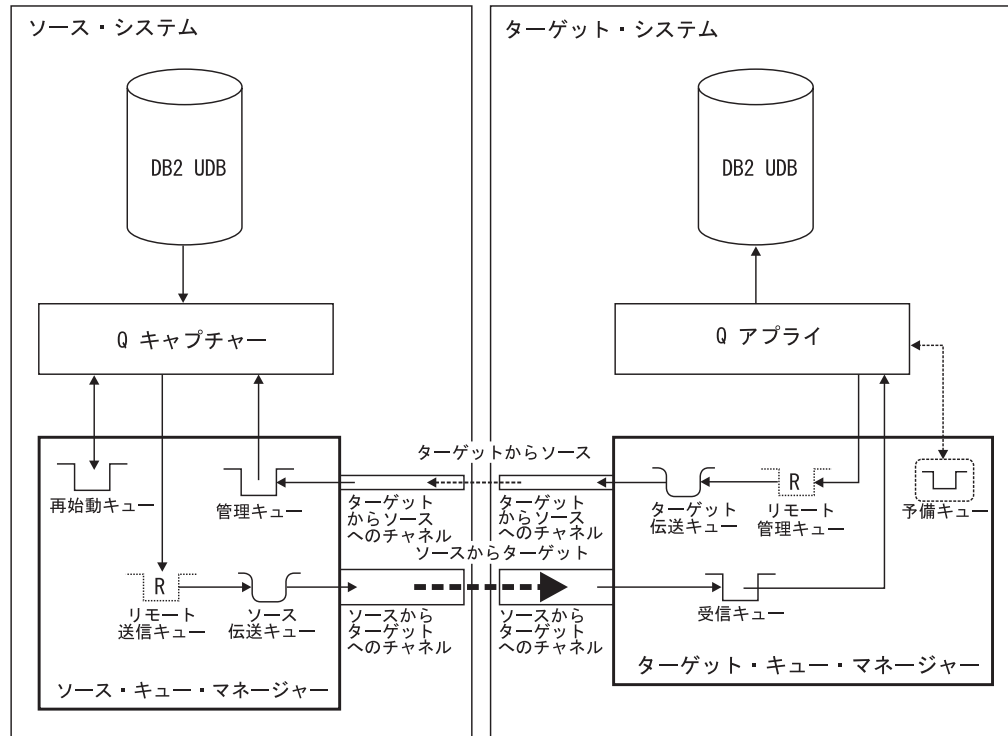


図5. リモート・サーバー間の単一方向 Q レプリケーションに必要な WebSphere MQ オブジェクト： Q キャプチャー・プログラムで必要なオブジェクトは、ソース・システム上のキュー・マネージャー内で定義されます。 Q アプライ・プログラムで必要なオブジェクトは、ターゲット・システム上のキュー・マネージャー内で定義されます。 Q キャプチャー・プログラムからのデータ・メッセージおよび情報メッセージ用の、ソース・システムとターゲット・システム間の伝送パスを作成するために、2 つのチャンネル・オブジェクトが必要です。 Q アプライ・プログラムからのコントロール・メッセージ用の、ターゲット・システムからソース・システムへの伝送パスを作成するために、2 つのチャンネル・オブジェクトも必要です。

Q キャプチャー・プログラムから Q アプライ・プログラムへの複数のチャンネルを作成する場合、送信を待機しているメッセージを保持するための、複数の伝送キューが必要になります。

関連概念:

- 47 ページの『Q レプリケーションおよびイベント発行に必要な WebSphere MQ オブジェクト -- 概要』
- 62 ページの『WebSphere MQ オブジェクトの設定要件』
- 91 ページの『ソースからターゲットへの (単一方向) レプリケーションのセットアップ -- 概要』
- 48 ページの『必須 WebSphere MQ オブジェクトの概観』

イベント発行に必要な WebSphere MQ オブジェクト

リモート・サーバー間のイベント発行は、キュー・マネージャー、および Q キャプチャー・プログラムとユーザー・アプリケーション用のキューを必要とします。サーバーが分散しているため、トランザクション・データを送信し、ネットワーク間で通信を行うために、伝送キューとチャンネルも必要です。

以下のチェックリストは、2 つのリモート・サーバー間のイベント発行に必要なオブジェクトを示しています。

ソース・システム上の非チャンネル・オブジェクト

- ___ キュー・マネージャー。
- ___ 送信キューとして機能するリモート・キュー (このキューは、ターゲット・システム上の受信キューを指します)。
- ___ 管理キューとして機能するローカル・キュー。
- ___ 再始動キューとして機能するローカル・キュー。

ターゲット・システム上の非チャンネル・オブジェクト

- ___ キュー・マネージャー。
- ___ 受信キューとして機能するローカル・キュー。
- ___ 管理キューとして機能するリモート・キュー (このキューは、ソース・システム上の管理キューを指します)。

ソースからターゲットへのチャンネル

- ___ ソース・キュー・マネージャー内で定義される送信側チャンネル。
- ___ 関連するローカル伝送キュー。
- ___ ターゲット・キュー・マネージャー内で定義される、対応する受信側チャンネル。

ターゲットからソースへのチャンネル

- ___ ターゲット・キュー・マネージャー内で定義される送信側チャンネル。
- ___ 関連するローカル伝送キュー。
- ___ ソース・キュー・マネージャー内で定義される、対応する受信側チャンネル。

55 ページの図 6 は、リモート・サーバー間のイベント発行に必要な WebSphere® MQ オブジェクトを示しています。

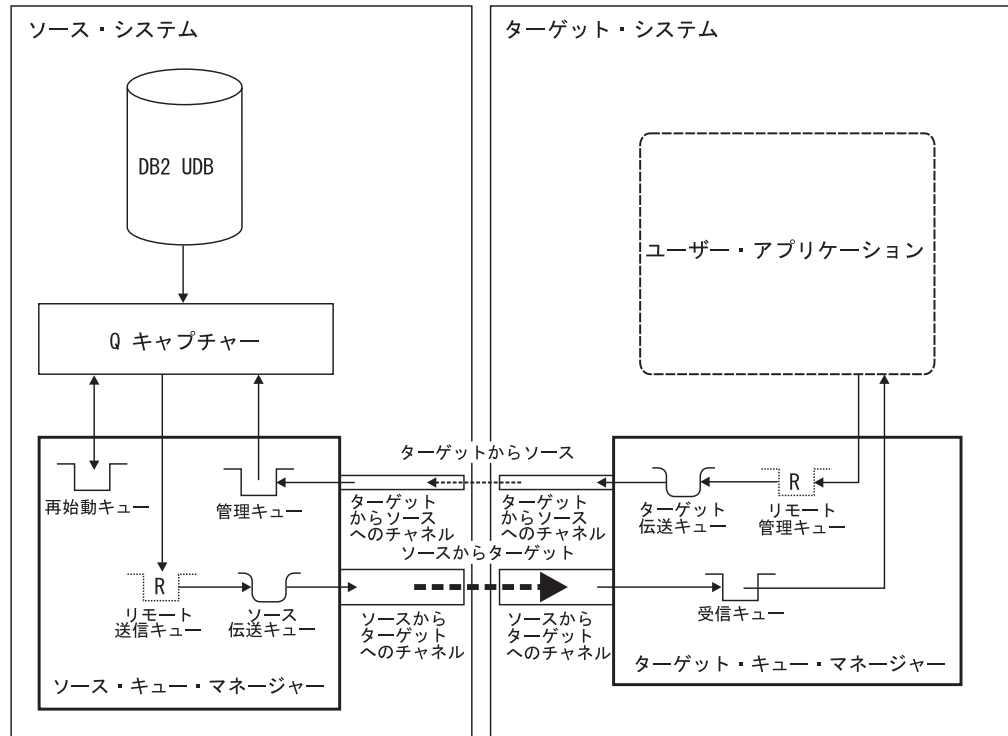


図6. リモート・サーバー間のイベント発行に必要な WebSphere MQ オブジェクト： Q キャプチャー・プログラムに必要なオブジェクトは、ソース・システム上のキュー・マネージャー内で定義されます。ユーザー・アプリケーションに必要なオブジェクトは、ターゲット・システム上のキュー・マネージャー内で定義されます。 Q キャプチャー・プログラムからのデータ・メッセージおよび情報メッセージ用の、ソース・システムとターゲット・システム間の伝送パスを作成するために、2 つのチャンネル・オブジェクトが必要です。また、ユーザー・アプリケーションからの制御メッセージ用の、ターゲット・システムからソース・システムへの伝送パスの作成にも必要です。

Q キャプチャー・プログラムからユーザー・アプリケーションへの複数のチャンネルを作成する場合、通過を待機するメッセージを保持するための複数の伝送キューが必要です。

関連概念:

- 47 ページの『Q レプリケーションおよびイベント発行に必要な WebSphere MQ オブジェクト -- 概要』
- 62 ページの『WebSphere MQ オブジェクトの設定要件』
- 179 ページの『ソースからの発行 (イベント発行) の設定 -- 概要』
- 48 ページの『必須 WebSphere MQ オブジェクトの概観』

双方向またはピアツーピア・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)

2 つのサーバー間で両方向にトランザクションを複製するには、サーバーごとにペアになっている Q キャプチャー・プログラムおよび Q アプライ・プログラムが必要です。単一方向レプリケーションに必要な、同一の WebSphere® MQ オブジェクトの 2 つのセットを定義します。1 つの例外として、各システムに必要なキュー・マネージャーは 1 つだけです。

たとえば、サーバー A とサーバー B の間で両方向にトランザクションを複製することを計画しているとして。サーバー A の Q キャプチャー・プログラムをサーバー B の Q アプライ・プログラムにリンクさせる WebSphere MQ オブジェクトを作成します。さらに、サーバー B の Q キャプチャー・プログラムをサーバー A の Q アプライ・プログラムにリンクさせる WebSphere MQ オブジェクトも作成します。サーバー A とサーバー B はそれぞれ、実行されるシステムの単一のキュー・マネージャーに接続します。

以下のチェックリストでは、2 つのリモート・サーバー間の双方向またはピアツーピア・レプリケーションに必要なオブジェクトが示されています。キュー・マネージャーは、レプリケーション・サーバーの一部ではなく、同じシステム上で実行するため、オブジェクトはシステムごとにグループ化されます。

システム A の非チャンネル・オブジェクト

- ___ キュー・マネージャー。
- ___ 送信キューとして機能するリモート・キュー (このキューはシステム B の受信キューを指します)。
- ___ 管理キューとして機能するローカル・キュー。
- ___ 再始動キューとして機能するローカル・キュー。
- ___ 受信キューとして機能するローカル・キュー。
- ___ 管理キューとして機能するリモート・キュー (このキューはシステム B の管理キューを指します)。
- ___ Q アプライ・プログラムが作成し、ターゲット表のロード中に使用する、任意の一時ローカル予備キュー用のモデル・キュー定義。

システム B の非チャンネル・オブジェクト

- ___ キュー・マネージャー。
- ___ 送信キューとして機能するリモート・キュー (このキューはシステム A の受信キューを指します)。
- ___ 管理キューとして機能するローカル・キュー。
- ___ 再始動キューとして機能するローカル・キュー。
- ___ 受信キューとして機能するローカル・キュー。
- ___ 管理キューとして機能するリモート・キュー (このキューはシステム A の管理キューを指します)。
- ___ Q アプライ・プログラムが作成し、ターゲット表のロード中に使用する、任意の一時ローカル予備キュー用のモデル・キュー定義。

チャンネル・オブジェクト

システム A の Q キャプチャーからシステム B の Q アプライへのチャンネル・オブジェクト

- ___ システム A のキュー・マネージャー内で定義される送信側チャンネル。

- ___ システム A の関連するローカル伝送キュー。
- ___ システム B のキュー・マネージャー内で定義される、対応する受信側チャンネル。

システム B の Q アプライからシステム A の Q キャプチャーへのチャンネル・オブジェクト

- ___ システム B のキュー・マネージャー内で定義される送信側チャンネル。
- ___ システム B の関連するローカル伝送キュー。
- ___ システム A のキュー・マネージャー内で定義される、対応する受信側チャンネル。

システム B の Q キャプチャーからシステム A の Q アプライへのチャンネル・オブジェクト

- ___ システム B のキュー・マネージャー内で定義される送信側チャンネル。
- ___ システム B の関連するローカル伝送キュー。
- ___ システム A のキュー・マネージャー内で定義される、対応する受信側チャンネル。

システム A の Q アプライからシステム B の Q キャプチャーへのチャンネル

- ___ システム A のキュー・マネージャー内で定義される送信側チャンネル。
- ___ システム A の関連するローカル伝送キュー。
- ___ システム B のキュー・マネージャー内で定義される、対応する受信側チャンネル。

58 ページの図 7 では、2 つのリモート・サーバー間の双方向またはピアツーピア Q レプリケーションに必要な WebSphere MQ オブジェクトを示しています。

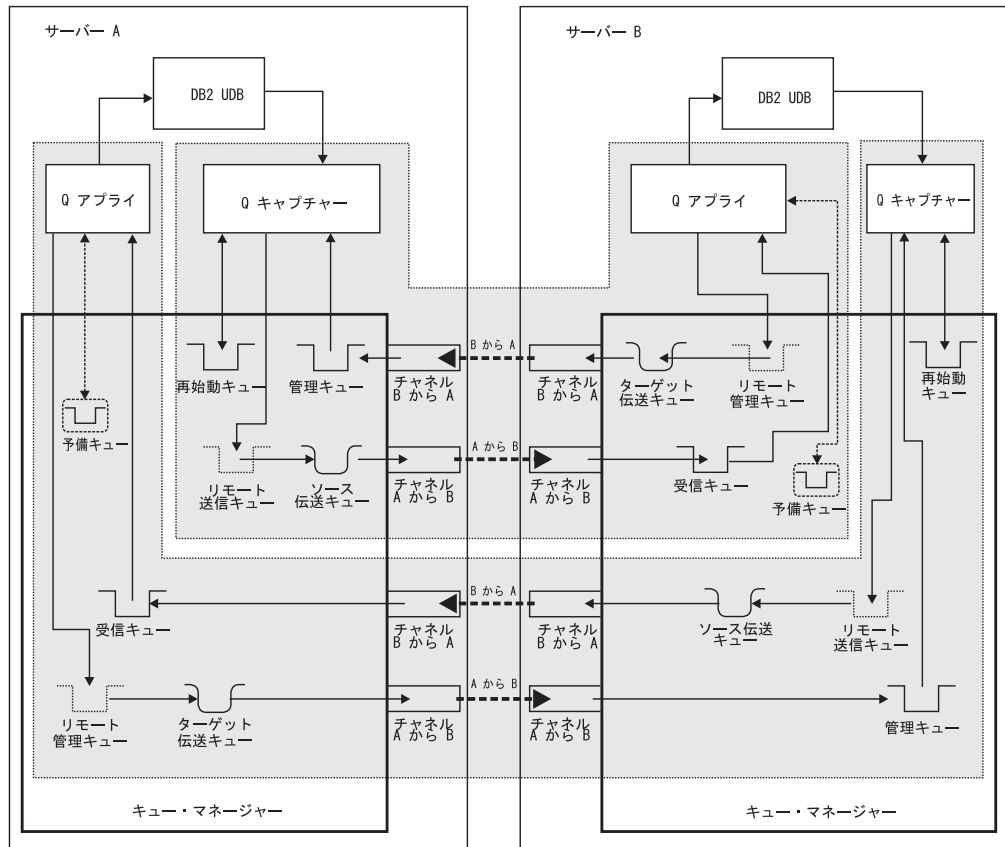


図7. 2つのリモート・サーバー間の双方向またはピアツーピア Q レプリケーションに必要な WebSphere MQ オブジェクト：単方向 Q レプリケーションでは、Q キャプチャー・プログラムおよび Q アプライ・プログラムへの接続に必要な、同じ WebSphere MQ オブジェクトの2つのセットを作成しなければなりません。オブジェクトの片方のセットは、一方方向でレプリケーションを処理し、オブジェクトのもう一方のセットは、その逆方向でレプリケーションを処理します。必要なキュー・マネージャーは、システムごとに1つだけです。

関連概念:

- 47 ページの『Q レプリケーションおよびイベント発行に必要な WebSphere MQ オブジェクト -- 概要』
- 62 ページの『WebSphere MQ オブジェクトの設定要件』
- 133 ページの『ソースからターゲットへのレプリケーションのセットアップ (複数方向) -- 概要』
- 48 ページの『必須 WebSphere MQ オブジェクトの概観』

ピアツーピア・レプリケーション (3 つ以上のリモート・サーバー) で必要な WebSphere MQ オブジェクト

3 つ以上のリモート・サーバーを持つピアツーピア・グループにおいて、各サーバーには、グループ内の各追加サーバーへの 1 つの発信チャンネルが必要です。また、グループ内の各追加サーバーからの着信チャンネルも必要です。

各サーバーの Q アプライ・プログラムには、1 つの発信チャンネルにつき 1 つのリモート管理キューが必要です。Q キャプチャー・プログラムはローカル管理キュー

を 1 つしか必要としません。なぜなら、Q アプライ・プログラムからのすべての着信メッセージは、1 つのキュー・マネージャーによって処理され、1 つのキューに送信されるからです。

送信キューと受信キューの数は、複製される論理表の数、およびグループ内のサーバーの数によって異なります。

たとえば、1 つの論理表が 3 つのリモート・サーバー間で複製される場合、サーバー A の Q キャプチャー・プログラムは、サーバー B に発信するトランザクション用とサーバー C に発信するトランザクション用の 2 つの送信キューを必要とします。サーバー A の Q アプライ・プログラムは、サーバー B から着信するトランザクション用とサーバー C から着信するトランザクション用の 2 つの受信キューを必要とします。

2 つの論理表が複製される場合、送信キューと受信キューの数を 2 倍にする必要があります。

以下のチェックリストは、3 つ以上のサーバーを持つピアツーピア・レプリケーション内の各システムに必要なオブジェクトを示しています。キュー・マネージャーは、レプリケーション・サーバーの一部ではなく、同じシステム上で実行するため、オブジェクトはシステムごとにグループ化されます。

各システムの非チャネル・オブジェクト

- ___ 1 つのキュー・マネージャー。
- ___ 各発信チャネルごと、および複製される各論理表ごとの 1 つのリモート送信キュー。
- ___ Q キャプチャー・プログラム用の管理キューとして機能する 1 つのローカル・キュー。
- ___ 再始動キューとして機能する 1 つのローカル・キュー。
- ___ 各着信チャネルごと、および複製される各論理表ごとの 1 つのローカル受信キュー。
- ___ 各発信チャネルごとの Q アプライ・プログラム用の 1 つのリモート管理キュー。
- ___ Q アプライ・プログラムが作成し、ターゲット表のロード中に使用する、任意の一時ローカル予備キュー用のモデル・キュー定義。

各システムの発信チャネル・オブジェクト

これらのオブジェクトをグループ内の追加サーバーごとに作成します。たとえば、3 つのサーバーを持つグループの場合、各サーバーは 2 つの発信チャネルを必要とします。

- ___ ローカル・キュー・マネージャー内で定義される送信側チャネル。
- ___ 関連するローカル伝送キュー。
- ___ このチャネルの接続先のリモート・サーバー上のキュー・マネージャー内で定義される、対応する受信側チャネル。

各システムの着信チャンネル・オブジェクト

これらのオブジェクトをグループ内の追加サーバーごとに作成します。たとえば、3つのサーバーを持つグループの場合、各サーバーは2つの着信チャンネルを必要とします。

- ___ ローカル・キュー・マネージャー内で定義される受信側チャンネル。
- ___ このチャンネルの接続先のリモート・サーバー上のキュー・マネージャー内で定義される、対応する送信側チャンネル。

61 ページの図 8 は、1つの論理表が複製される3つのリモート・サーバー間のピアツーピア・レプリケーションに関する、1つのサーバーで必要な WebSphere® MQ オブジェクトを示しています。

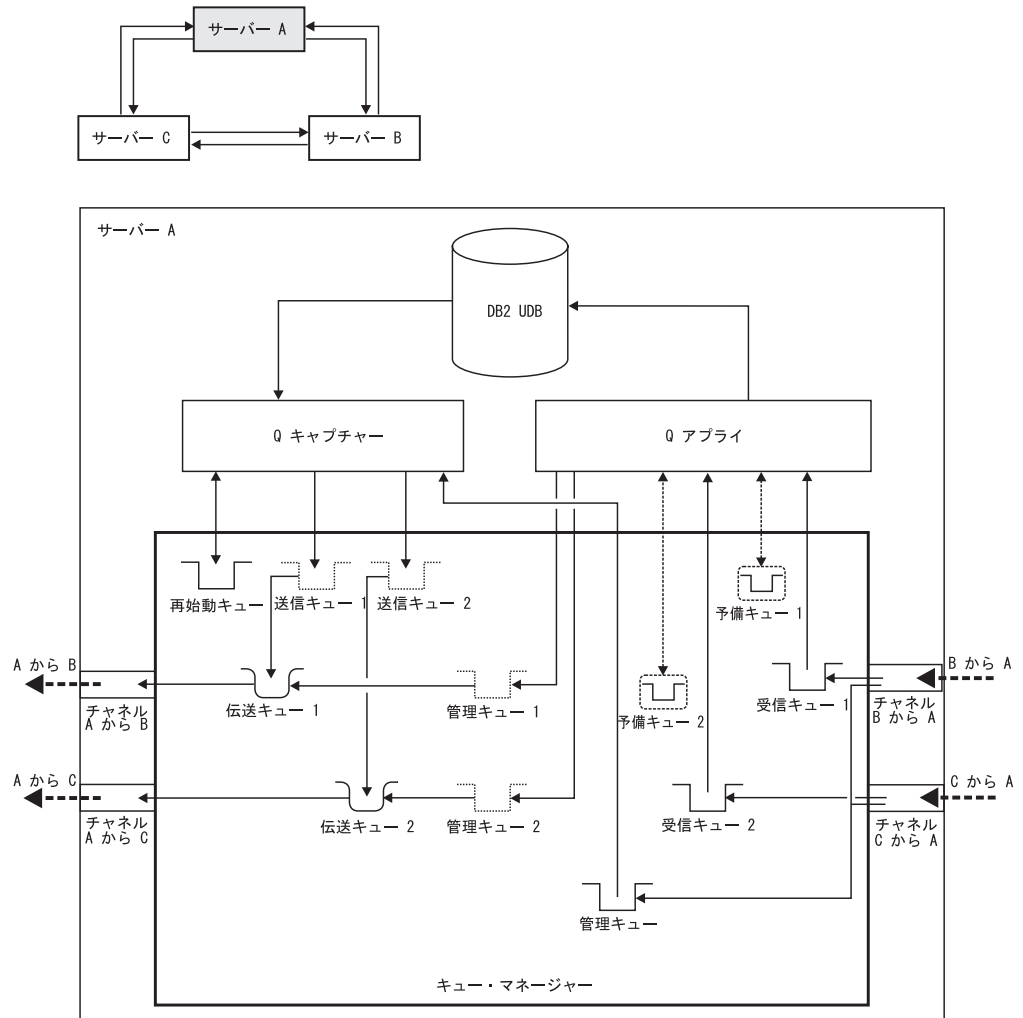


図 8. 2 つのリモート・サーバーを持つピアツーピア・レプリケーションに関する、1 つのサーバーに必要な WebSphere MQ オブジェクト：各システムで、1 つのキュー・マネージャーを作成します。Q キャプチャー・プログラムは、1 つの管理キューと 1 つの再始動キューを必要とします。各発信チャンネルごとに 1 つのリモート送信キューを作成します (1 つの論理表が複製されていると想定します)。Q アプライ・プログラムは、各発信チャンネルごとに 1 つのリモート管理キューと、各着信チャンネルごとに 1 つのローカル受信キューを必要とします (1 つの論理表が複製されていると想定します)。グループ内の追加サーバーごとに 1 つの発信チャンネルと 1 つの着信チャンネルを作成します。

関連概念:

- 47 ページの『Q レプリケーションおよびイベント発行に必要な WebSphere MQ オブジェクト -- 概要』
- 62 ページの『WebSphere MQ オブジェクトの設定要件』
- 133 ページの『ソースからターゲットへのレプリケーションのセットアップ (複数方向) -- 概要』
- 48 ページの『必須 WebSphere MQ オブジェクトの概観』

WebSphere MQ オブジェクトの設定要件

このトピックでは、さまざまな Q レプリケーションおよびイベント発行シナリオで使用される、WebSphere® MQ オブジェクトの設定要件を説明します。以下のようなシナリオがあります。

- 『ソースでの WebSphere MQ オブジェクト』
- 63 ページの『ターゲットでの WebSphere MQ オブジェクト』
- 65 ページの『WebSphere MQ チャネル・オブジェクト』

WebSphere MQ オブジェクトの構成に関する詳細については、ご使用のプラットフォームの「*WebSphere MQ* システム管理ガイド」、「*WebSphere MQ* 相互通信」、および「*WebSphere MQ Script (MQSC)* コマンド・リファレンス」を参照してください。

ソースでの WebSphere MQ オブジェクト

表 5 では、ソースでの WebSphere MQ オブジェクトの、選択されたパラメーターの必須の値を記載しています。

表 5. ソースでの *WebSphere MQ* オブジェクトの必須のパラメーター値

オブジェクト名	設定要件
キュー・マネージャー	MAXMSGL (このキュー・マネージャーのキューで許可されているメッセージの最大サイズ。) この値は、少なくとも、レプリケーション・キュー・マップまたは発行キュー・マップの作成時に定義する <code>MAX_MESSAGE_SIZE</code> と同じ大きさでなければなりません。 <code>MAX_MESSAGE_SIZE</code> は、Q キャプチャー・プログラムが各送信キューに割り振るメッセージ・バッファーを定義します。 <code>MAXMSGL</code> の値はまた、少なくとも、各送信キュー、伝送キュー、および管理キューに定義される <code>MAXMSGL</code> と同じ大きさでなければなりません。 注: このパラメーターは z/OS™ では無効です。
送信キュー	PUT (ENABLED) Q キャプチャー・プログラムが、データ・メッセージおよび情報メッセージをキューに入れられるようにします。 DEFPSIST(YES) キューは永続的であり、キュー・マネージャーが再始動してもそのまま残ります。メッセージが記録されるので、リカバリーが可能です。
管理キュー	GET (ENABLED) Q キャプチャー・プログラムが、キューからメッセージを受け取ることができます。 DEFPSIST(YES) キューは永続的であり、キュー・マネージャーが再始動してもそのまま残ります。メッセージが記録されるので、リカバリーが可能です。

表 5. ソースでの WebSphere MQ オブジェクトの必須のパラメーター値 (続き)

オブジェクト名	設定要件
再始動キュー	<p>PUT (ENABLED) Q キャプチャー・プログラムは、再始動メッセージをキューに入れることができます。</p> <p>GET (ENABLED) Q キャプチャー・プログラムは、キューから再始動メッセージを受け取ることができます。</p> <p>DEFPSIST(YES) キューは永続的であり、キュー・マネージャーが再始動してもそのまま残ります。キュー上のメッセージが記録されるので、リカバリーが可能です。</p>

ターゲットでの WebSphere MQ オブジェクト

表 6では、ターゲットでの WebSphere MQ オブジェクト、選択されたパラメーターの必須の値を記載しています。

表 6. ターゲットでの WebSphere MQ オブジェクトの必須のパラメーター値

オブジェクト名	設定要件
キュー・マネージャー	<p>MAXMSGL</p> <p>(このキュー・マネージャーのキューで許可されているメッセージの最大サイズ。) この値は、少なくとも、レプリケーション・キュー・マップの作成時に定義する MAX_MESSAGE_SIZE および MEMORY_LIMIT と同じでなければなりません。 MAX_MESSAGE_SIZE は、Q キャプチャー・プログラムが各送信キューに割り振るメッセージ・バッファーを定義します。 MEMORY_LIMIT は、各受信キューからのトランザクションをバッファーに入れるために Q アプライ・プログラムが使用する、メッセージ・バッファーを定義します。</p> <p>注: このパラメーターは z/OS では無効です。</p>

表 6. ターゲットでの WebSphere MQ オブジェクトの必須のパラメーター値 (続き)

オブジェクト名	設定要件
受信キュー	<p>GET (ENABLED) Q アプリケーション・プログラムが、キューからメッセージを受け取ることができます。</p> <p>MAXMSGL キューのメッセージの最大サイズが、少なくとも、ソース・システムでの伝送キューに定義するMAXMSGL、およびレプリケーション・キュー・マップの作成時に設定するMAX_MESSAGE_SIZE およびMEMORY_LIMIT と同じであることを確認してください。</p> <p>DEFPSIST(YES) キューは永続的であり、キュー・マネージャーが再始動してもそのまま残ります。メッセージが記録されるので、リカバリーが可能です。</p> <p>SHARE 複数の Q アプリケーション・スレッドがこのキューを処理できるようにします。</p> <p>MSGDLVSQ(PRIORITY) キュー上のメッセージが、優先順位内での先入れ先出し法の順序で送達されるように指定します。Q レプリケーション・メッセージが優先順位付けされていなくても、受信キューのこのデフォルトを受け入れることをお勧めします。</p>
管理キュー	<p>PUT (ENABLED) Q アプリケーション・プログラムまたはユーザー・アプリケーションが、コントロール・メッセージをキューに入れられるようにします。</p> <p>DEFPSIST(YES) キューは永続的であり、キュー・マネージャーが再始動してもそのまま残ります。メッセージが記録されるので、リカバリーが可能です。</p> <p>SHARE 複数の Q アプリケーション・スレッドがこのキューを処理できるようにします。</p>

表 6. ターゲットでの WebSphere MQ オブジェクトの必須のパラメーター値 (続き)

オブジェクト名	設定要件
予備キュー	<p>キュー名 (Queue name) 予備キューには、IBMQREP.SPILL.MODELQ という特定の名前を付けなければなりません。</p> <p>DEFTYPE(PERMDYN) 予備キューが永続的な動的キューになるように指定します。それらは、Q アプライ・プログラムの要求で作成および削除されますが、再始動が行われてもそのまま残ります。メッセージが記録されるので、リカバリーが可能です。</p> <p>DEFSOPT(SHARED) 複数のスレッド (さまざまなエージェント・スレッドおよび予備エージェント・スレッド) が、予備キュー上のメッセージに同時にアクセスできるようにします。</p> <p>MAXDEPTH(500000) これは、予備キュー上のメッセージの数として推奨されている上限です。ターゲット表がロードされる間にソース表で予期される変更の数に基づいて、この数を調整してください。</p> <p>MAXMSGL(100000) これは、最大メッセージ・サイズとして推奨されている制限です。この値が、ソース・キュー・マネージャーおよび伝送キューでの MAXMSGL と同じか、それより大きいことを確認してください。</p> <p>MSGDLVSQ(FIFO) 予備キュー上のメッセージが、先入れ先出し法の順序で送達されるように指定します。</p>

WebSphere MQ チャネル・オブジェクト

66 ページの表 7 では、ソース・システムとターゲット・システムの間でチャネルを作成するために使用される WebSphere MQ オブジェクトについて説明し、選択されるパラメーターの必須値を示しています。

表7. チャンネルで使用される WebSphere MQ オブジェクトの必須のパラメーター値

オブジェクト名	説明	設定要件
伝送キュー (Transmission queue) (Q キャプチャーから Q アプライまたはユーザー・アプリケーションへ)	リモート Q アプライ・プログラムまたはユーザー・アプリケーションに送信される、Q キャプチャー・プログラムからのトランザクション・メッセージおよび情報メッセージを保持します。	USAGE(XMITQ) 伝送キュー。 MAXDEPTH 複数の送信キューで単一の伝送キューを使用するように計画している場合、伝送キューで許可されているメッセージの最大数が、少なくとも、すべての送信キューの最大数の合計数と同じ大きさでなければなりません。 MAXMSGL キューのメッセージの最大サイズが、ターゲット・システムでの受信キューに定義する MAXMSGL、およびレプリケーション・キュー・マップの作成時に設定する MAX_MESSAGE_SIZE および MEMORY_LIMIT 以上であることを確認してください。 DEFPSIST(YES) キューは永続的であり、キュー・マネージャーが再始動してもそのまま残ります。メッセージが記録されるので、リカバリーが可能です。
伝送キュー (Transmission queue) (Q アプライまたはユーザー・アプリケーションから Q キャプチャーへ)	Q キャプチャー管理キュー送られる、Q アプライ・プログラムまたはユーザー・アプリケーションからのコントロール・メッセージを保持します。	USAGE(XMITQ) 伝送キュー。 DEFPSIST(YES) キューは永続的であり、キュー・マネージャーが再始動してもそのまま残ります。メッセージが記録されるので、リカバリーが可能です。
チャンネル (Channel) (Q キャプチャーから Q アプライまたはユーザー・アプリケーションへ)	アウトバウンド・データおよび情報メッセージ用に、ソース・キュー・マネージャー内で定義されます。	CHLTYPE(SDR) 送信側チャンネル。 DISCINT 切断インターバルが、複製されるトランザクションがない期間に、このチャンネルがタイムアウトにならないだけの長さになっているかを確認してください。 HBINT この値は、Q キャプチャーの heartbeat_interval パラメーターで調整します。 HBINT パラメーターを使用してハートビート・フローを送信する場合、 heartbeat_interval=0 を設定して、ハートビート・メッセージを除去することを考慮してください。

表7. チャンネルで使用される WebSphere MQ オブジェクトの必須のパラメーター値 (続き)

オブジェクト名	説明	設定要件
チャンネル (Channel) (Q キャプチャー から Q アプライ またはユーザー・ア プリケー ションへ)	インバウンド・トランザクションまたは 情報メッセージ用に、ターゲット・キュー ・マネージャー内で定義されます。	CHLTYPE(RCVR) 受信側チャンネル。 DISCINT 切断インターバルが、複製されるトランザク ションがない期間に、このチャンネルがタイムア ウトにならないだけの長さになっているかを確認 してください。 HBINT この値は、Q キャプチャーの heartbeat_interval パラメーターで調整しま す。 HBINT パラメーターを使用してハートビ ート・フローを送信する場合、 heartbeat_interval=0 を設定して、ハートビ ート・メッセージを除去することを考慮してくだ さい。
チャンネル (Channel) (Q アプライまた はユーザー・ア プリケーション から Q キャプチ ャーへ)	アウトバウンド・コントロール・メッセ ージ用に、ターゲット・キュー・マネー ジャー内で定義されます。	CHLTYPE(SDR) 送信側チャンネル。 DISCINT Q アプライ・プログラムまたはユーザー・アプ リケーションによって送信されるコントロ ール・メッセージがほとんどないと予想される非 アクティブの期間中、このチャンネルがタイムア ウトにならないだけの長さに切断インター バルがなっているかを確認してください。 HBINT Q アプライ・プログラムはハートビート・メッ セージを送信しないため、いずれかの Q レプリ ケーション・パラメーターを設定してこの値を 調整する必要はありません。
チャンネル (Channel) (Q アプライまた はユーザー・ア プリケーション から Q キャプチ ャーへ)	インバウンド・コントロール・メッセ ージ用に、ソース・キュー・マネージャー 内で定義されます。	CHLTYPE(RCVR) 受信側チャンネル。 DISCINT Q アプライ・プログラムまたはユーザー・アプ リケーションによって送信される、コントロ ール・メッセージがほとんどないことが予想され る非アクティブの期間に、このチャンネルがタイ ムアウトにならないだけの長さに切断インター バルがなっているかを確認してください。 HBINT Q アプライ・プログラムはハートビート・メッ セージを送信しないため、いずれかの Q レプリ ケーション・パラメーターを設定してこの値を 調整する必要はありません。

関連概念:

- 47 ページの『Q レプリケーションおよびイベント発行に必要な WebSphere MQ オブジェクト -- 概要』

- 47 ページの『Q レプリケーションおよびイベント発行用の WebSphere MQ のセットアップ -- 概要』
- 48 ページの『必須 WebSphere MQ オブジェクトの概観』

WebSphere MQ メッセージ・サイズ

WebSphere® MQ メッセージのサイズは、キューおよびキュー・マネージャーの作成時、さらにレプリケーションまたは発行の設定時に制限できます。 WebSphere MQ、Q レプリケーション、およびイベント発行の間で、メッセージ・サイズ限度を調整する必要があります。

WebSphere MQ では、MAXMSGL (最大メッセージ長) を定義してメッセージのサイズを制限します。次のリストは、MAXMSGL と、Q キャプチャー・プログラムおよび Q アプライ・プログラムのメモリー限度との関連を示しています。

Q キャプチャー・プログラム

Q キャプチャー・プログラムがメッセージを送信キューに書き込む前に、各メッセージをバッファーに入れるために使用するメモリーの量を制限できます。この MAX_MESSAGE_SIZE は、レプリケーション・キュー・マップまたは発行キュー・マップの作成時に定義します。デフォルトは 64 KB です。

重要: Q キャプチャー・プログラムに許可するメッセージ・バッファーが、キューで処理するように設定したものより大きい場合は、レプリケーションまたは発行は行われません。ほとんどの場合、Q キャプチャー MAX_MESSAGE_SIZE および MAXMSGL には、メッセージを保持するキューに定義するのと同じ値を使用するのが無難な方法です。

Q アプライ・プログラム

エージェント・スレッドがメッセージを再アセンブルしてトランザクションにする前に、Q アプライ・プログラムが受信キューから取得する複数のメッセージをバッファーに入れるために使用するメモリーの量を制限できます。この MEMORY_LIMIT は、レプリケーション・キュー・マップの作成時に定義します。デフォルトは 2 MB です。

重要: 受信キューとしての機能を果たすローカル・キューの MAXMSGL が、受信キューの MEMORY_LIMIT より大きくないことを確認してください。

メッセージ分割:

Q キャプチャー・プログラムは、トランザクションを行境界で分けることにより、MAX_MESSAGE_SIZE を超えるトランザクションを複数のメッセージに自動的に分割します。ソース表のラージ・オブジェクト (LOB) 列のデータを複製または発行する場合は、Q キャプチャー・プログラムは LOB データを複数のメッセージに自動的に分割します。こうすることにより、メッセージは、送信キューごとに定義された MAX_MESSAGE_SIZE を超えなくなります。

一部のオペレーティング・システムでは、キューまたはチャネルには大きすぎるメッセージが自動的に分割されるようにするために、WebSphere MQ でメッセージ分割を定義できます。Q レプリケーションおよびイベント発行は、このメッセージ分

割機能を使用しません。メッセージ分割を使用するよう WebSphere MQ オブジェクトを設定する場合でも、キューの MAXMSGL が Q キャプチャー・プログラムの MAX_MESSAGE_SIZE 以上であることを確認する必要があります。

メッセージ・サイズの詳細については、ご使用のプラットフォームの「*WebSphere MQ システム管理ガイド*」を参照してください。

関連概念:

- 72 ページの『ラージ・オブジェクト (LOB) 値のキュー項目数に関する考慮事項』
- 198 ページの『Q レプリケーションおよびイベント発行でのラージ・オブジェクト (LOB) データ・タイプに関する考慮事項』
- 13 ページの『発行キュー・マップ』
- 6 ページの『レプリケーション・キュー・マップ』

関連タスク:

- 180 ページの『発行キュー・マップの作成』
- 25 ページの『Q レプリケーションおよびイベント発行のメモリー要件の計画』
- 95 ページの『レプリケーション・キュー・マップの作成』

Q レプリケーションおよびイベント発行の WebSphere MQ のストレージ要件

WebSphere® MQ リソースが、ネットワーク障害および/またはターゲット障害に対して希望するレベルの回復力を実現するように計画してください。メッセージをトランスポートできない場合は、より多くのリソースがソースで使用されます。メッセージを適用できない場合は、より多くのリソースがターゲットで使用されます。

Q レプリケーションで使用されるすべてのメッセージは、永続メッセージです。WebSphere MQ は、すべての永続メッセージをログに書き込みます。キュー・マネージャーを障害後に再始動する場合は、キュー・マネージャーは、必要に応じてログに記録されたすべての永続メッセージを検索します。

AIX®, HP-UX, Linux, Sun™ Solaris™, および Windows® システム上の WebSphere MQ ログ・ファイルの詳細については、「*WebSphere MQ システム管理ガイド*」を参照してください。z/OS™ の場合は、「*WebSphere MQ for z/OS 概説および計画ガイド*」、他のプラットフォームの場合は、該当する「システム管理」を参照してください。

関連概念:

- 70 ページの『Q アプライ・プログラムの予備キューのストレージ要件』

関連タスク:

- 32 ページの『Q レプリケーションおよびイベント発行のストレージ要件の計画』

Q アプライ・プログラムの予備キューのストレージ要件

Q アプライ・プログラムがターゲットをロードするときに、Q サブスクリプションごとに一時予備キューが作成されます。予備キューは、ロードの完了後に除去されます。ターゲットのロード中に多くの更新が発生することが予想される場合は、予備キューをさらに大きくする必要があります。メモリー限度を超えた場合に Q アプライ・プログラムが使用する予備キューのために、十分なストレージを考慮しておく必要があります。予備キューの最大メッセージ・サイズおよび最大キュー項目数は、予備データを処理するために十分な大きさがなければなりません。

推奨: 最初は、Q アプライ・プログラムが予備キューに書き出す更新の数を制限するために、ターゲットをオフピーク時にロードしてください。

関連概念:

- 19 ページの『Q アプライ・プログラム』

関連タスク:

- 32 ページの『Q レプリケーションおよびイベント発行のストレージ要件の計画』

WebSphere MQ オブジェクトの接続および許可に関する要件

データを複製または発行する前に、Q レプリケーションおよびイベント発行プログラムが実行されるシステム上のキュー・マネージャー間の接続を構成する必要があります。また、レプリケーションおよび発行プログラムを実行するユーザー ID に、WebSphere® MQ オブジェクトに対して必要なアクションを実行する許可があることも確認します。

このトピックでは、接続要件と許可要件について説明します。

接続要件

レプリケーションまたは発行に関係する各システム上のキュー・マネージャーは、相互に接続が可能でなければなりません。分散環境では、Q キャプチャー・プログラム、Q アプライ・プログラム、およびユーザー・アプリケーションは、キュー・マネージャーに接続し、リモート・キュー定義、伝送キュー、およびチャネル経由でメッセージを送信します。

また、Q レプリケーションおよびイベント発行は、キュー・マネージャーのグループが、リモート・キュー定義、伝送キュー、およびチャネルを使わずにネットワーク上で直接通信するクラスタリングもサポートします。キュー・マネージャーと、そのキューを管理している Q キャプチャー・プログラムまたは Q アプライ・プログラムとが異なるシステム上で実行されている場合は、クライアント/サーバー接続はサポートされません。

さまざまなキュー・マネージャー構成やそれぞれの接続のセットアップ方法の詳細については、「*WebSphere MQ 相互通信*」を参照してください。

許可要件

WebSphere MQ キューは、Q キャプチャー・プログラムおよび Q アプライ・プログラムがデータ交換や通信を行うための主要な手段であり、これらのプログラムは、キュー上でデータにアクセス可能でなければなりません。

WebSphere MQ オブジェクトを作成するときは、レプリケーション・プログラムを操作するユーザー ID に、これらのオブジェクトに対して必要なアクションを実行する権限があることを確認してください。以下のリストは、Q キャプチャー・プログラム、Q アプライ・プログラム、レプリケーション・アラート・モニターに関するこれらの要件を要約しています。

Q キャプチャー・プログラムの許可要件

Q キャプチャー・プログラムを実行するユーザー ID には、次の権限が必要です。

- Q キャプチャー・プログラムが実行されるシステム上でキュー・マネージャーに接続するための権限 (MQCONN または MQCONNX)。
- 送信キュー上で以下のアクションを実行する権限: オープン (MQOPEN)、属性についての問い合わせ (MQINQ)、メッセージの挿入 (MQPUT)、メッセージのコミット (MQCMIT)、およびメッセージのロールバック (MQBACK)。
- Q キャプチャー管理キュー上で以下のアクションを実行する権限: オープン (MQOPEN)、属性についての問い合わせ (MQINQ)、およびメッセージの取得 (MQGET)。
- 再始動キュー上で以下のアクションを実行する権限: オープン (MQOPEN)、属性についての問い合わせ (MQINQ)、メッセージの挿入 (MQPUT)、およびメッセージの取得 (MQGET)。

Q アプライ・プログラムの許可要件

Q アプライ・プログラムを実行するユーザー ID には、次の権限が必要です。

- Q アプライ・プログラムが実行されるシステム上でキュー・マネージャーに接続するための権限 (MQCONN または MQCONNX)。
- 受信キュー上で以下のアクションを実行する権限: オープン (MQOPEN)、属性についての問い合わせ (MQINQ)、およびメッセージの取得 (MQGET)。
- Q アプライ管理キュー上で以下のアクションを実行する権限: オープン (MQOPEN)、属性についての問い合わせ (MQINQ)、およびメッセージの挿入 (MQPUT)。
- 一時予備キュー上で以下のアクションを実行する権限: オープン (MQOPEN)、メッセージの挿入 (MQPUT)、メッセージの取得 (MQGET)、およびキューの削除 (MQCLOSE)。

レプリケーション・アラート・モニターの許可要件

受信キュー上のメッセージ数のモニター (QAPPLY_QDEPTH アラート条件) または予備キュー上のメッセージ数のモニター (QAPPLY_SPILLQDEPTH アラート条件) にレプリケーション・アラート・モニターを使用する場合、モニターを実行するユーザー ID には、Q アプライ・プログラムが実行されるシステム上のキュー・マネージャーに対する接続権限が必要です。

Q キャプチャー・プログラムと Q アプライ・プログラムのどちらの場合も、メッセージ・チャンネル・エージェント (MCA) と関連付けられるユーザー ID には次の権限が必要です。

- ローカル・キュー・マネージャーに接続する権限。
- ローカル伝送キュー上で以下のアクションを実行する権限: オープン (MQOPEN) およびメッセージの挿入 (MQPUT)。

WebSphere MQ の許可と特権の詳細については、「*WebSphere MQ セキュリティー*」を参照してください。

関連概念:

- 75 ページの『Q レプリケーションおよびイベント発行の接続要件』
- 291 ページの『レプリケーション・アラート・モニター』
- 19 ページの『Q アプライ・プログラム』
- 15 ページの『Q キャプチャー・プログラム』
- 47 ページの『Q レプリケーションおよびイベント発行用の WebSphere MQ のセットアップ -- 概要』

ラージ・オブジェクト (LOB) 値のキュー項目数に関する考慮事項

ソース表からのラージ・オブジェクト (LOB) 値は、各送信キューのメッセージ・バッファとして Q キャプチャー・プログラムが割り振るメモリーの最大量を超える可能性があります。送信キューのデフォルトの MAX_MESSAGE_SIZE (メッセージ・バッファ) は 64 キロバイトです。DB2[®] UDB では、LOB 値は最大で 2 ギガバイトまで可能です。そのため、LOB 値は頻繁に複数のメッセージに分割されます。

大きい LOB 値が、比較的小さいメッセージ・バッファに基づいて分割される場合、非常に多くの LOB メッセージが作成されて、伝送キューまたは受信キューに設定するメッセージの最大量 (MAXDEPTH) を超える可能性があります。このような場合、キュー・エラーのプロンプトが出されます。リモート受信キューがエラー状態の場合、ターゲット・システム上のメッセージ・チャンネル・エージェントは、送達できないメッセージごとに WebSphere[®] MQ 例外レポートを送信します。これらの例外レポートには、Q キャプチャー・プログラムの管理キューが入れられている場合があります。

LOB データを複製することを計画している場合、ソース・システム上の伝送キューと管理キュー、およびターゲット・システム上の受信キューの MAXDEPTH 値が、分割された LOB メッセージが入るだけの大きさになっているかを確認してください。レプリケーション・キュー・マップまたは発行キュー・マップを作成する際に、送信キューの MAX_MESSAGE_SIZE を増やすことにより、LOB データを送信するのに必要なメッセージの数を減らすことができます。

関連概念:

- 62 ページの『WebSphere MQ オブジェクトの設定要件』
- 198 ページの『Q レプリケーションおよびイベント発行でのラージ・オブジェクト (LOB) データ・タイプに関する考慮事項』

- 47 ページの『Q レプリケーションおよびイベント発行用の WebSphere MQ のセットアップ -- 概要』

Q レプリケーションおよびイベント発行でのキュー・マネージャーのクラスタリング

Q レプリケーションは、キュー・マネージャーのクラスター環境で機能します。クラスタリングにより、リモート・キュー定義、伝送キュー、およびチャネルを必要とせずに、キュー・マネージャーのグループがネットワークを介して直接通信できます。

クラスター構成を使用することにより、同じクラスター内の複数のキュー・マネージャーに、同じ名前を使用してキューの複数のインスタンスを作成できます。ただし Q レプリケーションでは、ターゲット・システム上の受信キューを定義する回数は、クラスター内で必ず一度だけにしなければなりません。Q キャプチャーおよび Q アプライ・プログラムは、連続番号付けシステムを使用して、欠落しているメッセージを識別および検索します。(各メッセージの番号には、連続する正の整数が割り当てられます。) 受信キュー名は、Q キャプチャー・プログラムと Q アプライ・プログラムの特定のペアに固有でなければなりません。2 つの受信キューに同じ名前が付けられている場合、連続番号付けシステムは機能しません。

Q レプリケーションは、クラスター配布リストと同時に機能しません。これは、単一の MQPUT コマンドを使用して、複数の宛先に同じメッセージを送信します。

詳細については、「*WebSphere MQ* キュー・マネージャー・クラスター」を参照してください。

関連概念:

- 19 ページの『Q アプライ・プログラム』
- 15 ページの『Q キャプチャー・プログラム』
- 47 ページの『Q レプリケーションおよびイベント発行用の WebSphere MQ のセットアップ -- 概要』

第 8 章 Q レプリケーションおよびイベント発行用のサーバーの構成

Q レプリケーションおよびイベント発行用のサーバーの構成 -- 概要

データを複製または発行する前に、サーバーを作成および構成して、相互に接続できるようにしなければなりません。以下のトピックでは、環境をセットアップおよび構成する方法について説明しています。

- 『Q レプリケーションおよびイベント発行の接続要件』
- 76 ページの『Q レプリケーションおよびイベント発行用のデータベースの構成 (Linux、UNIX、Windows)』
- 85 ページの『Q レプリケーションおよびイベント発行用のデータベースの構成 (z/OS)』
- 86 ページの『レプリケーション・センターのソフトウェア前提条件』
- 87 ページの『Q キャプチャー・プログラムと Q アプライ・プログラムのコントロール表の作成』

DB2[®] Information Integrator Replication for z/OS[™] 用のサーバーの構成については、「*IBM DB2 Information Integrator Replication Installation and Customization Guide for z/OS*」を参照してください。

Q レプリケーションおよびイベント発行の接続要件

分散環境でデータを複製または発行する場合、接続をセットアップして構成しなければなりません。ほとんどの場合、レプリケーション・センターまたはレプリケーション・アラート・モニターを使用したり、ターゲット表をロードしたり、さらには、シグナルを挿入して Q サブスクリプションまたは XML 発行をアクティブまたは非アクティブにするために、リモート DB2[®] データベースまたはサブシステムに接続できなければなりません。

DB2 データベースまたはサブシステムの接続要件は、レプリケーションまたは発行環境によって異なります。

- レプリケーション・センターでは、以下へ接続できなければなりません。
 - Q キャプチャー・サーバー。これはイベント発行を管理するためです。
 - Q キャプチャー・サーバーおよび Q アプライ・サーバー。これは Q レプリケーションを管理するためです。
 - モニター・コントロール・サーバー。これはレプリケーション・アラート・モニターをセットアップするためです。
- EXPORT ユーティリティを使用したソースからのデータを含むターゲットを、Q アプライ・プログラムに自動的にロードさせる場合、Q アプライ・プログラムが Q キャプチャー・プログラムに接続できなければなりません。この接続では、**asnpwd** コマンドで作成される、パスワード・ファイルが必要です。

- DB2 UDB Extended Server Edition 上の Q キャプチャー・プログラムは、パーティション・データベースに接続できなければなりません。この接続では、**asnpwd** コマンドで作成される、パスワード・ファイルが必要です。
- レプリケーション・センター、システム・コマンド、または SQL を使用してリモート・ワークステーションからのレプリケーションを管理する場合、そのリモート・ワークステーションが、Q キャプチャー・サーバー、Q アプライ・サーバー、またはモニター・コントロール・サーバーに接続できなければなりません。

関連概念:

- 75 ページの『Q レプリケーションおよびイベント発行用のサーバーの構成 -- 概要』
- 291 ページの『レプリケーション・アラート・モニター』
- 19 ページの『Q アプライ・プログラム』
- 15 ページの『Q キャプチャー・プログラム』
- 70 ページの『WebSphere MQ オブジェクトの接続および許可に関する要件』

Q レプリケーションおよびイベント発行用のデータベースの構成 (Linux、UNIX、Windows)

Q レプリケーションおよびイベント発行用のデータベースの構成 (Linux、UNIX、Windows)

データを複製する前に、環境変数を設定し、ソースおよびターゲット・データベースを構成しなければなりません。イベント発行では、構成が必要なのはソース・データベースだけです。

手順:

Q レプリケーションおよびイベント発行用にデータベースを構成するには、以下のようになります。

1. 環境変数を設定する。詳細については、77 ページの『環境変数の設定 (Linux、UNIX、Windows)』を参照してください。
2. Q キャプチャー・プログラムを処理するように、ソース・データベースを構成する。詳細については、78 ページの『Q キャプチャー・プログラムが動作するソース・データベースの構成 (Linux、UNIX、Windows)』を参照してください。
3. Q アプライ・プログラムを処理するように、ターゲット・データベースを構成する (Q レプリケーションのみ)。詳細については、80 ページの『Q アプライ・プログラムが動作するターゲット・データベースの構成 (Linux、UNIX、Windows)』を参照してください。
4. オプション: プログラム・パッケージをバインドする。詳細については、81 ページの『オプション : プログラム・パッケージのバインド (Linux、UNIX、Windows) -- 概要』を参照してください。

関連概念:

- 75 ページの『Q レプリケーションおよびイベント発行用のサーバーの構成 -- 概要』

- 85 ページの『Q レプリケーションおよびイベント発行用のデータベースの構成 (z/OS)』

環境変数の設定 (Linux、UNIX、Windows)

Q キャプチャー・プログラム、Q アプライ・プログラム、またはレプリケーション・アラート・モニター・プログラムを操作する前、レプリケーション・センターを使用する前、システム・コマンドを発行する前に、環境変数を設定する必要があります。

環境変数の設定は、データベースの構成というさらに大きなタスクの一部です。

手順:

環境変数を設定するには、以下のようにします。

1. 以下のように、Q キャプチャー・サーバー、Q アプライ・サーバー、またはモニター・コントロール・サーバーを含む、DB2 インスタンス名 (DB2INSTANCE) の環境変数を設定する。

Windows:

```
SET DB2INSTANCE=db2_instance_name
```

Linux および UNIX:

```
export DB2INSTANCE=db2_instance_name
```

2. デフォルトのコード・ページ値以外のコード・ページを使用してソース・データベースを作成した場合は、DB2CODEPAGE 環境変数をそのコード・ページに設定する。
3. オプション: 環境変数 DB2DBDFT を Q キャプチャー・サーバーまたは Q アプライ・サーバーに設定する。
4. **Linux および UNIX:** Q レプリケーション・ライブラリーおよび実行可能ファイルがインストールされているディレクトリーが、システム変数に含まれていることを確認する。DB2 インスタンス・ホーム・ディレクトリーで、デフォルトのライブラリー・パスは LIBPATH=SQLLIB/LIB、デフォルトの実行可能ファイル・パスは PATH=SQLLIB/BIN です。ファイルを移動させた場合は、新しいパスを含むように環境変数を更新してください。
5. **AIX および DB2 Extended Server Edition:** 以下のコマンドを入力し、AIX ではソースおよびターゲット・データベースで、DB2 Extended Server Edition では (Q キャプチャー・プログラムが複数のデータベース・パーティションに接続しなければならない場合) ソース・データベースだけで、EXTSHM 環境変数を ON に設定する。

```
$ export EXTSHM=ON  
$ db2set DB2ENVLIST=EXTSHM
```

DB2 を始動するたびに、EXTSHM 環境変数が設定されていることを確認します。これを行うには、`/home/db2inst1/sqllib/profile.env` ファイルを編集し、以下の行を追加または変更します。

```
DB2ENVLIST='EXTSHM'
```


ここで、*db2inst* はターゲット・データベースを含んでいる DB2 インスタンスの名前です。

さらに、以下の行を */home/db2inst/sqllib/userprofile* ファイルに追加します。

```
export EXTSHM=ON
```

次のタスク:

- Q キャプチャー・プログラムを処理するように、ソース・データベースを構成する。詳細については、『Q キャプチャー・プログラムが動作するソース・データベースの構成 (Linux、UNIX、Windows)』を参照してください。

関連概念:

- 291 ページの『レプリケーション・アラート・モニター』
- 19 ページの『Q アプライ・プログラム』
- 15 ページの『Q キャプチャー・プログラム』

Q キャプチャー・プログラムが動作するソース・データベースの構成 (Linux、UNIX、Windows)

ソース・データベースでアーカイブ・ロギングが有効になっていない場合、ソース表への変更に関して、Q キャプチャー・プログラムがログ・ファイルを読み取れるように、アーカイブ・ロギングを有効にしなければなりません。この変更を有効にするために、ソース・データベースのオフライン・バックアップも実行しなければなりません。

重要: 大規模なデータベースのバックアップには長い時間がかかります。このプロセス中、アプリケーションはデータベースから切断され、新しい接続は許可されません。

Q キャプチャー・プログラムと協働するソース・データベースの構成は、データベースの構成というさらに大きなタスクの一部です。

手順:

レプリケーション・センターまたはコマンド行からソース・データベースを構成できます。

レプリケーション・センター

「アーカイブ・ロギングをオン」ウィンドウを使用して、ソース・データベースでアーカイブ・ロギングを有効にします。このウィンドウをオープンするには、有効にする Q キャプチャー・サーバーを右マウス・ボタンでクリックして、「アーカイブ・ロギングをオン」を選択します。詳細については、オンライン・ヘルプを参照してください。

コマンド行

Q キャプチャー・プログラムを実行するために DB2 データベースを構成するには、以下のようにします。

1. 以下を入力して、現在の設定を検査する。

```
db2 get database manager configuration for database
```

ここで、*database* はソース・データを含んでいるデータベースです。

出力で (LOGRETAIN) = RECOVERY が示されている場合、データベースはすでに Q キャプチャー・プログラムを実行するように構成されています。出力で (LOGRETAIN) = OFF が示されている場合、次のステップに進んでください。

2. 次のコマンドを発行する。

```
update database configuration for database using LOGRETAIN RECOVERY
```

exit ルーチンを使用して、アーカイブ・ログを管理する場合、USEREXIT データベース構成パラメーターを ON に設定しなければなりません。

3. オプション: **update database configuration** コマンドを使用し、レプリケーションまたは発行での必要に応じて、ソース・データベースの値を増やす。以下のデータベース構成値は、多くの大規模なレプリケーション・シナリオに適したものです (データベースが他のアプリケーションと協働するように構成されている場合、値がこれらよりもすでに大きくなっている場合があります): APPLHEAPSZ 4096、 LOGFILSIZ 2000、 LOGPRIMARY 20、 LOCKLIST 200、 DBHEAP 1000、 STMTHEAP 4096、 LOGBUFSZ 64、 MAXAPPLS 300、 LOCKTIMEOUT 30。

以下の公式を使用して、MAXLOCKS パラメーターの値を決定します。

$$n \geq 200 / \text{average number of concurrent connections}$$

ここで、*n* は MAXLOCKS の値です。

LOGSECOND パラメーターでは、値 20 がほとんどのシナリオに適しています。長いトランザクションのレプリケーションを想定している場合、ほとんどのログ・スペースの問題を避けるために、DB2 UDB バージョン 8.1 以降では LOGSECOND = -1 を設定するようにお勧めします。

4. ソース・データベースに適切なパラメーターを使用し、**backup database** コマンドを発行する。

次のタスク:

- Q アプライ・プログラムを処理するように、ターゲット・データベースを構成する (Q レプリケーションのみ)。詳細については、80 ページの『Q アプライ・プログラムが動作するターゲット・データベースの構成 (Linux、UNIX、Windows)』を参照してください。

関連概念:

- 15 ページの『Q キャプチャー・プログラム』

Q アプライ・プログラムが動作するターゲット・データベースの構成 (Linux、UNIX、Windows)

Q アプライ・プログラムは、高度な並列処理であり、さまざまなレプリケーション・ワークロードに合わせて構成できます。Q アプライ・プログラムの構成の仕方に応じて、MAXAPPLS (アクティブ・アプリケーションの最大数) パラメーターが適切に設定されているかを確認する必要があります。Q アプライ・プログラムは、ターゲットへのトランザクションのアプライのワークロードを分割するために指定した数に基づいて、複数のエージェントを使用します。データベースは各エージェントを、接続を試行する個々のアプリケーションとして扱います。

Q アプライ・プログラムと協働するターゲット・データベースの構成は、データベースの構成というさらに大きなタスクの一部です。

手順:

特定のレプリケーション・シナリオに基づいて MAXAPPLS パラメーターを設定するには、以下のようにします。

1. 次のコマンドを発行する。

```
update database configuration for database using MAXAPPLS n
```

ここで、*database* はターゲット・データベースを、*n* はターゲット・データベースで同時に実行できるアプリケーションの最大数です。

n を判別するには、以下の公式を使用します。

$$n \geq (\text{number of applications other than Q Apply that can use the database at the same time}) + (3 * \text{the number of Q Apply programs on the database}) + (\text{number of receive queues, each with a browser thread} + \text{total number of Q Apply agents for all receive queues})$$

これは、以下のシナリオに基づいた、この計算結果の例です。

- Q アプライ・プログラム以外にアプリケーションが 2 つある
- Q アプライ・プログラムが 2 つあり、1 つは 3 つのエージェントを使用して単一の受信キューを処理し、もう 1 つは 12 のエージェントを使用して 4 つの受信キューを処理します (そのため、5 つの受信キューを処理するために合計で 5 つのブラウザー・スレッドがあります)。

$$n \geq 2 + (3 * 2) + (5 + 15)$$

このシナリオでは、MAXAPPLS を 28 以上に設定する。

2. オプション: LOAD ユーティリティの CURSOR からの LOAD オプションを使用して、Q アプライ・プログラムに自動的にターゲットをロードさせる場合、以下のコマンドを発行する。

```
update dbm cfg using FEDERATED YES
```

3. オプション: **update database configuration** コマンドを使用し、レプリケーションでの必要に応じて、ターゲット・データベースの値を増やす。以下のデータベース構成値は、多くの大規模なレプリケーション・シナリオに適したものです

(データベースが他のアプリケーションと協働するように構成されている場合、値がこれらよりもすでに大きくなっている場合があります): APPLHEAPSZ 4096、LOGFILSIZ 2000、LOGPRIMARY 20、LOCKLIST 200、DBHEAP 1000、STMTHEAP 4096、LOCKTIMEOUT 30。

以下の公式を使用して、MAXLOCKS パラメーターの値を決定します。

$n \geq 200 / \text{average number of concurrent connections}$

ここで、 n は MAXLOCKS の値です。

LOGSECOND パラメーターでは、値 20 がほとんどのシナリオに適しています。長いトランザクションのレプリケーションを想定している場合、ほとんどのログ・スペースの問題を避けるために、DB2 UDB バージョン 8.1 以降では LOGSECOND = -1 を設定するようにお勧めします。

LOGBUFSZ では、64 から 512 の間の値が推奨されています。

次のタスク:

- オプション: プログラム・パッケージをバインドする。詳細については、『オプション : プログラム・パッケージのバインド (Linux、UNIX、Windows) -- 概要』を参照してください。

関連概念:

- 19 ページの『Q アプライ・プログラム』

オプション : プログラム・パッケージのバインド (Linux、UNIX、Windows)

オプション : プログラム・パッケージのバインド (Linux、UNIX、Windows) -- 概要

Linux、UNIX[®]、および Windows[®] では、Q キャプチャー・プログラム、Q アプライ・プログラム、またはレプリケーション・アラート・モニターが初めてデータベースに接続した時に、プログラム・パッケージが自動的にバインドされます。しかしながらこれらのパッケージのバインドは、必要に応じて、バインド・オプションを指定して手動で行ったり、データベースでの競合が少ないと思われる時間に行うこともできます。

以下のトピックでは、各プログラムの手動でのバインド・プロセスについて説明します。

- 82 ページの『オプション : Q キャプチャー・プログラム・パッケージのバインド (Linux、UNIX、Windows)』
- 83 ページの『オプション : Q アプライ・プログラム・パッケージのバインド (Linux、UNIX、Windows)』
- 84 ページの『オプション : レプリケーション・アラート・モニター・パッケージのバインド (Linux、UNIX、Windows)』

関連概念:

- 326 ページの『Q レプリケーションおよびイベント発行のパッケージおよびプランの再バインドに関する考慮事項』

関連タスク:

- 76 ページの『Q レプリケーションおよびイベント発行用のデータベースの構成 (Linux、UNIX、Windows)』

オプション : Q キャプチャー・プログラム・パッケージのバインド (Linux、UNIX、Windows)

Linux、UNIX、および Windows では、Q キャプチャー・プログラムが初めて Q キャプチャー・サーバーに接続した時に、Q キャプチャー・プログラム・パッケージが自動的にバインドされます。バインド・オプションを指定するか、またはこのデータベースでの競合が少ないと思われる時間に、パッケージを手動でバインドするかを選択できます。以下の手順では、Q キャプチャー・プログラム・パッケージを手動でバインドする方法について説明します。

Q キャプチャー・プログラム・パッケージのバインドは、Q レプリケーションおよびイベント発行のためのデータベースの構成というさらに大きなタスクの一部です (Linux、UNIX、Windows)。

手順:

Q キャプチャー・プログラム・パッケージをバインドするには、以下のようになります。

1. 以下を入力して、Q キャプチャー・サーバーに接続する。

```
db2 connect to database
```

ここで、*database* は Q キャプチャー・サーバーです。

2. Q キャプチャーのバインド・ファイルがあるディレクトリに変更する。

Windows:

```
drive:¥...¥sqllib¥bnd
```

Linux および UNIX:

```
db2homedir/sqllib/bnd
```

ここで、*db2homedir* は DB2 インスタンスのホーム・ディレクトリです。

3. 以下のコマンドを入力し、Q キャプチャー・パッケージを作成してデータベースにバインドする。

```
db2 bind @qcapture.lst isolation ur blocking all
```

ここで *ur* は、パフォーマンスの向上のため、非コミット読み取りを指定している。

これらのコマンドは、パッケージを作成します。その名前は、*qcapture.lst* ファイル内で見つけることができます。

関連概念:

- 81 ページの『オプション : プログラム・パッケージのバインド (Linux、UNIX、Windows) -- 概要』

- 15 ページの『Q キャプチャー・プログラム』

オプション : Q アプライ・プログラム・パッケージのバインド (Linux、UNIX、Windows)

Linux、UNIX、および Windows では、Q アプライ・プログラムが初めてターゲット・データベースに接続した時に、Q アプライ・プログラム・パッケージが自動的にバインドされます。Q アプライ・プログラムがターゲット表のロードを処理する場合は、ソース・データベースに接続した時になります。バインド・オプションを指定するか、これらのデータベースでの競合が少ないと思われる時間にパッケージを手動でバインドするかを選択できます。以下の手順では、Q アプライ・プログラム・パッケージを手動でバインドする方法について説明します。

Q アプライ・プログラム・パッケージのバインドは、Q レプリケーションおよびイベント発行のためのデータベースの構成というさらに大きなタスクの一部です (Linux、UNIX、Windows)。

手順:

Q アプライ・プログラム・パッケージをバインドするには、以下のようになります。

1. Q アプライ・プログラムのバインド・ファイルがあるディレクトリーに変更する。

Windows:

```
drive:¥...¥sqllib¥bnd
```

Linux および UNIX:

```
db2homedir/sqlib/bnd
```

ここで、*db2homedir* は DB2 インスタンスのホーム・ディレクトリーです。

2. ソース・データベースとターゲット・データベースの両方で、以下のステップを行う。
 - a. 以下のように入力して、データベースに接続する。

```
db2 connect to database
```

ここで、*database* はソースまたはターゲット・データベースです。データベースがリモート・データベースとしてカタログされている場合は、**db2 connect to** コマンドでユーザー ID とパスワードを指定する必要があります。たとえば、次のようになります。

```
db2 connect to database user userid using password
```

- b. 以下のコマンドを入力し、Q アプライ・プログラム・パッケージを作成してデータベースにバインドする。

```
db2 bind @qapply.lst isolation ur blocking all grant public
```

この *ur* は非コミット読み取りを意味している。

これらのコマンドは、パッケージを作成します。その名前は、*qapply.lst* ファイル内で見つけることができます。

関連概念:

- 81 ページの『オプション：プログラム・パッケージのバインド (Linux、UNIX、Windows) -- 概要』
- 19 ページの『Q アプライ・プログラム』

オプション：レプリケーション・アラート・モニター・パッケージのバインド (Linux、UNIX、Windows)

Linux、UNIX、および Windows では、レプリケーション・アラート・モニター・プログラムが初めてモニター・コントロール・サーバーに接続した時に、レプリケーション・アラート・モニター・パッケージが自動的にバインドされます。または、いずれかの Q キャプチャー・サーバーまたは Q アプライ・サーバーをモニターするように選択した場合は、そのサーバーに接続した時になります。バインド・オプションを指定するか、これらのデータベースでの競合が少ないと思われる時間にパッケージを手動でバインドするかを選択できます。以下の手順では、レプリケーション・アラート・モニター・パッケージを手動でバインドする方法について説明します。

レプリケーション・アラート・モニター・パッケージのバインドは、Q レプリケーションおよびイベント発行のためのデータベースの構成というさらに大きなタスクの一部です (Linux、UNIX、Windows)。

手順:

レプリケーション・アラート・モニター・パッケージをバインドするには、以下のようになります。

1. レプリケーション・アラート・モニターのバインド・ファイルが配置されているディレクトリーに変更する。

Windows:

```
drive:¥...¥sqllib¥bnd
```

Linux および UNIX:

```
db2homedir/sql/lib/bnd
```

ここで、*db2homedir* は DB2 インスタンスのホーム・ディレクトリーです。

2. それぞれのモニター・コントロール・サーバーごとに以下のステップを実行する。

- a. 次のように入力して、モニター・コントロール・サーバーのデータベースに接続する。

```
db2 connect to database
```

ここで、*database* はモニター・コントロール・サーバーです。データベースがリモート・データベースとしてカタログされている場合は、**db2 connect to** コマンドでユーザー ID とパスワードを指定する必要があります。たとえば、次のようになります。

```
db2 connect to database user userid using password
```

- b. 以下のコマンドを入力し、レプリケーション・アラート・モニター・パッケージを作成してデータベースにバインドする。

```
db2 bind @asnmnocs.lst isolation cs blocking all grant public
```



```
db2 bind @asnmonur.lst isolation ur blocking all grant public
```

なお、cs はカーソル固定を意味し、ur は非コミット読み取りを意味している。

これらのコマンドは、パッケージを作成します。その名前は、asnmoncs.lst および asnmonur.lst ファイル内で見つけることができます。

3. 現在モニターの対象としており、レプリケーション・アラート・モニターの接続先となっているそれぞれのサーバーごとに、以下のステップを実行する。

- a. 以下のように入力して、データベースに接続する。

```
db2 connect to database
```

ここで、*database* はモニターされているサーバーです。データベースがリモート・データベースとしてカタログされている場合は、**db2 connect to** コマンドでユーザー ID とパスワードを指定する必要があります。たとえば、次のようになります。

```
db2 connect to database user userid using password
```

- b. 以下のコマンドを入力し、レプリケーション・アラート・モニター・パッケージを作成して、データベースにバインドする。

```
db2 bind @asnmonit.lst isolation ur blocking all grant public
```

この ur は非コミット読み取りを意味している。

これらのコマンドは、パッケージを作成します。その名前は、asnmonit.lst ファイル内で見つけることができます。

関連概念:

- 81 ページの『オプション：プログラム・パッケージのバインド (Linux、UNIX、Windows) -- 概要』
- 291 ページの『レプリケーション・アラート・モニター』

Q レプリケーションおよびイベント発行用のデータベースの構成 (z/OS)

DB2® Information Integrator Replication for z/OS™ のインストール時に、レプリケーション・プログラムをセットアップしてカスタマイズする必要があります。「*IBM DB2 Information Integrator Replication Installation and Customization Guide for z/OS*」の手順を参照してください。

関連概念:

- 75 ページの『Q レプリケーションおよびイベント発行用のサーバーの構成 -- 概要』

関連タスク:

- 76 ページの『Q レプリケーションおよびイベント発行用のデータベースの構成 (Linux、UNIX、Windows)』

レプリケーション・センターのソフトウェア前提条件

レプリケーション・センターを実行するには、システムに正しい Java™ Runtime Environment (JRE) がインストールされていなければなりません。DB2® UDB のインストール時に、JRE をインストールするオプションがあります。これを行うことを強くお勧めします。JRE をインストールしないことを選択した場合は、システムに IBM® 承認済みの Java 2 Runtime Environment または Java 2 Software Development Kit のバージョン 1.3.1 以上があることを確認しなければなりません。

z/OS™: レプリケーション・センターから Q キャプチャー・プログラム、Q アプライ・プログラム、または Q レプリケーション・アラート・モニター・プログラムを操作するには、DB2 Administration Server (DAS) for z/OS をインストールしなければなりません。z/OS™ バッファ・プールを表示するには、390 Enablement パッケージをインストールしなければなりません。

- DB2 Administration Server 用の DB2 for OS/390® and z/OS バージョン 7 FMID は HDAS810 です。
- 390 Enablement パッケージ用の DB2 for OS/390 and z/OS バージョン 7 FMID は JDB771D です。このパッケージには、DB2 にインストールしなければならぬインストール・プロシージャが組み込まれています。
- 390 Enablement パッケージ用の DB2 for OS/390 バージョン 6 FMID は JDB661D です。

レプリケーション・センターを使用して、リモート・システム上の Q キャプチャー・プログラム、Q アプライ・プログラム、またはレプリケーション・アラート・モニター・プログラムを操作する場合は、レプリケーション・センターを実行するローカル・システムおよび、Q キャプチャー・プログラムまたは Q アプライ・プログラムを実行するそれぞれのリモート DB2 システム上で、DB2 Administration Server が実行されていることを確認してください。DAS がなくとも、構成タスクのみであれば、レプリケーション・センターの利用は可能です。DAS for z/OS は、DB2 for OS/390 and z/OS バージョン 7 またはそれ以降でのみ使用可能です。DAS のインストール後は、DB2 for OS/390 バージョン 6 サブシステムで使用することも可能です。

関連概念:

- 「管理ガイド: インプリメンテーション」の『DB2 Administration Server』
- 75 ページの『Q レプリケーションおよびイベント発行用のサーバーの構成 -- 概要』

関連タスク:

- 「管理ガイド: インプリメンテーション」の『DB2 Administration Server の作成』

Q キャプチャー・プログラムと Q アプライ・プログラムのコントロール表の作成

データを発行または複製する前に、Q キャプチャー・プログラムまたは Q アプライ・プログラム、あるいはその両方のコントロール表を作成しなければなりません。コントロール表には、Q サブスクリプションと XML 発行に関する情報、メッセージ・キュー、稼働パラメーター、およびユーザー設定が保管されています。Q キャプチャー・サーバーまたは Q アプライ・サーバーとして使用される、データベースまたは z/OS サブシステムにコントロール表を作成します。

Q キャプチャー・プログラムまたは Q アプライ・プログラムの各インスタンスには、Q キャプチャー・スキーマまたは Q アプライ・スキーマで示される、コントロール表の独自のセットがあります。たとえば、QC1 というスキーマを含む、Q キャプチャー・プログラムの稼働パラメーターを保管しているコントロール表は、QC1.IBMQREP_CAPPARMS という名前になります。

ピアツーピア・レプリケーションの場合、Q キャプチャー・プログラムまたは Q アプライ・プログラムはサーバーごとのペアとして実行されます。レプリケーション・センターを使用して、サーバーで同時にコントロール表の両方のセットを作成できます。この場合、コントロール表の両方のセットのスキーマが同じでなければなりません。

前提条件:

- コントロール表を作成するサーバーにレプリケーション・センターが接続できなければなりません。
- 以下の WebSphere MQ オブジェクトの名前を把握しておかなければなりません。

Q キャプチャー・プログラムの場合

- Q キャプチャー・プログラムが実行するシステム上のキュー・マネージャー。
- 管理キューとしての役割を果たす、ローカルの永続的なキュー。
- 再始動キューとしての役割を果たす、ローカルの永続的なキュー。

Q アプライ・プログラムの場合

- Q アプライ・プログラムが実行するシステム上のキュー・マネージャー。

重要: レプリケーション・センターは、コントロール表の作成時に指定する WebSphere MQ キュー・マネージャーおよびキュー名の妥当性検査は行いません。指定する名前は WebSphere MQ オブジェクト名と必ず一致させてください。そうしないと、Q キャプチャーまたは Q アプライ・プログラムは実行されません。

制約事項:

複数のパーティション・データベースの場合、コントロール表によって使用されるすべての表スペースが、カタログ・パーティションに置かれていなければなりません。既存の表スペースを使用する場合、その表スペースは、非パーティションであり、カタログ・パーティションに置かれていなければなりません。

手順:

Q キャプチャー・プログラムまたは Q アプライ・プログラム、あるいはその両方のコントロール表を作成するには、以下のようにします。

レプリケーション・センターの「コントロール表の作成 (Create Control Tables)」ウィザードを使用して、Q キャプチャー・プログラムまたは Q アプライ・プログラム、あるいはその両方のコントロール表を作成します。これらのウィザードをオープンするには、「**Q キャプチャー・サーバー**」フォルダーを右マウス・ボタンでクリックして、「**Q キャプチャー・コントロール表の作成**」を選択するか、または「**Q アプライ・サーバー**」フォルダーを右マウス・ボタンでクリックして、「**Q アプライ・コントロール表の作成**」を選択します。

デフォルトでは、Linux、UNIX、および Windows の場合は 1 つの表スペースに、z/OS の場合は 2 つの表スペースにコントロール表が置かれます。各コントロール表が作成される場所をカスタマイズできます。さらに、既存の表スペースを指定するか、あるいは新規の表スペースを作成できます。

「コントロール表の作成 (Create Control Tables)」ウィザードによって生成された SQL スクリプトを保管し、それを使用して、コントロール表の追加セットを作成できます。このウィザードを完了すると、「今実行するか、SQL を保管する」ウィンドウが表示されます。このウィンドウを使用して、スクリプトを即時に実行するか、ファイルとして保管するか、またはタスク・センターでタスクとしてスケジュールする (z/OS 以外) ことができます。

詳細については、オンライン・ヘルプを参照してください。

関連概念:

- 75 ページの『Q レプリケーションおよびイベント発行用のサーバーの構成 -- 概要』
- 19 ページの『Q アプライ・プログラム』
- 15 ページの『Q キャプチャー・プログラム』
- 21 ページの『Q アプライ・プログラムと Q キャプチャー・プログラムのスキーマ』
- 415 ページの『Q レプリケーションおよびイベント発行のコントロール表 -- 概要』

関連タスク:

- 200 ページの『レプリケーション・センターからの SQL スクリプトの実行および保管』

第 3 部 Q レプリケーションおよびイベント発行のセットアップ

本書の第 3 部は、以下の章で構成されています。

- 91 ページの『第 9 章 ソースからターゲットへの (単一方向の) レプリケーションのセットアップ』では、単一方向レプリケーション用の Q サブスクリプションの作成方法について説明し、変更がソース表からターゲット表に一方向で複製されるようにします。データの転送に使用する WebSphere MQ キューと、複製するデータを指定するオプションの指定方法を説明しています。
- 133 ページの『第 10 章 ソースからターゲットへのレプリケーションのセットアップ (複数方向)』では、Q サブスクリプションの作成方法について説明し、変更が表間で双方向に複製されるようにします。この章では、双方向レプリケーションとピアツーピア・レプリケーションについて説明しており、ビジネス・ニーズを満たすために適したレプリケーションのタイプを選択できます。データの転送に使用する WebSphere MQ キューと、複製するデータを指定するオプションの指定方法を説明しています。
- 163 ページの『第 11 章 Q レプリケーションにターゲット表をロードするためのオプション』では、ソース表からのデータを最初にロードするターゲット表のオプションについて説明しています。
- 179 ページの『第 12 章 ソースからの発行 (イベント発行) のセットアップ』では、XML 発行の作成方法について説明しています。これによってソース表からの変更は、任意のユーザー・アプリケーションで使用するために XML フォーマットで発行されます。この章では、変更点を転送する WebSphere MQ キューの指定方法も説明しています。
- 197 ページの『第 13 章 データ・タイプの複製および発行に関する考慮事項』では、複製および発行の一般的なデータの制限事項と、ラージ・オブジェクトの処理方法について説明しています。
- 199 ページの『第 14 章 レプリケーション・センターからの SQL スクリプトおよび操作可能コマンドの実行』では、レプリケーション・センターが生成する SQL スクリプトの変更、保管、および実行方法を説明しています。

第 9 章 ソースからターゲットへの (単一方向の) レプリケーションのセットアップ

ソースからターゲットへの (単一方向) レプリケーションのセットアップ -- 概要

単一方向レプリケーションでは、ソース表からターゲット表への単一の方向でデータを複製したり、ストアード・プロシージャを使用してターゲットでデータを操作したりできます。Q キャプチャー・プログラムは、ソース表のトランザクションを複製して、それらのトランザクションを圧縮形式で送信キューに書き込みます。その後、Q アプライ・プログラムが、その圧縮メッセージを受信キューから読み取って、トランザクションをターゲット表にアプライするか、またはストアード・プロシージャに渡します。

以下のトピックでは、単一方向レプリケーションについて、必要とするキューの指定方法、単一方向レプリケーションのセットアップ方法、およびソース・データの操作方法を説明します。

- 『単一方向レプリケーション』
- 95 ページの『レプリケーション・キュー・マップの作成』
- 98 ページの『単一方向レプリケーションの Q サブスクリプションの作成 -- 概要』
- 121 ページの『Q レプリケーションのターゲットに適用されるデータを操作するストアード・プロシージャの使用 -- 概要』

関連概念:

- 133 ページの『ソースからターゲットへのレプリケーションのセットアップ (複数方向) -- 概要』
- 179 ページの『ソースからの発行 (イベント発行) の設定 -- 概要』

単一方向レプリケーション

単一方向レプリケーションは、以下の特性のある Q レプリケーション構成です。

- ソース表で生じたトランザクションは、WebSphere® MQ キューを使用してターゲット表に複製されるか、データ操作のために入力パラメーターとしてストアード・プロシージャに渡されます。
- ターゲット表で生じたトランザクションは、ソース表に複製されて戻されることはありません。
- 通常、ターゲット表は読み取り専用ですが、そうでない場合、Q アプライ・プログラム以外のアプリケーションで更新されることはありません。

Q キャプチャー・プログラムは、ソース表からトランザクションを複製し、このトランザクションを圧縮形式で送信キューに書き込みます。その後、Q アプライ・プ

プログラムが受信キューから圧縮メッセージを受け取り、トランザクションをターゲット表にアプライするか、ストアード・プロシージャに渡します。

単一方向レプリケーションを使用して、ソース表からターゲット表にデータを複製します。どのソース表からでも、すべての列と行あるいは列と行のサブセットだけのいずれでも複製できます。データを変換する場合、Q アプライ・プログラムを指定して、ソース表からトランザクションを入力パラメーターとして規定のストアード・プロシージャに受け渡すことができます。

単一方向レプリケーションでは、以下のオブジェクトがサーバー間にあります。

レプリケーション・キュー・マップ

少なくとも 1 つのレプリケーション・キュー・マップを作成し、ソース・サーバーにある Q キャプチャー・プログラムから、ターゲット・サーバーにある Q アプライ・プログラムにデータを転送しなければなりません。

Q サブスクリプション

ソース表とターゲット表のペアごとまたはソース表とストアード・プロシージャのペアごとに 1 つの Q サブスクリプションがあります。たとえば、SERVER_RED にソース表、SERVER_GREEN にターゲット表、そして、SERVER_BLUE に別のターゲット表がある場合、以下のように 2 つの Q サブスクリプションがあります。

- SERVER_RED 上のソース表から SERVER_GREEN 上のターゲット表への Q サブスクリプション
- SERVER_RED 上のソース表から SERVER_BLUE 上のターゲット表への Q サブスクリプション

93 ページの図 9 は、単一方向レプリケーションで 3 つのソース表を同時に 3 つのターゲット表にマッピングする場合に入手するものを示しています。この例では、3 つの別個の Q サブスクリプションが存在します。ソース A からの変更はターゲット A に複製され、ソース B からの変更はターゲット B に複製され、それ以降も同様になります。ソース A からの変更をターゲット B に複製することはできません。このソース - ターゲットのペアは同一のレプリケーション・キュー・マップ、Q キャプチャー・プログラム、および Q アプライ・プログラムを使用します。

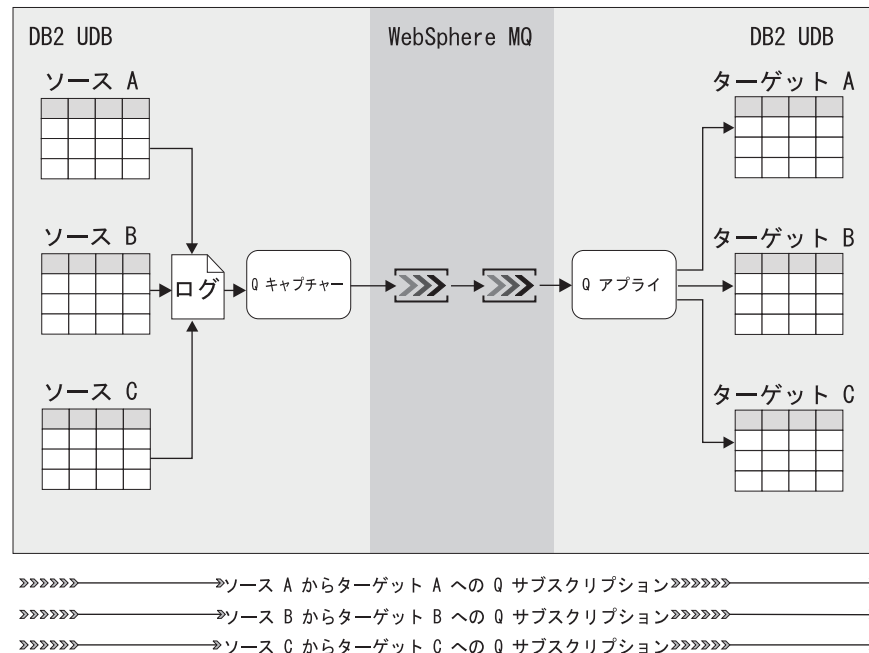


図9. 単一方向レプリケーションを行う複数の Q サブスクリプション：単一方向レプリケーションで、各ソース表からの変更は WebSphere MQ キューで、特定のターゲット表に複製します。

レプリケーション・キュー・マップと Q サブスクリプションのグループ化

Q サブスクリプションとレプリケーション・キュー・マップを定義する前に、まず Q サブスクリプションとレプリケーション・キュー・マップをどのようにグループ化するかを計画する必要があります。各 Q サブスクリプションでは、単一のソース表が、単一のターゲット表またはストアード・プロシージャと組み合わせられます。Q サブスクリプションを定義する際は、ソース表からターゲット表またはストアード・プロシージャへのデータの転送に使用するレプリケーション・キュー・マップも定義する必要があります。

特に、各レプリケーション・キュー・マップは、Q キャプチャー・プログラムが変更を送信する WebSphere® MQ キューと、Q アプライ・プログラムが、変更をターゲット表に適用したり、ストアード・プロシージャに渡したりする前に、それらの変更の受信元の WebSphere MQ キューを識別します。単一のレプリケーション・キュー・マップを、複数の Q サブスクリプションのデータの転送に使用できるため、どの Q サブスクリプションが同一のレプリケーション・キュー・マップを使用してデータを転送するかを決定する必要があります。

Q サブスクリプションとレプリケーション・キュー・マップをグループ化する場合は、以下の規則に従います。

- WebSphere MQ キューを、複数の Q キャプチャー・プログラムまたは複数の Q アプライ・プログラムで共有することはできません。
- 単一の Q キャプチャー・プログラムまたは Q アプライ・プログラムが、複数のキューに対して読み取りや書き込みを行えます。たとえば、単一の Q キャプチ

ャー・プログラムが、多数の送信キューにデータを書き込むことができ、単一の Q アプライ・プログラムが、多数の受信キューからデータを読み取ってアプライすることができます。

- Q キャプチャー・プログラムと Q アプライ・プログラムのペアとの間に、1 つ以上のレプリケーション・キュー・マップを作成できます。各 Q キャプチャー・プログラムと Q アプライ・プログラムは、複数のレプリケーション・キュー・マップを処理できます。たとえば、単一の Q キャプチャー・プログラムが、複数の送信キューにメッセージを送信することができ、Q アプライ・プログラムが、複数の受信キューからメッセージを取り込むことができます。

Q キャプチャー・プログラムによる送信キューの処理方法

レプリケーション・キュー・マップの場合、Q キャプチャー・プログラムは、アクティブ Q サブスクリプションが存在するすべての表のデータベース・ログから変更をキャプチャーします。Q キャプチャー・プログラムはこれらの変更を、データベース・ログから、対応するコミット・レコードまたはアポート・レコードを読み取るまでメモリー内に保管します。その後 Q キャプチャー・プログラムは、コミットされたトランザクションに関する情報を、その Q サブスクリプションで定義されているすべての送信キューに送信します。

Q アプライ・プログラムによる受信キューの処理方法

Q アプライ・プログラムは、特定の Q アプライ・スキーマに対して定義されたすべての受信キューに対してブラウザー・スレッドを開始します。ブラウザーごとに、Q アプライのブラウザー・スレッドが受信キューから読み取るトランザクション・メッセージが、定義したエージェント最大数までの 1 つ以上の Q アプライ・エージェントによってアプライされます。受信キューに関しては、トランザクション間にユニーク制約や外部キー間の関係に基づく従属関係がある場合、トランザクションは直列で実行されます。トランザクション間に制約の従属関係がない場合、トランザクションは可能な限り並列で実行されます。

類似した Q サブスクリプションのレプリケーション・キュー・マップによるグループ化の提案

一般的に、1 つ以上のアプリケーションによるトランザクションに関係する表については、それらの表に対して Q サブスクリプションを作成して、共通のレプリケーション・キュー・マップがそれらの表すべてによって共有されるようにする必要があります。類似した Q サブスクリプションを同一のレプリケーション・キュー・マップによってグループ化すれば、Q アプライ・プログラムがデータをターゲット表にアプライしたり、ストアード・プロシージャに渡したりする際に、トランザクションのデータ整合性が保証されます。Q アプライ・プログラムは、すでにこれらのトランザクションの変更を並列でアプライしているため、データのアプライの並列性を高めるために複数のレプリケーション・キュー・マップを作成する必要はありません。

独立したレプリケーション・キュー・マップによってデータを送信する関連トランザクションに関係する Q サブスクリプションを定義している場合、Q キャプチャー・プログラムは複数の送信キューの間でデータを分割します。受信キューに関連付けられた複数の Q アプライ・ブラウザーが独立してデータをアプライします。

従属関係のある Q サブスクリプションは、同一のレプリケーション・キュー・マップを共有する必要があります。受信キューの Q アプライ・ブラウザーはトランザクション間の従属関係を検出しますが、従属トランザクションが別個のレプリケーション・キュー・マップを使用するように Q サブスクリプションを定義していると、このような検出ができないため、望ましくない結果が生じる可能性があります。従属トランザクションが異なるレプリケーション・キュー・マップによって異なる受信キューに送信されると、ターゲット・データベースから、ソース・データベースとのトランザクションの整合性が失われる可能性があります。

複数のアプリケーションがソース・サーバーを更新しているが、これらが同一の表を更新することはない場合、ソース・サーバーからターゲット・サーバーにデータを複製するための単一の Q キャプチャー・プログラムと Q アプライ・プログラムのペアを構成する場合は、この Q キャプチャー・プログラムと Q アプライ・プログラムのペアが使用する複数のレプリケーション・キュー・マップを定義することを検討できます。各アプリケーションのトランザクションと関連した Q サブスクリプションはすべて、これらのいずれかのレプリケーション・キュー・マップによって複製されます。このような構成には、障害分離やスループットの向上などの利点があります。それぞれ独自のレプリケーション・キュー・マップを持つ Q キャプチャー・プログラムと Q アプライ・プログラムのペアを複数構成すれば、さらに高いスループットと障害分離が得られる可能性があります。ただし、これらの利点を、CPU 使用量が増えること、およびレプリケーション環境がさらに複雑になることと比較してバランスを取る必要があります。

関連概念:

- 153 ページの『ピアツーピア・レプリケーションの Q サブスクリプションの作成 -- 概要』
- 6 ページの『レプリケーション・キュー・マップ』
- 98 ページの『単一方向レプリケーションの Q サブスクリプションの作成 -- 概要』

関連タスク:

- 144 ページの『双方向レプリケーションの Q サブスクリプションの作成』
- 95 ページの『レプリケーション・キュー・マップの作成』

レプリケーション・キュー・マップの作成

Q サブスクリプションを作成する際、各 Q サブスクリプションをレプリケーション・キュー・マップと関連付けることによって、データの送信に使用する WebSphere MQ キューを指定します。レプリケーション・キュー・マップは、Q サブスクリプションを作成する前、または Q サブスクリプションの作成中のステップの 1 つとして作成することができます。

各レプリケーション・キュー・マップは、以下の WebSphere MQ キューを識別します。

送信キュー

Q キャプチャー・プログラムがソース・トランザクションおよび通知メッ

セージを送信する WebSphere MQ キュー。レプリケーション・キュー・マップを定義する際、圧縮メッセージを転送するように構成されている送信キューを選択する必要があります。

受信キュー

Q アプライ・プログラムのソース・トランザクションおよび通知メッセージの受信元 WebSphere MQ キュー。

管理キュー

Q アプライ・プログラムは、この管理キューを使用して Q キャプチャー・プログラムに制御メッセージを送信します。このキュー上で Q アプライ・プログラムが送信するメッセージにはいくつかの目的があります。たとえば、Q キャプチャー・プログラムにメッセージの送信の開始を命令する、ターゲット表のフル・リフレッシュを開始する、Q キャプチャー・プログラムに特定の Q サブスクリプションの複製の停止を命令する、などです。

レプリケーション・キュー・マップは、以下のオプションも識別します。

最大メッセージ長

Q キャプチャー・プログラムがこの送信キューに置けるメッセージの最大サイズ (KB)。この最大メッセージ長は、キューまたはキュー・マネージャーに対して定義される WebSphere MQ 最大メッセージ・サイズ属性 (MAXMSGL) 以下でなければなりません。

キュー・エラー・アクション

このレプリケーション・キュー・マップに関する WebSphere MQ キューの 1 つでエラーが発生する場合に実行されるアクション。Q キャプチャー・プログラムが以下を実行するかどうかを決める必要があります。

- キューを使用する Q サブスクリプションを非アクティブにする
- 実行を停止する

キュー・エラー・アクションは、このキューに固有の WebSphere MQ エラーが発生する際に使用されます。Q キャプチャー・プログラムは、選択オプションに関係なく、エラー・メッセージをその診断ログおよび IBMQREP_CAPTRACE 表に記録します。

Q アプライ・エージェントの数

Q プログラムがこの受信キューからのトランザクションを並行して適用するために使用するスレッド、またはエージェントの数。ソース表から受信した順にトランザクションを適用することを要求するには、Q アプライ・エージェントを 1 つだけ指定します。変更を並行してターゲット・サーバーに適用するには、複数の Q アプライ・エージェントを指定します。

Q アプライの最大メモリー使用量

Q アプライ・プログラムがこの受信キューからのメッセージのバッファとして使用するメモリーの最大量 (MB)。

ハートビート・インターバル

複製トランザクションがないときに、Q キャプチャー・プログラムが依然実行中であることを Q アプライ・プログラムに知らせるために、Q キャプチャー・プログラムがこのキュー上でメッセージを送信する頻度 (秒単位)。ハートビートは、ハートビート・インターバルが満了した後の最初のコミット

ト・インターバルで送信されます。値 0 を指定すると、Q キャプチャー・プログラムはハートビート・メッセージを送信しません。

注: このハートビート・インターバルは、WebSphere MQ チャンネルに対して定義できる WebSphere MQ パラメーター HBINT (ハートビート・インターバル) とは異なります。

前提条件:

- レプリケーション・キュー・マップおよび Q サブスクリプションをグループ化する方法を計画します。詳細については、93 ページの『レプリケーション・キュー・マップと Q サブスクリプションのグループ化』を参照してください。
- Q サブスクリプションのソース表があるサーバー上に、Q キャプチャー・プログラムのコントロール表を作成し、Q サブスクリプションのターゲット表があるサーバー上に、Q アプライ・プログラムのコントロール表を作成します。詳細については、87 ページの『Q キャプチャー・プログラムと Q アプライ・プログラムのコントロール表の作成』を参照してください。
- WebSphere MQ で適切なオブジェクトが定義されているかを確認する。詳細については、62 ページの『WebSphere MQ オブジェクトの設定要件』を参照してください。

制約事項:

Q レプリケーションとイベント発行の両方に対して同じ送信キューを使用することはできません。なぜなら、圧縮メッセージ (Q レプリケーション用) を転送するか XML メッセージ (イベント発行用) を転送するかを指定するキュー・レベルの属性が存在するからです。

手順:

レプリケーション・キュー・マップを定義するには、レプリケーション・センターの「レプリケーション・キュー・マップの作成」ウィンドウを使用します。このウィンドウをオープンするには、以下のいずれかを行います。

- キュー・マップを使用させたい Q キャプチャー・プログラムを識別する Q キャプチャー・スキーマを展開する。「レプリケーション・キュー・マップ」フォルダーを右マウス・ボタンでクリックし、「作成」を選択します。
- キュー・マップを使用させたい Q アプライ・プログラムを識別する Q アプライ・スキーマを展開する。「レプリケーション・キュー・マップ」フォルダーを右マウス・ボタンでクリックし、「作成」を選択します。

詳細については、オンライン・ヘルプを参照してください。

関連概念:

- 6 ページの『レプリケーション・キュー・マップ』

関連タスク:

- 211 ページの『レプリケーション・キュー・マップの属性の変更』
- 215 ページの『レプリケーション・キュー・マップの削除』
- 180 ページの『発行キュー・マップの作成』

単一方向レプリケーションの Q サブスクリプションの作成

単一方向レプリケーションの Q サブスクリプションの作成 -- 概要

Q レプリケーションでは、Q サブスクリプションを作成することにより、ソース表からターゲット表へのデータの複製をセットアップしたり、ストアード・プロシージャを使用してターゲットでデータを操作したりできます。Q サブスクリプションは、ソース対ターゲットの各ペアに対して作成する必要があります。各 Q サブスクリプションは、以下の情報を識別する単一のオブジェクトです。

- 変更の複製元にするソース表
- 変更内容の複製先とするターゲット表またはストアード・プロシージャ
- 複製するソース表の列および行
- レプリケーション・キュー・マップ。これは、ソース・サーバーとターゲット・サーバーの間で情報を転送する WebSphere® MQ キューを指定する。

Q レプリケーションでは、一度に 1 つ以上の Q サブスクリプションを作成できません。

重要: Q サブスクリプションは XML 発行とは異なるオブジェクトです。XML 発行では、データは Q アプライ・プログラムに対してではなく、ユーザーが選択したアプリケーションに対して発行されます。Q サブスクリプションは、データを複製するためのものであり、XML 発行は、データを発行するためのものです。ソース表の変更内容を複製し、Q アプライ・プログラムによってそれらのソースの変更内容をターゲット表にアプライしたり、データ操作のためにストアード・プロシージャに渡したりする場合は、XML 発行ではなく、Q サブスクリプションを定義してください。

以下のトピックでは、単一方向レプリケーションのための Q サブスクリプションを一度に 1 つ以上作成する方法と、その Q サブスクリプションをさらにカスタマイズするために選択できるオプションについて説明します。

- 99 ページの『Q レプリケーションのターゲット・オブジェクト・プロファイル』
- 100 ページの『単一方向レプリケーションの単一の Q サブスクリプションの作成』
- 103 ページの『単一方向レプリケーションの複数の Q サブスクリプションの作成』
- 105 ページの『Q サブスクリプションのソース列 (単一方向レプリケーション)』
- 106 ページの『Q キャプチャー・プログラムがメッセージを送信する頻度 (単一方向レプリケーション)』
- 108 ページの『行のフィルタリングのための検索条件 (単一方向レプリケーション)』
- 111 ページの『ソース列からターゲット列にマップする方法 (単一方向レプリケーション)』
- 112 ページの『ターゲットの索引列またはキー列 (単一方向レプリケーション)』

- 114 ページの『ターゲット表での予期しない条件のオプション (単一方向レプリケーション)』
- 118 ページの『Q レプリケーションのエラー・オプション』

関連概念:

- 5 ページの『Q サブスクリプション』
- 91 ページの『ソースからターゲットへの (単一方向) レプリケーションのセットアップ -- 概要』

関連タスク:

- 144 ページの『双方向レプリケーションの Q サブスクリプションの作成』
- 153 ページの『2 つのサーバーによるピアツーピア・レプリケーションの Q サブスクリプションの作成』
- 155 ページの『3 つ以上のサーバーによるピアツーピア・レプリケーションの Q サブスクリプションの作成』

Q レプリケーションのターゲット・オブジェクト・プロファイル

ターゲット・サーバーにオブジェクトを作成するときに、レプリケーション・センターが使用するレプリケーション・センターのプロファイルを設定できます。このオブジェクトとしては、表 (ターゲット表)、表スペース、索引、またはニックネームなどがあります。ターゲット・オブジェクト・プロファイルを使用すると、すべてのターゲット表、ニックネーム、およびターゲット・サーバー上に作成される、関連データベース・オブジェクトのユーザーのデフォルト命名規則を指定できます。レプリケーション・センターが使用する命名規則には、ターゲット・オブジェクト・プロファイルが入っており、作成するオブジェクトに名前を付けます。ターゲット・オブジェクト・プロファイルを修正し、レプリケーション・センターの作成したすべてのオブジェクトが企業の命名規則に従って命名されるようにする規則が入るようにします。

レプリケーション・センターには、デフォルトのターゲット・オブジェクト・プロファイルがありますが、このターゲット・オブジェクト・プロファイルを変更することができます。また、レプリケーション・センターで新規の Q サブスクリプションを作成するとき、ターゲット・オブジェクト・プロファイルのオーバーライドができます。既存のオブジェクトは、リネームされません。

レプリケーション・センターは、各ターゲット・サーバーごとに、1 つのターゲット・オブジェクト・プロファイルを保管します。Q サブスクリプションを作成するとき、レプリケーション・センターは、ターゲット・サーバーのプロファイルを使用し、ターゲット表の所有者と名前を決定します。その所有者と名前をもつ表がある場合、レプリケーション・センターは Q サブスクリプションのターゲットを既存の表として使用します。その所有者と名前をもつ表がない場合、レプリケーション・センターはその所有者と名前をもつ表を作成します。

ターゲット・オブジェクトの命名規則は、以下の 3 つの部分から構成されています。

- 接頭部
- ベース (関連データベース・オブジェクトまたはタイム・スタンプの名前のいずれか)。

- 接尾部

ターゲット表を既存の表スペースまたは新規の表スペースのどちらに作成するかも指定できます。さらに、ターゲット・サーバーが z/OS™ 上の DB2® UDB サブシステムである場合、ターゲット表スペースがソース表スペースと同一の区切りを使用しているかどうかということに加えて、表スペースの稼働パラメーターも指定できます。

最後に、これらのオブジェクト名の短縮形規則を定義できます。オブジェクト名(接頭部、ベース、接尾部)が、オペレーティング・システムの最大長を超えると、短縮形規則は、レプリケーション・センターに名前がオペレーティング・システムの許容する最大長になるように、左または右のいずれかから、名前のベースを短縮するように指示します。

単一方向レプリケーションの単一の Q サブスクリプションの作成

単一方向レプリケーションの単一の Q サブスクリプションを作成することにより、単一のソース表のデータを、単一のターゲット表に複製する方法、またはデータ操作のためにストアード・プロシージャのパラメーターに渡す方法を定義できます。

前提条件:

単一方向レプリケーションの Q サブスクリプションを作成する前に、以下のアクションを実行する必要があります。

- レプリケーション・キュー・マップと Q サブスクリプションをグループ化する方法を計画する。詳細については、93 ページの『レプリケーション・キュー・マップと Q サブスクリプションのグループ化』を参照してください。
- Q サブスクリプションのソース表があるサーバー上に、Q キャプチャー・プログラムのコントロール表を作成し、Q サブスクリプションのターゲット表があるサーバー上に、Q アプライ・プログラムのコントロール表を作成する。詳細については、87 ページの『Q キャプチャー・プログラムと Q アプライ・プログラムのコントロール表の作成』を参照してください。
- レプリケーション・キュー・マップを作成することにより、複製用のキューとそれらの属性を指定する。(このタスクは、Q サブスクリプションを作成する前か、Q サブスクリプションを定義する際に行うことができます。)詳細については、95 ページの『レプリケーション・キュー・マップの作成』を参照してください。
- Q アプライ・プログラムによって、ソースの変更をターゲット表にではなく、ストアード・プロシージャに渡す場合は、使用するストアード・プロシージャを作成する。詳細については、124 ページの『Q レプリケーションのソース・データを操作するストアード・プロシージャの作成』を参照してください。

制約事項:

Q サブスクリプションのソースやターゲットは、表示できません。

手順:

1 つのソース表から 1 つのターゲット表、またはストアード・プロシージャーへの単一方向レプリケーションの Q サブスクリプションを作成するには、次のようにします。

1. ウィザードを使用して、単一方向レプリケーションの Q サブスクリプションを作成することを指定する。

a. レプリケーション・センターから、「Q サブスクリプションの作成」ウィザードをオープンします。

このウィザードは、以下のいずれかの方法でオープンする。

- Q サブスクリプションの変更をキャプチャーする Q キャプチャー・プログラムを示す Q キャプチャー・スキーマを展開する。「サブスクリプション」フォルダーを右マウス・ボタンでクリックし、「作成」を選択します。
- Q サブスクリプションの変更をアプライする Q アプライ・プログラムを示す Q アプライ・スキーマを展開します。「サブスクリプション」フォルダーを右マウス・ボタンでクリックし、「作成」を選択します。

詳細については、オンライン・ヘルプを参照してください。

b. 「レプリケーション」ページで、単一方向レプリケーションを指定する。

2. 「サーバー」ページで、次のようにする。

a. ソース・サーバーに関する情報を指定する。

b. ターゲット・サーバーに関する情報を指定する。

c. 使用するレプリケーション・キュー・マップを指定する。適切なレプリケーション・キュー・マップがない場合は、新しいものを作成します。詳細については、95 ページの『レプリケーション・キュー・マップの作成』を参照してください。

3. 「ソース表」ページで、複製元のソース表を選択する。

4. 「ターゲット」ページで、複製先にするターゲットのタイプを指定する。以下のオプションのいずれかを指定することができます。

- 新規ターゲット表を自動的に作成する。
- 既存の表をターゲットとして使用する。
- ストアード・プロシージャーを使用してソース・データを操作する。

ターゲット・オブジェクト・プロファイルによって、既存のターゲット表が使用されるか、または新しく作成されるかが決まります。レプリケーション・センターは、プロファイルで定義されている命名スキーマに一致するオブジェクトを探し、なければそのオブジェクトを作成します。

5. 「行と列」ページで、複製するデータを指定する。

a. 複製する列を指定する。

1) 「列の選択」ウィンドウを使用して、複製する列のサブセットを指定する。詳細については、105 ページの『Q サブスクリプションのソース列 (単一方向レプリケーション)』を参照してください。

2) ソース表にある列のサブセットを発行する場合は、Q キャプチャー・プログラムが Q アプライ・プログラムにメッセージを送信するときに指定す

る。詳細については、106 ページの『Q キャプチャー・プログラムがメッセージを送信する頻度 (単一方向レプリケーション)』を参照してください。

- b. 複製する行を指定する。詳細については、108 ページの『行のフィルタリングのための検索条件 (単一方向レプリケーション)』を参照してください。
 - c. ソース列からターゲット列へのマップ方法を指定するか、どのソース列をどのターゲット列にマップするかを手動で選択する。詳細については、111 ページの『ソース列からターゲット列にマップする方法 (単一方向レプリケーション)』を参照してください。
 - d. ターゲット索引またはキーを指定する。詳細については、112 ページの『ターゲットの索引列またはキー列 (単一方向レプリケーション)』を参照してください。
6. 「エラー」ページで、エラーに対する Q アプライ・プログラムの応答方法を指定する。
- a. ターゲットでの予期しない条件に対する Q アプライ・プログラムの応答方法を指定する。詳細については、114 ページの『ターゲット表での予期しない条件のオプション (単一方向レプリケーション)』を参照してください。
 - b. エラーに対する Q アプライ・プログラムの応答方法を指定する。詳細については、118 ページの『Q レプリケーションのエラー・オプション』を参照してください。
7. 「ターゲット表のロード」ページで、次のようにする。
- a. ターゲット表のロードのオプションを指定する。詳細については、163 ページの『Q レプリケーションのターゲット表のロードのオプション -- 概要』を参照してください。
 - b. Q サブスクリプションを作成後すぐにアクティブにするかどうかを指定する。
8. 「Q サブスクリプションの検討」ページで、この Q サブスクリプションが有効なことを確認する。
- Q サブスクリプションに関して何らかの変更を加える場合、その Q サブスクリプションのプロパティを変更する。
 - Q サブスクリプションの情報が欠落している場合は、そのプロパティを変更して、これを完全にする。
9. 「サマリー」ページで、「完了」をクリックする。

関連概念:

- 98 ページの『単一方向レプリケーションの Q サブスクリプションの作成 -- 概要』
- 121 ページの『Q レプリケーションのターゲットに適用されるデータを操作するストアード・プロシージャの使用 -- 概要』
- 99 ページの『Q レプリケーションのターゲット・オブジェクト・プロファイル』

関連タスク:

- 205 ページの『Q サブスクリプションの属性の変更』
- 213 ページの『Q サブスクリプションの削除』

- 103 ページの『単一方向レプリケーションの複数の Q サブスクリプションの作成』

単一方向レプリケーションの複数の Q サブスクリプションの作成

時間を節約するため、単一方向レプリケーションのための Q サブスクリプションを一度に複数定義することができます。作成するすべての Q サブスクリプションに対する属性を定義したら、レプリケーション・センターによってそれらが作成される前に、個々の Q サブスクリプションのプロパティを変更することも選択できます。

たとえば、単一方向レプリケーションのために、一度に 3 つのソース表を 3 つのターゲット表にマップできます。この例では、3 つの別個の Q サブスクリプションが存在します。ソース A からの変更はターゲット A に複製され、ソース B からの変更はターゲット B に複製され、それ以降も同様になります。ソース A の変更内容をターゲット B に複製することはできません。これらのソース対ターゲットのペアが、同一のレプリケーション・キュー・マップ、Q キャプチャー・プログラム、および Q アプライ・プログラムを使用することはできます。

ヒント: 定義すべき多数の Q サブスクリプションがあり、それらのほとんどが同じ属性を共有する場合は、複数の Q サブスクリプションを一度に作成します。

前提条件:

単一方向レプリケーションの Q サブスクリプションを作成する前に、以下のアクションを実行する必要があります。

- レプリケーション・キュー・マップと Q サブスクリプションをグループ化する方法を計画する。詳細については、93 ページの『レプリケーション・キュー・マップと Q サブスクリプションのグループ化』を参照してください。
- Q サブスクリプションのソース表があるサーバー上に、Q キャプチャー・プログラムのコントロール表を作成し、Q サブスクリプションのターゲット表があるサーバー上に、Q アプライ・プログラムのコントロール表を作成する。詳細については、87 ページの『Q キャプチャー・プログラムと Q アプライ・プログラムのコントロール表の作成』を参照してください。
- レプリケーション・キュー・マップを作成することにより、複製用のキューとそれらの属性を指定する。(このタスクは、Q サブスクリプションを作成する前か、Q サブスクリプションを定義する際に行うことができます。)詳細については、95 ページの『レプリケーション・キュー・マップの作成』を参照してください。

制約事項:

Q サブスクリプションのソースやターゲットは、表示できません。

手順:

単一方向レプリケーションのために Q サブスクリプションを一度に複数作成するには、次のようにする。

1. ウィザードを使用して、単一方向レプリケーションの Q サブスクリプションを作成することを指定する。

- a. レプリケーション・センターから、「Q サブスクリプションの作成」ウィザードをオープンする。

このウィザードは、以下のいずれかの方法でオープンします。

- Q サブスクリプションの変更をキャプチャーする Q キャプチャー・プログラムを示す Q キャプチャー・スキーマを展開する。「サブスクリプション」フォルダーを右マウス・ボタンでクリックし、「作成」を選択します。
- Q サブスクリプションの変更をアプライする Q アプライ・プログラムを示す Q アプライ・スキーマを展開する。「サブスクリプション」フォルダーを右マウス・ボタンでクリックし、「作成」を選択します。

詳細については、オンライン・ヘルプを参照してください。

- b. 「レプリケーション」ページで、単一方向レプリケーションを指定する。
2. 「サーバー」ページで、次のようにする。
 - a. ソース・サーバーに関する情報を指定する。
 - b. ターゲット・サーバーに関する情報を指定する。
 - c. 使用するレプリケーション・キュー・マップを指定する。適切なレプリケーション・キュー・マップがない場合は、新しいものを作成します。詳細については、95 ページの『レプリケーション・キュー・マップの作成』を参照してください。
 3. 「ソース表」ページで、複製元のソース表を選択する。

一度に複数の Q サブスクリプションを作成する場合、レプリケーション・センターは、各ソース表のすべての列および行を複製するものとみなします。レプリケーション・センターが Q サブスクリプションを作成する前に、ウィザードの終わりで、ソースの列および行のサブセットだけが複製されるように個々の Q サブスクリプションを変更できます。

4. 「ターゲット表」ページで、ターゲット・オブジェクト・プロファイルを検討する。Q サブスクリプションのターゲット表が必要を満たすものとなるように、必要に応じてプロファイルを変更する。

ターゲット・オブジェクト・プロファイルによって、既存のターゲット表が使用されるか、または新しく作成されるかが決まります。レプリケーション・センターは、プロファイルで定義されている命名スキーマに一致するオブジェクトを探し、なければそのオブジェクトを作成します。

5. 「列のマップ」ページで、ソース列からターゲット列へのマップ方法を指定する。詳細については、111 ページの『ソース列からターゲット列にマップする方法 (単一方向レプリケーション)』を参照してください。
6. 「エラー」ページで、エラーに対する Q アプライ・プログラムの応答方法を指定する。
 - a. ターゲットでの予期しない条件に対する Q アプライ・プログラムの応答方法を指定する。詳細については、114 ページの『ターゲット表での予期しない条件のオプション (単一方向レプリケーション)』を参照してください。
 - b. エラーに対する Q アプライ・プログラムの応答方法を指定する。詳細については、118 ページの『Q レプリケーションのエラー・オプション』を参照してください。

7. 「ターゲット表のロード」ページで、次のようにする。
 - a. ターゲット表のロードのオプションを指定する。詳細については、163ページの『Q レプリケーションのターゲット表のロードのオプション -- 概要』を参照してください。
 - b. Q サブスクリプションが作成された時に、それらをすぐにアクティブにするかどうかを指定する。
8. 「Q サブスクリプションの検討」ページで、各 Q サブスクリプションが有効なことを確認する。
 - 個々の Q サブスクリプションに何らかの変更を加える場合は、その Q サブスクリプションのプロパティを変更します。たとえば、ある Q サブスクリプションで、ソース表の行または列のサブセットのみを複製することができます。
 - いずれかの Q サブスクリプションの情報が欠落している場合は、そのプロパティを変更して必要な情報を追加します。
9. 「サマリー」ページで、「完了」をクリックする。

関連概念:

- 98 ページの『単一方向レプリケーションの Q サブスクリプションの作成 -- 概要』
- 99 ページの『Q レプリケーションのターゲット・オブジェクト・プロファイル』

関連タスク:

- 205 ページの『Q サブスクリプションの属性の変更』
- 213 ページの『Q サブスクリプションの削除』
- 100 ページの『単一方向レプリケーションの単一の Q サブスクリプションの作成』

Q サブスクリプションのソース列 (単一方向レプリケーション)

デフォルトでは、Q サブスクリプションを作成すると、ソース表にあるすべての列の変更が、ターゲット表またはストアド・プロシージャに複製されます。しかし、単一方向レプリケーションの Q サブスクリプションを作成する場合は、ソース表にあるすべての列ではなく、列のサブセットを複製することができます。

以下の状況では、列のサブセットを複製することができます。

- ソース表にある列のすべてを、ターゲット表やストアド・プロシージャで利用可能にはしたくない。
- Q サブスクリプションのターゲットが、ソース表に定義されているデータ・タイプのすべてをサポートしているわけではない。
- ターゲット表がすでに存在し、それに含まれている列が、ソース表より少ない。

106 ページの図 10 は、列のサブセットの複製方法を示しています。

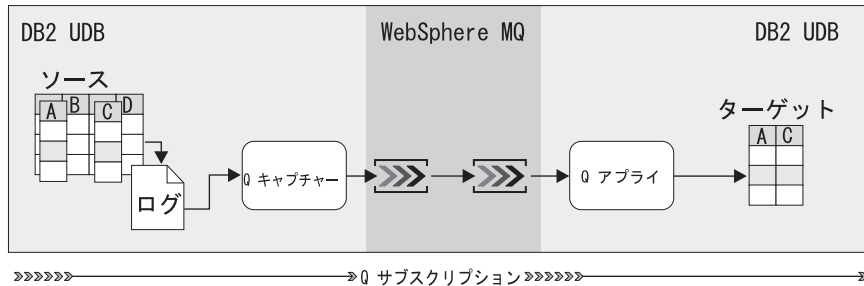


図 10. 単一方向レプリケーションの Q サブスクリプションにおける列のサブセット化：レプリケーション用に列 A および C のみが選択されている。ソース表からターゲット表へ行が複製される時、Q キャプチャー・プログラムは列 A および C の値のみを複製する。

列のサブセットを複製する場合は、ターゲットに複製するソース列のみを選択します。単一の Q サブスクリプションを作成する場合は、レプリケーション・センターの「Q サブスクリプションの作成」ウィザードに、ソース表から列のサブセットを複製する方法のオプションが示されます。複数の Q サブスクリプションを一度に作成する場合は、「Q サブスクリプションの作成」ウィザードの「レビュー」ページで、列をサブセット化する個々の Q サブスクリプションを選択し、その Q サブスクリプションのプロパティを編集してください。

キー列に関連した重要事項: ターゲットには一意性を保つための十分なキー・データが含まれていなければならないので、ターゲット表のキー列や、キー列にマップするストアド・プロシージャ内のパラメーターを構成するソース列は必ず選択してください。

LOB 列に関連した重要事項: Q サブスクリプションに対して LOB データ・タイプを含む列を選択する場合は、ソース表が少なくとも 1 つの固有のデータベース制約 (たとえば、ユニーク索引や主キー) を強制していることを確認してください。Q サブスクリプションに対して、この一意性のプロパティを構成する列を選択する必要はありません。

関連概念:

- 188 ページの『XML 発行のソース列』
- 98 ページの『単一方向レプリケーションの Q サブスクリプションの作成 -- 概要』

関連タスク:

- 103 ページの『単一方向レプリケーションの複数の Q サブスクリプションの作成』
- 100 ページの『単一方向レプリケーションの単一の Q サブスクリプションの作成』

Q キャプチャー・プログラムがメッセージを送信する頻度 (単一方向レプリケーション)

Q サブスクリプションを作成する際に、Q キャプチャー・プログラムが Q アプライ・プログラムにメッセージを送信するタイミングを指定できます。Q キャプチャー・プログラムは、Q サブスクリプションの一部である列が変更された場合のみ、

またはソース表内の列が変更されるたびに、メッセージを送信できます。以下のセクションでは、Q キャプチャー・プログラムによってメッセージが送信されることになる、2 つの異なるタイプのイベントについて説明します。

- 『Q サブスクリプションの列が変更されたときにのみメッセージを送信する』
- 『ソース表が変更されるたびにメッセージを送信する』

ソース表にあるすべての列を複製する場合、これらの 2 つのオプションの結果は、同じアクションとなります。

Q サブスクリプションの列が変更されたときにのみメッセージを送信する

デフォルトで、Q キャプチャー・プログラムは、Q サブスクリプション用に選択した列に変更があった場合にのみメッセージを送信します。

例: ソース表に 100 の列があり、それらの列うちの 25 の列を、Q サブスクリプションで複製する列として選択しているとします。Q サブスクリプションの列が変更された場合にのみメッセージを送信することを指定している場合は、Q サブスクリプションの一部である 25 の列のいずれかに変更が加えられると、Q キャプチャー・プログラムがメッセージを送信します。Q サブスクリプションの一部ではない 75 の列のいずれかに変更が加えられても、Q キャプチャー・プログラムはメッセージを送信しません。

推奨: ソース表の中のQ サブスクリプションの対象ではない列が頻繁に変更される場合は、Q サブスクリプションの対象である列の変更のみを複製してください。このオプションは、ソース表で発生するすべての変更の履歴を保持するのは望ましくない場合に使用してください。このオプションは、キューを介して送信されるデータの量を最小化します。

ソース表が変更されるたびにメッセージを送信する

ソース表で変更が発生するたびに Q キャプチャー・プログラムがメッセージを送信するよう Q サブスクリプションを定義できます。ソース表の列のサブセットのみを複製している場合、Q キャプチャー・プログラムは、Q サブスクリプションの一部でない列が変更された場合にも、メッセージを送信します。

例: ソース表に 100 の列があり、それらの列うちの 25 の列を、Q サブスクリプションで複製する列として選択しているとします。ソース表が変更されるたびにメッセージを送信することを指定している場合は、ソース表の 100 の列のいずれかに変更が加えられると、Q キャプチャー・プログラムがメッセージを送信します。

推奨事項: このオプションは、ソース表で発生するすべての変更を監査する目的で履歴を残す場合に使用してください。

関連概念:

- 98 ページの『単一方向レプリケーションの Q サブスクリプションの作成 -- 概要』

関連タスク:

- 103 ページの『単一方向レプリケーションの複数の Q サブスクリプションの作成』

- 100 ページの『単一方向レプリケーションの単一の Q サブスクリプションの作成』

行のフィルタリングのための検索条件 (単一方向レプリケーション)

デフォルトでは、単一方向レプリケーションの Q サブスクリプションを作成すると、ソース表のすべての行が、ターゲット表またはストアード・プロシージャに複製されます。しかし、単一方向レプリケーションの Q サブスクリプションを作成する場合は、検索条件を含む WHERE 文節を指定して、複製する行を識別できます。Q キャプチャー・プログラムは、DB2® リカバリー・ログ内でソース表と関連した変更を検出すると、その変更を検索条件に照らして評価し、ターゲット表やストアード・プロシージャにその変更を複製するかどうかを決定します。

単一の Q サブスクリプションを作成する場合は、レプリケーション・センターの「Q サブスクリプションの作成」ウィザードを用いて、ソース表から行のサブセットを複製するための WHERE 文節を追加することができます。複数の Q サブスクリプションを一度に作成する場合は、「Q サブスクリプションの作成」ウィザードの「レビュー」ページで、行をサブセット化する個々の Q サブスクリプションを選択し、その Q サブスクリプションのプロパティを編集して WHERE 文節を追加してください。

ターゲット表に初めからソース・データがロードされるように Q サブスクリプションを定義している場合は、Q サブスクリプションの検索条件は、ターゲット表がロードされるときに評価されます。ターゲット表のロード時に行のフィルタリングが使用されるため、ターゲット表には初めからソース表の行のサブセットが含まれません。

WHERE 文節を指定すると、現在のログ・レコードの値を使用して列を評価するかどうかを指定できます。現在のログ・レコードの値を使用して WHERE 文節の列を評価したい場合は、列名の直前にコロンを 1 つ挿入します。

現在のログ・レコードの値を使用して列を評価する WHERE 文節の例

```
WHERE :LOCATION = 'EAST' AND :SALES > 100000
```

この例で、LOCATION および SALES は、現在のログ・レコードの値を使用して評価されるソース表の列名です。ここで、Q キャプチャー・プログラムは、East での \$100,000 を超える売上を含むソース表の変更だけを送信します。列名を入力する場合、名前を二重引用符で囲んでいないと、文字は大文字に変換されます。たとえば、列名に大文字と小文字が混在している場合は、"Location" と入力します。

Q キャプチャー・プログラムは、WHERE 文節に含まれる列を複製する場合、ターゲット表やストアード・プロシージャに送信する必要のある操作のタイプを変更することが必要になる場合があります。

WHERE 文節が原因で Q キャプチャー・プログラムが操作のタイプを変更しなければならない場合の例

```
WHERE :LOCATION = 'EAST'  
AND :SALES > 100000
```

ソース表で以下の変更が発生したとします。

```
INSERT VALUES ('EAST', 50000)
UPDATE SET SALES = 200000 WHERE LOCATION = 'EAST'
```

変更前の値が WHERE 文節の検索条件に合致しないので、Q キャプチャー・プログラムは、操作を UPDATE ではなく INSERT として送信します。

同様に、変更前の値が検索条件と合致するが、変更後の値が合致しない場合、Q キャプチャー・プログラムは UPDATE を DELETE に変更します。たとえば、上記と同じ以下の WHERE 文節があるとします。

```
WHERE :LOCATION = 'EAST'
AND :SALES > 100000
```

ここで、ソース表で以下の変更が発生したとします。

```
INSERT VALUES ('EAST', 200000)
UPDATE SET SALES = 50000 WHERE LOCATION = 'EAST'
```

最初の変更である挿入は、WHERE 文節の検索条件と合致する (200000 > 100000 は成立する) ので、ターゲット表またはストアード・プロシージャに送信されます。しかし、2 番目の変更である更新は、検索条件と合致しません (50000 > 100000 は不成立)。Q キャプチャー・プログラムは、変更を DELETE として送信することにより、ターゲット表またはストアード・プロシージャからこの値が削除されるようにします。

複雑な検索条件:

Q レプリケーションでは、さらに複雑な WHERE 文節を指定できます。ただし、検索条件を複雑にするとパフォーマンスが影響を受ける場合があります。たとえば、他の表や、ソース表または他の表のレコードを参照する副選択を含むさらに複雑な WHERE 文節を指定できます。

副選択を含む WHERE 文節の例:

```
WHERE :LOCATION = 'EAST'
AND :SALES > (SELECT SUM(EXPENSE) FROM STORES WHERE STORES.DEPTNO = :DEPTNO)
```

上記の例では、Q キャプチャー・プログラムは、利益を生んだ East の変更 (売上の値が経費の合計より大きい場合) だけを送信します。副選択は STORES 表と、ソース表の LOCATION、SALES、および DEPTNO という列を参照します。

WHERE 文節に副選択が含まれた Q サブスクリプションを定義する場合は、以下の問題が発生する可能性があります。

- ソース表で変更が発生するたびに、Q キャプチャー・プログラムが STORES 表に対する大規模な選択を実行して SUM(EXPENSE) 値を計算するため、パフォーマンスが低下する場合があります。また、このタイプの選択では、表に対するロックが競合する場合があります。
- 副選択により、予期しない結果が生じる可能性があります。たとえば、副選択は現在のデータベース値と照らして評価されるため、データベースの中で EXPENSE 値が変わっても、WHERE 文節の列は古い状態であり、異なった結果となります。副選択の照会結果が変わらない場合は、検索条件は正しい結果を示します。副選択で参照される表名が変わらない場合は、検索条件は正しい結果を示します。

検索条件の制約事項:

- 列関数を副選択ステートメント内で使用しない場合は、検索条件に列関数を含めることはできません。

列関数を含む無効な WHERE 文節:

```
#-----
# Incorrect: Don't do this
#-----

WHERE :LOCATION = 'EAST' AND SUM(:SALES) > 1000000
```

レプリケーション・センターは、レプリケーション・センターが Q サブスクリプションを作成するときではなく、Q キャプチャー・プログラムが検索条件を評価するときに、検索条件の妥当性検査をします。Q サブスクリプションに無効な検索条件が含まれている場合、その無効な条件が評価されると、その Q サブスクリプションは失敗し、非アクティブにされます。

- ORDER BY または GROUP BY 文節を副選択ステートメント内で使用しない場合は、これらの文節を検索条件に含めることはできません。

GROUP BY を含む無効な WHERE 文節:

```
#-----
# Incorrect: Don't do this
#-----

WHERE :COL1 > 3 GROUP BY COL1, COL2
```

GROUP BY を含む有効な WHERE 文節:

```
WHERE :COL2 = (SELECT COL2 FROM T2 WHERE COL1=1 GROUP BY COL1, COL2)
```

- 検索条件で、列名を変更の複製元のソース表名で修飾することはできません。WHERE 文節では、列名にソース表の schema.tablename.colname、や tablename.colname、という表記を使用しないでください。ただし、副選択では、この表記を使用して列を修飾できます。

スキーマ名・表名で修飾した列名を含む無効な WHERE 文節:

```
#-----
# Incorrect: Don't do this
#-----

WHERE :ADMINISTRATOR.SALES > 100000
```

上記の WHERE 文節で、ADMINISTRATOR は複製される表であり、SALES は列名です。この無効な WHERE 文節は、administrator 表の SALES 列の値のうち、SALES が 100000 を超えるもののみを選択することを目的としています。

列名を含む有効な WHERE 文節:

```
WHERE :SALES > 100000
```

上記の WHERE 文節で、SALES は列名です。

- 検索条件で、変更前の列にあった値を参照することはできません。検索条件では、変更後の値のみを参照できます。
- 検索条件に、全選択を使用する IN 述部を含めることはできません。

IN 述部を含む無効な WHERE 文節:

```
#-----  
# Incorrect: Don't do this  
#-----  
  
WHERE :COL1 IN (SELECT .... )
```

IN 述部を含む有効な WHERE 文節:

```
WHERE :COL1 IN ('CA', 'IL')
```

- 検索条件に、EXISTS 述部を含めることはできません。
- 検索条件に比較述部 (SOME、ANY、または ALL を使用する述部) を含めることはできません。
- 検索条件で、LOB 値を参照することはできません。

関連概念:

- 189 ページの『XML 発行の行をフィルター処理する検索条件』
- 98 ページの『単一方向レプリケーションの Q サブスクリプションの作成 -- 概要』

関連タスク:

- 103 ページの『単一方向レプリケーションの複数の Q サブスクリプションの作成』
- 100 ページの『単一方向レプリケーションの単一の Q サブスクリプションの作成』

ソース列からターゲット列にマップする方法 (単一方向レプリケーション)

既存のターゲット表の場合やストアード・プロシージャの場合は、ソース列のデータをターゲット列またはストアード・プロシージャのパラメーターにマップする方法を指定できます。(ターゲット表が存在しない場合は、レプリケーション・センターが、ソース表と同じ列が含まれたターゲット表を作成します。)

重要: ソース表は、ターゲット表にあるより多くの列、またはストアード・プロシージャ・パラメーターより多くの列を複製することはできません。しかし、ターゲット表には、レプリケーション用に選択したソース列の数より多くの列が含まれていても構いません。(ストアード・プロシージャには、ソース列にマップしない余分のパラメーターを含めることはできません。) ターゲット表に、ソース列へマップしない余分の列が含まれている場合、それらの余分のターゲット列は、ターゲット・キーの一部にすることはできず、NULL 可能か、デフォルト値を持つ非 NULL のいずれかでなければなりません。

単一の Q サブスクリプションを作成する場合、レプリケーション・センターは、同一の列名とデータ・タイプを結び付けることによって、ソース表の列を、ターゲット表の列またはストアード・プロシージャのパラメーターにマップしようとします。いずれかの列が同一の列名およびデータ・タイプにマップしない場合、それらの余分の列は、手動でマップする必要があります。レプリケーション・センターでは、任意で互換列のマッピングを変更することもできます。

複数の Q サブスクリプションを一度に作成する場合は、レプリケーション・センターの「Q サブスクリプションの作成」ウィザードを使用して、ソース表の列をター

ゲット表の列やストアード・プロシージャのパラメーターにマップする方法の規則を指定できます。一度に複数の Q サブスクリプションを作成する場合、レプリケーション・センターは、ソース表のすべての列を複製するものとみなし、列マッピング規則をすべての列に適用します。いずれかの列が指定した規則どおりにマップしない場合、それらの余分の列は、手動でマップする必要があります。

Q サブスクリプションを作成する際は、ターゲット列へのソース列のマップのための以下のオプションを選択できます。

列名およびデータ・タイプによってマップ (Map by column name and data type) ソース表の各列は、同一の名前およびデータ・タイプを持つターゲット表内の列 (またはストアード・プロシージャのパラメーター) にマップされます。

列位置およびデータ・タイプによってマップ (Map by column position and data type) レプリケーション用に選択したソース表の最初の列が、ターゲット表の最初の列、またはストアード・プロシージャの最初のパラメーターにマップされ、レプリケーション用に選択したソース表の 2 番目の列が、2 番目のターゲット列またはパラメーターにマップされ、それ以降も同様になります。ターゲットの列は、左から右の順にマップされます。マップされた列の各ペアのデータ・タイプは、同じでなければなりません。

関連概念:

- 98 ページの『単一方向レプリケーションの Q サブスクリプションの作成 -- 概要』

関連タスク:

- 103 ページの『単一方向レプリケーションの複数の Q サブスクリプションの作成』
- 100 ページの『単一方向レプリケーションの単一の Q サブスクリプションの作成』

ターゲットの索引列またはキー列 (単一方向レプリケーション)

Q アプライ・プログラムは、ターゲット表のまたはストアード・プロシージャ・パラメーターに行をアプライする際に、各行を固有にするための何らかのメカニズムを必要とします。レプリケーション・センターは、ターゲットで固有性を識別するために用いる列を選択します。ターゲットの行を一意に識別するためにレプリケーション・センターが選択した列は、ソース表をターゲット表またはストアード・プロシージャにマップする際にオーバーライドできます。Q アプライ・プログラムは、ターゲットで主キー、ユニーク制約、またはユニーク索引を構成するこれらの列を使用して、すでにアプライ済みの変更をトラッキングします。

単一の Q サブスクリプションを作成する場合は、レプリケーション・センターの「Q サブスクリプションの作成」ウィザードを用いて、ターゲットの行を一意に識別するための列を選択できます。複数の Q サブスクリプションを一度に作成する場合は、「Q サブスクリプションの作成」ウィザードの「レビュー」ページで、ターゲットで固有性を識別するために用いる列をカスタマイズできます。

レプリケーション・センターは、ソース表で固有性を検出すると、ターゲットの索引列またはキーを推奨します。レプリケーション・センターがソースで固有性を検出

できない場合は、ターゲットのユニーク索引とする列を指定する必要があります。推奨を受け入れるか、ターゲットの主キー、ユニーク制約、またはユニーク索引とする列を指定することができます。

レプリケーション・センターが行う推奨は、ターゲット表がすでに存在しているかどうかによって異なります。以下のセクションでは、レプリケーション・センターが、ターゲットのキーまたは索引を推奨する際に使用するロジックについて説明します。

- 『新規ターゲット表のターゲット・キーまたはターゲット索引』
- 『Q サブスクリプションの既存のターゲット表のターゲット・キーまたはターゲット索引』

新規ターゲット表のターゲット・キーまたはターゲット索引

レプリケーション・センターは、新規ターゲット表用の主キー、ユニーク制約、またはユニーク索引を推奨するとき、ソース表に以下のいずれかの定義があるかどうかをこの順序でチェックします。

1. 主キー
2. ユニーク制約
3. ユニーク索引

レプリケーション・センターがソース表のこれらのいずれかの定義を検出し、それらのソース列がレプリケーション用に選択されている場合、レプリケーション・センターはソース表の主キー（またはユニーク制約かユニーク索引）をターゲットのキーとして使用します。レプリケーション・センターは、SQL を生成して新規ターゲット表を作成するとき、ユーザーが指定するキーまたは索引を使用して作成します。デフォルトの索引名とスキーマを使用できますが、デフォルトをユーザーの命名規則に一致するように変更することもできます。レプリケーション・センターが主キー、ユニーク制約、またはユニーク索引を検出できない場合は、キー列を選択する必要があります。

Q サブスクリプションの既存のターゲット表のターゲット・キーまたはターゲット索引

レプリケーション・センターは、既存のターゲット表用のキーまたは索引を推奨するとき、まずターゲット表に主キー、ユニーク制約、またはユニーク索引がすでに存在するかどうかをチェックします。レプリケーション・センターは、ターゲット表で固有性を検出すると、それらの列が、ソース表から複製する列として選択した列に含まれていることを確認します。さらにレプリケーション・センターは、ソース表でこれらの列が使用されているかどうかにもチェックし、ソース表での固有性を強制的に適用します。ソース表とターゲット表の間に、キー列または索引列の完全一致が少なくとも 1 つあれば、レプリケーション・センターは、それらの列を使用してターゲット行の固有性を確立するよう推奨します。

ターゲット表に固有性がない場合、レプリケーション・センターは、ソース表に以下のいずれかの定義がないか、この順でチェックします。

1. 主キー
2. ユニーク制約
3. ユニーク索引

レプリケーション・センターがソース表のこれらのいずれかの定義を検出し、それらのソース列がレプリケーション用に選択されている場合、レプリケーション・センターは、このソース表のソース表の主キー、ユニーク制約、またはユニーク索引を推奨します。

関連概念:

- 98 ページの『単一方向レプリケーションの Q サブスクリプションの作成 -- 概要』

関連タスク:

- 103 ページの『単一方向レプリケーションの複数の Q サブスクリプションの作成』
- 100 ページの『単一方向レプリケーションの単一の Q サブスクリプションの作成』

ターゲット表での予期しない条件のオプション (単一方向レプリケーション)

Q アプライ・プログラムは、ソース表で発生する変更でターゲットを更新します。他のアプリケーションもそのターゲットを変更していると、Q アプライ・プログラムは、予期していなかった行をターゲットで検出する可能性があります。たとえば、Q アプライ・プログラムが、他のアプリケーションによってすでに削除されたターゲット行を更新しようとしていることがあります。選択するオプションは、どの程度細かく問題を分離して修正するか依存します。

多くのシナリオでは、Q アプライ・プログラムに、変更を強制的に適用させるか、あるいはターゲット・データ内の予期しない条件を無視させることができます。しかし、あるシナリオでは、ターゲット・データ内で問題が起こるとは思われないために、問題のトラブルシューティングのレベルに必要なと思われるレベルに応じた別のアクションを選択することもできます。

ストアード・プロシージャの場合の推奨事項: Q アプライ・プログラムがストアード・プロシージャにデータを渡す場合は、ストアード・プロシージャの動作に応じたアクションを選択する必要があります。Q アプライ・プログラムがストアード・プロシージャにデータを渡すほとんどのシナリオにおいては、行操作を変えることにより変更を強制的に適用することは、ほとんどの場合、望ましくありません。データを操作するストアード・プロシージャが関係するほとんどのシナリオでは、Q アプライ・プログラムがターゲット・データ内の予期しない条件を無視するように指定できます。ただし、その行の SQL ステートメントを再ビルドして一部のデータのみを変換する (たとえば、米ドルから欧州ユーロへの変換のみを行う) ストアード・プロシージャがある場合は、変更を強制的に適用することを検討できます。

Q アプライ・プログラムがターゲット・データ内で予期しない条件を検出したときに、以下のいずれかのアクションを取るよう指定できます。

- 115 ページの『変更の強制的な適用』
- 115 ページの『予期しない条件の無視』
- 115 ページの『対応する Q サブスクリプションの非アクティブ化』

- 116 ページの『Q アプライ・プログラムによる、対応する受信キューからの読み取りの停止』

予期しない条件に対して選択されているオプションに関係なく、Q アプライ・プログラムは、行の処理中に問題を検出すると、予期しない条件を IBMQREP_APPLYTRACE 表と診断ログ・ファイルに記録します。また、エラー行の XML バージョンが、IBMQREP_EXCEPTIONS 表に挿入されます。

変更の強制的な適用

Q アプライ・プログラムは、ターゲット・データ内で予期しない条件を検出すると、ソース表の変更を、ターゲット表またはストアド・プロシージャ・パラメーターへ強制的に適用します。Q アプライ・プログラムは、変更をターゲット表にアプライしたり、ストアド・プロシージャ・パラメーターに渡したりできるように、操作を (たとえば、挿入から更新へ) 変更します。その後、変更の再アプライを試行します。

推奨事項: Q アプライ・プログラムがソース表から受け取るデータをすべてターゲットに反映したい場合は、変更を強制的に適用できます。

予期しない条件の無視

Q アプライ・プログラムは、ターゲット・データ内で予期しない条件を検出した場合、その予期しない条件を無視し、その行をアプライせずに、そのエラーとアプライされなかった行をログに記録してから、残りのトランザクションを完了してコミットします。ターゲットにあるデータがすべて優先され、ターゲットのデータは上書きされません。ただし、Q アプライ・プログラムが予期しない条件を無視するよう選択している場合は、一部のデータがターゲットにアプライされません。アプライされなかった行はすべて IBMQREP_EXCEPTIONS 表に記録されます。

推奨事項: 次の場合は、Q アプライ・プログラムがターゲット・データでの予期しない条件を無視するようことができます。

- シナリオにおいて、ソース表およびターゲット表でのデータ収束は重要でない。
- すべての SQL 状態が予想され、容認できる。

対応する Q サブスクリプションの非アクティブ化

Q アプライ・プログラムは、ターゲット・データ内で予期しない条件を検出すると、予期しない条件が発生した Q サブスクリプションのみを非アクティブ化 (または停止) し、他の Q サブスクリプションの変更は引き続きアプライします。Q アプライ・プログラムは、そのエラーとアプライされなかった行を記録してから、残りのトランザクションを完了してコミットします。Q キャプチャー・プログラムは、非アクティブ化された Q サブスクリプション用のソース表で発生する変更のキャプチャーを停止します。このオプションでは、特定の表の問題が最も細かい細分度でトラブルシューティングされます。この場合は、その Q サブスクリプションの問題をチェックし、それを修正してから、その Q サブスクリプションをアクティブ化できます。Q サブスクリプションが非アクティブ化された場合、その Q サブスクリプションを再びアクティブ化する際には、その Q サブスクリプションのターゲットへの再ロードが必要ということに留意してください。

推奨事項:

- 次の場合は、ターゲット・データ内で予期しない条件が発生したときに、対応する Q サブスクリプションを非アクティブ化することを選択できます。
 - 同一のレプリケーション・キュー・マップを使用して複製される、非関連の複数の表に対して定義された複数の Q サブスクリプションがある。
 - 表がほとんど変更されない (特にこの表が親で、親キー列がまれにしか変更されない場合)。
- 次の場合は、ターゲット・データ内で予期しない条件が発生したときに、対応する Q サブスクリプションを非アクティブ化しないことを選択できます。
 - Q サブスクリプションの表に、他の表との参照保全があるため、1 つの Q サブスクリプションを非アクティブ化すると、他の表が影響を受ける可能性がある。表が子表である場合は、(DELETE RESTRICT がオンの場合に) 参照保全エラーのために、親表で発生する削除または更新が、子表で失敗する可能性があります。すべての Q サブスクリプションが参照保全に関係したエラーで終わり、エラーの結果として、すべてが事実上使用不可になる可能性があります。
 - Q サブスクリプションを再びアクティブ化するとき、Q サブスクリプションを再ロードする必要がある場合に、ターゲットを再ロードしたくない。
 - Q サブスクリプションの非アクティブ化は、シナリオにおける、ターゲット・データ内で予期しない条件が発生した場合の結果としては重大すぎる。

Q アプライ・プログラムによる、対応する受信キューからの読み取りの停止

Q アプライ・プログラムは、ターゲット・データ内で予期しない条件を検出すると、エラーのあった Q サブスクリプションだけでなく、受信キュー上にあるすべての Q サブスクリプションの変更のアプライを停止します。Q アプライ・プログラムは、そのエラーとアプライされなかった行を記録し、残りのトランザクションの完了やコミットはしません。ターゲット・データ内の予期しない条件の影響を受けるトランザクションは、ロールバックされます。

Q アプライ・プログラムは、他の受信キューからデータを読み取る場合、引き続きそれら他の受信キューのデータを処理します。他の受信キューからデータを読み取らない場合、Q アプライ・プログラムはシャットダウンします。これらの Q サブスクリプションの Q キャプチャー・プログラムは、引き続き受信キューにデータを送信するので、受信キューが満杯になる前に、問題を訂正して受信キューを再始動する必要があります。Q キャプチャー・プログラムは、送信キューにデータを書き込めなくなると、レプリケーション・キュー・マップに対して指定されているエラー・オプションに応じて、その送信キューを使用するすべての Q サブスクリプションを非アクティブ化するか、またはシャットダウンします。

推奨事項:

- 次の場合は、ターゲット・データ内で予期しない条件が発生したときに、Q アプライ・プログラムによる受信キューからのトランザクションのアプライを停止することができます。
 - シナリオにおいて、ターゲット・データ内の予期しない条件は許容できず、頻繁には発生しないものと予期される。
 - 関連する複数の表に対して定義された複数の Q サブスクリプションがあり、それらの Q サブスクリプションが同一のレプリケーション・キュー・マップ

を共有しているため、1 つの Q サブスクリプションのターゲット・データに予期しない条件がある場合は、すべての Q サブスクリプションを停止したい。この場合は、その 1 つの Q サブスクリプションの何が悪かったのかをチェックし、その Q サブスクリプションを修正してから、受信キューを再始動できます。このオプションでは、トランザクションを保持し、関連する表の参照保全を維持することができるので、ターゲット表に複製する変更がほとんどない場合は、このオプションの使用が適している可能性があります。

- ターゲット・データ内で予期しない条件が発生したときに、Q アプライ・プログラムによる受信キューからのトランザクションのアプライが停止されるという結果がシナリオにおいて重大すぎる場合は、これを停止しないようにできます。

Q アプライ・プログラムの停止

Q アプライ・プログラムは、ターゲット・データ内で予期しない条件を検出するとシャットダウンしますが、受信キューは引き続き Q キャプチャー・プログラムからソース・データを受信します。Q アプライ・プログラムは、そのエラーとアプライされなかった行を記録し、残りのトランザクションの完了やコミットはしません。ターゲット・データ内の予期しない条件の影響を受けるトランザクションは、ロールバックされます。Q アプライ・プログラムが 1 つの受信キューからしか読み取らない場合、このオプションの結果は、Q アプライ・プログラムによる、受信キュー上のすべての Q サブスクリプションの変更のアプライを停止する場合と同じになります。

シャットダウンは、Q アプライ・プログラムに対して設定できる、ターゲット・データで予期しない条件が発生した場合の最も重大な応答です。Q キャプチャー・プログラムは、引き続き受信キューにデータを送信するので、受信キューが満杯になる前に、問題を訂正して受信キューを再始動する必要があります。Q キャプチャー・プログラムは、送信キューにデータを書き込めなくなると、レプリケーション・キュー・マップに対して指定されているエラー・オプションに応じて、その送信キューを使用するすべての Q サブスクリプションを非アクティブ化するか、またはシャットダウンします。

推奨事項:

- 次の場合は、ターゲット・データ内で予期しない条件が発生したときに Q アプライ・プログラムを停止することができます。
 - 関連する複数の表に対して定義された複数の Q サブスクリプションがあり、複数のレプリケーション・キュー・マップによってデータが転送される。1 つの Q サブスクリプションのターゲット・データ内に予期しない条件がある場合は、すべての Q サブスクリプションを停止したい。この場合は、その 1 つの Q サブスクリプションの何が悪かったのかをチェックし、その Q サブスクリプションを修正してから、受信キューを再始動できます。このオプションでは、トランザクションを保持し、関連した表の参照保全を維持することができるので、ターゲット表に複製する変更がほとんどない場合は、このオプションの使用が適している可能性があります。
 - 構成を簡単にモニターしたい。この Q アプライ・プログラムの停止オプションは、Q アプライ・プログラムによる受信キューからのトランザクションのアプライの停止と似ています。しかし、単に特定の受信キューからの読み取りを

停止する代わりに Q アプライ・プログラムがシャットダウンする場合は、さらに簡単に環境をモニターできる可能性があります。このオプションは、たとえば、テストの際に選択できます。

- ターゲット・データ内で予期しない条件が発生したときに、Q アプライ・プログラムが停止されるという結果がシナリオにおいて重大すぎる場合は、これを停止しないようにできます。

関連概念:

- 98 ページの『単一方向レプリケーションの Q サブスクリプションの作成 -- 概要』

関連タスク:

- 103 ページの『単一方向レプリケーションの複数の Q サブスクリプションの作成』
- 100 ページの『単一方向レプリケーションの単一の Q サブスクリプションの作成』

Q レプリケーションのエラー・オプション

Q レプリケーションでは、環境で Q アプライ・プログラムが SQL エラーなどのエラーを検出したときに取るアクションを指定できます。選択するオプションは、どの程度細かく問題を分離して修正するか依存します。単一方向と複数方向のどちらのレプリケーションにも、同一のエラー・オプションが適用されます。

Q アプライ・プログラムがエラーを検出したときに取るアクションとして、以下のいずれかを選択できます。

- 『対応する Q サブスクリプションの非アクティブ化』
- 119 ページの『Q アプライ・プログラムによる、対応する受信キューからの読み取りの停止』
- 120 ページの『Q アプライ・プログラムの停止』

選択されているエラー・オプションに関係なく、Q アプライ・プログラムは、エラーを検出すると、エラーを IBMQREP_APPLYTRACE 表とその診断ログ・ファイルに記録し、アプライされなかった行の XML バージョンとエラーについての詳細を IBMQREP_EXCEPTIONS 表に挿入します。

対応する Q サブスクリプションの非アクティブ化

Q アプライ・プログラムはエラーを検出すると、エラーが発生した Q サブスクリプションのみを非アクティブ化 (または停止) し、他の Q サブスクリプションの変更は引き続きアプライします。Q アプライ・プログラムは、そのエラーとアプライされなかった行を記録してから、残りのトランザクションを完了してコミットします。Q キャプチャー・プログラムは、非アクティブ化された Q サブスクリプション用のソース表で発生する変更のキャプチャーを停止します。このオプションでは、特定の表の問題が最も細かい細分度でトラブルシューティングされます。その Q サブスクリプションの問題をチェックし、それを修正してから、その Q サブスクリプションをアクティブ化できます。Q サブスクリプションが非アクティブ化された場合、その Q サブスクリプションを再びアクティブ化する際には、その Q サブスクリプションのターゲットへの再ロードが必要ということに留意してください。

推奨事項:

- 次の場合は、エラーが発生したときに、対応する Q サブスクリプションを非アクティブ化できます。
 - 同一のレプリケーション・キュー・マップを使用して複製される、非関連の複数の表に対して定義された 2 つの Q サブスクリプションがある。
 - 表がほとんど変更されない (特にこの表が親で、親キー列がまれにしか変更されない場合)。
- 次の場合は、エラーが発生したときに、対応する Q サブスクリプションを非アクティブ化しないようにできます。
 - Q サブスクリプションの表に、他の表との参照保全があるため、1 つの Q サブスクリプションを非アクティブ化すると、他の表が影響を受ける可能性がある。表が子表である場合は、(DELETE RESTRICT がオンの場合に) 参照保全エラーのために、親表で発生する削除または更新が、子表で失敗する可能性があります。すべての Q サブスクリプションが参照保全に関係したエラーで終わり、エラーの結果として、すべてが事実上使用不可になる可能性があります。
 - Q サブスクリプションを再びアクティブ化するときに、Q サブスクリプションを再ロードする必要がある場合に、ターゲットを再ロードしたくない。
 - Q サブスクリプションを非アクティブ化するというのは、シナリオにおける、エラーが発生した場合の結果としては重大すぎる。

Q アプライ・プログラムによる、対応する受信キューからの読み取りの停止

Q アプライ・プログラムは、エラーを検出すると、エラーのあった Q サブスクリプションだけでなく、受信キュー上にあるすべての Q サブスクリプションの変更のアプライを停止します。Q アプライ・プログラムは、そのエラーとアプライされなかった行を記録し、残りのトランザクションの完了やコミットはしません。エラーの影響を受けるトランザクションは、ロールバックされます。Q アプライ・プログラムは、他の受信キューからデータを読み取る場合、それら他の受信キューのデータの処理を継続します。他の受信キューからデータを読み取らない場合、Q アプライ・プログラムはシャットダウンします。Q キャプチャー・プログラムは受信キューへのデータの送信を継続するため、受信キューが満杯になる前に、問題を訂正して受信キューを再始動する必要があります。Q キャプチャー・プログラムは、送信キューに書き込めなくなると、レプリケーション・キュー・マップに対して指定されているエラー・オプションに応じて、その送信キューを使用するすべての Q サブスクリプションを非アクティブ化するか、またはシャットダウンします。

推奨事項:

- 次の場合は、エラーが発生したときに、Q アプライ・プログラムによる受信キューからのトランザクションのアプライを停止できます。
 - シナリオにおいて、エラーは許容できず、頻繁には発生しないものと予期される。
 - 関連する複数の表に対して定義された複数の Q サブスクリプションがあり、それらの Q サブスクリプションが同一のレプリケーション・キュー・マップを共有しているので、1 つの Q サブスクリプションにエラーがある場合は、すべての Q サブスクリプションを停止したい。この場合は、その 1 つの Q

サブスクリプションの問題をチェックし、その Q サブスクリプションを修正してから、受信キューを再始動できます。このオプションでは、トランザクションを保持し、関連した表の参照保全を維持することができるので、ターゲット表に複製する変更がほとんどない場合は、このオプションの使用が適している可能性があります。

- エラーが発生したときに、Q アプライ・プログラムによる受信キューからのトランザクションのアプライが停止されるという結果が自分のシナリオにおいて重大すぎる場合は、これを停止しないようにできます。

Q アプライ・プログラムの停止

Q アプライ・プログラムは、エラーを検出するとシャットダウンしますが、受信キューは引き続き Q キャプチャー・プログラムからソース・データを受信します。

Q アプライ・プログラムは、そのエラーとアプライされなかった行を記録し、残りのトランザクションの完了やコミットはしません。エラーの影響を受けるトランザクションは、ロールバックされます。Q アプライ・プログラムが 1 つの受信キューからしか読み取らない場合、このオプションの結果は、Q アプライ・プログラムによる受信キュー上のすべての Q サブスクリプションの変更のアプライを停止する場合と同じになります。

シャットダウンは、Q アプライ・プログラムに対して設定できる、エラーが発生した場合の最も重大な応答です。これらの Q サブスクリプションの Q キャプチャー・プログラムは受信キューへのデータの送信を継続するため、受信キューが満杯になる前に、問題を訂正して受信キューを再始動する必要があります。Q キャプチャー・プログラムは、送信キューにデータを書き込めなくなると、レプリケーション・キュー・マップに対して指定されているエラー・オプションに応じて、その送信キューを使用するすべての Q サブスクリプションを非アクティブ化するか、またはシャットダウンします。

推奨事項:

- 次の場合は、エラーが発生したときに、Q アプライ・プログラムによる受信キューからのトランザクションのアプライを停止できます。
 - シナリオにおいて、エラーは許容できず、頻繁には発生しないものと预期される。
 - 関連する複数の表に対して定義された複数の Q サブスクリプションがあり、複数のレプリケーション・キュー・マップによってデータが転送される。1 つの Q サブスクリプションにエラーがある場合は、すべての Q サブスクリプションを停止したい。この場合は、その 1 つの Q サブスクリプションの問題をチェックし、その Q サブスクリプションを修正してから、受信キューを再始動できます。このオプションでは、トランザクションを保持し、関連した表の参照保全を維持することができるので、ターゲット表に複製する変更がほとんどない場合は、このオプションの使用が適している可能性があります。
- エラーが発生したときに、Q アプライ・プログラムによる受信キューからのトランザクションのアプライが停止されるという結果がシナリオにおいて重大すぎる場合は、これを停止しないようにできます。

関連タスク:

- 144 ページの『双方向レプリケーションの Q サブスクリプションの作成』

- 153 ページの『2 つのサーバーによるピアツーピア・レプリケーションの Q サブスクリプションの作成』
- 155 ページの『3 つ以上のサーバーによるピアツーピア・レプリケーションの Q サブスクリプションの作成』
- 103 ページの『単一方向レプリケーションの複数の Q サブスクリプションの作成』
- 100 ページの『単一方向レプリケーションの単一の Q サブスクリプションの作成』

ターゲットに適用されるデータを操作するストアード・プロシージャの使用

Q レプリケーションのターゲットに適用されるデータを操作するストアード・プロシージャの使用 -- 概要

単一方向レプリケーションの Q サブスクリプションを作成する際、変更がソース表からターゲット表に直接ではなく、ストアード・プロシージャに複製されることを指定できます。データがターゲット表に適用される前に、ストアード・プロシージャを使用して、ソースからキャプチャーされるデータを操作することができます。

以下のトピックでは、ストアード・プロシージャの作成方法について説明します。

- 122 ページの『Q レプリケーションのソース・データを操作するストアード・プロシージャ』
- 124 ページの『Q レプリケーションのソース・データを操作するストアード・プロシージャの作成』
- 126 ページの『Q レプリケーションの操作タイプを識別するストアード・プロシージャ・パラメーター』
- 128 ページの『各ソース列が抑制されたかどうかを識別するストアード・プロシージャ・パラメーター』
- 129 ページの『Q レプリケーションのトランザクションを識別するストアード・プロシージャ・パラメーター』
- 129 ページの『Q レプリケーションのソース列にマップするストアード・プロシージャ・パラメーター』

関連概念:

- 91 ページの『ソースからターゲットへの (単一方向) レプリケーションのセットアップ -- 概要』

関連タスク:

- 100 ページの『単一方向レプリケーションの単一の Q サブスクリプションの作成』

Q レプリケーションのソース・データを操作するストアード・プロシージャ

一般的に Q サブスクリプションでは、ソース表からのデータはターゲット表にマップされます。しかし、単一方向レプリケーションの Q サブスクリプションの場合、Q アプライ・プログラムにストアード・プロシージャの呼び出しを実行させて、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、ターゲット列ではなく、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。ソース列をストアード・プロシージャ内のパラメーターに直接マップすることにより、着信データを構文解析する必要がなくなり、クリーンで単純なプログラミング・モデルを保つことができます。

ターゲット表がある Q サブスクリプションに対して有効なオプションはすべて同様に有効です。たとえば、ストアード・プロシージャでも同様にエラー・オプションを選択できます。たとえば、ストアード・プロシージャで引き続きエラー・オプションを選択できます。

ストアード・プロシージャは Q アプライ・プログラムに、データがターゲット表にアプライされたかどうかを示す戻りコードを戻します。ストアード・プロシージャを作成する開発者は、ストアード・プロシージャが Q アプライ・プログラムに適切な SQL 戻りコードを戻すことを確認しなければなりません。ストアード・プロシージャは、ソース・データをその最終宛先に到達させなければなりません。

例: 以下の例は、ソース表内の 5 つの列 (そのうち 2 つはキー列) からの値を受け取るストアード・プロシージャのシグニチャーを示しています。

```
CREATE PROCEDURE storedprocedure_name(  
    INOUT operation integer,  
    IN suppression_ind VARCHAR(size),  
    IN src_commit_lsn CHAR(10),  
    IN src_trans_time TIMESTAMP,  
    XParm1,  
    Parm1,  
    XParm2,  
    Parm2,  
    Parm3,  
    Parm4,  
    Parm5  
)
```

この例は、4 つの必須パラメーターを示しています。

operation

この INOUT パラメーターは、操作のタイプを識別します。

suppression_ind

この IN パラメーターは、抑制されたパラメーターを識別します。

src_commit_lsn

この IN パラメーターは、ソース・サーバーがトランザクションに対して COMMIT を発行したときのログのシーケンス番号を識別します。

src_trans_time

この IN パラメーターは、ソース・サーバーがトランザクションに対して COMMIT を発行したときのタイム・スタンプを識別します。

シグニチャーには、キー列に関する変更前の値 (before value) と変更後の値 (after value)、および非キー列に関する変更後の値 (after value) も含まれています。以下のパラメーターはソース列からの値を受け入れます。

- パラメーター Parm1 および Parm2 は、ソース表内のキー列にマップします。
- パラメーター XParm1 および XParm2 には、ソース表内のキー列の変更前の値 (before value) が入ります。
- パラメーター Parm3、Parm4、および Parm5 は、ソース表内の非キー列にマップします。

ヒント: C および SQL で作成されている Q レプリケーションのストアード・プロシージャの例については、サンプル・プログラムを参照してください。

重要: ストアード・プロシージャは COMMIT または ROLLBACK ステートメントを実行してはなりません。Q アプライ・プログラムがトランザクションをコミットまたはロールバックするからです。

LOB データに関して重要: Q アプライ・プログラムは、ターゲット表の LOB データと同じ方法で、ストアード・プロシージャの LOB データを処理します。ストアード・プロシージャを使用する Q サブスクリプションの場合、LOB 変更の複製は、Q キャプチャー・プログラムに送信を許可した LOB のサイズとメッセージのサイズに応じて、複数のメッセージで送信することによって行われます。単一の LOB 変更に対する最初の LOB メッセージを含むストアード・プロシージャ呼び出しは、元になる行操作です。単一の LOB 変更に対する残りの LOB データはすべて、追加呼び出しによってストアード・プロシージャに送信されます。Q アプライ・プログラムは、操作を更新にトランスフォームします。suppression_ind 索引は、ソース表の LOB 列にマップするパラメーター以外のすべてのパラメーターに抑制済みとしてマークを付けます。ストアード・プロシージャは LOB メッセージが受信キューに到着するたびに呼び出されます。

関連概念:

- 198 ページの『Q レプリケーションおよびイベント発行でのラージ・オブジェクト (LOB) データ・タイプに関する考慮事項』
- 32 ページの『Q レプリケーションおよびイベント発行の LOB データ・タイプのためのメモリー』
- 121 ページの『Q レプリケーションのターゲットに適用されるデータを操作するストアード・プロシージャの使用 -- 概要』

関連タスク:

- 100 ページの『単一方向レプリケーションの単一の Q サブスクリプションの作成』

関連資料:

- 529 ページの『Q レプリケーションおよびイベント発行をセットアップするためのサンプル (Linux、UNIX、Windows)』
- 532 ページの『Q レプリケーションおよびイベント発行をセットアップするためのサンプル (z/OS)』

Q レプリケーションのソース・データを操作するストアード・プロシージャの作成

このトピックでは、ストアード・プロシージャを作成してそれを Q サブスクリプションのターゲットとして使用する方法、およびストアード・プロシージャのセットアップ方法を説明します。ストアード・プロシージャを Q サブスクリプションのターゲットとして宣言するには、まずストアード・プロシージャを作成しなければなりません。

ストアード・プロシージャは 4 つの必須パラメーター (操作を識別する 1 つのパラメーターとトランザクションを識別する 3 つのパラメーター)、およびソース列にマップする追加パラメーターで構成されます。

制約事項:

- ストアード・プロシージャは COMMIT または ROLLBACK ステートメントを実行してはなりません。
- 非キー列からストアード・プロシージャのパラメーターに変更前の値 (before value) を送信することはできません。
- Q アプライ・プログラムはストアード・プロシージャに関してのみ認識し、ストアード・プロシージャが操作済みデータを渡す可能性のある表に関しては認識しないため、ストアード・プロシージャと関連付けられている Q サブスクリプションは参照保全を保守できません。
- ストアード・プロシージャがターゲット表を保守しており、Q アプライ・プログラムに競合を解決させたい場合は、ターゲット表上に 2 次固有索引を作成しないでください。

手順:

Q サブスクリプションのターゲットとなるようにストアード・プロシージャを作成し、そのストアード・プロシージャをセットアップするには、以下のようになります。

1. 4 つの必須パラメーターとソース列にマップする追加パラメーターを含むストアード・プロシージャを作成します。ストアード・プロシージャのパラメーターには定位置があります。最初のパラメーターのパラメーター・モードは INOUT でなければなりません。その他のパラメーターのパラメーター・モードはすべて IN でなければなりません。次のトピックを参照してください。
 - 126 ページの『Q レプリケーションの操作タイプを識別するストアード・プロシージャ・パラメーター』
 - 128 ページの『各ソース列が抑制されたかどうかを識別するストアード・プロシージャ・パラメーター』
 - 129 ページの『Q レプリケーションのトランザクションを識別するストアード・プロシージャ・パラメーター』。これには、以下の 2 つの必須パラメーターが含まれます。
 - ソース・サーバーがトランザクションに対してコミット・ステートメントを発行したときのログのシーケンス番号
 - ソース・サーバーがトランザクションに対してコミット・ステートメントを発行したときのタイム・スタンプ

- 129 ページの『Q レプリケーションのソース列にマップするストアード・プロシージャ・パラメーター』

ヒント: C および SQL で作成されている Q レプリケーションのストアード・プロシージャの例については、サンプル・プログラムを参照してください。

2. CREATE PROCEDURE ステートメントを正しく実行し、ストアード・プロシージャの名前と適切なパラメーターを DB2 データベースに登録します。

ヒント:

- ストアード・プロシージャを宣言するときは GENERAL WITH NULLS パラメーター・スタイルを使用します。このスタイルを使用することにより、DB2 Universal Database ファミリー間のストアード・プロシージャの移植性が向上します。
 - **Linux、UNIX、Windows:** ストアード・プロシージャが sqllib/function ディレクトリーに存在することを確認してください。
3. ターゲット・サーバーでストアード・プロシージャのコンパイルおよびバインドを実行する。

z/OS のヒント: ストアード・プロシージャの DBRM を Q アプライ・パッケージ・リストに組み込みます。

4. Q アプライ・プログラムにストアード・プロシージャを呼び出すための適切な権限があることを確認する。

ストアード・プロシージャを呼び出した後、Q アプライ・プログラムは、戻りコードを検査してストアード・プロシージャが正常かどうかを妥当性検査する前に、必ず SQL コードを検査します。

5. ストアード・プロシージャがターゲットを保持する場合、ターゲットが存在することを確認する。
6. ストアード・プロシージャを Q サブスクリプションのターゲットとして宣言します。

関連概念:

- 81 ページの『オプション：プログラム・パッケージのバインド (Linux、UNIX、Windows) -- 概要』
- 98 ページの『単一方向レプリケーションの Q サブスクリプションの作成 -- 概要』
- 121 ページの『Q レプリケーションのターゲットに適用されるデータを操作するストアード・プロシージャの使用 -- 概要』
- 122 ページの『Q レプリケーションのソース・データを操作するストアード・プロシージャ』

関連資料:

- 529 ページの『Q レプリケーションおよびイベント発行をセットアップするためのサンプル (Linux、UNIX、Windows)』
- 532 ページの『Q レプリケーションおよびイベント発行をセットアップするためのサンプル (z/OS)』

Q レプリケーションの操作タイプを識別するストアード・プロシ ージャー・パラメーター

ストアード・プロシージャー内の最初のパラメーターは INOUT パラメーターとして宣言します。まず、Q アプライ・プログラムが操作タイプ (INSERT、UPDATE、DELETE、または KEY UPDATE) を渡すために使用し、また、操作が成功したかどうかをストアード・プロシージャーからQ アプライ・プログラムに戻りコードを戻すために使用します。Q サブスクリプションに対して指定する属性は、ターゲット内 (この場合ストアード・プロシージャー内) での予期しない条件を Q アプライ・プログラムが処理する方法を決定します。

ヒント: C および SQL で作成されているストアード・プロシージャーの operation パラメーターの例については、サンプル・プログラムを参照してください。operation パラメーターを使用して Q アプライ・プログラムに戻りコードを戻す方法に注目してください。

表 8 は、Q アプライ・プログラムがストアード・プロシージャーに渡す操作値とそれぞれの値の意味を示しています。

表 8. Q アプライ・プログラムがストアード・プロシージャーに渡す SQL 戻りコード

操作値	操作タイプ
16	削除
32	挿入
64	非キー列の更新
128	キー列の更新

評価する Q アプライ・プログラムの SQL コードに INOUT operation integer パラメーターを設定し、変更がターゲットに対して正常に適用されたかどうか確認します。ストアード・プロシージャーが Q アプライ・プログラムに SQL コードを戻さない場合、Q アプライ・プログラムはストアード・プロシージャーの内部での処理を認識していません。

重要: ストアード・プロシージャーは COMMIT または ROLLBACK ステートメントを実行してはなりません。Q アプライ・プログラムがトランザクションをコミットまたはロールバックするからです。

ストアード・プロシージャーが変更を適用している表に他のアプリケーションも変更を行っている場合、ストアード・プロシージャーはその表で予期しない条件を検出する可能性があります。たとえば、ストアード・プロシージャーは、別のアプリケーションがすでに削除した行に対して更新を試みるかもしれません。ストアード・プロシージャーがターゲット表に操作されたソース・データを挿入しようとするときに起きる可能性のある予期しない条件を解決する方法に関して、次の選択肢があります。

- 127 ページの『Q アプライ・プログラムによる予期しない条件の処理方法』
- 127 ページの『ストアード・プロシージャーによる予期しない条件の処理方法』

Q アプライ・プログラムによる予期しない条件の処理方法

Q アプライ・プログラムにストアード・プロシージャが保持するターゲットの予期しない条件の処理を行わせたい場合、以下にリストしている (表 9) SQL コードの 1 つを Q アプライ・プログラムに戻すようにストアード・プロシージャを作成し、ストアード・プロシージャが保持するターゲットに対する変更を強制するように Q アプライ・プログラムの Q サブスクリプションを定義します。Q アプライ・プログラムはソースの変更を強制します。SQL 戻りコードの中には、Q アプライ・プログラムが行操作をトランスフォームし、例外を IBMQREP_EXCEPTIONS table 表に記録し、トランスフォームされた行操作をストアード・プロシージャに戻すものもあります。

ストアード・プロシージャによる予期しない条件の処理方法

ターゲットに対してソース変更を強制する方法で、Q アプライ・プログラムにターゲット内の予期しない条件の処理を行わせたくない場合、ストアード・プロシージャに別のエラー処理ロジックを構築することができます。ストアード・プロシージャは Q アプライ・プログラムに対して常に SQL コードを戻さなければなりません。Q アプライ・プログラムがターゲット表内の予期しない条件または実際の障害を検出しないようにするために 0 (ゼロ) を戻すことができます。ストアード・プロシージャが operation パラメーターの Q アプライ・プログラムに SQL 戻りコードを渡した後、Q アプライ・プログラムはそれぞれの SQL コードを解釈し、(Q アプライ・プログラムによる予期しない条件の処理方法の指定に応じて) 適切なアクションを開始します。Q アプライ・プログラムは戻りコード +100 および -803 を競合として、また他のすべての戻りコードをエラーとして処理します。表 9 は、ストアード・プロシージャが出力する +100 および -803 タイプの戻りコード、Q アプライ・プログラムによるそのタイプの戻りコードの解釈、および結果として Q アプライ・プログラムが取るアクションを示しています。以下の表にある情報は、Q アプライ・プログラムに対してターゲット変更の強制を指定したと想定しています。

表 9. ストアード・プロシージャが Q アプライ・プログラムに渡す SQL 戻りコードと、Q アプライ・プログラムの反応。

SQL 戻りコード	操作タイプ	戻りコードの意味	Q アプライの反応
0	挿入	行は正常に挿入されました。	Q アプライは次の行を処理します。
0	更新	行は正常に更新されました。	Q アプライは次の行を処理します。
0	削除	行は正常に削除されました。	Q アプライは次の行を処理します。
+100	削除	行はターゲット内に見つかりませんでした。	Q アプライは呼び出しを再試行しません。
+100	更新	行はターゲット内に見つかりませんでした。	Q アプライに対して変更の強制を指定した場合、Q アプライは更新を挿入に変更し、呼び出しを再試行します。
-803	挿入	行はターゲット内にすでに存在します。	Q アプライに対して変更の強制を指定した場合、Q アプライは挿入を更新にトランスフォームし、呼び出しを再試行します。

表9. ストアド・プロシージャが Q アプライ・プログラムに渡す SQL 戻りコードと、Q アプライ・プログラムの反応。(続き)

SQL 戻りコード	操作タイプ	戻りコードの意味	Q アプライの反応
-803	キー更新	新しいキーはすでにターゲット内に存在します。	Q アプライに対して変更の強制を指定した場合、Q アプライは操作を新しいキーを持つ更新操作にトランスフォームし、呼び出しを再試行します。

関連概念:

- 114 ページの『ターゲット表での予期しない条件のオプション (単一方向レプリケーション)』

関連タスク:

- 124 ページの『Q レプリケーションのソース・データを操作するストアド・プロシージャの作成』

関連資料:

- 529 ページの『Q レプリケーションおよびイベント発行をセットアップするためのサンプル (Linux、UNIX、Windows)』
- 532 ページの『Q レプリケーションおよびイベント発行をセットアップするためのサンプル (z/OS)』

各ソース列が抑制されたかどうかを識別するストアド・プロシージャ・パラメーター

操作を識別するパラメーター (operation) の後、ストアド・プロシージャ内の 2 番目のパラメーター (suppression_ind) を、各ソース列が抑制されたかどうかを識別する IN パラメーターとして宣言します。

IN パラメーター suppression_ind は、4 つの必須パラメーターを除く各パラメーターごとに文字を保持する文字配列 (colsupind) です。文字は、ソース列の値が抑制されたかどうかを示します。

- 1** パラメーターと相関するソース列の値は抑制されました。
- 0** パラメーターと相関するソース列の値は抑制されませんでした。

配列は、関数パラメーターを評価するためにストアド・プロシージャの内部で使用されなければなりません。Q アプライ・プログラムは、キー列にマップする各パラメーターごとにちょうど 1 つの変更前の値 (before value) が存在するかどうかを検査します。

ヒント: C および SQL で作成されているストアド・プロシージャの抑制された列を識別するパラメーターの例については、サンプル・プログラムを参照してください。

関連タスク:

- 124 ページの『Q レプリケーションのソース・データを操作するストアド・プロシージャの作成』

関連資料:

- 529 ページの『Q レプリケーションおよびイベント発行をセットアップするためのサンプル (Linux、UNIX、Windows)』
- 532 ページの『Q レプリケーションおよびイベント発行をセットアップするためのサンプル (z/OS)』

Q レプリケーションのトランザクションを識別するストアード・プロシージャ・パラメーター

ソース列を識別するパラメーター (`suppression_ind`) の後、ストアード・プロシージャ内の次の 2 つのパラメーターを、トランザクションを識別する Q アプライ・プログラムのメッセージ・ヘッダーからの情報を取り込む IN パラメーターとして宣言します。Q アプライ・プログラムはトランザクションではなく行をストアード・プロシージャに渡すため、トランザクション・パラメーターからこの行変更が属するトランザクションの識別に役立つ情報を得られます。また、タイム・スタンプやログのシーケンス番号もエラーの報告や診断に役立ちます。以下の 2 つのパラメーターはトランザクションを識別します。

IN src_commit_lsn CHAR(10)

ソース・サーバーがトランザクションに対してコミット・ステートメントを発行したときのログのシーケンス番号。

IN src_trans_time TIMESTAMP

ソース・サーバーがトランザクションに対してコミット・ステートメントを発行したときのタイム・スタンプ。

ヒント: C および SQL で作成されているストアード・プロシージャの 2 つのトランザクション・パラメーターの例については、サンプル・プログラムを参照してください。

関連タスク:

- 124 ページの『Q レプリケーションのソース・データを操作するストアード・プロシージャの作成』

関連資料:

- 529 ページの『Q レプリケーションおよびイベント発行をセットアップするためのサンプル (Linux、UNIX、Windows)』
- 532 ページの『Q レプリケーションおよびイベント発行をセットアップするためのサンプル (z/OS)』

Q レプリケーションのソース列にマップするストアード・プロシージャ・パラメーター

トランザクションを識別するストアード・プロシージャ内のパラメーター (`src_commit_lsn` および `src_trans_time`) の後、ストアード・プロシージャ内の次のパラメーターを、各列からのソース・データを取り込む IN パラメーターとして宣言します。列にマップする追加パラメーターは任意の順序でリストできます。

稼働パラメーターおよびトランザクション・パラメーターの後、
 IBMQREP_TRG_COLS 表の TARGET_COLNAME 値にマップするストアード・プロシージャ内のパラメーターのみを組み込んでください。これ以外のパラメーターは、OUT または INOUT パラメーターとして宣言したとしても、ストアード・プロシージャ内で許可されません。

TARGET_COLNAME 列にリストされている名前は、ストアード・プロシージャ・パラメーターと同じ名前であればなりません。表 10 は、ソース列名と IBMQREP_TRG_COLS 表内のパラメーター名との対応の例を示しています。表にはコントロール表からの関連する 3 つの列のみが示されています。

表 10. ソース列名と IBMQREP_TRG_COLS 表内に保管されているストアード・プロシージャ内のパラメーターの名前

...	SOURCE_COLNAME	TARGET_COLNAME	IS_KEY	...
	Col1	Parm1	Y	
	Col2	Parm2	Y	
	Col3	Parm3	N	
	Col4	Parm4	N	
	Col5	Parm5	N	

ストアード・プロシージャ内のいくつかのパラメーターは、キーの一部となるように宣言しなければなりません。ターゲット・キー列として宣言するパラメーターは、ソースのキー列にマップする必要があります。ソース・キーとターゲット・キーは一致していなければなりません。Q アプライ・プログラムはこれらのキー列を使用して、同じ行を扱うトランザクションを正しく配列します。

パラメーターがソース表内のキー列にマップする場合、以下のアクションを実行する必要があります。

- 対応する変更前の値 (before-value) 列をパラメーターとして宣言します。
- キーの変更前の値 (before value) のパラメーターが、キーの変更後の値 (after value) のパラメーターの直前であることを確認する。
- キーの変更前の値 (before value) のパラメーター名を 'X' で開始し、その後パラメーター名を続けます。たとえば、Parm1 という名前のキー列の場合、変更前の値 (before value) のパラメーターは XParm1 になります。

非キー列の変更前の値 (before values) のパラメーターは宣言しないでください。

ヒント: ストアード・プロシージャのパラメーターにマップしているソース列が多数ある場合、ソース列の名前を使用してパラメーターを命名することを検討してください。パラメーターの名前を対応するソース列の名前とまったく同じにする場合、レプリケーション・センターはソース列名を自動的にターゲット・パラメーター名にマップすることができます。

キー・パラメーターを使用したストアード・プロシージャの有効なシグニチャーの例: この例の Parm1 と Parm2 は、キー列にマップするパラメーターです。

```
CREATE PROCEDURE storedprocedure_name(
    INOUT operation integer,
    IN suppression_ind VARCHAR(size),
    IN src_commit_tsn CHAR(10),
    IN src_trans_time TIMESTAMP,
    XParm1,
```

```
    Parm1,  
    XParm2,  
    Parm2,  
    Parm3,  
    Parm4,  
    Parm5  
)
```

キー・パラメーターを使用した無効なストアード・プロシージャの例: この無効な例では、Parm1 and Parm2 は、キー列にマップするパラメーターです。キー・パラメーター Parm2 の変更前の値 (before value) を受け取るパラメーターがないので、この例は無効です。

```
#-----  
# Incorrect: Don't do this  
#-----  
  
CREATE PROCEDURE storedprocedure_name(  
    INOUT operation integer,  
    IN suppression_ind VARCHAR(size) ,  
    IN src_commit_lsn CHAR(10),  
    IN src_trans_time TIMESTAMP,  
    XParm1,  
    Parm1,  
    Parm2,  
    XParm3,  
    Parm3,  
    Parm4,  
    Parm5  
)
```

ヒント: C および SQL で作成されているストアード・プロシージャのソース列にマップするパラメーターの例については、サンプル・プログラムを参照してください。

関連タスク:

- 124 ページの『Q レプリケーションのソース・データを操作するストアード・プロシージャの作成』

関連資料:

- 529 ページの『Q レプリケーションおよびイベント発行をセットアップするためのサンプル (Linux、UNIX、Windows)』
- 532 ページの『Q レプリケーションおよびイベント発行をセットアップするためのサンプル (z/OS)』

第 10 章 ソースからターゲットへのレプリケーションのセットアップ (複数方向)

ソースからターゲットへのレプリケーションのセットアップ (複数方向) -- 概要

Q レプリケーションを使うと、複数のサーバー上の表間でデータを複製することができます。

以下のトピックでは、2 つのタイプの複数方向レプリケーション (双方向またはピアツーピア)、および双方向またはピアツーピア・レプリケーションのセットアップ方法を説明します。

- 『双方向レプリケーション』
- 136 ページの『ピアツーピア・レプリケーション』
- 141 ページの『双方向レプリケーションとピアツーピア・レプリケーション』
- 144 ページの『双方向レプリケーションの Q サブスクリプションの作成』
- 153 ページの『ピアツーピア・レプリケーションの Q サブスクリプションの作成 -- 概要』
- 158 ページの『2 つのサーバーを使用した双方向およびピアツーピア・レプリケーションの開始』
- 159 ページの『2 つのサーバーを使用した双方向およびピアツーピア・レプリケーションの停止』
- 160 ページの『3 つ以上のサーバーを使用したピアツーピア・グループでのレプリケーションの開始』
- 161 ページの『3 つ以上のサーバーを使用したピアツーピア・グループでのレプリケーションの停止』

関連概念:

- 91 ページの『ソースからターゲットへの (単一方向) レプリケーションのセットアップ -- 概要』

双方向レプリケーション

双方向レプリケーションは、以下の特性を持つ Q レプリケーション構成です。

- レプリケーションは 2 つのサーバーの表の間で行われる。表のコピーの 1 つに加えられた変更が、その表の 2 番目のコピーに複製され、さらに 2 番目のコピーへ行われた変更が最初のコピーに複製して戻されます。
- どちらのサーバー上のアプリケーションも、それらの表の中の同じ行を同時に更新できる。ただし、複製された表の中の同一のデータが両方のサーバーによって同時に更新される可能性はほとんどあるいはまったくありません。一度に 1 つの

サーバーによって同じ行が更新されるか、あるいは、あるサーバーがデータの特
定の列または行だけを更新し、他のサーバーが別の列または行を更新するかのど
ちらかです。

- 競合が発生した場合は、表のどちらのコピーを優先するか選択できる。

双方向レプリケーションでは、2 つのサーバー間で表のコピーを複製します。単一
の表の両方のコピーの集合を論理表と呼びます。各サーバーに表のコピーがありま
す。表の各コピーについては、以下ようになります。

- 列および行の数が同じでなければならない。
- 列名が同一でなければならない。
- 互換データ・タイプが含まれていなければならない。
- 名前およびスキーマが異なってもよい。

このタイプのレプリケーションでは、Q アプライ・プログラムを使用してデータを
ストアード・プロシージャに渡すことによってデータを操作することはできません。
双方向構成の一部であるサーバーごとに、1 つ以上の Q キャプチャー・プログラ
ムと 1 つ以上の Q アプライ・プログラムを実行しなければなりません。

重要: 個々のサーバー上の Q キャプチャー・プログラムと Q アプライ・プログラ
ムのコントロール表のスキーマ名は同じでなければなりません。たとえば、
SERVER_RED というサーバーと SERVER_GREEN というサーバーがある場合、
SERVER_RED に置かれている Q キャプチャーと Q アプライ・プログラムの両方
でスキーマが同じでなければならない、SERVER_GREEN に置かれている Q キャ
プチャーと Q アプライ・プログラムの両方でスキーマが同じでなければなりません。

双方向レプリケーションでのレプリケーション・オブジェクト

双方向構成では、以下のように適切な数のレプリケーション・キュー・マップおよ
び Q サブスクリプションが必要です。

レプリケーション・キュー・マップの数

双方向レプリケーションに加わるサーバーの各ペア間で、2 つのレプリケー
ション・キュー・マップが必要です。たとえば、SERVER_RED と
SERVER_GREEN という名前の 2 つのサーバーがある場合、以下のような
2 つのレプリケーション・キュー・マップが必要です。

- データを SERVER_RED から SERVER_GREEN に転送する WebSphere[®]
MQ キューを識別するためのレプリケーション・キュー・マップ。
- データを SERVER_GREEN から SERVER_RED に転送する WebSphere
MQ キューを識別するためのレプリケーション・キュー・マップ。

Q サブスクリプションの数

双方向レプリケーションで複製される論理表ごとに、2 つのサーバー間での
Q サブスクリプションのペアが必要です。たとえば、SERVER_RED と
SERVER_GREEN という名前の 2 つのサーバーがある場合、以下のような
2 つの Q サブスクリプションが作成されます。

- SERVER_RED 上のソース表から SERVER_GREEN 上のターゲット表へ
の Q サブスクリプション。
- SERVER_GREEN 上のソース表から SERVER_RED 上のターゲット表へ
の Q サブスクリプション。

2つの論理表がある場合は4つのQサブスクリプションが必要です。論理表が3つの場合は6つのQサブスクリプションが必要で、以降も同様です。

図11では、2つのサーバー間での1つの論理表の双方向レプリケーションを示しています。論理表が1つの場合、2つのQサブスクリプションと2つのレプリケーション・キュー・マップが必要です。

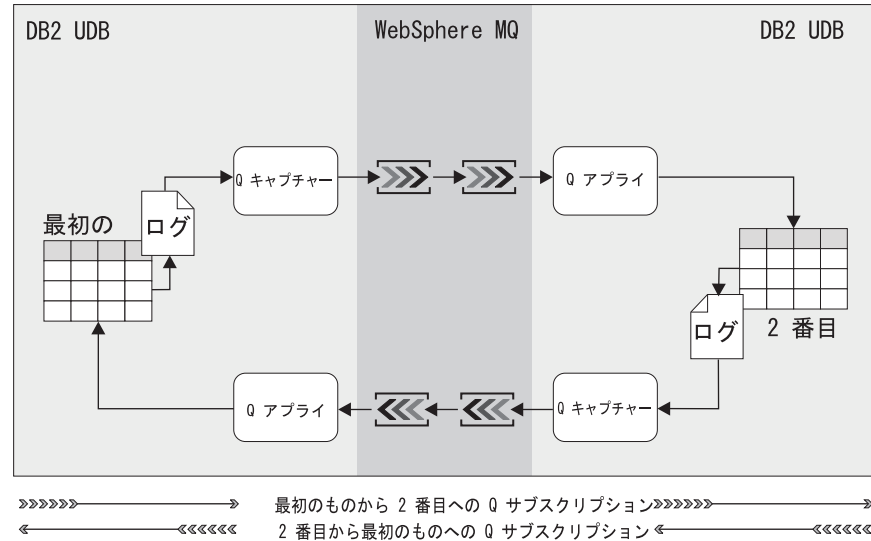


図11. 双方向レプリケーションでの表の2つのコピー間でのQサブスクリプション：変更は、WebSphere MQキューを使用して、表の各コピーからその表の他のコピーに複製される。

双方向レプリケーションでの競合の検出

双方向レプリケーションでは、あるQサブスクリプション内のソース表から複製されたデータが、Qアプライ・プログラム以外のアプリケーションによって、対応するターゲット表に行われた変更と競合する可能性があります。双方向レプリケーションはデータ値を使用して、競合の検出および解決を行います。競合の検出に使用するデータ値を選択できます。これらのデータ値は、キー列値のみ、変更された列値、またはすべての列値にすることができます。

たとえば、あるシステム上のアプリケーションがサーバー (SERVER_RED) 内の表への変更を行い、そのサーバーがスタンバイ・システム上のサーバー (SERVER_GREEN) 内の同一の表へそれらの変更を複製する、というシナリオを想像してください。最初のシステム SERVER_RED に障害が起きたら、アプリケーションは SERVER_GREEN の表の使用を開始します。最初のシステムがオンラインに戻ったら、SERVER_GREEN から SERVER_RED へ変更を複製します。ただし、最初のシステムがシャットダウンされた場合、データの一部が2番目のシステムに複製できなかった可能性があります。そのデータ (この時点では古いデータ) を、SERVER_GREEN から複製されたデータと置換する必要があります。新しいデータを複製する場合、SERVER_RED でのQアプライ・プログラムは競合を認識し、SERVER_GREEN からの変更を SERVER_RED に強制的に適用します。

両方のサーバーの Q アプライ・プログラムが、表の両方のコピーへのデータのアプライを試行する際に、どのように競合を検査するかを指定できます。また、競合を検出した場合に、それらのプログラムの行うアクションを選択できます。競合の規則および競合のアクションに関して行う選択は、行がアプライされる仕方に関する動作に影響を与えるため、重要な決定といえます。

関連概念:

- 133 ページの『ソースからターゲットへのレプリケーションのセットアップ (複数方向) -- 概要』
- 5 ページの『Q サブスクリプション』
- 6 ページの『レプリケーション・キュー・マップ』
- 147 ページの『競合検出のオプション (双方向レプリケーション)』

関連タスク:

- 144 ページの『双方向レプリケーションの Q サブスクリプションの作成』

ピアツーピア・レプリケーション

ピアツーピア・レプリケーション (マルチマスター・レプリケーションとも呼ばれる) は、次の特性を持つ構成です。

- 複数のサーバー上の表間でレプリケーションが行われる。
- あるサーバー上での更新は他のすべての関連するサーバーに複製される。
- どのサーバー上のアプリケーションでも、表の同じ行および列を同時に更新できる。
- すべてのサーバーは、データに対して等しい所有権を持つ対等の関係であり、どのサーバーも「マスター」や、データのソース所有者ではない。

ピアツーピア・レプリケーションでは、複数のサーバー間で表のコピーを複製します。単一表のすべてのコピーの集合を論理表と呼びます。各サーバーに表のコピーがあります。表の各コピーについては、以下のようになります。

- 列および行の数が同じでなければならない。
- 列名が同一でなければならない。
- 互換データ・タイプが含まれていなければならない。
- 名前およびスキーマが異なってもよい。

ピアツーピア・レプリケーションでは、論理表のすべてのコピー間でデータ収束が保証されます。つまり、表の各コピーは、最終的に他のコピーと同じ状態になり、最新のコミット値を持つようになります。ピアツーピア・レプリケーションは非同期であるため、アプリケーションがすべての表に対する変更を停止し、すべての変更が複製され、すべてのメッセージが処理されるまで、表のコピーは収束しません。

このタイプのレプリケーションでは、Q アプライ・プログラムを使用してデータをストアード・プロシージャに渡すことによってデータを操作することはできません。ピアツーピア構成の一部である各サーバーでは、少なくとも 1 つの Q キャプチャー・プログラムと 1 つの Q アプライ・プログラムが実行されています。

重要: 個々のサーバー上の Q キャプチャー・プログラムと Q アプライ・プログラムのコントロール表のスキーマ名は同じでなければなりません。たとえば、SERVER_RED という名前のサーバーと SERVER_GREEN という名前のサーバーがある場合、SERVER_RED 上の 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両方とも同じでなければならず、SERVER_GREEN 上の 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でなければなりません。

ピアツーピア構成では、競合検出および解決は、データ収束を保証する仕方で Q アプライ・プログラムによって自動的に管理されるため、競合検出および解決のための構成は不要です。Q レプリケーションは追加情報を保守して各データ変更のバージョンをトラッキングし、Q レプリケーション・システムはこの追加バージョン管理情報を使用して競合を検出および解決します。

ピアツーピア・レプリケーションで複製されるすべての表は、Q レプリケーションだけで使用される 2 つの列 (タイム・スタンプ列と短整数列) を含むように変更されます。これらの列は両方ともトリガーによって保守されます。これらの追加のレプリケーション列およびトリガーは、ピアツーピア・レプリケーション用の Q サブスクリプションの作成時に作成されます。バージョン管理列は、行のどのバージョンが最新であるかを示します。バージョン管理列の値を調べることによって、行が最後に更新された時刻や、更新を行ったサーバーを判別できます。

競合検出および解決は、これらのバージョン管理列の内容を基にして行われます。競合が検出されると、行の最新バージョン、つまり最新の (時間帯に合わせて時刻を訂正した後の) タイム・スタンプ値を含む行が保持されます。

LOB データについての注: バージョン管理列はピアツーピア・レプリケーションでの競合を検出するために使用されるため、LOB データ・タイプの列は、他のデータ・タイプの列と同じように処理されます。

以下のトピックでは、ピアツーピア・レプリケーションに必要なレプリケーション・キュー・マップおよび Q サブスクリプションの数と、ピアツーピア・レプリケーションが参照保全を処理する方法について説明します。

- 『2 つのサーバーが関係するピアツーピア・レプリケーション用のレプリケーション・オブジェクト』
- 139 ページの 『3 つのサーバーが関係するピアツーピア・レプリケーション用のレプリケーション・オブジェクト』
- 140 ページの 『競合解決および参照保全』

2 つのサーバーが関係するピアツーピア・レプリケーション用のレプリケーション・オブジェクト

2 つのサーバーが関係するピアツーピア構成では、適切な数のレプリケーション・キュー・マップと Q サブスクリプションが必要です。

レプリケーション・キュー・マップの数

ピアツーピア・レプリケーションに参加するサーバーの各ペア間に、2 つのレプリケーション・キュー・マップが必要です。たとえば、SERVER_RED と SERVER_GREEN という名前の 2 つのサーバーがある場合、以下のような 2 つのレプリケーション・キュー・マップが必要です。

- データを SERVER_RED から SERVER_GREEN に転送する WebSphere® MQ キューを識別するためのレプリケーション・キュー・マップ。
- データを SERVER_GREEN から SERVER_RED に転送する WebSphere MQ キューを識別するためのレプリケーション・キュー・マップ。

Q サブスクリプションの数

ピアツーピア・レプリケーションで複製される論理表ごとに、2 つのサーバー間に Q サブスクリプションのペアが必要です。たとえば、SERVER_RED と SERVER_GREEN という名前の 2 つのサーバーがある場合、以下のような 2 つの Q サブスクリプションが作成されます。

- SERVER_RED 上のソース表から SERVER_GREEN 上のターゲット表への Q サブスクリプション。
- SERVER_GREEN 上のソース表から SERVER_RED 上のターゲット表への Q サブスクリプション。

2 つの論理表がある場合は 4 つの Q サブスクリプションが必要です。論理表が 3 つの場合は 6 つの Q サブスクリプションが必要で、以降も同様です。

図 12 は、2 つのサーバー間での 1 つの論理表のピアツーピア・レプリケーションを示しています。2 つのサーバー間で複製される 1 つの論理表ごとに 2 つの Q サブスクリプションが必要です。つまり、対等表 A から対等表 B にデータを複製するための Q サブスクリプションと、対等表 B から対等表 A にデータを複製するための Q サブスクリプションです。また、少なくとも 2 つのレプリケーション・キュー・マップが必要です。

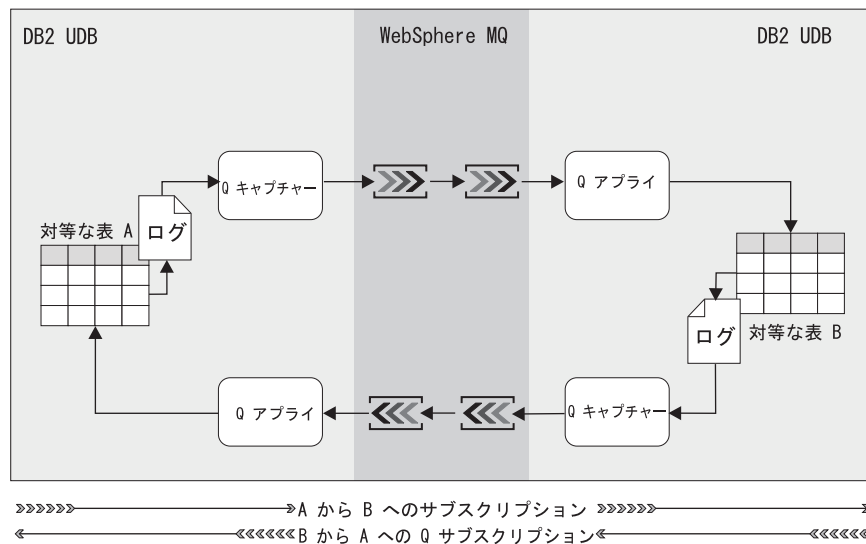


図 12. 2 つのサーバーが関係するピアツーピア・レプリケーションでの Q サブスクリプション：変更は、WebSphere MQ キューを使用して、表の各コピーからその表の他のコピーに複製されます。

3 つのサーバーが関係するピアツーピア・レプリケーション用のレプリケーション・オブジェクト

3 つのサーバーが関係するピアツーピア構成では、適切な数のレプリケーション・キュー・マップと Q サブスクリプションが必要です。

レプリケーション・キュー・マップの数

ピアツーピア・レプリケーションに参加するサーバーの各ペア間に、2 つのレプリケーション・キュー・マップが必要です。必要なレプリケーション・キュー・マップの数は、式 $n*(n-1)$ を使用して計算できます。ここで、 n はサーバーの数です。たとえば、SERVER_RED、SERVER_BLUE、および SERVER_GREEN という名前の 3 つのサーバーがある場合は、次の 6 つのレプリケーション・キュー・マップが必要です。

- データを SERVER_RED から SERVER_GREEN に転送する WebSphere MQ キューを識別するためのレプリケーション・キュー・マップ。
- データを SERVER_GREEN から SERVER_RED に転送する WebSphere MQ キューを識別するためのレプリケーション・キュー・マップ。
- データを SERVER_RED から SERVER_BLUE に転送する WebSphere MQ キューを識別するためのもの。
- データを SERVER_BLUE から SERVER_RED に転送する WebSphere MQ キューを識別するためのもの。
- データを SERVER_BLUE から SERVER_GREEN に転送する WebSphere MQ キューを識別するためのもの。
- データを SERVER_GREEN から SERVER_BLUE に転送する WebSphere MQ キューを識別するためのもの。

Q サブスクリプションの数

ピアツーピア・レプリケーションで複製される論理表ごとに、2 つのサーバー間に Q サブスクリプションのペアがあります。構築される Q サブスクリプションの数は、式 $n*(n-1)$ を使用して計算できます。ここで、 n はサーバーの数です。たとえば、SERVER_RED、SERVER_GREEN、および SERVER_BLUE という名前の 3 つのサーバーがある場合は、次の 6 つの Q サブスクリプションが構築されます。

- SERVER_RED 上のソース表から SERVER_GREEN 上のターゲット表への Q サブスクリプション。
- SERVER_GREEN 上のソース表から SERVER_RED 上のターゲット表への Q サブスクリプション。
- SERVER_RED 上のソース表から SERVER_BLUE 上のターゲット表への Q サブスクリプション。
- SERVER_BLUE 上のソース表から SERVER_RED 上のターゲット表への Q サブスクリプション。
- SERVER_BLUE 上のソース表から SERVER_GREEN 上のターゲット表への Q サブスクリプション。
- SERVER_GREEN 上のソース表から SERVER_BLUE 上のターゲット表への Q サブスクリプション。

2つの論理表がある場合は12のQサブスクリプションが必要で、論理表が3つの場合は18のQサブスクリプションが必要であり、以下同様です。

図13は、3つのサーバー間での1つの論理表のピアツーピア・レプリケーションを示しています。このケースでは、6つのQサブスクリプションが必要です。つまり、サーバーの各ペア間に2つです。また、少なくとも6つのレプリケーション・キュー・マップが必要です。

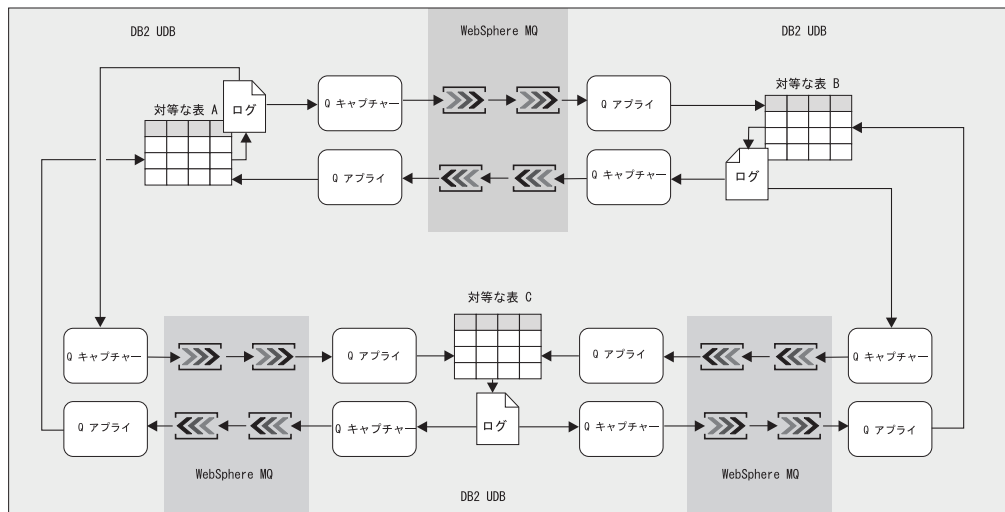


図13. 3つのサーバーが関係するピアツーピア・レプリケーションでのQサブスクリプション：変更は、WebSphere MQキューを使用して、表の各コピーからその表の他のすべてのコピーに複製されます。

競合解決および参照保全

ほとんどの場合、ピアツーピア・レプリケーションでは、さまざまなコピーで競合の変更が発生しても、複製された表が同じ状態に収束することが保証されます。ただし、キー列以外の列に定義されているユニーク制約の重複値、または参照制約違反に競合の原因がある場合は、解決不能の競合が発生する場合があります。解決不能の競合が発生すると、競合する行が `IBMQREP_EXCEPTIONS` 表に記録され、Qアプライ・プログラムは、Qサブスクリプションに指定されたエラー・アクションを実行します。

特定の解決不能の競合を許容し、Qサブスクリプションに指定したエラー・アクションをQアプライ・プログラムが実行しないようにする場合は、Qサブスクリプションに `OKSQLSTATES` を設定することによって許容可能な `SQLSTATE` 値を指定できます。ただし、`OKSQLSTATES` に特定の `SQL` 状態を指定しても、ピアツーピア・レプリケーションでは、参照制約違反または非キー列に定義されたユニーク制約の重複値に起因する競合の場合は、表のすべてのコピーが確実に収束されるわけではないことに注意してください。表相違検出ユーティリティおよび表修復ユーティリティを使用して、許可する解決不能の競合に起因する差異を見つけて修復できます。

複製された表のさまざまなコピーで変更が発生し、Qサブスクリプションのキー列以外の列のユニーク制約が同じ値になる場合は、競合を解決できません。Qサブスクリプションの `OKSQLSTATES` に値を追加することによって `SQLSTATE 23505`

が許可されることを指定する場合は、解決不能のユニーク・キー競合が発生しても、Q アプライ・プログラムは Q サブスクリプションに指定したエラー・アクションを実行しません。

参照制約が定義されている表のさまざまなコピーの行で変更が発生する場合は、競合を解決できません。これらの競合の原因として、その行が関係するメッセージ伝搬に遅延があるか、実際に競合が発生しているかのどちらかが考えられます。実際の競合の例は、あるコピーで親行が削除され、同時に子行が別のコピーに挿入されるという場合です。Q アプライ・プログラムが、親行が同時に削除されたコピーで子行を挿入しようとする、解決不能の競合が発生し、Q アプライ・プログラムは子行を SQLSTATE 23503 で IBMQREP_EXCEPTIONS 表に記録します。Q アプライ・プログラムが、子行が同時に挿入されたコピーで親行を削除しようとする、参照制約の削除規則で削除が制限されている場合 (DELETE RESTRICT) は、削除は失敗します。Q アプライ・プログラムは、親行を SQLSTATE 23504 または SQLSTATE 23001 で IBMQREP_EXCEPTIONS 表に記録します。

実際の競合の別の例は、子行が同時に挿入され、親行の削除が適用されるときに参照保全の削除規則 (CASCADE DELETE) によって子行が除去されるという場合です。このケースでは、子行のカスケード削除が表の他のコピーに複製されると、他のコピーはその子行を検出できないため、SQLSTATE 02000 が報告されます。同じ SQLSTATE 02000 は、その行が関係するメッセージ伝搬の遅延に起因する場合があります。コピー 1 での親行の挿入がコピー 3 に到達する前に、コピー 2 での子行の挿入がコピー 3 に到達するような場合です。

関連概念:

- 133 ページの『ソースからターゲットへのレプリケーションのセットアップ (複数方向) -- 概要』
- 153 ページの『ピアツーピア・レプリケーションの Q サブスクリプションの作成 -- 概要』
- 3 ページの『Q レプリケーション』
- 5 ページの『Q サブスクリプション』
- 6 ページの『レプリケーション・キュー・マップ』

双方向レプリケーションとピアツーピア・レプリケーション

2 つのサーバー上の表間でデータを複製する場合、複数方向レプリケーションの選択肢には、双方向レプリケーションとピアツーピア・レプリケーションの 2 つがあります。以下の情報は、ビジネス・ニーズによりよく応えるために、双方向レプリケーションまたはピアツーピア・レプリケーションのどちらを選択すべきかを決定するのに役立ちます。ご使用の構成で 3 つ以上のサーバーが必要な場合、複数方向レプリケーションとして提示される選択肢は、ピアツーピア・レプリケーションだけです。

双方向レプリケーションの機能に最適のシナリオ

次の状況では、双方向レプリケーションの選択を考慮してください。

- 構成中で競合が発生することが予期されず、競合が発生したかどうかを検査する必要がない。最小のオーバーヘッドおよびネットワーク・トラフィックについては、両方のサーバーが競合を無視するように指定する。
- 構成中で競合が発生することは予期されないが、安全策として競合が発生したかどうかを検査し、かつ予期しないデータの衝突が発生した場合に 1 つのサーバーを優先させることが許容できる。
- 1 つのサーバーがデータの特定の列だけを更新し、他のサーバーが他の列を更新する。Q アプライ・プログラムがキーと変更された列の両方で競合を検査するように指定する場合、Q アプライ・プログラムは同じ行の別の列に影響を与える更新をマージします。
- 1 つのサーバーがデータの特定の行だけを更新し、他のサーバーが他の行を更新する。

ピアツーピア・レプリケーションの機能に最適なシナリオ

競合が LOB データ・タイプの列で発生する可能性がある場合、ピアツーピア・レプリケーションの選択を考慮してください。バージョン管理列はピアツーピア・レプリケーションでの競合の検出に使用されるので、LOB データ・タイプの列は他のデータ・タイプの列と同じように処理されます。

高可用性シナリオ用の双方向レプリケーションとピアツーピア・レプリケーション

高可用性シナリオ用に 2 つのサーバーを構成している場合は、ビジネス・ニーズに応じて、双方向レプリケーションまたはピアツーピア・レプリケーションのどちらを選択できます。このようなシナリオでは、どちらのタイプも使用できます。以下のトレードオフを考慮することが必要です。

- 1 次サーバーに障害が起こった場合に 2 次サーバーが引き継ぐのにかかる時間、または 1 次サーバーに切り替えて戻すのにかかる時間
- 引き継ぎのプロセスを制御するための手動または自動手順の必要
- 各メソッドで必要なオーバーヘッド

高可用性シナリオ用の双方向レプリケーション

競合発生時に、最新のタイム・スタンプでの変更が優先するようになりたい場合は、高可用性シナリオに双方向レプリケーションを選択すると良いでしょう。

推奨: 高可用性に双方向レプリケーションを選択する場合、構成を次のようにセットアップします。

- 読み取りおよび書き込みアプリケーションに使用可能な 1 次サーバーをセットアップします。
- 読み取りアプリケーションにのみ使用可能な 2 次サーバーをセットアップします。
- 1 次サーバーが敗者になるように、1 次サーバーには競合ルール「強制適用」を設定する。2 次サーバーから来る競合があれば、1 次サーバーに対して強制されます。
- 2 次サーバーが勝者になるように 2 次サーバーには競合ルール「無視」を設定する。1 次サーバーから来る競合があれば、2 次サーバーによって無視されます。

構成をこのようにセットアップすると、1 次サーバーはアプリケーションによって更新され、通常はアプリケーションによって更新されることのない 2 次サーバーにこれらの変更を複製します。1 次サーバーが使用不可になるイベントでは、アプリケーションをリダイレクトして 2 次サーバーを更新します。1 次サーバーが再び使用可能になったら、2 次サーバーからの最新の変更が 1 次サーバーに複製しなおされ、2 次サーバーからのこれらの変更が 1 次サーバーの古い変更を上書きします。

1 次サーバーが使用不可の間、サーバーが障害を起こす前に複製されていないと、データの一部は障害が発生したサーバーに残ったままになる場合があります。このデータは、後で 1 次サーバーが再度使用可能になったときに 2 次サーバーにアプライされます。複製された他の変更は、2 次サーバーの受信キューに入っている可能性があります。2 次サーバーの表のコピーにはアプライされていません。これらの変更は、アプリケーションが 2 次サーバーにリダイレクトされる際に加えられる、新しい変更と競合する場合があります。キューに入れられたデータと、2 次サーバー上の最新の変更の間に衝突がある場合、新規にリダイレクトされたアプリケーションのアクティビティが優先します。これは、2 次サーバーが 1 次サーバーからの競合を無視するように設定されているためです。

トラフィックが 2 次サーバーにリダイレクトされる前に 2 次サーバー上の受信キューを空にしてテークオーバーを行うことが可能です。このようにすると、フェイルオーバー中にデータの衝突の可能性を排除できます。したがって、フェイルオーバー中にデータの衝突の可能性を除去します。1 次サーバーに再度切り替えると、1 次サーバーから取り込まれていなかった古いデータが取り込まれて 2 次サーバーにアプライされます。アプリケーションがリダイレクトされた先の 2 次サーバーに最新の変更があるので、衝突があっても優先されません。その後 2 次サーバーに取り込まれたデータが 1 次サーバーにアプライされ、ここでも衝突があれば、このさらに新しいデータが優先されます。

アプリケーションを 1 次サーバーに再度リダイレクトする前に、1 次サーバーでのアプリケーションからの新規の変更が、2 次サーバーで発生した古いトランザクションに優先され始めないようにするステップを実行する必要があります。2 次サーバーでデータベース・アクティビティを静止させ、確実にすべてのデータ変更を 1 次サーバーにアプライします。

高可用性シナリオ用のピアツーピア・レプリケーション

高可用性にピアツーピア・レプリケーションを選択する場合、いつでもすべてのサーバーが読み取りおよび書き込みに使用可能です。(ピアツーピア・レプリケーションでは、3 つ以上のサーバーを構成できます。)したがって、1 つのサーバーが使用不可になると、他のサーバーがすぐに使用可能になり、引き継ぎまたは切り替えを実行します。この構成は、最も堅固な競合検出向けに提供されます。しかし、複製された表の各コピー上にある余分のバージョン管理列、およびこれらのバージョン管理列を保守するのに必要なトリガーによってシステムが引き起こす、追加のオーバーヘッドに対してこの操作性、停止時間の不足、および堅固な競合検出を評価する必要があります。

高可用性シナリオ用の競合検出に関する考慮事項

双方向レプリケーションでは、値に基づく競合検出 (すべての列の検査) の最も高いレベルでも、ピアツーピア・レプリケーションでのバージョンに基づく競合検出と同じほど堅固ではありません。いくつかの状況では、競合が検出されない場合があります。

推奨: アプリケーション設計による競合を予期している場合、ピアツーピア構成を選択してください。

アプリケーション開発者が、どのサーバーでも更新可能な表の分散コピーを含むアプリケーションを設計する場合、開発者は競合の可能性、競合の影響、および競合の解決方法を完全に調査する必要があります。競合の結果、トランザクションの永続性の喪失が起こりうるので、アプリケーションは競合の可能性を最小限にして設計する必要があります。

重要: Q アプライ・プログラムは、指定された行に競合を検出すると、その行に対してだけ動作します。双方向レプリケーションでもピアツーピア・レプリケーションでも、競合する行はターゲットへの変更として受け入れられるか、または無視されてアプライされないかのどちらかです。ピアツーピア・レプリケーションの目的は、トランザクション全体でなく、競合する行を処理する、データベース表のコピーの収束セットを提供することです。トランザクション全体の拒否または受け入れを実行すると、すぐにその結果は収束しないデータベース・コピーのセットになる場合があります。

Q アプライ・プログラムは、IBMQREP_EXCEPTIONS 表のすべての競合する行を報告します。複数のサーバーが関係することのあるピアツーピア構成では、Q レプリケーションは競合する行を一度だけ報告しようとしています。しかし場合によっては、複数のサーバー上の IBMQREP_EXCEPTIONS 表で競合が現れることがあります。データ値およびバージョン管理情報は同一なので、これらの重複は簡単に見分けることができます。ピアツーピアまたは双方向構成において競合アクティビティ全体を確認するには、すべてのサーバーの IBMQREP_EXCEPTIONS 表を見てください。

関連概念:

- 133 ページの『双方向レプリケーション』
- 136 ページの『ピアツーピア・レプリケーション』
- 147 ページの『競合検出のオプション (双方向レプリケーション)』

双方向レプリケーションの Q サブスクリプションの作成

双方向レプリケーションの Q サブスクリプションの作成

双方向レプリケーションでは、表のコピーの 1 つに加えられた変更が、その表の 2 番目のコピーに複製され、さらに 2 番目のコピーからの変更が最初のコピーに戻されて複製されます。双方向レプリケーションでは、すべての行および列が複製され、表の各コピー内の列名が一致していなければなりません。

ある Q サブスクリプションが作成されて、表の最初のコピーから表の 2 番目のコピーにトランザクションが複製され、別の Q サブスクリプションが作成されて、表の 2 番目のコピーから表の最初のコピーにトランザクションが戻されて複製されます。レプリケーション・センターを使用して、双方向レプリケーション用に Q サブスクリプションを作成すると、レプリケーション・センターは一度に両方の Q サブスクリプションを作成します。

前提条件:

双方向レプリケーション用に Q サブスクリプションを作成する前に、以下のアクションを実行しなければなりません。

- レプリケーション・キュー・マップおよび Q サブスクリプションをグループ化する方法を計画します。詳細については、93 ページの『レプリケーション・キュー・マップと Q サブスクリプションのグループ化』を参照してください。
- 表の最初のコピーがあるサーバーで、Q キャプチャーおよび Q アプライ・プログラム用のコントロール表を作成します。詳細については、87 ページの『Q キャプチャー・プログラムと Q アプライ・プログラムのコントロール表の作成』を参照してください。

重要: 個々のサーバー上の Q キャプチャー・プログラムと Q アプライ・プログラムのコントロール表のスキーマ名は同じでなければなりません。たとえば、SERVER_RED というサーバーと SERVER_GREEN というサーバーがある場合、SERVER_RED に置かれている Q キャプチャーと Q アプライ・プログラムの両方でスキーマが 1 つでなければならず、SERVER_GREEN に置かれている Q キャプチャーと Q アプライ・プログラムの両方でスキーマが 1 つでなければなりません。

- 表の 2 番目のコピーがあるサーバーで、Q キャプチャーおよび Q アプライ・プログラム用のコントロール表を作成します。詳細については、87 ページの『Q キャプチャー・プログラムと Q アプライ・プログラムのコントロール表の作成』を参照してください。

重要: このサーバーに置かれている Q キャプチャーおよび Q アプライ・プログラム用のコントロール表は、スキーマ名が同じでなければなりません。

- 各サーバー間でデータを転送する、2 つのレプリケーション・キュー・マップを作成します。表の最初のコピーから 2 番目のものにデータを複製するためのレプリケーション・キュー・マップを 1 つと、表の 2 番目のコピーから表の最初のものにデータを複製するためのレプリケーション・キュー・マップが 1 つ必要です。(このタスクは、Q サブスクリプションの作成前か、または Q サブスクリプションの定義中に行うことができます。) 詳細については、95 ページの『レプリケーション・キュー・マップの作成』を参照してください。

制約事項:

- 双方向レプリケーションでの Q サブスクリプション用の表をビューにすることはできません。
- 双方向レプリケーションのターゲットを、ストアド・プロシージャにすることはできません。
- 各サーバーで、単一の表が加わることのできる、双方向レプリケーションのトポロジーは 1 つだけです。たとえば、Server 1 が双方向レプリケーションを使用し

て表を Server 2 に複製する場合、Server 1 か Server 2 のいずれかにある表を、双方向レプリケーションを使用して Server 3 に複製することはできません。

- 双方向レプリケーションでは LOB 列の変更前の値が複製されないため、LOB 列の競合が検出されません。

手順:

表の 2 つのコピー間の双方向レプリケーション用の Q サブスクリプションを作成する場合、以下を行います。

1. ウィザードをオープンし、双方向レプリケーション用の Q サブスクリプションを作成するように指定する。

- a. レプリケーション・センターから「Q サブスクリプションの作成」ウィザードをオープンする。

このウィザードをオープンするには、以下のいずれかを行います。

- Q サブスクリプションの変更をキャプチャーする Q キャプチャー・プログラムを示す Q キャプチャー・スキーマを展開する。「サブスクリプション」フォルダーを右マウス・ボタンでクリックして、「作成」を選択します。
- Q サブスクリプションの変更をアプライする Q アプライ・プログラムを示す Q アプライ・スキーマを展開する。「サブスクリプション」フォルダーを右マウス・ボタンでクリックして、「作成」を選択します。

詳細については、オンライン・ヘルプを参照してください。

- b. 「レプリケーション」ページで、双方向レプリケーションが必要であることを指定する。
2. 「サーバー」ページで以下を行う。
 - 最初のサーバーおよびスキーマ (複製する表を含むもの) に関する情報を指定する。
 - 2 番目のサーバーおよびスキーマに関する情報を指定する。複製する表のコピーがすでに存在する場合、レプリケーション・センターはそれらを使用します。表が存在しない場合は、レプリケーション・センターがそれらを作成します。
 - 使用するレプリケーション・キュー・マップを指定する。適切なレプリケーション・キュー・マップがない場合は、新しいものを作成します。詳細については、95 ページの『レプリケーション・キュー・マップの作成』を参照してください。
 3. 「ソース表」ページで、複製元のソース表を選択する。
 4. 「ターゲット表」ページで、ターゲット・オブジェクト・プロファイルを検討する。必要に応じてプロファイルを変更し、Q サブスクリプション用のターゲット表を必要に合わせるようにします。

ターゲット・オブジェクト・プロファイルは、既存のターゲット表を使用するかどうか、または新しいものを作成するかどうかを決定します。レプリケーション・センターは、プロファイルで定義されている命名スキーマに一致するオブジェクトを探し、なければそのオブジェクトを作成します。

5. 「競合」ページで、Q アプライ・プログラムが競合を処理する方法を指定する。詳細については、『競合検出のオプション (双方向レプリケーション)』を参照してください。
6. 「エラー」ページで、Q アプライ・プログラムがエラーにตอบสนองする方法を指定する。詳細については、118 ページの『Q レプリケーションのエラー・オプション』を参照してください。
7. 「ターゲット表のロード」ページで以下を行う。
 - a. ターゲット表のロードのオプションを指定する。詳細については、163 ページの『Q レプリケーションのターゲット表のロードのオプション -- 概要』を参照してください。
 - b. Q サブスクリプションが作成された時に、それらをすぐにアクティブにするかどうかを指定する。
8. 「Q サブスクリプションの検討」ページで、各 Q サブスクリプションが有効かどうかを確認する。
 - Q サブスクリプションに関して何らかの変更を加える場合、その Q サブスクリプションのプロパティを変更する。
 - いずれかの Q サブスクリプションの情報が欠落している場合は、そのプロパティを変更して必要な情報を追加します。
9. 「サマリー」ページで、「完了」をクリックします。

関連概念:

- 133 ページの『双方向レプリケーション』
- 133 ページの『ソースからターゲットへのレプリケーションのセットアップ (複数方向) -- 概要』
- 153 ページの『ピアツーピア・レプリケーションの Q サブスクリプションの作成 -- 概要』
- 5 ページの『Q サブスクリプション』
- 98 ページの『単一方向レプリケーションの Q サブスクリプションの作成 -- 概要』
- 99 ページの『Q レプリケーションのターゲット・オブジェクト・プロファイル』

競合検出のオプション (双方向レプリケーション)

競合規則および競合アクションの選択は、行アプライの動作に影響します。競合規則は、競合を検出するために検査されるデータの量と、検出される競合のタイプを決定します。より多くのデータの競合を検査しようとする、Q キャプチャー・プログラムは、より多くのデータを検査できるようにするために Q アプライ・プログラムに送信しなければならない、これによりパフォーマンスとネットワーク・トラフィックに影響を受ける可能性があります。

双方向レプリケーションの競合検出の場合は、ソース・サーバーにある変更前の値が、ターゲット・サーバーにある現行値と比較されます。Q キャプチャー・プログラムは、選択する競合検出のレベルに基づいて、さまざまな変更前の値と変更後の値の組み合わせを Q アプライ・プログラムに送信します。ここに記載されている情報は、競合検出のレベルに関してさらに詳細な情報を得る上で役立ちます。

LOB データ・タイプの制約事項: 双方向レプリケーションでの競合の検出には変更前の値が使用されますが、Q レプリケーションではLOB データ・タイプの変更前の値は複製されないため、LOB 列の競合は検出されません。

以下のセクションでは、双方向レプリケーションにおける競合検出のオプションと、さまざまな競合オプションの組み合わせの結果について説明します。

- 『Q アプライ・プログラムによる競合のチェック方法』
- 149 ページの『各サーバーでの競合の解決方法』
- 149 ページの『競合の検査と解決のためのさまざまな選択の結果』

Q アプライ・プログラムによる競合のチェック方法

Q アプライ・プログラムが競合を判別するときにチェックする列のグループとして、以下のいずれかを選択できます。

- キー列のみ
- キー列と変更された列
- 全列

キー列のみ

Q アプライ・プログラムは、キー列の値を検査することにより、行を更新または削除しようとします。Q アプライ・プログラムは、以下の競合を検出します。

- 行がターゲット表にない。
- 行が、ターゲット表にすでに存在する行の重複である。

この競合規則では、Q キャプチャー・プログラムによって競合検査のために Q アプライ・プログラムに送信されるデータの量が最小になります。変更前の値は送信されず、変更された列の変更後の値のみが送信されます。

キー列と変更された列

Q アプライ・プログラムは、キー列と、更新で変更された列を検査することにより、行を更新または削除しようとします。Q アプライ・プログラムは、以下のような競合を検出します。

- 行がターゲット表にない。
- 行が、ターゲット表にすでに存在する行の重複である。
- 行が両方のサーバーで同時に更新され、同一の列値が変更された。

行が両方のサーバーで同時に更新され、異なる列値が変更された場合は、競合はありません。この競合規則では、異なる列に影響を及ぼしている更新が Q アプライ・プログラムによって同一の行にマージされます。Q アプライ・プログラムは、この競合アクションのために、変更された列の変更前の値を必要とするため、Q キャプチャー・プログラムは、変更された列の変更前の値を送信します。

全列

Q アプライ・プログラムは、ターゲット表にあるすべての列を検査することにより、行を更新または削除しようとします。この競合規則では、Q キャプチャー・プログラムによって競合検査のために Q アプライ・プログラムに送信されるデータの量が最大になります。

各サーバーでの競合の解決方法

サーバーごとに、競合の発生時に各サーバーが取るアクションを選択できます。各サーバーは、競合行をそのターゲット表に強制的に適用するか、または競合を無視できます。これらの強制適用および無視のオプションを、次の 2 つの異なる方法で組み合わせることにより、Q アプライ・プログラムに対して異なる動作を規定できます。

一方のサーバーが競合を強制的に適用し、他方のサーバーが競合を無視する

競合が発生した場合は、一方のサーバー（無視の競合アクションを持つサーバー）が優先されます。このサーバーが「マスター」またはデータのソースの所有者となります。行が両方のサーバーで同時に更新され、同一の列値が変更された場合、マスター・サーバー（無視の競合アクションを持つサーバー）は競合を無視し、マスター・サーバーの行が他方のサーバー（強制適用の競合アクションを持つサーバー）上にあるターゲット表に強制的に適用されます。この競合アクションでは、Q キャプチャー・プログラムは、すべての列の変更前の値を Q アプライ・プログラムに送信します。Q アプライ・プログラムは、すべての競合を IBMQREP_EXCEPTIONS 表に記録します。

両方のサーバーが競合を無視する

行が見つからないため、または行がターゲット表の既存の行の重複であるために競合が発生した場合、Q アプライ・プログラムはそのたびにその競合を IBMQREP_EXCEPTIONS 表に記録しますが、そうでない場合は競合を無視します。この競合アクションでは、Q キャプチャー・プログラムは、変更前の値を競合検査のために Q アプライ・プログラムに送信しません。変更された列の変更後の値のみが送信されます。

推奨事項: 2 つのサーバーの間で競合が発生すると予期されない場合、および Q キャプチャー・プログラムと Q アプライ・プログラムによる競合検出に使用されるオーバーヘッドを最小限に抑える場合は、両方のサーバーで競合を無視するよう設定してください。

競合の検査と解決のためのさまざまな選択の結果

150 ページの表 11 では、競合検出のための選択できるさまざまなオプションの組み合わせの結果を示しています。すべてのケースにおいて、最初のサーバーは、ウィザードを開いたサーバーです。

表 11. さまざまな競合検出オプションの組み合わせの結果

競合の検査方法	競合の解決方法	結果
すべての列で競合を検査する。	最初のサーバーを優先する。	最初のサーバーから 2 番目のサーバーへの Q サブスクリプションの場合: 最初のサーバーにあるソース表に加えられた変更が、2 番目のサーバーにあるターゲット表のデータと競合する場合、Q アプライ・プログラムはソースの変更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、その競合を IBMQREP_EXCEPTIONS 表に記録し、ターゲット表の競合行を削除して、ソース表の行を挿入します。
		2 番目のサーバーから最初のサーバーへの Q サブスクリプションの場合: ソース表に加えられた変更が、ターゲット表のデータと競合する場合、Q アプライ・プログラムはその競合を記録しますが、変更をターゲット表に強制的に適用することはありません。
	2 番目のサーバーを優先する。	最初のサーバーから 2 番目のサーバーへの Q サブスクリプションの場合: ソース表に加えられた変更がターゲット表のデータと競合する場合、Q アプライ・プログラムはその競合を記録しますが、変更をターゲット表に強制的に適用することはありません。
		2 番目のサーバーから最初のサーバーへの Q サブスクリプションの場合: ソース表に加えられた変更が、ターゲット表のデータと競合する場合、Q アプライ・プログラムはソースの変更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、ターゲット表の競合行を削除して、ソース表の行を挿入します。
	いずれのサーバーも優先しない。	Q アプライ・プログラムは、すべての競合を IBMQREP_EXCEPTIONS 表に記録し、処理を続行します。時間の経過とともに、論理表の 2 つのコピーは異なるものになっていきます。

表 11. さまざまな競合検出オプションの組み合わせの結果 (続き)

競合の検査方法	競合の解決方法	結果
変更された非キー列でのみ競合を検査する。	最初のサーバーを優先する。	<p>最初のサーバーから 2 番目のサーバーへの Q サブスクリプションの場合: ソース表の非キー列に対する変更が、ターゲット表の対応する列に加えられた変更と競合する場合、Q アプライ・プログラムはソースの変更をターゲット表に何らかの仕方でアプライします。Q アプライ・プログラムは、ターゲット表の競合行を削除して、ソース表の行を挿入します。</p> <p>2 番目のサーバーから最初のサーバーへの Q サブスクリプションの場合: ソース表の非キー列に対する変更が、ターゲット表の対応する列に加えられた変更と競合する場合、Q アプライ・プログラムはその競合を記録しますが、その変更をターゲット表に強制的に適用することはありません。</p>
	2 番目のサーバーを優先する。	<p>最初のサーバーから 2 番目のサーバーへの Q サブスクリプションの場合: ソース表の非キー列に対する変更が、ターゲット表の対応する列に加えられた変更と競合する場合、Q アプライ・プログラムはその競合を記録しますが、その変更をターゲット表に強制的に適用することはありません。</p> <p>2 番目のサーバーから最初のサーバーへの Q サブスクリプションの場合: ソース表の非キー列に対する変更が、ターゲット表の対応する列に加えられた変更と競合する場合、Q アプライ・プログラムはソースの変更をターゲット表に何らかの仕方でアプライします。Q アプライ・プログラムは、ターゲット表の競合行を削除して、ソース表の行を挿入します。</p>
	いずれのサーバーも優先しない。	Q アプライ・プログラムは、すべての競合を IBMQREP_EXCEPTIONS 表に記録し、処理を続行します。時間の経過とともに、論理表の 2 つのコピーは異なるものになっていきます。

表 11. さまざまな競合検出オプションの組み合わせの結果 (続き)

競合の検査方法	競合の解決方法	結果
キー列でのみ競合を検査する。	最初のサーバーを優先する。	<p>最初のサーバーから 2 番目のサーバーへの Q サブスクリプションの場合: ソース表のキーに対する変更がターゲット表のキーと競合する場合、Q アプライ・プログラムはソースの変更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、ターゲット表の競合行を削除して、ソース表の行を挿入します。</p> <p>2 番目のサーバーから最初のサーバーへの Q サブスクリプションの場合: ソース表のキーに対する変更が、ターゲット表のキーと競合する場合、Q アプライ・プログラムはその競合を記録しますが、変更をターゲット表に強制的に適用することはありません。</p>
	2 番目のサーバーを優先する。	<p>最初のサーバーから 2 番目のサーバーへの Q サブスクリプションの場合: ソース表のキーに対する変更が、ターゲット表のキーと競合する場合、Q アプライ・プログラムはその競合を記録しますが、変更をターゲット表に強制的に適用することはありません。</p> <p>2 番目のサーバーから最初のサーバーへの Q サブスクリプションの場合: ソース表のキーに対する変更がターゲット表のキーと競合する場合、Q アプライ・プログラムはソースの変更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、ターゲット表の競合行を削除して、ソース表の行を挿入します。</p>
	いずれのサーバーも優先しない。	Q アプライ・プログラムは、すべての競合を IBMQREP_EXCEPTIONS 表に記録し、処理を続行します。時間の経過とともに、論理表の 2 つのコピーは異なるものになっていきます。

関連概念:

- 133 ページの『双方向レプリケーション』

関連タスク:

- 144 ページの『双方向レプリケーションの Q サブスクリプションの作成』

ピアツーピア・レプリケーションの Q サブスクリプションの作成

ピアツーピア・レプリケーションの Q サブスクリプションの作成 -- 概要

対等表を相互にマップするように Q サブスクリプションを作成すると、変更を各表との間で複製できます。このタスクは、ソースからターゲット へのレプリケーション (複数方向) をセットアップするという、さらに大きなタスクの一部です。

以下のトピックでは、Q サブスクリプションと、必要に応じて選択するオプションの作成方法について説明します。

- 『2 つのサーバーによるピアツーピア・レプリケーションの Q サブスクリプションの作成』
- 155 ページの『3 つ以上のサーバーによるピアツーピア・レプリケーションの Q サブスクリプションの作成』
- 118 ページの『Q レプリケーションのエラー・オプション』

関連概念:

- 133 ページの『ソースからターゲットへのレプリケーションのセットアップ (複数方向) -- 概要』
- 136 ページの『ピアツーピア・レプリケーション』
- 5 ページの『Q サブスクリプション』

2 つのサーバーによるピアツーピア・レプリケーションの Q サブスクリプションの作成

2 つのサーバーによるピアツーピア・レプリケーションでは、表のコピーの 1 つに加えられた変更内容は、その表の 2 つ目のコピーに複製されます。2 つ目のコピーの変更内容は、1 つ目のコピーに複製して戻されます。ピアツーピア・レプリケーションでは、すべての行および列が複製され、表の各コピーの列名が一致していなければなりません。

1 つ目の対等表から 2 つ目の対等表への方向に 1 つの Q サブスクリプションを作成し、別の Q サブスクリプションを 2 つ目の対等表から 1 つ目の対等表への方向に作成します。レプリケーション・センターを使用してピアツーピア・レプリケーションの Q サブスクリプションを作成する場合、レプリケーション・センターは、両方の Q サブスクリプションを一度に作成します。

前提条件:

ピアツーピア・レプリケーションの Q サブスクリプションを作成する前に、以下のアクションを実行する必要があります。

- ピアツーピア・レプリケーションに参加する各サーバーごとに、Q キャプチャー・プログラムと Q アプライ・プログラム用のコントロール表を作成する。詳細については、87 ページの『Q キャプチャー・プログラムと Q アプライ・プログラムのコントロール表の作成』を参照してください。

重要: 個々のサーバー上にある Q キャプチャー・プログラム用のコントロール表と Q アプライ・プログラム用のコントロール表は、同じスキーマ名を持つ必要があります。たとえば、SERVER_RED という名前のサーバーと

SERVER_GREEN という名前のサーバーがある場合は、SERVER_RED 上にある Q キャプチャー・プログラムと Q アプライ・プログラムの両方が同一のスキーマを持ち、SERVER_GREEN 上にある Q キャプチャー・プログラムと Q アプライ・プログラムの両方が同一のスキーマを持っている必要があります。

- サーバー・ペア間でデータを転送する 2 つのレプリケーション・キュー・マップを作成する。表の 1 つ目のコピーから 2 つ目のコピーへのデータ・レプリケーション用に 1 つ、表の 2 つ目のコピーから 1 つ目のコピーへのデータ・レプリケーション用に 1 つのレプリケーション・キュー・マップが必要です。(このタスクは、Q サブスクリプションを作成する前か、Q サブスクリプションを作成する際に行うことができます。) 詳細については、95 ページの『レプリケーション・キュー・マップの作成』を参照してください。

制約事項:

- ピアツーピア・レプリケーションの Q サブスクリプションの表は表示できません。
- ストアード・プロシージャを、ピアツーピア・レプリケーションのターゲットとすることはできません。

手順:

2 つのサーバーによるピアツーピア・レプリケーションの Q サブスクリプションを作成するには、次のようにします。

1. ウィザードをオープンし、2 つのサーバーによるピアツーピア・レプリケーションの Q サブスクリプションを作成することを指定する。
 - a. レプリケーション・センターから、「Q サブスクリプションの作成」ウィザードをオープンする。

このウィザードは、以下のいずれかの方法でオープンします。

- Q サブスクリプションの変更をキャプチャーする Q キャプチャー・プログラムを示す Q キャプチャー・スキーマを展開する。「サブスクリプション」フォルダーを右マウス・ボタンでクリックして、「作成」を選択します。
- Q サブスクリプションの変更をアプライする Q アプライ・プログラムを示す Q アプライ・スキーマを展開する。「サブスクリプション」フォルダーを右マウス・ボタンでクリックして、「作成」を選択します。

詳細については、オンライン・ヘルプを参照してください。

- b. 「レプリケーション」ページで、2 つのサーバーによるピアツーピア・レプリケーションを指定する。
2. 「サーバー」ページで、次のようにする。
 - 第 1 の対等サーバーとスキーマ (複製する表が含まれているもの) に関する情報を指定する。
 - 第 2 の対等サーバーとスキーマに関する情報を指定する。複製する表のコピーがすでに存在する場合、レプリケーション・センターはそれらを使用します。表が存在しない場合は、レプリケーション・センターがそれらを作成します。

- 使用するレプリケーション・キュー・マップを指定する。適切なレプリケーション・キュー・マップがない場合は、新しいものを作成します。詳細については、95 ページの『レプリケーション・キュー・マップの作成』を参照してください。
3. 「ソース表」 ページで、複製元のソース表を選択する。
 4. 「ターゲット表」 ページで、ターゲット・オブジェクト・プロファイルを検討する。 Q サブスクリプションのターゲット表が必要に合ったものとなるよう、必要に応じてプロファイルを変更する。

ターゲット・オブジェクト・プロファイルによって、既存のターゲット表が使用されるか、または新しく作成されるかが決まります。レプリケーション・センターは、プロファイルで定義されている命名スキーマに一致するオブジェクトを探し、なければそのオブジェクトを作成します。

5. 「エラー」 ページで、エラーに対する Q アプライ・プログラムの応答方法を指定する。詳細については、118 ページの『Q レプリケーションのエラー・オプション』を参照してください。
6. 「ターゲット表のロード」 ページで、次のようにする。
 - a. ターゲット表のロードのオプションを指定する。詳細については、163 ページの『Q レプリケーションのターゲット表のロードのオプション -- 概要』を参照してください。
 - b. Q サブスクリプションが作成された時に、それらをすぐにアクティブにするかどうかを指定する。
7. 「Q サブスクリプションの検討」 ページで、各 Q サブスクリプションが有効なことを確認する。
 - Q サブスクリプションに関して何らかの変更を加える場合、その Q サブスクリプションのプロパティを変更する。
 - いずれかの Q サブスクリプションの情報が欠落している場合は、そのプロパティを変更して必要な情報を追加する。
8. 「サマリー」 ページで、「完了」をクリックする。

関連概念:

- 98 ページの『単一方向レプリケーションの Q サブスクリプションの作成 -- 概要』
- 99 ページの『Q レプリケーションのターゲット・オブジェクト・プロファイル』

関連タスク:

- 155 ページの『3 つ以上のサーバーによるピアツーピア・レプリケーションの Q サブスクリプションの作成』

3 つ以上のサーバーによるピアツーピア・レプリケーションの Q サブスクリプションの作成

3 つ以上のサーバーによるピアツーピア・レプリケーションでは、表の各コピーに対する変更が、その表の他のすべてのコピーに複製されます。すべての行および列が複製され、表の各コピーの列名は一致していなければなりません。

ソースからターゲットへの各ペアに対して 1 つの Q サブスクリプションが作成されます。3 つのサーバー間で相互に複製する表が 1 つある場合は、6 つの Q サブスクリプションが作成されます。3 つのサーバー間で相互に複製する表が 4 つある場合は、12 の Q サブスクリプションが作成されます。Q サブスクリプション数を決定する公式は、 $n*(n-1)$ (n は、サーバーの数) です。レプリケーション・センターを使用してピアツーピア・レプリケーションの Q サブスクリプションを作成する場合、レプリケーション・センターは、必要なすべての Q サブスクリプションを一度に作成します。

前提条件:

ピアツーピア・レプリケーションの Q サブスクリプションを作成する前に、以下のアクションを実行する必要があります。

- ピアツーピア・レプリケーションに参加する各サーバーごとに、Q キャプチャー・プログラムと Q アプライ・プログラム用のコントロール表を作成する。詳細については、87 ページの『Q キャプチャー・プログラムと Q アプライ・プログラムのコントロール表の作成』を参照してください。

重要: 個々のサーバー上にある Q キャプチャー・プログラム用のコントロール表と Q アプライ・プログラム用のコントロール表は、同じスキーマ名を持つ必要があります。たとえば、SERVER_RED という名前のサーバーと SERVER_GREEN という名前のサーバー、および SERVER_BLUE という名前のサーバーがある場合は、SERVER_RED 上にある Q キャプチャー・プログラムと Q アプライ・プログラムの両方が同一のスキーマを持ち、SERVER_GREEN 上にある Q キャプチャー・プログラムと Q アプライ・プログラムの両方が同一のスキーマを持ち、SERVER_BLUE 上にある Q キャプチャー・プログラムと Q アプライ・プログラムの両方が同一のスキーマを持っている必要があります。

- 各サーバー・ペア間でデータを転送するレプリケーション・キュー・マップを作成する。ソースからターゲットへのペアそれぞれに対して 1 つのレプリケーション・キュー・マップが必要です。3 つのサーバー間で複製する表が 1 つある場合は、6 つのレプリケーション・キュー・マップが必要です。4 つのサーバー間で複製する表が 1 つある場合は、12 のレプリケーション・キュー・マップが必要です。(このタスクは、Q サブスクリプションを作成する前か、Q サブスクリプションを定義する際に行うことができます。)詳細については、95 ページの『レプリケーション・キュー・マップの作成』を参照してください。

制約事項:

- ピアツーピア・レプリケーションの Q サブスクリプションの表は表示できません。
- ストアード・プロシージャーを、ピアツーピア・レプリケーションのターゲットとすることはできません。

手順:

3 つ以上のサーバーによるピアツーピア・レプリケーションの Q サブスクリプションを作成するには、次のようにします。

1. ウィザードをオープンし、3 つ以上のサーバーによるピアツーピア・レプリケーションの Q サブスクリプションを作成することを指定する。
 - a. レプリケーション・センターから、「Q サブスクリプションの作成」ウィザードをオープンする。

このウィザードは、以下のいずれかの方法でオープンします。

- Q サブスクリプションの変更をキャプチャーする Q キャプチャー・プログラムを示す Q キャプチャー・スキーマを展開する。「サブスクリプション」フォルダーを右マウス・ボタンでクリックして、「作成」を選択します。
- Q サブスクリプションの変更をアプライする Q アプライ・プログラムを示す Q アプライ・スキーマを展開する。「サブスクリプション」フォルダーを右マウス・ボタンでクリックして、「作成」を選択します。

詳細については、オンライン・ヘルプを参照してください。

- b. 「レプリケーション」ページで、3 つ以上のサーバーによるピアツーピア・レプリケーションを指定する。
2. 「サーバー」ページで、次のようにする。
 - 第 1 の対等サーバーとスキーマ (複製する表が含まれているもの) に関する情報を指定する。
 - それぞれの追加の対等サーバーとスキーマに関する情報を指定する。複製する表のコピーがすでにサーバー上に存在する場合、レプリケーション・センターはそれらを使用します。表が存在しない場合は、レプリケーション・センターがそれらを作成します。

ピアツーピア・レプリケーションに参加するサーバーを指定するたびに、その新しいサーバーから他の選択済みのそれぞれのサーバーへのレプリケーション・キュー・マップと、それぞれの既存のサーバーからその新しいサーバーへのレプリケーション・キュー・マップを指定する必要があります。

- 使用するレプリケーション・キュー・マップを指定する。適切なレプリケーション・キュー・マップがない場合は、新しいものを作成します。詳細については、95 ページの『レプリケーション・キュー・マップの作成』を参照してください。

3. 「ソース表」ページで、複製元のソース表を選択する。
4. 「ターゲット表」ページで、既存の (基本) 表を含むサーバー以外の各サーバーのターゲット・オブジェクト・プロファイルを検討する。Q サブスクリプションのターゲット表が必要に合ったものとなるよう、必要に応じてプロファイルを変更する。

たとえば、SERVER_RED に既存の表が含まれており、ピアツーピア構成に SERVER_RED、SERVER_BLUE、および SERVER_GREEN という名前のサーバーが含まれている場合、レプリケーション・センターは、SERVER_BLUE および SERVER_GREEN のターゲット・オブジェクト・プロファイルを表示しません。

5. 「エラー」ページで、エラーに対する Q アプライ・プログラムの応答方法を指定する。詳細については、118 ページの『Q レプリケーションのエラー・オブション』を参照してください。
6. 「ターゲット表のロード」ページで、ターゲット表のロードのオプションを指定する。詳細については、163 ページの『Q レプリケーションのターゲット表のロードのオプション -- 概要』を参照してください。

3 つ以上のサーバーによるピアツーピア・レプリケーションの Q サブスクリプションを作成する場合、Q サブスクリプションは常に非アクティブ状態で作成されるので、それらをアクティブ化する必要があります。

7. 「Q サブスクリプションの検討」 ページで、各 Q サブスクリプションが有効なことを確認する。
 - Q サブスクリプションに関して何らかの変更を加える場合、その Q サブスクリプションのプロパティを変更する。
 - いずれかの Q サブスクリプションの情報が欠落している場合は、そのプロパティを変更して必要な情報を追加する。
8. 「サマリー」 ページで、「完了」をクリックする。

関連概念:

- 98 ページの『単一方向レプリケーションの Q サブスクリプションの作成 -- 概要』
- 99 ページの『Q レプリケーションのターゲット・オブジェクト・プロファイル』

関連タスク:

- 153 ページの『2 つのサーバーによるピアツーピア・レプリケーションの Q サブスクリプションの作成』

2 つのサーバーを使用した双方向およびピアツーピア・レプリケーションの開始

デフォルトでは、2 つのサーバーを使って双方向またはピアツーピア・レプリケーションに対して Q サブスクリプションを作成する場合、新規の Q サブスクリプションは、Q キャプチャー・プログラムの開始時または再初期化時に自動的にアクティブになります。以下の状況では、2 つのサーバーを使った双方向またはピアツーピア・レプリケーションで、レプリケーションを手動で開始することが必要です。

- Q サブスクリプションは、新規の場合は自動的にアクティブにならないように作成されています。
- 論理表のレプリケーションは停止されており、再びレプリケーションを開始する予定です。(Q サブスクリプションは、新規の場合のみ自動的にアクティブにされます。)

前提条件:

- 論理表に対する両方の Q サブスクリプションが定義されている必要があります。
- 両方の Q サブスクリプションが I (非アクティブ) 状態でなければなりません。

手順:

2 つのサーバーを使った双方向またはピアツーピア・レプリケーションで、論理表に対するレプリケーションを手動で開始するには、論理表に対する 2 つの Q サブスクリプションのうち 1 つをアクティブにします。Q キャプチャー・プログラムおよび Q アプライ・プログラムが、他の Q サブスクリプションを自動的にアクティブにします。

Q サブスクリプションがロード・フェーズを指定する場合、アクティブにする Q サブスクリプションのソース表が、他のサーバーでターゲット表をロードするために使用されます。

関連概念:

- 163 ページの『Q レプリケーションのターゲット表のロードのオプション -- 概要』

関連タスク:

- 144 ページの『双方向レプリケーションの Q サブスクリプションの作成』
- 253 ページの『Q サブスクリプションまたは XML 発行のアクティブ化』
- 153 ページの『2 つのサーバーによるピアツーピア・レプリケーションの Q サブスクリプションの作成』

2 つのサーバーを使用した双方向およびピアツーピア・レプリケーションの停止

2 つのサーバーを使用した双方向またはピアツーピア・レプリケーションでは、Q キャプチャーまたは Q アプライ・プログラムを停止しなくても、論理表のレプリケーションを停止することができます。これを実行するには、論理表に対して Q サブスクリプションを非アクティブにします。他の論理表のレプリケーションは、2 つのサーバー間で継続します。

前提条件:

論理表に対する Q サブスクリプションが A (アクティブ) 状態である必要があります。

制約事項:

2 つの Q サブスクリプションのうち 1 つだけを論理表に対して非アクティブにすることはできません。1 つの Q サブスクリプションを非アクティブにすると、もう一方も自動的に非アクティブになります。

手順:

2 つのサーバーを使った双方向またはピアツーピア・レプリケーションで、論理表に対するレプリケーションを停止するには、2 つの Q サブスクリプションのうち 1 つを非アクティブにします。

関連タスク:

- 254 ページの『Q サブスクリプションまたは XML 発行の非アクティブ化』

3 つ以上のサーバーを使用したピアツーピア・グループでのレプリケーションの開始

3 つ以上のサーバーを使ってピアツーピア・レプリケーションで論理表に対する Q サブスクリプションを作成した後、Q サブスクリプションのグループをアクティブにしてレプリケーションを開始することが必要です。

また、論理表に対するすべての Q サブスクリプションを非アクティブにしてあり、再びレプリケーションを開始したい場合も、グループをアクティブにすることが必要です。

3 つ以上のサーバーを使用するピアツーピア・グループをアクティブにするプロセスは、いくつかの段階に分かれています。まず、2 つのサーバー間の論理表に対して Q サブスクリプションをアクティブにしてから、すべてのサーバーがアクティブに論理表を複製できるようになるまで、一度に 1 つずつ新規サーバーを追加します。

前提条件:

Q キャプチャーおよび Q アプライ・プログラムは、グループ中のすべてのサーバーで実行していることが必要です。

手順:

3 つ以上のサーバーを使用してピアツーピア・グループでのレプリケーションを開始するには、次のようにします。

1. グループ中の 2 つのサーバーを選択し、アクティブ化のプロセスを開始します。
2. 2 つのサーバー間で、2 つの Q サブスクリプションのうち 1 つを論理表に対してアクティブにします。

Q キャプチャー・プログラムおよび Q アプライ・プログラムが、2 つのサーバー間で、この論理表に対して他の Q サブスクリプションを自動的にアクティブにします。

注: Q サブスクリプションがロード・フェーズを指定する場合、アクティブにする Q サブスクリプションは、ロード元にする表をソース表とする Q サブスクリプションでなければなりません。この表は、他のサーバーで表をロードするために使用されます。

3. 両方の Q サブスクリプションがアクティブになったら、グループに入れる新規サーバーを選択します。
4. アクティブに論理表を複製しているサーバーの 1 つを選択します。
5. ステップ 4 で選択したサーバーをソースとして、また新規のサーバーをターゲットとして指定する論理表の Q サブスクリプションをアクティブにします。

Q キャプチャー・プログラムおよび Q アプライ・プログラムが、新規サーバーとアクティブに複製しているサーバーとの間で、論理表に対して他の Q サブス

クリプションをアクティブにします。新規サーバーおよびアクティブに複製している他のサーバー間の、論理表に対する Q サブスクリプションもアクティブになります。

この時点で、すべてのサーバー間で、すべての方向でレプリケーションが開始します。

6. グループ中のすべての Q サブスクリプションがアクティブになるまで、ステップ 3、4、および 5 を繰り返します。

重要: グループに追加できる新規サーバーは、一度に 1 つだけです。他のサーバーがアクティブに論理表を複製し始めてから、新規サーバー追加のプロセスを開始してください。

関連概念:

- 163 ページの『Q レプリケーションのターゲット表のロードのオプション -- 概要』

関連タスク:

- 253 ページの『Q サブスクリプションまたは XML 発行のアクティブ化』
- 155 ページの『3 つ以上のサーバーによるピアツーピア・レプリケーションの Q サブスクリプションの作成』

3 つ以上のサーバーを使用したピアツーピア・グループでのレプリケーションの停止

3 つ以上のサーバーを使用したピアツーピア・レプリケーションでは、Q キャプチャーまたは Q アプライ・プログラムを停止したり、ピアツーピア・グループ中の他の論理表のレプリケーションを停止したりしなくても、論理表のレプリケーションを停止することができます。

論理表のレプリケーションを、グループ中の 1 つのサーバーで、またはすべてのサーバーで停止できます。

前提条件:

論理表に対する Q サブスクリプションが A (アクティブ) 状態である必要があります。

手順:

- 1 つのサーバーで論理表のレプリケーションを停止するには、次のようにします。
 1. アクティブに論理表を複製している、グループ中のサーバーを選択する。
 2. このサーバーをソースとして指定し、レプリケーションを停止させるサーバーをターゲットとして指定している Q サブスクリプションを非アクティブにする。

Q キャプチャー・プログラムおよび Q アプライ・プログラムが、残りのサーバー間で、この論理表に対して他の Q サブスクリプションを自動的に非アクティブにします。

グループ中のすべてのサーバーで論理表のレプリケーションを停止するには、論理表に対するすべての Q サブスクリプションが非アクティブになるまで、一度の 1 つのサーバーずつ、同じ手順を繰り返します。

関連タスク:

- 254 ページの『Q サブスクリプションまたは XML 発行の非アクティブ化』

第 11 章 Q レプリケーションにターゲット表をロードするためのオプション

Q レプリケーションのターゲット表のロードのオプション -- 概要

Q サブスクリプションを作成するときは、ターゲット表にソースからのデータをロードするための以下のオプションを選択できます。

自動ロード

Q アプライ・プログラムが、ターゲット表のロードを管理します。Q アプライ・プログラムが呼び出すロード・ユーティリティーを選択するか、ご使用のオペレーティング・システムとバージョンで利用できる最善のユーティリティーを Q アプライ・プログラムに選択させることができます。

手動ロード

ユーザーがターゲット表のロードを処理し、ロードが完了したらレプリケーション・プログラムに知らせます。このオプションは、外部ロードと呼ばれます。

ロードなし

ターゲット表をロードしないか、レプリケーション・プログラムのコンテキスト外でターゲット表をロードします。

以下のトピックでは、ロード・オプションについて詳しく説明します。

- 164 ページの『Q レプリケーションのターゲット表をロードする際の推奨事項』
- 165 ページの『Q レプリケーションの自動ロード・オプションで使用するユーティリティー』
- 169 ページの『Q レプリケーションの手動ロード・オプション』
- 172 ページの『Q レプリケーションのロードなしオプション』
- 173 ページの『さまざまなタイプの Q レプリケーションのロード・オプション』
- 166 ページの『z/OS プラットフォーム用の自動ロードに関する考慮事項』

関連概念:

- 98 ページの『単一方向レプリケーションの Q サブスクリプションの作成 -- 概要』

関連タスク:

- 144 ページの『双方向レプリケーションの Q サブスクリプションの作成』
- 153 ページの『2 つのサーバーによるピアツーピア・レプリケーションの Q サブスクリプションの作成』
- 155 ページの『3 つ以上のサーバーによるピアツーピア・レプリケーションの Q サブスクリプションの作成』
- 103 ページの『単一方向レプリケーションの複数の Q サブスクリプションの作成』

- 100 ページの『単一方向レプリケーションの単一の Q サブスクリプションの作成』

Q レプリケーションのターゲット表をロードする際の推奨事項

Q レプリケーションはターゲット表の自動ロード用に構成することができ、データを失うことなくロード・プロセス中にレプリケーションを継続できるように設計されています。以下に、確実にこのプロセスを円滑に実行するための推奨事項を挙げます。

ターゲットでのアプリケーション

ターゲット表のロード中に、アプリケーションにターゲット表を更新させないようにします。表の中のデータが、表のロード中に不整合になり、Q アプライ・プログラムが、ターゲット表および他の関連したターゲット表がロードされるまで参照保全制約を除去してしまいます。

アプリケーションは、Q アプライ・プログラムが `IBMQREP_TARGETS` 表に以下の変更を加えると、再びターゲット表を安全に使用できます。

- `STATE` 列を `A` (アクティブ) に設定する。
- `STATE_INFO` 列を `ASN7606I` に設定する。つまり、ターゲット表に参照保全制約がある場合、それらはリストアされているということです。

レプリケーション・センターの「Q サブスクリプションの管理 (Manage Q Subscriptions)」ウィンドウを使用して、これらの 2 つの変更が加えられていることを検査することができます。このウィンドウをオープンするには、Q サブスクリプションのソース表が置かれている Q キャプチャー・サーバーを右マウス・ボタンでクリックし、「管理 (Manage)」→「Q サブスクリプション (Q Subscriptions)」を選択します。詳細については、オンライン・ヘルプを参照してください。

ロード・プロセスが失敗した場合、または Q アプライ・プログラムがロード中に停止した場合、ロードの開始前にターゲット表にあったデータは削除されます。ロード・プロセス中にソース表に加えられた変更は失われず、正常にロードされた後のターゲット表に適用されます。

ソースでのアプリケーション

ソースでは、相対的な非活動時にターゲット表をロードしてください。

関連概念:

- 163 ページの『Q レプリケーションのターゲット表のロードのオプション -- 概要』

Q レプリケーションの自動ロード・オプション

Q レプリケーションの自動ロード・オプション -- 概要

Q サブスクリプションがアクティブにされるときに、Q アプライ・プログラムに Q サブスクリプションのターゲット表をロードさせることができます。自動ロード と呼ばれるこのオプションは、Q レプリケーションのデフォルトです。

自動ロードを指定している場合、デフォルトでは、Q アプライ・プログラムが、ご使用のオペレーティング・システムとバージョンで利用できる最善のユーティリティーを選択します。必要なら、Q サブスクリプションの作成時に Q アプライ・プログラムが使用するロード・ユーティリティーを指定することができます。

自動ロード・プロセス中、キャプチャーされて Q アプライ・プログラムに送信されるソース・トランザクションは、Q アプライ・プログラムによって一時予備キューに置かれます。このため、ロード・プロセス中もレプリケーションは続行できます。Q アプライ・プログラムは、ターゲット表がロードされた後に、これらのトランザクションをアプライします。

以下のトピックでは、自動ロードについて詳しく説明します。

- Q レプリケーションの自動ロード・オプションで使用するユーティリティー
- z/OS プラットフォーム用の自動ロードに関する考慮事項
- Q レプリケーションの自動ロード・オプションでのニックネームの指定

関連概念:

- 173 ページの『さまざまなタイプの Q レプリケーションのロード・オプション』
- 163 ページの『Q レプリケーションのターゲット表のロードのオプション -- 概要』
- 169 ページの『Q レプリケーションの手動ロード・オプション』
- 172 ページの『Q レプリケーションのロードなしオプション』

Q レプリケーションの自動ロード・オプションで使用するユーティリティー

自動ロードを選択する場合は、Q アプライ・プログラムに使用可能な最善のロード・ユーティリティーを選択させるか、ユーザーがユーティリティーを指定できます。

以下のリストは、使用可能なロード・ユーティリティーを示しています。

CURSOR からの **LOAD**

DB2® **LOAD** ユーティリティーのオプションを使用して、中間エクスポート・ファイルを作成せずに、ソース表からターゲット表へデータを移動します。Linux、UNIX®, または Windows® オペレーティング・システム上でこのユーティリティーを指定する場合、ソース表とターゲット表が同一のデータベース内になれば、ソース表のニックネームを指定する必要がありません。

EXPORT ユーティリティーおよび **LOAD** ユーティリティー

DB2 **EXPORT** ユーティリティーと DB2 **LOAD** ユーティリティーの組み合わせを使用します。

EXPORT ユーティリティーおよび **IMPORT** ユーティリティー

DB2 **EXPORT** ユーティリティーと DB2 **IMPORT** ユーティリティーの組み合わせを使用します。

EXPORT ユーティリティーを使用する場合、ソース・サーバーとターゲット・サーバーが同じでなければ、Q アプライ・プログラムは、Q キャプチャー・サーバーに接続するためのパスワード・ファイルを必要とします。パスワード・ファイルを作成する場合は、**asnpwd** コマンドを使用します。 **apply_path** パラメーターで指定されたパスに、IXF ファイルが作成されます。

表 12 では、DB2 UDB のバージョンおよびオペレーティング・システムに応じた使用可能なロード・ユーティリティーを示しています。ロード・ユーティリティーを指定しない場合、DB2 UDB for Linux、UNIX、Windows バージョン 8.1 以上、および DB2 UDB for z/OS™ バージョン 7.1 以上では、Q アプライ・プログラムはデフォルトで CURSOR からの LOAD を使用します。CURSOR からの LOAD は、DB2 UDB for z/OS バージョン 6 ではサポートされていません。使用できないロード・ユーティリティーを指定すると、Q アプライ・プログラムは Q サブスクリプションを非アクティブにします。

表 12. Q レプリケーションのロード・オプション

DB2 のバージョン	オペレーティング・システム	ロード・オプション
バージョン 8	Linux、UNIX、Windows	<ul style="list-style-type: none"> • CURSOR からの LOAD • EXPORT ユーティリティーおよび IMPORT ユーティリティー • EXPORT ユーティリティーおよび LOAD ユーティリティー
	z/OS	CURSOR からの LOAD
バージョン 7	z/OS	CURSOR からの LOAD
バージョン 6	z/OS	ロードはユーザーが行わなければならないので、利用できるオプションは、手動ロードかロードなしのみです。

関連概念:

- 166 ページの『z/OS プラットフォーム用の自動ロードに関する考慮事項』
- 164 ページの『Q レプリケーションの自動ロード・オプション -- 概要』

z/OS プラットフォーム用の自動ロードに関する考慮事項

z/OS プラットフォームで実行している Q アプライ・プログラムがターゲット表の自動ロードを実行する際、以下の問題を考慮する必要があります。

NUMTCB パラメーターの設定

Q アプライ・プログラムは、CURSOR からの LOAD ユーティリティーを使用して z/OS でターゲット表の自動ロードを実行します。ユーティリティーを呼び出すため、Q アプライ・プログラムは DB2 UDB for z/OS に付属する DSNUTILS ストアード・プロシージャーを呼び出します。

DSNUTILS は、ワークロード・マネージャー (WLM) 環境で実行することが必要です。WLM の開始に使用される NUMTCB パラメーターは、以下のように設定する必要があります。

NUMTCB=1

DSNUTILS の詳細については、ご使用のバージョンの「*DB2 Universal Database (OS/390 および z/OS 版) ユーティリティー・ガイドおよび解説書*」を参照してください。

並列ロードの表スペースに関する考慮事項

z/OS では、複数の Q サブスクリプションを同時にアクティブにし、Q アプライ・プログラムがターゲット表の自動ロードを実行している場合、Q アプライ・プログラムはターゲット表を並列でロードします。この場合、各ターゲット表は必ず別個の表スペースになければなりません。

各ターゲット表を別個の表スペースに入れる代わりに、1 つの Q サブスクリプションのロードが次の Q サブスクリプションのロード開始前に終了するように、各 Q サブスクリプションを順番にアクティブにすることもできます。

並列ロードを回避するには、次のようにします。

1. 最初の Q サブスクリプションをアクティブにします。
2. Q サブスクリプション状態が A (アクティブ) に変更するのを待機します。

レプリケーション・センターの「Q サブスクリプションの管理 (Manage Q Subscriptions)」ウィンドウを使用するか、または IBMQREP_TARGETS コントロール表の STATE 列を確認することによって、Q サブスクリプションを検査できます。

3. 次の Q サブスクリプションをアクティブにします。

関連概念:

- 165 ページの『Q レプリケーションの自動ロード・オプションで使用するユーティリティー』
- 164 ページの『Q レプリケーションの自動ロード・オプション -- 概要』
- 335 ページの『システム・サービスを使用した Q レプリケーションおよびイベント発行プログラムの操作 (z/OS) -- 概要』

Q レプリケーションの自動ロード・オプションでのニックネームの指定

CURSOR ユーティリティーの LOAD を使用してターゲット表をロードする一部の Q サブスクリプションには、ニックネームが必要です。ニックネームは、Q アプライ・サーバー上で定義されるもので、Q キャプチャー・サーバー上のソース表を参照します。

ターゲット表が Linux、UNIX、または Windows オペレーティング・システム上にある場合、および Q キャプチャー・サーバーが Q アプライ・サーバーに対してリモートの場合は、ニックネームが必要です。

Q サブスクリプションを作成する際は、レプリケーション・センターでニックネームを作成するか、既存のニックネームを指定することができます。

前提条件:

- Q アプライ・サーバーは、フェデレーテッド・サーバーでなければなりません。
- レプリケーション・センターでニックネームを作成する場合は、サーバー定義、ラッパー、およびユーザー・マッピングを作成する必要があります。

手順:

単一方向 Q サブスクリプションに対してニックネームを指定するには、次のようにします。

単一の Q サブスクリプションを作成している場合:

レプリケーション・センターで、「Q サブスクリプションの作成」ウィザードの「ターゲット表のロード」ページを使用して、既存のニックネームを指定するか、または新規ニックネームを作成します。既存のニックネームを指定する場合は、ニックネームの名前と所有者を指定します。新規ニックネームを作成する場合は、Q アプライ・サーバー上の登録済みのサーバー定義を選択し、新規ニックネームの所有者と名前を指定します。

複数の Q サブスクリプションを作成している場合:

1. レプリケーション・センターで、「Q サブスクリプションの作成」ウィザードの「ターゲット表のロード」ページを使用して、新規ニックネームを作成するか、既存のニックネームを使用するかを指定する。新規ニックネームを作成する場合は、Q アプライ・サーバー上の登録済みのサーバー定義を選択します。
2. 「Q サブスクリプションの作成」ウィザードの「Q サブスクリプションの検討」ページから立ち上げられる「Q サブスクリプションのプロパティ」ノートブックを使用して、ニックネームが必要な各 Q サブスクリプションを選択する。各 Q サブスクリプションのプロパティ・ノートブックで、既存のニックネームの名前と所有者か、新規ニックネームを作成するために用いる名前と所有者を指定します。

「Q サブスクリプションの作成」ウィザードは、以下のいずれかの方法でオープンします。

- Q サブスクリプションの変更をキャプチャーする Q キャプチャー・プログラムを示す Q キャプチャー・スキーマを展開する。「サブスクリプション」フォルダーを右マウス・ボタンでクリックし、「作成」を選択します。
- Q サブスクリプションの変更をアプライする Q アプライ・プログラムを示す Q アプライ・スキーマを展開する。「サブスクリプション」フォルダーを右マウス・ボタンでクリックし、「作成」を選択します。

詳細については、オンライン・ヘルプを参照してください。

関連概念:

- 「フェデレーテッド・システム・ガイド」の『サーバー定義およびサーバー・オプション』
- 165 ページの『Q レプリケーションの自動ロード・オプションで使用するユーティリティー』
- 164 ページの『Q レプリケーションの自動ロード・オプション -- 概要』

関連タスク:

- 80 ページの『Q アプライ・プログラムが動作するターゲット・データベースの構成 (Linux、UNIX、Windows)』

関連資料:

- 「フェデレーテッド・システム・ガイド」の『フェデレーテッド・システムのニックネーム・オプション』
- 「フェデレーテッド・システム・ガイド」の『フェデレーテッド・システムの利用者・マッピング・オプション』
- 「フェデレーテッド・システム・ガイド」の『フェデレーテッド・システムのラッパー・オプション』

Q レプリケーションの手動ロード・オプション

Q サブスクリプションに対して手動ロード を指定する場合は、選択するユーティリティーを使用してターゲット表をロードし、その表がロードされた時点で Q レプリケーション・プログラムに通知します。

170 ページの図 14 は、手動ロード・プロセスの各段階を表しています。

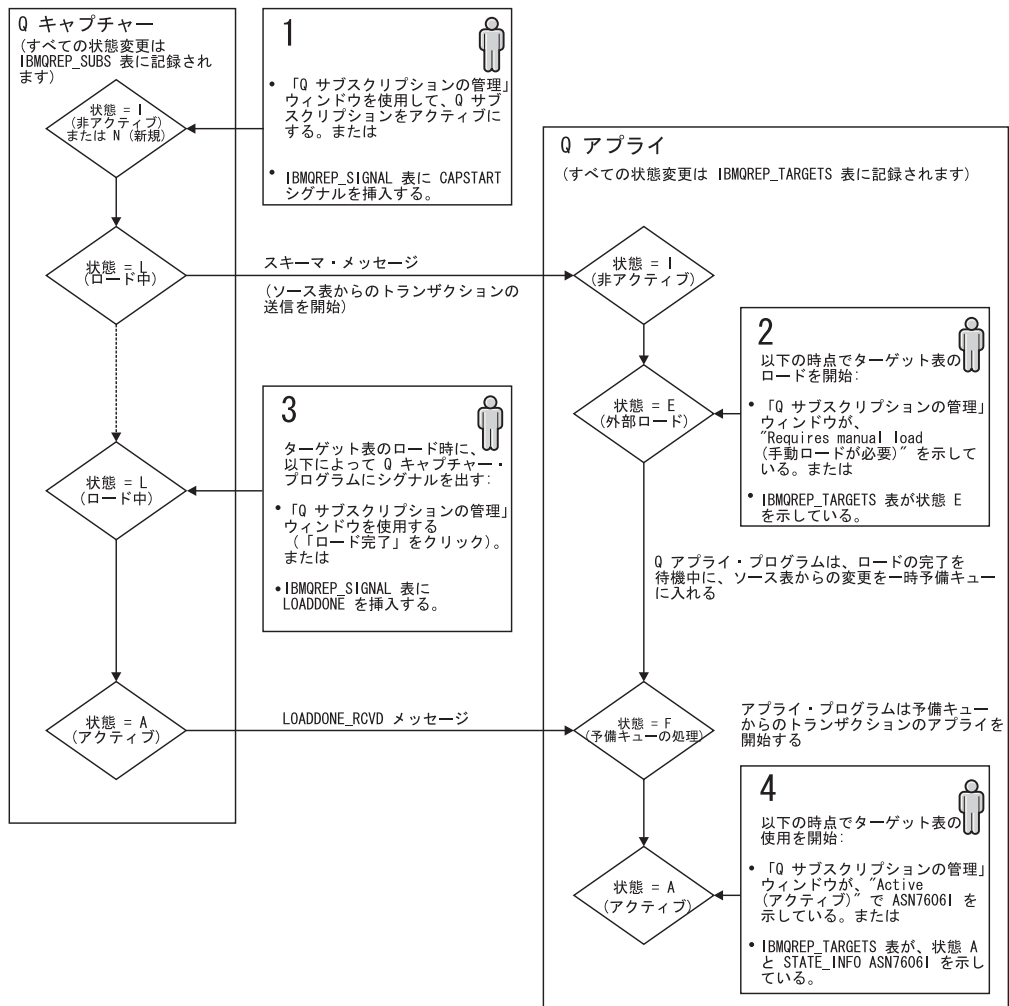


図 14. 手動ロード・プロセスの各段階: 手動ロードでは、プロセスを完了するために、次の示すいくつかの段階でアクションをとる必要があります。1) Q サブスクリプションをアクティブ化する。2) レプリケーション・センターで「手動ロードが必要」を (または IBMQREP_TARGETS 表の STATE 列で E を) 探す。3) ロードが完了した時点で Q キャプチャー・プログラムに通知する。4) レプリケーション・センターに「アクティブ」および「ASN7606I」が (または IBMQREP_TARGETS 表の STATE 列に A、STATE_INFO 列に ASN7606I が) 表示されるまで、ターゲット表を使用しないようにする。

手動ロード・プロセスは、以下のとおりです。

- Q サブスクリプションがアクティブ化されると、Q キャプチャー・プログラムが、ターゲット表が手動でロードされることを示す subscription schema メッセージを送信します。以下の変更が行われます。
 - Q キャプチャー・プログラムが、IBMQREP_SUBS コントロール表で、Q サブスクリプションの状態を I (非アクティブ) または N (新規) から L (ロード中) に変更する。
 - Q アプライ・プログラムが、IBMQREP_TARGETS コントロール表で、Q サブスクリプションの状態を I (非アクティブ) から E (外部ロード) に変更する。
 - レプリケーション・センターの「Q サブスクリプションの管理」ウィンドウに、状況が「手動ロードが必要」として表示されます。このウィンドウをオー

プンするには、Q サブスクリプションのソース表がある Q キャプチャー・サーバーを右マウス・ボタンでクリックして、「管理」→「Q サブスクリプション」の順に選択します。

- Q アプライ・プログラムが、ターゲット表に対して定義されている参照保全制約をドロップします。
2. 「Q サブスクリプションの管理」ウィンドウに「手動ロードが必要」が表示されたら、または IBMQREP_TARGETS 表に対する SELECT ステートメントにより STATE 列の値が E であることが検証されたら、選択するユーティリティーを使用してターゲット表のロードを開始できます。
 3. ターゲット表がロードされる間、Q キャプチャー・プログラムが、ソース表のトランザクションを、変更前と変更後の両方の値と共に送信します。Q アプライ・プログラムは、それらのトランザクションを、一時予備キューに書き込みます。
 4. ターゲット表がロードされた後、ロードが完了したことを Q キャプチャー・プログラムに通知します。以下のいずれかの方法を使用できます。
 - レプリケーション・センターの「Q サブスクリプションの管理」ウィンドウを使用して、ロードが完了したことを通知する。
 - 次のように、IBMQREP_SIGNAL 表に LOADDONE シグナルを挿入する。

```
insert into schema.IBMQREP_SIGNAL(  
    SIGNAL_TIME,  
    SIGNAL_TYPE,  
    SIGNAL_SUBTYPE,  
    SIGNAL_INPUT_IN,  
    SIGNAL_STATE  
) values (  
    CURRENT_TIMESTAMP,  
    'CMD',  
    'LOADDONE',  
    'subname',  
    'P');
```

schema は通知先の Q キャプチャー・プログラムを示し、*subname* は手動ロードを実行する Q サブスクリプションの名前です。

5. Q キャプチャー・プログラムに手動ロードが完了したことが通知された後、Q キャプチャー・プログラムが IBMQREP_SUBS 表で Q サブスクリプションの状態を A (アクティブ) に変更し、Q サブスクリプションに対して定義されている列サブセット化オプションの使用を開始します。Q キャプチャー・プログラムが load done received メッセージを Q アプライ・プログラムに送信します。
6. Q アプライ・プログラムが、Q サブスクリプションの状態を F (予備キューを処理中) に変更し、予備キューからトランザクションのアプライを開始します。Q アプライ・プログラムは、終了すると、予備キューを削除します。
7. Q アプライ・プログラムは、従属の Q サブスクリプションがそのロード・フェーズを完了するまで待ってから、参照保全制約をターゲット表に戻します。
8. Q アプライ・プログラムが、Q サブスクリプションの状態を A (アクティブ) に、IBMQREP_TARGETS 表の STATE_INFO 列を ASN7606I に変更し、受信キューからのトランザクションのアプライを開始します。「Q サブスクリプションの管理」ウィンドウに、状態が「アクティブ」として表示されます。

関連概念:

- 163 ページの『Q レプリケーションのターゲット表のロードのオプション -- 概要』

関連タスク:

- 144 ページの『双方向レプリケーションの Q サブスクリプションの作成』
- 253 ページの『Q サブスクリプションまたは XML 発行のアクティブ化』
- 153 ページの『2 つのサーバーによるピアツーピア・レプリケーションの Q サブスクリプションの作成』
- 155 ページの『3 つ以上のサーバーによるピアツーピア・レプリケーションの Q サブスクリプションの作成』
- 103 ページの『単一方向レプリケーションの複数の Q サブスクリプションの作成』
- 100 ページの『単一方向レプリケーションの単一の Q サブスクリプションの作成』

関連資料:

- 437 ページの『IBMQREP_SIGNAL 表』

Q レプリケーションのロードなしオプション

ロードなしオプションは、Q サブスクリプションをアクティブにしてレプリケーションを開始する前に、ソース表およびターゲット表を同期する場合に適しています。

Q サブスクリプションに対してロードなしオプションを指定している場合、Q アプリ・プログラムは、Q サブスクリプションがアクティブになるとすぐにターゲット表へのトランザクションのアプリを開始します。

ロードなしオプションを選択する場合は、ソース表の主キーまたはユニーク索引の値が、ターゲット表の主キーまたはユニーク索引にも存在することを確認してください。

ソースおよびターゲットのデータベースまたはサブシステムのアクティビティが比較的少ない期間に多数の新規表を追加する場合は、ロードなしオプションを指定できます。ソース表を静止してからターゲット表をロードし、その後で Q サブスクリプションをアクティブにします。

ソース・データベースをバックアップしてから、そのデータベースをターゲット・サーバー上にリストアする場合も、ロードなしオプションを指定できます。

関連概念:

- 163 ページの『Q レプリケーションのターゲット表のロードのオプション -- 概要』

さまざまなタイプの Q レプリケーションのロード・オプション

ターゲット表のロードのオプションは、セットアップしている Q レプリケーションのタイプによって異なります。以下のセクションでは、Q レプリケーションの各タイプごとに使用可能なロード・オプション、初期ロードに使用されるサーバー、およびロードの開始に必要なステップを説明します。

- 『単一方向のレプリケーションのロード・オプション』
- 174 ページの『2 つのサーバーでの双方向およびピアツーピアのレプリケーションのロード・オプション』
- 175 ページの『3 つ以上のサーバーでのピアツーピア・レプリケーションのロード・オプション』

単一方向のレプリケーションのロード・オプション

このセクションでは、単一方向のレプリケーションのロード・オプションを説明します。

ロード・オプション

すべてのロード・オプションが使用可能です。

初期ロードに使用されるサーバー

Q キャプチャー・サーバー

実行すべきこと

デフォルトでは、Q サブスクリプションのソース表に対して Q キャプチャー・プログラムを開始すると、Q アプライ・プログラムはロード・プロセスを開始します。Q キャプチャー・プログラムの開始時に自動的にアクティブにされないように Q サブスクリプションを作成する場合は、ロードが開始するように Q サブスクリプションをアクティブにすることが必要です。

例

サーバー A の MANAGERS 表からサーバー B の MANAGERS 表に、単一方向でデータのレプリカを生成し、最も自動化された方法を使用すると想定します。Q アプライ・プログラムがロードを処理し、使用可能な最適のユーティリティーを使用するようにします。

1. 使用可能な最適のユーティリティーを使用する自動ロードを指定する、MANAGERS 表に対して Q サブスクリプションを作成します。
2. サーバー A で Q キャプチャー・プログラムを、サーバー B で Q アプライ・プログラムを開始します。

Q アプライ・プログラムが、サーバー A の MANAGERS 表からサーバー B の MANAGERS 表にデータをコピーするロード・ユーティリティーを呼び出します。ロード・プロセスが終了すると、サーバー A からサーバー B に単一方向でレプリケーションが開始します。

2 つのサーバーでの双方向およびピアツーピアのレプリケーションのロード・オプション

このセクションでは、2 つのサーバーでのピアツーピアおよび双方向のレプリケーションのロード・オプションを説明します。

ロード・オプション

すべてのロード・オプションが使用可能です。ただし、自動ロードを指定すると、ご使用のオペレーティング・システムおよびバージョンに応じて、デフォルトで Q アプライ・プログラムが EXPORT と LOAD ユーティリティの組み合わせ、および EXPORT と IMPORT ユーティリティの組み合わせのどちらかを選択します。この振る舞いをオーバーライドし、個々の Q サブスクリプションに対して「Q サブスクリプション・プロパティー (Q Subscription Properties)」ノートブックをオープンすることによって、Q アプライ・プログラムが CURSOR からの LOAD ユーティリティを使用するように指示できます。

初期ロードに使用されるサーバー

2 つのサーバーでの双方向またはピアツーピア・レプリケーションに対して 2 つの Q サブスクリプションを作成する場合、初期ロード・ソースになるサーバーを選択します。このサーバーには、そのデータを他のサーバーの表にコピーする表が入っています。

初期ロード以降のロードの場合 (たとえば、Q サブスクリプション・グループを非アクティブにしてからアクティブにする場合)、2 つの Q サブスクリプションのうちどちらをアクティブにするか決定する際に、ロード・ソースになるサーバーを指定します。アクティブにする Q サブスクリプションのソース表がロード・ソースになります。

実行すべきこと

ロードの開始のプロセスは、自動ロードか手動ロードのどちらを指定するかによって異なります。

自動ロード

Q キャプチャー・プログラムの開始時に自動的にアクティブにされるように Q サブスクリプションを作成した場合、ロード・プロセスを開始するために必要なのは、両方のサーバーで Q キャプチャーおよび Q アプライ・プログラムを開始することだけです。

Q サブスクリプションが自動的にアクティブにされないようにした場合、以下のアクションを実行する必要があります。

- Q キャプチャーおよび Q アプライ・プログラムを両方のサーバーで開始します。
- ソース表をロード・ソースとして指定した Q サブスクリプションをアクティブにします。

手動ロード

1. Q キャプチャーおよび Q アプライ・プログラムを両方のサーバーで開始します。
2. ソース表をロード・ソースとする Q サブスクリプションをアクティブにします。

Q サブスクリプションがロード・ペンディング状態になります。

3. 任意の方法で、Q サブスクリプションのターゲット表をロードします。
4. ロードが終了したら、レプリケーション・センターにロードが終了したことを通知するか、または Q サブスクリプションのソース・サーバーで IBMQREP_SIGNAL 表に LOADDONE シグナルを挿入します。

例

サーバー A およびサーバー B 上で、双方向セットアップで EMPLOYEES 表のレプリカを生成し、最も自動化された方法を使用すると想定します。サーバー A を初期ロード・ソースとします。

1. 2 つの Q サブスクリプション EMP_A2B および EMP_B2A を作成します。EMP_A2B の作成時、初期ロード・ソースとしてサーバー A を指定し、さらに Q アプライ・プログラムの使用可能な最適のロード・ユーティリティを選択する、自動ロードを指定する。
2. Q キャプチャーおよび Q アプライ・プログラムをサーバー A およびサーバー B で開始することによって、ロードを開始します。

サーバー B の Q アプライ・プログラムが、サーバー A の EMPLOYEES 表からサーバー B の EMPLOYEES 表にデータをコピーするロード・ユーティリティを呼び出すことによって EMP_A2B のロードを開始します。ロードが完了すると、サーバー A とサーバー B との間で双方向でレプリケーションが開始します。

3 つ以上のサーバーでのピアツーピア・レプリケーションのロード・オプション

このセクションでは、3 つ以上のサーバーでのピアツーピア・レプリケーションのロード・オプションを説明します。

ロード・オプション

すべてのロード・オプションが使用可能です。ただし、自動ロードを指定すると、ご使用のオペレーティング・システムおよびバージョンによって、デフォルトで Q アプライ・プログラムが EXPORT と LOAD ユーティリティの組み合わせ、および EXPORT と IMPORT ユーティリティの組み合わせのどちらかを選択します。この振る舞いをオーバーライドし、個々の Q サブスクリプションに対して「Q サブスクリプション・プロパティ (Q Subscription Properties)」ノートブックをオープンすることによって、Q アプライ・プログラムが CURSOR からの LOAD ユーティリティを使用するように指示できます。

初期ロードに使用されるサーバー

3 つ以上のサーバーを持つピアツーピア・グループでは、段階に分けてレプリケーションを開始します。まず、2 つのサーバー間でレプリケーションを

開始してから、アクティブなサーバーと新規サーバー間でレプリケーションを開始することによって、追加のサーバーをグループに入れます。以下のガイドラインに従います。

- 最初の 2 つのサーバー間でレプリケーションを開始する際、1 つのサーバーをロード・ソースとして選択します。このサーバーをソースとして指定する Q サブスクリプションをアクティブにします。2 番目のサーバーの Q アプライ・プログラムが、2 番目のサーバーの表に対してロード・プロセスを開始します。
- 新規のサーバーを追加するために、ロード・ソースとしてアクティブなサーバーの 1 つを選択します。このサーバーをソースとして、また新規のサーバーをターゲットとして指定する Q サブスクリプションをアクティブにします。新規のサーバーの Q アプライ・プログラムが、新規のサーバーの表に対してロード・プロセスを開始します。

実行すべきこと

3 つ以上のサーバーを持つピアツーピア構成では、自動的にアクティブにされる Q サブスクリプションを作成することはできません。手動で、いくつかの段階に分けて Q サブスクリプションをアクティブにすることが必要です。以下のステップに従います。

1. グループ中の最初の 2 つのサーバーで、Q キャプチャーおよび Q アプライ・プログラムを開始します。
2. サーバー間で、2 つの Q サブスクリプションのうち 1 つをアクティブにします。アクティブにする Q サブスクリプションのソース表がロード・ソースになり、ターゲット表がロードされます。
3. Q キャプチャーおよび Q アプライ・プログラムを新規のサーバーで開始します。
4. アクティブなサーバーをソースとして、また新規のサーバーをターゲットとして指定する Q サブスクリプションをアクティブにします。アクティブにする Q サブスクリプションのソース表がロード・ソースになり、新規サーバーの表がロードされます。
5. グループ中のすべてのサーバーがロードされるまでステップ 3 および 4 を繰り返します。

手動ロード: 手動ロードの選択する場合、各 Q サブスクリプションをアクティブにした後にターゲット表をロードしてから、そのターゲット表がロードされた時点でレプリケーション・プログラムに通知します。

例

単一の論理表の DEPARTMENTS 表を持つサーバー A、サーバー B、およびサーバー C を含む、ピアツーピアの Q サブスクリプション・グループに対してロード・プロセスを開始すると想定します。Q アプライ・プログラムがロードを処理し、その際使用可能な最適のロード・ユーティリティを使用するようにします。サーバー A を、サーバー B およびサーバー C の両方で、表のロード・ソースとして使用します。

1. 6 つの Q サブスクリプション DEP_A2B、DEP_B2A、DEP_A2C、DEP_C2A、DEP_B2C、および DEP_C2B を、使用可能な最適のユーティリティを使い、すべて自動ロードを指定して作成します。

2. Q キャプチャーおよび Q アプライ・プログラムをサーバー A およびサーバー B で開始します。
3. Q サブスクリプション DEP_A2B をアクティブにします。

サーバー B の Q アプライ・プログラムが、サーバー A の DEPARTMENTS 表からのデータを使って、サーバー B の DEPARTMENTS 表にデータをロードするユーティリティを呼び出します。ロードが完了すると、サーバー A とサーバー B との間で双方向でレプリケーションが開始します。

4. サーバー C でロードを開始するには、まずサーバー C で Q キャプチャーおよび Q アプライ・プログラムを開始します。
5. 次に、Q サブスクリプション DEP_A2C をアクティブにします。

サーバー C の Q アプライ・プログラムが、サーバー A の DEPARTMENTS 表からのデータを使って、サーバー C の DEPARTMENTS 表にデータをロードするユーティリティを呼び出します。ロードが完了すると、3 つすべてのサーバー間で全方向でレプリケーションが開始します。

関連概念:

- 163 ページの『Q レプリケーションのターゲット表のロードのオプション -- 概要』

関連タスク:

- 144 ページの『双方向レプリケーションの Q サブスクリプションの作成』
- 153 ページの『2 つのサーバーによるピアツーピア・レプリケーションの Q サブスクリプションの作成』
- 155 ページの『3 つ以上のサーバーによるピアツーピア・レプリケーションの Q サブスクリプションの作成』
- 103 ページの『単一方向レプリケーションの複数の Q サブスクリプションの作成』
- 100 ページの『単一方向レプリケーションの単一の Q サブスクリプションの作成』

第 12 章 ソースからの発行 (イベント発行) のセットアップ

ソースからの発行 (イベント発行) の設定 -- 概要

イベント発行を使用して、変更された行またはトランザクションをソース表からユーザー・アプリケーションに発行できます。Q キャプチャー・プログラムは、ソース表から変更内容を発行し、それらの変更内容を XML 形式で送信キューに挿入します。その後、ユーザーが選択したアプリケーションに、それらの XML メッセージを検索させます。

ヒント: Q キャプチャー・プログラムが発行する XML メッセージを検索する WebSphere® アプリケーションの例については、サンプル・プログラムを参照してください。サンプルは、XML 発行をビジネス・シナリオで使用する方法を示しています。

以下のトピックでは、イベント発行とその設定方法について説明します。

- 『発行キュー・マップおよび XML 発行のグループ化』
- 180 ページの『発行キュー・マップの作成』
- 182 ページの『XML 発行の作成 -- 概要』

関連資料:

- 529 ページの『Q レプリケーションおよびイベント発行をセットアップするためのサンプル (Linux、UNIX、Windows)』

発行キュー・マップおよび XML 発行のグループ化

XML 発行および発行キュー・マップを定義する前に、それらをどのようにグループ化するかを計画する必要があります。各 XML 発行は、XML 形式での変更の発行元となる単一ソース表を識別します。XML 発行の定義時に、そのソース表用のデータのトランスポートに使用する発行キュー・マップも定義します。特に、各発行キュー・マップは、Q キャプチャー・プログラムが変更を送信する WebSphere® MQ キューを識別します。単一発行キュー・マップを使用して、複数の XML 発行用のデータをトランスポートできるので、どの XML 発行が同じ発行キュー・マップを使用してデータをトランスポートするかを決定する必要があります。

XML 発行および発行キュー・マップをグループ化する方法を計画する場合は、以下の規則に留意してください。

- 複数の Q キャプチャー・プログラムは、WebSphere MQ キューを共用できない。
- 単一 Q キャプチャー・プログラムは、複数の送信キューに書き込める。
- 単一 Q キャプチャー・プログラムから、1 つ以上の発行キュー・マップを作成できる。

Q キャプチャー・プログラムによる送信キューの処理方法

発行キュー・マップの場合、Q キャプチャー・プログラムは、アクティブ XML 発行があるすべての表のデータベース・ログから変更をキャプチャーします。これらの変更は、対応するコミットまたは異常終了レコードがデータベース・ログから読み取られるまでメモリーに保管されます。次いで、コミットされたトランザクションについての情報が、XML 発行用に定義されたすべての WebSphere MQ 送信キューに送信されます。

発行キュー・マップを使用した類似の XML 発行のグループ化の提案

一般に、1 つ以上のアプリケーションとのトランザクションに関係する表については、それらすべてが共通の発行キュー・マップを共有するように、これらの表用の XML 発行を作成する必要があります。類似の XML 発行を同じ発行キュー・マップを使用してグループ化することにより、送信キューに送信されるデータのトランザクション整合性が保証されます。

従属関係にある XML 発行に同じ発行キュー・マップを共有させることは重要です。独立した発行キュー・マップを使用してデータを送信するように、関連トランザクションに関係する XML 発行を定義すると、Q キャプチャー・プログラムは、データを複数の送信キューの間で分割します。

複数のアプリケーションがソース・サーバーを更新しても、同じ表は更新せず、データをソース・サーバーからターゲット・サーバーに発行するよう単一 Q キャプチャー・プログラムを構成する場合は、この Q キャプチャー・プログラムが使用する複数の発行キュー・マップを定義することを考慮できます。このようにすると、各アプリケーションのトランザクションに関連したすべての XML 発行は、これらの発行キュー・マップのいずれかを使用して発行されます。このような構成には、障害分離やスループットの向上などの利点があります。スループットの向上や障害分離は、発行キュー・マップを使用して各 Q キャプチャー・プログラムを構成することによっても実現できます。ただし、これらの利点と、CPU 使用量や複雑な発行環境とのバランスを取る必要があります。

関連概念:

- 182 ページの『XML 発行の作成 -- 概要』
- 13 ページの『発行キュー・マップ』
- 12 ページの『XML 発行』

関連タスク:

- 180 ページの『発行キュー・マップの作成』

発行キュー・マップの作成

XML 発行の作成時に、各 XML 発行を発行キュー・マップに関連付けることによって、データの送信先の WebSphere MQ キューを指定します。発行キュー・マップは、XML 発行の作成を開始する前に、または XML 発行の作成時のステップの一部として作成できます。

各発行キュー・マップは、以下のオプションを示します。

送信キュー

Q キャプチャー・プログラムがソース・データおよび通知メッセージを送信する WebSphere MQ キュー。

メッセージ内容のタイプ

Q キャプチャー・プログラムが以下のいずれかのタイプの内容を含むメッセージを送信することを指定できます。

- 個々の行操作。(Q キャプチャー・プログラムからのこのタイプの XML メッセージを、行操作メッセージと呼びます。)
- 全トランザクション。(Q キャプチャー・プログラムからのこのタイプの XML メッセージを、トランザクション・メッセージと呼びます。)

どちらのタイプのメッセージ内容の場合も、操作が属するトランザクションがコミットされるまで操作は送信されません。選択するメッセージ内容のタイプによって、Q キャプチャー・プログラムがこの発行キュー・マップを使用するすべての XML 発行にデータを送信する方法が決まります。

LOB データ・タイプの場合: 選択するオプションに関係なく、LOB データ・タイプは、トランザクション・メッセージまたは行操作メッセージに関連した個別の物理メッセージとして別個に送信されます。

最大メッセージ長

Q キャプチャー・プログラムがこの送信キューに書き込めるメッセージの最大サイズ (KB 単位)。この最大メッセージ長は、キューまたはキュー・マネージャーに定義された WebSphere MQ 最大メッセージ・サイズ属性 (MAXMSGL) 以下でなければなりません。

キュー・エラー・アクション (Queue error action)

送信キューでエラーが発生する場合の結果アクション。Q キャプチャー・プログラムがとる以下のアクションを決定する必要があります。

- キューを使用する XML 発行の非アクティブ化
- 実行の停止

キュー・エラー・アクションは、このキューに固有の WebSphere MQ エラーが発生する場合に使用されます。選択するオプションに関係なく、Q キャプチャー・プログラムはすべてのエラー・メッセージを診断ログおよび IBMQREP_CAPTRACE 表に記録します。

ハートビート・インターバル

発行する変更がない場合に、Q キャプチャー・プログラムがこのキューでメッセージを送信し、Q キャプチャー・プログラムがまだ稼働していることをユーザー・アプリケーションに通知する頻度 (秒単位)。ハートビート・メッセージは、ハートビート・インターバルの満了後の最初のコミット・インターバルで送信されます。値 0 を指定すると、Q キャプチャー・プログラムはハートビート・メッセージを送信しません。

注: このハートビート・インターバルは、WebSphere MQ チャンネルに定義できる WebSphere MQ パラメーター HBINT (ハートビート・インターバル) とは異なります。

前提条件:

- 発行キュー・マップおよび XML 発行をグループ化する方法を計画する。詳細については、93 ページの『レプリケーション・キュー・マップと Q サブスクリプションのグループ化』を参照してください。
- XML 発行のソース表を含むサーバー上で、Q キャプチャー・プログラムのコントロール表を作成する。詳細については、87 ページの『Q キャプチャー・プログラムと Q アプライ・プログラムのコントロール表の作成』を参照してください。
- WebSphere MQ で適切なオブジェクトが定義されているかを確認する。詳細については、62 ページの『WebSphere MQ オブジェクトの設定要件』を参照してください。

制約事項:

Q レプリケーションとイベント発行の両方に対して同じ送信キューを使用することはできません。なぜなら、圧縮メッセージ (Q レプリケーション用) を転送するか XML メッセージ (イベント発行用) を転送するかを指定するキュー・レベルの属性が存在するからです。

手順:

発行キュー・マップを定義するには、レプリケーション・センターの「発行キュー・マップの作成」ウィンドウを使用します。このウィンドウをオープンするには、キュー・マップを識別するために使用したい Q キャプチャー・プログラムを識別する Q キャプチャー・スキーマを展開します。「発行キュー・マップ」フォルダーを右マウス・ボタンでクリックして、「作成」を選択します。詳細については、オンライン・ヘルプを参照してください。

関連概念:

- 13 ページの『発行キュー・マップ』

関連タスク:

- 223 ページの『発行キュー・マップの属性の変更』
- 226 ページの『発行キュー・マップの削除』
- 95 ページの『レプリケーション・キュー・マップの作成』

関連資料:

- 497 ページの『トランザクション・メッセージ』
- 506 ページの『行操作メッセージ』

XML 発行の作成

XML 発行の作成 -- 概要

イベント発行を使用して、XML 発行を作成することにより、変更された行またはトランザクションをソース表からユーザー・アプリケーションに発行できます。各 XML 発行は、以下を識別する単一オブジェクトです。

- 変更の発行元となるソース表
- 発行するソース表の列および行
- 変更の発行先となる WebSphere® MQ キューを指定する発行キュー・マップ

イベント発行では、1 つ以上の XML 発行を同時に作成できます。

重要: XML 発行は、Q サブスクリプションとは別のオブジェクトです。XML 発行は、データを Q アプライ・プログラムではなく、ユーザーが選択したアプリケーションに発行します。XML 発行はデータの発行のためのものであり、Q サブスクリプションはデータを複製するためのものです。ソース表から変更を複製し、Q アプライ・プログラムがそれらのソースの変更をターゲット表に適用するか、データ操作を目的としてストアード・プロシージャに渡すようにするには、XML 発行ではなく Q サブスクリプションを作成します。

以下のトピックでは、1 つ以上の XML 発行を作成する方法について説明しています。

- 『単一 XML 発行の作成』
- 185 ページの『複数の XML 発行の作成』

XML 発行をさらにカスタマイズしたい場合は、オプションで他のプロパティを設定できます。以下のトピックは、ビジネス・ニーズに基づいて選択するオプションを決定するのに役立ちます。

- 188 ページの『XML 発行のソース列』
- 188 ページの『Q キャプチャー・プログラムが XML 発行用のメッセージを発行する時期』
- 189 ページの『XML 発行の行をフィルター処理する検索条件』
- 192 ページの『XML 発行のキー列』
- 193 ページの『XML 発行用のメッセージに未変更の列を組み込むためのオプション』
- 194 ページの『XML 発行用のメッセージに変更前の値を組み込むためのオプション』

関連概念:

- 179 ページの『ソースからの発行 (イベント発行) の設定 -- 概要』
- 98 ページの『単一方向レプリケーションの Q サブスクリプションの作成 -- 概要』
- 12 ページの『XML 発行』

単一 XML 発行の作成

単一 XML 発行を作成することにより、データが XML 形式でソース表から送信キューに発行される方法を定義し、任意のユーザー・アプリケーションが XML メッセージを検索および使用できるようにします。

184 ページの図 15 は、単一 XML 発行がソース表を WebSphere MQ 送信キューに接続する方法を示しています。

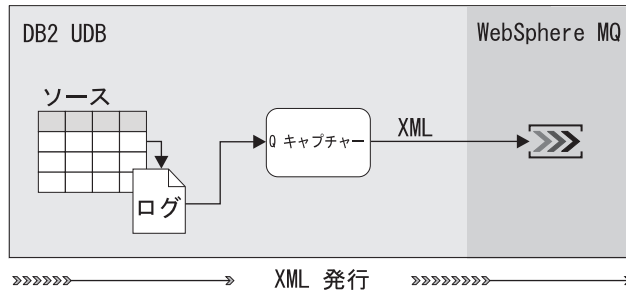


図 15. 単一 XML 発行：ソース表の変更が、XML 形式で WebSphere MQ 送信キューに発行されます。

前提条件:

XML 発行を作成する前に、以下の処置を実行する必要があります。

- 発行キュー・マップおよび XML 発行をグループ化する方法を計画する。詳細については、179 ページの『発行キュー・マップおよび XML 発行のグループ化』を参照してください。
- XML 発行のソース表を含むサーバー上で、Q キャプチャー・プログラムのコントロール表を作成する。詳細については、87 ページの『Q キャプチャー・プログラムと Q アプライ・プログラムのコントロール表の作成』を参照してください。
- 発行キュー・マップを作成する。(このタスクは、XML 発行の作成前、または XML 発行の作成中に行うことができます。) 詳細については、180 ページの『発行キュー・マップの作成』を参照してください。

制約事項:

ビューを XML 発行のソースとすることはできません。

手順:

1 つのソース表を使用して XML 発行を作成するには、次のようにします。

1. レプリケーション・センターから「XML 発行の作成」ウィザードをオープンする。

このウィザードをオープンするには、XML 発行の変更をキャプチャーしたい Q キャプチャー・プログラムを識別する Q キャプチャー・スキーマを展開します。「XML 発行」フォルダーを右マウス・ボタンでクリックして、「作成」を選択します。詳細については、オンライン・ヘルプを参照してください。

2. 「サーバー」および「キュー・マップ」ページで、次のようにする。
 - a. ソース・サーバーについての情報を指定する。
 - b. 使用したい発行キュー・マップを指定する。適切な発行キュー・マップが存在しない場合は、新規発行キュー・マップを作成します。詳細については、180 ページの『発行キュー・マップの作成』を参照してください。

重要: 選択する発行キュー・マップは、Q キャプチャー・プログラムがこの XML 発行用に送信するメッセージ内容のタイプ (コミットされた行操作またはコミットされたトランザクション) を使用して定義されていることを確認してください。

3. 「ソース表」 ページで、変更の発行元にしたいソース表を選択する。
4. 「列と行」 ページで、発行するデータを次のように指定する。
 - a. 発行したい列を次のように指定する。
 - 1) 「列の選択」 ウィンドウを使用して、発行したい列のサブセットを選択する。詳細については、188 ページの『XML 発行のソース列』を参照してください。
 - 2) ソース表の中の列のサブセットを発行する場合は、Q キャプチャー・プログラムがメッセージを発行する時期を指定する。詳細については、188 ページの『Q キャプチャー・プログラムが XML 発行用のメッセージを発行する時期』を参照してください。
 - b. XML 発行のキー列を指定する。詳細については、192 ページの『XML 発行のキー列』を参照してください。
 - c. 発行したい行を指定する。詳細については、189 ページの『XML 発行の行をフィルター処理する検索条件』を参照してください。
5. 「メッセージ内容 (Message Contents)」 ページで、Q キャプチャー・プログラムがメッセージを構築する方法を次のように指定する。
 - a. メッセージの送信時に組み込む (選択された) 列を指定する。このオプションは、更新操作にのみ適用されます。詳細については、193 ページの『XML 発行用のメッセージに未変更の列を組み込むためのオプション』を参照してください。
 - b. 列の更新時に Q キャプチャー・プログラムが非キー列の変更前の値および変更後の値を発行するかどうかを指定する。詳細については、194 ページの『XML 発行用のメッセージに変更前の値を組み込むためのオプション』を参照してください。
6. 「XML 発行の検討」 ページで、XML 発行が有効であることを次のように確認する。
 - XML 発行について変更したいことがあれば、その XML 発行のプロパティーを変更する。
 - XML 発行に欠落情報があれば、プロパティーを変更して完成させる。
7. 「サマリー」 ページで、「完了」をクリックする。

関連概念:

- 182 ページの『XML 発行の作成 -- 概要』
- 12 ページの『XML 発行』

関連タスク:

- 219 ページの『XML 発行の属性の変更』
- 225 ページの『XML 発行の削除』
- 185 ページの『複数の XML 発行の作成』

複数の XML 発行の作成

時間を節約するために、複数の XML 発行を同時に定義できます。作成したいすべての XML 発行の属性を定義した後で、レプリケーション・センターが XML 発行を作成する前に、個々の XML 発行のプロパティーを変更できます。

ヒント: 定義する XML 発行の数が多く、そのほとんどが同じ属性を共有する場合は、複数の XML 発行を一度に作成します。

図 16 は、複数の XML 発行が、どのように同じ発行キュー・マップおよび Q キャプチャー・プログラムを使用できるかを示しています。

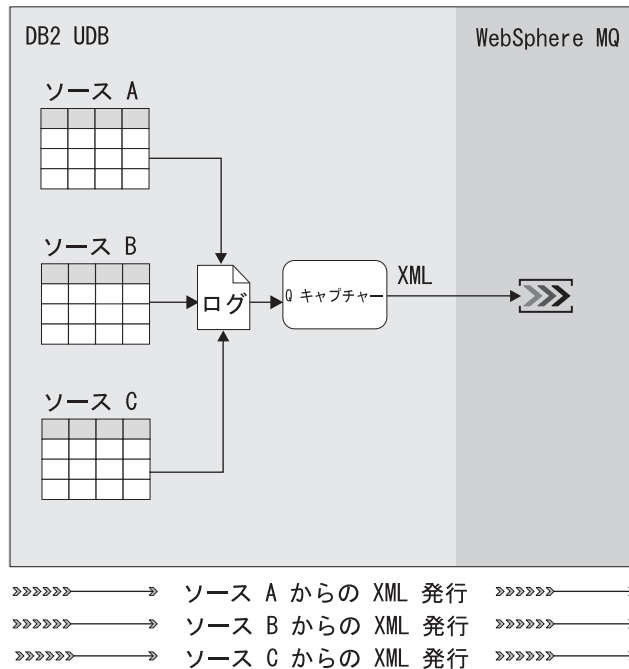


図 16. 同じ発行キュー・マップおよび Q キャプチャー・プログラムを使用する複数の XML 発行：各ソース表の変更が WebSphere MQ 送信キューに発行され、ユーザー・アプリケーションがそれらのメッセージを検索および使用できるようになります。

前提条件:

XML 発行を作成する前に、以下のアクションを実行する必要があります。

- 発行キュー・マップおよび XML 発行をグループ化する方法を計画する。詳細については、179 ページの『発行キュー・マップおよび XML 発行のグループ化』を参照してください。
- XML 発行のソース表を含むサーバー上で、Q キャプチャー・プログラムのコントロール表を作成する。詳細については、87 ページの『Q キャプチャー・プログラムと Q アプライ・プログラムのコントロール表の作成』を参照してください。
- 発行キュー・マップを作成する。(このタスクは、XML 発行の作成前、または XML 発行の定義中に行うことができます。) 詳細については、180 ページの『発行キュー・マップの作成』を参照してください。

制約事項:

ビューを XML 発行のソースとすることはできません。

手順:

複数のソース表を使用して XML 発行を同時に作成するには、次のようにします。

1. レプリケーション・センターから「XML 発行の作成」ウィザードをオープンする。

このウィザードをオープンするには、XML 発行の変更をキャプチャーしたい Q キャプチャー・プログラムを識別する Q キャプチャー・スキーマを展開します。「XML 発行」フォルダーを右マウス・ボタンでクリックして、「作成」を選択します。詳細については、オンライン・ヘルプを参照してください。

2. 「サーバー」および「キュー・マップ」ページで、次のようにする。
 - a. ソース・サーバーについての情報を指定する。
 - b. 使用したい発行キュー・マップを指定する。適切な発行キュー・マップが存在しない場合は、新規発行キュー・マップを作成します。詳細については、180 ページの『発行キュー・マップの作成』を参照してください。

重要: 選択する発行キュー・マップは、Q キャプチャー・プログラムがこの XML 発行用に送信するメッセージ内容のタイプ (コミットされた行操作またはコミットされたトランザクション) を使用して定義されていることを確認してください。

3. 「ソース表」ページで、変更の発行元にしたいソース表を選択する。

複数の XML 発行を一度に作成する場合、レプリケーション・センターは、各ソース表からすべての列および行が発行されるものと想定します。ウィザードの最後で、レプリケーション・センターが XML 発行を構築する前に、ソース列および行のサブセットのみが発行されるように個々の XML 発行を変更できます。

4. 「メッセージ内容 (Message Contents)」ページで、Q キャプチャー・プログラムがメッセージを構築する方法を次のように指定する。
 - a. メッセージの送信時に組み込む (選択された) 列を指定する。このオプションは、更新操作にのみ適用されます。詳細については、193 ページの『XML 発行用のメッセージに未変更の列を組み込むためのオプション』を参照してください。
 - b. 列の更新時に Q キャプチャー・プログラムが非キー列の変更前の値および変更後の値を発行するかどうかを指定する。詳細については、194 ページの『XML 発行用のメッセージに変更前の値を組み込むためのオプション』を参照してください。
5. 「XML 発行の検討」ページで、各 XML 発行が有効であることを次のように確認する。
 - XML 発行について変更したいことがあれば、その XML 発行のプロパティを変更する。たとえば、特定の XML 発行について、ソース表から行または列のサブセットのみを発行することを選択できます。
 - XML 発行に欠落情報があれば、プロパティを変更して必要な情報を追加する。
6. 「サマリー」ページで、「完了」をクリックする。

関連概念:

- 182 ページの『XML 発行の作成 -- 概要』
- 12 ページの『XML 発行』

関連タスク:

- 219 ページの『XML 発行の属性の変更』
- 225 ページの『XML 発行の削除』
- 183 ページの『単一 XML 発行の作成』

XML 発行のソース列

デフォルトでは、XML 発行を作成すると、ソース表のすべての列に対する変更が発行されます。ただし、ユーザー・アプリケーションに対して、ソース表のすべての列を使用可能にはしたくない場合は、列のサブセットを発行できます。また、XML 発行用のユーザー・アプリケーションが、ソース表に定義されているデータ・タイプのすべてをサポートしていない場合にも、列のサブセットを発行することができます。

列のサブセットを発行するには、ユーザー・アプリケーションに対して発行したいソース列だけを選択します。単一 XML 発行を作成する場合は、レプリケーション・センターの「XML 発行の作成」ウィザードで、ソース表から列のサブセットを発行する方法を選択できます。複数の XML 発行を同時に作成する場合は、「XML 発行の作成」ウィザードの「レビュー」ページで、サブセット列のみを発行したい個々の XML 発行を選択してから、その XML 発行のプロパティを編集します。

LOB 列の場合に重要: XML 発行用に LOB データ・タイプを含む列を選択する場合は、ソース表で少なくとも 1 つの固有データベース制約 (ユニーク索引や主キーなど) が課されていることを確認してください。XML 発行では、このような固有のプロパティを構成する列を選択する必要はありません。

関連概念:

- 182 ページの『XML 発行の作成 -- 概要』
- 105 ページの『Q サブスクリプションのソース列 (単一方向レプリケーション)』

関連タスク:

- 185 ページの『複数の XML 発行の作成』
- 183 ページの『単一 XML 発行の作成』

Q キャプチャー・プログラムが XML 発行用のメッセージを発行する時期

XML 発行の作成時に、Q キャプチャー・プログラムは、ソース表の列が変更されるたびにメッセージを発行するのか、XML 発行を構成する列が変更された場合にのみメッセージを発行するのかを指定できます。以下のセクションでは、Q キャプチャー・プログラムがメッセージを発行する要因となる 2 つの異なるタイプのイベントについて説明します。

- 189 ページの『XML 発行の列が変更される場合にのみメッセージを送信する』
- 189 ページの『ソース表で変更が発生するたびにメッセージを送信する』

ソース表のすべての列を発行する場合は、以下の 2 つのオプションのアクションは同じになります。

推奨: 一般に、選択した列に影響を与える変更だけを発行するのが適切です。ただし、変更が発生するたびに行の一部だけ (キー列など) を必要とするアプリケーション

ンもあります。このパブリッシュされた情報は、イベント通知として機能し、他のアクションを起動させることができます。

XML 発行の列が変更される場合にのみメッセージを送信する

デフォルトでは、Q キャプチャー・プログラムは、XML 発行用に選択した列で変更が発生する場合にのみメッセージを発行します。

例: ソース表には 100 個の列があり、それらの列のうち 25 個を XML 発行で発行するように選択するとします。XML 発行の列が変更される場合にのみメッセージを送信することを指定すると、XML 発行を構成する 25 個の列に変更が加えられるたびに、Q キャプチャー・プログラムは XML メッセージを発行します。XML 発行を構成しない 75 個の列が変更されても、Q キャプチャー・プログラムは XML メッセージを発行しません。

ソース表で変更が発生するたびにメッセージを送信する

ソース表で変更が発生するたびに Q キャプチャー・プログラムがメッセージを送信するように、XML 発行を定義できます。ソース表の列のサブセットのみを発行する場合は、XML 発行を構成していない列で変更が発生した場合でも、Q キャプチャー・プログラムはメッセージを発行します。

例: ソース表には 100 個の列があり、それらの列のうち 25 個を XML 発行で発行するように選択するとします。ソース表で変更が発生するたびにメッセージを送信することを選択すると、ソース表の 100 個の列のいずれかに変更が加えられるたびに、Q キャプチャー・プログラムは XML メッセージを発行します。

関連概念:

- 182 ページの『XML 発行の作成 -- 概要』

関連タスク:

- 185 ページの『複数の XML 発行の作成』
- 183 ページの『単一 XML 発行の作成』

XML 発行の行をフィルター処理する検索条件

デフォルトでは、XML 発行を作成すると、ソース表のすべての行が発行されます。ただし、XML 発行の作成時に、検索条件で WHERE 文節を指定して、発行したい行を識別できます。Q キャプチャー・プログラムがソース表に関連した DB2® リカバリー・ログ内で変更を検出すると、Q キャプチャー・プログラムはその変更を検索条件に照らして評価し、変更を発行するかどうかを判別します。

単一 XML 発行を作成する場合は、レプリケーション・センターの「XML 発行の作成」ウィザードで、ソース表から行のサブセットを発行するための WHERE 文節を追加できます。複数の XML 発行を一度に作成する場合は、「XML 発行の作成」ウィザードの「レビュー」ページで、行のサブセットのみを発行したい個々の XML 発行を選択してから、その XML 発行のプロパティを編集して WHERE 文節を追加します。

WHERE 文節を指定すると、現在のログ・レコードの値を使用して列を評価するかどうかを指定できます。現在のログ・レコードの値を使用して WHERE 文節の列を評価したい場合は、列名の直前にコロンを 1 つ挿入します。

現在のログ・レコードの値を使用して列を評価する WHERE 文節の例

```
WHERE :LOCATION = 'EAST' AND :SALES > 100000
```

この例で、LOCATION および SALES は、現在のログ・レコードの値を使用して評価されるソース表の列名です。ここで、Q キャプチャー・プログラムは、East での \$100,000 を超える売上を含むソース表の変更だけを送信します。列名を入力する場合、名前を二重引用符で囲んでいないと、文字は大文字に変換されます。たとえば、列名に大文字と小文字が混在している場合は、"Location" と入力します。

Q キャプチャー・プログラムが WHERE 文節の一部である列を発行する場合は、ターゲット表またはストアード・プロシージャに送信する必要がある操作のタイプを変更することが必要になる場合があります。

WHERE 文節が原因で Q キャプチャー・プログラムが操作のタイプを変更しなければならない場合の例

```
WHERE :LOCATION = 'EAST'  
AND :SALES > 100000
```

ソース表で以下の変更が発生したとします。

```
INSERT VALUES ('EAST', 50000)  
UPDATE SET SALES = 200000 WHERE LOCATION = 'EAST'
```

変更前の値が WHERE 文節の検索条件に合致しないので、Q キャプチャー・プログラムは、操作を UPDATE ではなく INSERT として送信します。

同様に、変更前の値が検索条件を満たしていて、変更後の値が検索条件を満たしていない場合は、Q キャプチャー・プログラムは UPDATE を DELETE に変更します。たとえば、上記と同じ以下の WHERE 文節があるとします。

```
WHERE :LOCATION = 'EAST'  
AND :SALES > 100000
```

ここで、ソース表で以下の変更が発生したとします。

```
INSERT VALUES ('EAST', 200000)  
UPDATE SET SALES = 50000 WHERE LOCATION = 'EAST'
```

最初の変更である挿入は、WHERE 文節の検索条件と合致する (200000 > 100000 は成立する) ので、ターゲット表またはストアード・プロシージャに送信されます。ただし、2 番目の変更 (更新) は、検索条件を満たしていません (50000 > 100000 が偽)。Q キャプチャー・プログラムは変更を DELETE として送信し、ターゲット表またはストアード・プロシージャから値が削除されるようにします。

複雑な検索条件:

イベント発行では、より複雑な WHERE 文節を指定できます。ただし、検索条件を複雑にするとパフォーマンスが影響を受ける場合があります。たとえば、他の表や、ソース表または他の表のレコードを参照する副選択を含むさらに複雑な WHERE 文節を指定できます。

副選択を含む WHERE 文節の例

```
WHERE :LOCATION = 'EAST'  
AND :SALES > (SELECT SUM(EXPENSE) FROM STORES WHERE STORES.DEPTNO = :DEPTNO)
```

上記の例では、Q キャプチャー・プログラムは、利益を生んだ East の変更 (売上の値が経費の合計より大きい場合) だけを送信します。副選択は STORES 表と、ソース表の LOCATION、SALES、および DEPTNO という列を参照します。

副選択を含む WHERE 文節を使用して XML 発行を定義すると、以下の問題が生じる場合があります。

- ソース表で変更が発生するたびに、Q キャプチャー・プログラムが STORES 表に対する大規模な選択を実行して SUM(EXPENSE) 値を計算するため、パフォーマンスが低下する場合があります。また、このタイプの選択では、表に対するロックが競合する場合があります。
- 副選択で予期しない結果となる場合があります。たとえば、副選択は現在のデータベース値と照らして評価されるため、データベースの中で EXPENSE 値が変わっても、WHERE 文節の列は古い状態であり、異なった結果となります。副選択の照会結果が変わらない場合は、検索条件は正しい結果を示します。副選択で参照される表名が変わらない場合は、検索条件は正しい結果を示します。

検索条件の制約事項:

- 列関数を副選択ステートメント内で使用しない場合は、検索条件に列関数を含めることはできない。

列関数を含む無効な WHERE 文節

```
#-----  
# Incorrect: Don't do this  
#-----
```

```
WHERE :LOCATION = 'EAST' AND SUM(:SALES) > 1000000
```

レプリケーション・センターが検索条件を検証するのは、レプリケーション・センターが XML 発行を作成するときではなく、Q キャプチャー・プログラムが検索条件を評価するときです。XML 発行に無効な検索条件が含まれている場合は、その XML 発行は無効な条件が評価されるときに失敗し、XML 発行は非活動されます。

- ORDER BY または GROUP BY 文節を副選択ステートメント内で使用しない場合は、これらの文節を検索条件に含めることはできない。

GROUP BY を含む無効な WHERE 文節

```
#-----  
# Incorrect: Don't do this  
#-----
```

```
WHERE :COL1 > 3 GROUP BY COL1, COL2
```

GROUP BY を含む有効な WHERE 文節

```
WHERE :COL2 = (SELECT COL2 FROM T2 WHERE COL1=1 GROUP BY COL1, COL2)
```

- 検索条件は変更の発行元となるソース表の実際の名前を参照できない。WHERE 文節では、ソース表の実際の名前に schema.tablename という表記を使用しないでください。ただし、副選択では、schema.tablename という表記を使用して別の表名を参照できます。

スキーマ名・表名で修飾した列名を含む無効な WHERE 文節:

```
#-----  
# Incorrect: Don't do this  
#-----
```

```
WHERE :ADMINISTRATOR.SALES > 100000
```

ソース表および列の実際の名前を含む WHERE 文節のこの例では、発行される表は ADMINISTRATOR で、SALES は列名です。この無効な WHERE 文節は、ADMINISTRATOR 表の SALES 列の値のうち、SALES が 100000 を超えるもののみを選択することを目的としています。

列名を含む有効な WHERE 文節

```
WHERE :SALES > 100000
```

列名を含む WHERE 文節のこの例では、SALES が列名です。

- 検索条件は、変更が発生する前の列の値を参照できない。検索条件が参照できるのは、変更後の値だけです。
- 検索条件に、全選択を使用する IN 述部を含めることはできない。

IN 述部を含む無効な WHERE 文節

```
#-----  
# Incorrect: Don't do this  
#-----
```

```
WHERE :COL1 IN (SELECT .... )
```

IN 述部を含む有効な WHERE 文節

```
WHERE :COL1 IN ('CA', 'IL')
```

- 検索条件に EXISTS 述部を含めることはできない。
- 検索条件に比較述部 (SOME、ANY、または ALL を使用する述部) を含めることはできない。
- 検索条件は LOB 値を参照できない。

関連概念:

- 182 ページの『XML 発行の作成 -- 概要』
- 108 ページの『行のフィルタリングのための検索条件 (単一方向レプリケーション)』

関連タスク:

- 185 ページの『複数の XML 発行の作成』
- 183 ページの『単一 XML 発行の作成』

XML 発行のキー列

XML 発行ごとに、ソース表のどの列がキー列であるかを指定する必要があります。イベント発行では、キー列で各行が固有になるようにする必要があります。レプリケーション・センターに、固有であることを識別するために使用するソース表の列を推奨させることができます。また、自分でキー列を選択することもできます。

単一 XML 発行を作成する場合は、レプリケーション・センターの「XML 発行の作成」ウィザードは、ソース表からキー列を選択できる「キー列の選択」ウィンド

ウを起動します。複数の XML 発行を同時に作成する場合は、「XML 発行の作成」ウィザードの「レビュー」ページを使用して、使用するキー列をカスタマイズできます。

関連概念:

- 182 ページの『XML 発行の作成 -- 概要』

関連タスク:

- 185 ページの『複数の XML 発行の作成』
- 183 ページの『単一 XML 発行の作成』

XML 発行用のメッセージに未変更の列を組み込むためのオプション

ソース列のサブセットを発行する XML 発行の作成時に、Q キャプチャー・プログラムが発行する XML メッセージに組み込む各行の列値を指定できます。以下のセクションでは、Q キャプチャー・プログラムが XML メッセージに組み込むことのできる値について説明します。

- 『変更された列のみの送信』
- 『変更された列と未変更の列の両方の送信』

このオプションは非キー列の値にのみ適用され、Q キャプチャー・プログラムは常にキー列の値を発行します。

変更された列のみの送信

デフォルトでは、XML 発行を作成すると、Q キャプチャー・プログラムは、列値が変更された場合にのみ、XML 発行用に選択した列の値を送信します。

例: ソース表には 100 個の列があり、それらの列のうち 25 個を XML 発行で発行するように選択するとします。変更された列のみの送信することを指定する場合は、25 個の選択した列で変更が発生するたびに、Q キャプチャー・プログラムは変更された列のみの値を発行します。たとえば、25 個の選択した列のうち 17 個で変更が発生する場合は、Q キャプチャー・プログラムは 17 個の変更された値を送信します。

推奨: このオプションを使用して、キュー間で送信される不要なデータの量を最小限にしてください。

変更された列と未変更の列の両方の送信

Q キャプチャー・プログラムが、値が変更されたかどうかに関係なく、XML 発行用に選択した列の値を必ず送信するように、XML 発行を定義することもできます。

例: ソース表には 100 個の列があり、それらの列のうち 25 個を XML 発行で発行するように選択するとします。変更された列と未変更の列の両方を送信することを指定する場合は、25 個の選択した列で変更が発生するたびに、Q キャプチャー・プログラムは選択したすべての列を発行します。たとえば、25 個の選択した列のうち 17 個で変更が発生する場合は、Q キャプチャー・プログラムは 25 個のすべての列の値を送信します。

関連概念:

- 182 ページの『XML 発行の作成 -- 概要』

関連タスク:

- 185 ページの『複数の XML 発行の作成』
- 183 ページの『単一 XML 発行の作成』

XML 発行用のメッセージに変更前の値を組み込むためのオプション

ターゲット・キーを構成しない列で更新が発生すると、Q キャプチャー・プログラムは、変更が発生した後の列の値を送信するか、変更が発生する前の列の値と変更が発生した後の列の値の両方を送信します。キー列で更新が発生すると、変更前の値と変更後の値が必ず送信されます。

削除操作は、必ず特定の列値ではなく行に適用されるため、削除が処理される方法が異なります。削除の場合、変更前の値だけが送信されます。キー列の変更前の値が必ず送信されます。変更前および変更後の列値をメッセージに組み込むことを指定して、非キー列の値が削除されると、非キー列の変更前の値が送信されます。

以下のセクションでは、変更前の値と変更後の値のための 2 つのオプションについて説明します。

- 『新しいデータ値のみの送信』
- 『古いデータ値と新しいデータ値の両方の送信』

新しいデータ値のみの送信

デフォルトでは、ソース表で更新が発生すると、Q キャプチャー・プログラムは、変更が発生した後の非キー列の値を発行します。新しいデータ値 (変更後の値) だけをメッセージに組み込むことを指定する場合は、変更が発生する前の非キー列の値はメッセージに組み込まれません。

推奨: XML 発行用に XML メッセージを受信するアプリケーションが変更前の各非キー列の値を使用することがない場合は、Q キャプチャー・プログラムが変更後の列値だけを送信することを指定してください。

古いデータ値と新しいデータ値の両方の送信

古いデータ値と新しいデータ値の両方 (変更前の値と変更後の値の両方) をメッセージに組み込むことを指定する場合は、非キー列が更新されると、Q キャプチャー・プログラムは、変更が発生する前の列の値と変更が発生した後の列の値を発行します。

推奨: XML 発行用に XML メッセージを受信するアプリケーションが変更前の各列の値を使用する場合は、Q キャプチャー・プログラムが変更前の列値と変更後の列値を送信することを指定してください。

LOB データ・タイプの制限: LOB データ・タイプの列の変更前の値は、XML メッセージで送信されません。変更前の値と変更後の値の両方をメッセージに組み込むことを指定しても、このオプションは LOB データ・タイプの列には適用されず、変更前の値は送信されません。

関連概念:

- 182 ページの『XML 発行の作成 -- 概要』

関連タスク:

- 185 ページの『複数の XML 発行の作成』
- 183 ページの『単一 XML 発行の作成』

第 13 章 データ・タイプの複製および発行に関する考慮事項

Q レプリケーションおよびイベント発行でのデータ・タイプの複製および発行に関する考慮事項 -- 概要

LONG VARCHAR や LOB のデータ・タイプなど、特定のデータ・タイプを複製または発行するときには、一定の条件および制約事項がありますので注意してください。以下のトピックでは、特定のデータ・タイプが Q レプリケーションおよびイベント発行でどのように処理されるかについて説明します。

- 『Q レプリケーションおよびイベント発行での一般的なデータ・タイプに関する考慮事項』
- 198 ページの『Q レプリケーションおよびイベント発行でのラージ・オブジェクト (LOB) データ・タイプに関する考慮事項』

Q レプリケーションおよびイベント発行での一般的なデータ・タイプに関する考慮事項

このトピックでは、Q レプリケーションおよびイベント発行でサポートされていないデータ・タイプ、および特定の状況でのみ使用できるデータ・タイプについて説明しています。データ・タイプ、長さ属性、およびヌル属性は、ソース列とターゲット列で同じでなければならないため、データ・タイプに関するこれらの考慮事項はソース表とターゲット表の両方に適用されます。

現時点では、以下のデータの複製または発行は行えません。

- DATALINK データ・タイプ
- 空間データ・タイプ
- 以下のいずれかのプロシージャが定義されている DB2® UDB for z/OS™ の列
 - EDITPROC
 - FIELDPROC
 - VALIDPROC

以下のタイプのデータの複製および発行は、特定の状況でのみ行えます。

LONG VARCHAR および LONG VARGRAPHIC

long 可変長文字 (LONG VARCHAR) および long 可変長グラフィック (LONG VARGRAPHIC) データ・タイプを含む列は、DB2 UDB for Linux、UNIX®、および Windows® から DB2 UDB for z/OS へ複製することはできません。long 可変長文字を含む DB2 UDB for z/OS 内のフィールドは、DB2 UDB for Linux、UNIX、および Windows 内のフィールドよりも最大長が短くなっています。そのため、これらのタイプのフィールドを、DB2 UDB for Linux、UNIX、および Windows から DB2 UDB for z/OS へ複製した場合、切り捨てが行われる可能性があります。

ユーザー定義のデータ・タイプ

ユーザー定義の特殊データ・タイプは複製または発行できますが、ユーザー定義の構造化データ・タイプおよび参照データ・タイプはそうすることはできません。ユーザー定義の特殊データ・タイプ (DB2 Universal Database™ の特殊データ・タイプ) は、複製される前に基本データ・タイプに変換されます。Q サブスクリプションの作成時にターゲット表が作成される場合、ユーザー定義の特殊データ・タイプは、新しいターゲット表の基本データ・タイプに変換されます。

関連概念:

- 197 ページの『Q レプリケーションおよびイベント発行でのデータ・タイプの複製および発行に関する考慮事項 -- 概要』

Q レプリケーションおよびイベント発行でのラージ・オブジェクト (LOB) データ・タイプに関する考慮事項

DB2® Universal Database では、以下のラージ・オブジェクト (LOB) データ・タイプをサポートしています。

- バイナリー LOB (BLOB)
- 文字 LOB (CLOB)
- 2 バイト文字 LOB (DBCLOB)

LOB または ROWID 列の変更前の値の複製または発行は行われません。Q キャプチャー・プログラムが DB2 UDB ログ内の LOB の変更 (LOB 記述子) の指示を見つけると、Q キャプチャー・プログラムはソース表から現在の LOB 値を送信します。

LOB のごく一部が変更される場合でも、LOB 全体が複製または発行されます。

一部の DB2 エクステンダー™は、Q レプリケーションおよびイベント・パブリッシングでサポートされていません。それはテキスト、音声、ビデオ、イメージ、その他のエクステンダーで、エクステンダーの LOB 列データに関連付けられた追加のコントロール・ファイルがデータベースの外部で保守されているようなエクステンダーです。

関連概念:

- 72 ページの『ラージ・オブジェクト (LOB) 値のキュー項目数に関する考慮事項』
- 197 ページの『Q レプリケーションおよびイベント発行でのデータ・タイプの複製および発行に関する考慮事項 -- 概要』
- 32 ページの『Q レプリケーションおよびイベント発行の LOB データ・タイプのためのメモリー』

関連資料:

- 508 ページの『ラージ・オブジェクト (LOB) メッセージ』

第 14 章 レプリケーション・センターからの SQL スクリプトおよび操作可能コマンドの実行

レプリケーション・センターからの SQL スクリプトおよび操作可能コマンドの実行 -- 概要

レプリケーション・センターは、操作可能コマンドおよび SQL スクリプトを生成し、その後実行することにより、レプリケーション・オブジェクトを作成および操作します。以下のトピックでは、レプリケーション・センターが生成する SQL スクリプトおよび操作可能コマンドを使用することによって実行できるタスクについて説明します。

- 『レプリケーション・センターが生成する SQL スクリプトおよび操作可能コマンド』
- 200 ページの『レプリケーション・センターからの SQL スクリプトの実行および保管』
- 201 ページの『レプリケーション・センターからのコマンドの実行および保管』

レプリケーション・センターが生成する SQL スクリプトおよび操作可能コマンド

レプリケーション・センターは、操作可能コマンドおよび SQL スクリプトを生成し、実行することにより、レプリケーション・オブジェクトを作成および操作します。たとえば、レプリケーション・センターを使用して Q キャプチャー・プログラムを開始する際、レプリケーション・センターは操作可能コマンドを生成して実行します。別の例として、レプリケーション・センターを使用して Q サブスクリプションを削除する際、レプリケーション・センターは SQL スクリプトを生成して実行します。

レプリケーション・センターは、Q キャプチャー・プログラム、Q アプライ・プログラム、およびレプリケーション・アラート・モニターの多数のタスクを実行するための操作可能コマンドを生成することができます。たとえば、プログラムを開始または停止したり、コントロール表を整理したり、パラメーターを変更したり、あるいはプログラム状況をチェックしたりするコマンドがあります。

また、レプリケーション・センターは、オブジェクトを作成、変更、および削除するための SQL スクリプトを生成することもできます。たとえば、Q サブスクリプションまたは XML 発行を作成、変更、削除、またはアクティブにするスクリプトがあります。また、スクリプトは、コントロール表を作成、変更、またはドロップしたり、アラート条件を作成したり削除したりすることもできます。

カスタマイズされたタスクを実行するためにコマンドおよびスクリプトを変更することもできます。たとえば、レプリケーション・センターが生成する SQL スクリプトを編集することによって、複数の Q サブスクリプションを同時に定義することができます。別の例として、レプリケーション・センターが生成する SQL スクリ

プトを変更することにより、異なるサーバー上で同じレプリケーション・アクションを作成したり、各サーバーごとにレプリケーションをカスタマイズすることができます。コマンドの変更は、レプリケーション・センターの「今実行するか、コマンドを保管する」ウィンドウで行います。SQL スクリプトの変更は、レプリケーション・センターの「今実行するか、SQL を保管する」ウィンドウで行います。

コマンドおよび SQL スクリプトの実行は、レプリケーション・センター、タスク・センター (z/OS オペレーティング・システムを除く)、またはコマンド行から行うことができます。コマンドまたはスクリプトは、タスクとして保管してタスク・センターから実行することもできますし、ファイルとして保管してコマンド行から実行することもできます。また、保管せずにレプリケーション・センターから直接実行することもできます。

関連タスク:

- 199 ページの『レプリケーション・センターからの SQL スクリプトおよび操作可能コマンドの実行 -- 概要』
- 200 ページの『レプリケーション・センターからの SQL スクリプトの実行および保管』
- 201 ページの『レプリケーション・センターからのコマンドの実行および保管』

レプリケーション・センターからの SQL スクリプトの実行および保管

SQL スクリプトは、レプリケーション・センターから次の 3 つの方法で実行することができます。これは、レプリケーション・センターから直接実行する方法、タスクとして保管して実行する方法、およびコマンド行から実行する方法の 3 つです。

レプリケーション・センターから SQL スクリプトを直接実行するには、「今実行するか、SQL を保管する」ウィンドウを使用します。

手順:

SQL スクリプトを直接実行するには、次のようにします。

1. オプション: SQL スクリプトを変更する。
2. 「今実行するか、SQL を保管する」ウィンドウで「今実行する」ラジオ・ボタン・オプションを選択することにより、スクリプトを実行する。

SQL スクリプトをタスクとしてタスク・センターから実行することもできます (z/OS オペレーション・システムを除く)。

手順:

SQL スクリプトをタスクとして実行するには、次のようにします。

1. オプション: SQL スクリプトを変更する。
2. 「今実行するか、コマンドを保管する」ウィンドウで「タスクとして保管」ラジオ・ボタン・オプションを選択することにより、スクリプトをタスク・オブジェクトとして保管する。
3. タスク・センターを使用することにより、タスク・オブジェクトを実行する。

SQL スクリプトを保管し、後にコマンド行から実行することができます。

前提条件:

コマンド行から SQL スクリプトを実行する前に、サーバーに接続し、サーバー用のユーザー ID およびパスワードを指定しなければなりません。以下のステートメントを使用します。

```
CONNECT TO MYDB USER XXXX USING XXXX ;
```

手順:

コマンド行から SQL スクリプトを実行するには、次のようにします。

1. オプション: スクリプトを変更する。
2. 「今実行するか、SQL を保管する」ウィンドウを使用してスクリプトをファイルとして保管する。
3. 以下のコマンドのいずれかを使用することにより、コマンド行からスクリプトを実行する。

- SQL スクリプトの終了文字がセミコロン (;) の場合、以下のコマンドを使用する。

```
db2 -tvf filename
```

- SQL スクリプトが区切り文字としてセミコロン以外を使用する場合、以下のコマンドを使用する。

```
db2 -td# -vf filename
```

関連概念:

- 199 ページの『レプリケーション・センターが生成する SQL スクリプトおよび操作可能コマンド』

レプリケーション・センターからのコマンドの実行および保管

操作可能コマンドは、レプリケーション・センターから次の 3 つの方法で実行することができます。レプリケーション・センターから直接実行する方法、タスクとして保管して実行する方法、およびコマンド行からバッチ・ファイルとして実行する方法の 3 つです。

レプリケーション・センターからコマンドを直接実行するには、「今実行するか、コマンドを保管する」ウィンドウを使用します。

手順:

コマンドを直接実行するには、次のようにします。

1. オプション: コマンドを変更する。
2. 「今実行するか、SQL を保管する」ウィンドウで「今実行する」ラジオ・ボタン・オプションを選択することにより、コマンドを実行する。

コマンドをタスクとしてタスク・センターから実行することもできます (z/OS オペレーション・システムを除く)。

手順:

コマンドをタスクとして実行するには、次のようにします。

1. オプション: コマンドを変更する。
2. 「今実行するか、コマンドを保管する」ウィンドウで「タスクとして保管」ラジオ・ボタン・オプションを選択することにより、コマンドをタスク・オブジェクトとして保管する。
3. タスク・センターを使用することにより、タスク・オブジェクトを実行する。

コマンドをバッチ・ファイルとして保管し、後にコマンド行から実行することができます。

手順:

コマンド行からコマンドを実行するには、次のようにします。

1. オプション: コマンドを変更する。
2. 「今実行するか、SQL を保管する」ウィンドウで「ファイルに保管」ラジオ・ボタン・オプションを選択することにより、コマンドをバッチ・ファイルとして保管する。
3. バッチ・ファイルをコマンド行から実行する。レプリケーション・センターが実行されているクライアント・マシンからリモートのシステム上でコマンドを実行するには、次のようにします。
 - a. コマンドが実行されるシステムにファイルを保管する。
 - b. リモート・システムに Telnet でログインするか、ログオンして、コマンドを実行する。
 - c. コマンドをクライアント・マシン上で保管した後にリモート・システム上でそのコマンドを実行するには、コマンド・ファイルを、コマンドが実行されるシステムに FTP 経由で送信する。

関連概念:

- 199 ページの『レプリケーション・センターが生成する SQL スクリプトおよび操作可能コマンド』

第 4 部 Q レプリケーションおよびイベント発行環境の管理およびモニター

本書の第 4 部は、以下の章で構成されています。

- 205 ページの『第 15 章 Q レプリケーション環境の変更』では、Q サブスクリプションと、それを処理してモニターするプログラムを変更または停止して、Q レプリケーション構成を変更する方法を説明しています。
- 219 ページの『第 16 章 イベント発行環境の変更』では、XML 発行と、それを処理してモニターするプログラムを変更または停止して、イベント発行構成を変更する方法を説明しています。
- 229 ページの『第 17 章 Q キャプチャー・プログラムの操作』では、Q レプリケーションまたはイベント発行環境でソースから変更を複製または発行するプログラムの操作方法を説明しています。
- 259 ページの『第 18 章 Q アプライ・プログラムの操作』では、Q レプリケーション環境でターゲットに変更を複製するプログラムの操作方法を説明しています。
- 279 ページの『第 19 章 Q レプリケーションおよびイベント発行プログラムに関するレポートの表示』では、最新の統計を表示する Q キャプチャー・プログラムおよび Q アプライ・プログラムで、レポートをチェックする方法を説明しています。
- 291 ページの『第 20 章 レプリケーション・アラート・モニターによるレプリケーションのモニター』では、Q レプリケーションおよびイベント発行環境をモニターし、さまざまな条件が満たされる場合に自動的に通知するプログラムのセットアップおよび操作方法を説明しています。この章では、SQL レプリケーション環境のモニター方法についても説明しています。
- 315 ページの『第 21 章 Q レプリケーションおよび発行環境の保守』では、ソース表、ターゲット、およびコントロール表を保守する方法、また複製と発行に使用する WebSphere MQ オブジェクトを保守する方法について説明します。
- 329 ページの『第 22 章 ソース表とターゲット表間の相違検出および修復』では、ソースおよびターゲット表を `tdiff` および `trep` ユーティリティを使用して同期する方法を説明しています。
- 335 ページの『第 23 章 システム・サービスを使用してレプリケーション・プログラムを操作する方法』では、さまざまなオペレーティング・システムで使用可能なコマンドおよびサービスを使用した、Q レプリケーションおよびイベント発行のスケジュール方法と開始方法を説明しています。

第 15 章 Q レプリケーション環境の変更

Q レプリケーション環境の変更 -- 概要

以下のトピックでは、Q レプリケーション環境への日常の変更に関する点およびその手順について説明します。

- 『Q サブスクリプションの属性の変更』
- 207 ページの『既存の Q サブスクリプションへの列の追加』
- 209 ページの『Q サブスクリプションでの許容可能な SQL 状態の変更』
- 211 ページの『レプリケーション・キュー・マップの属性の変更』
- 213 ページの『Q サブスクリプションの削除』
- 215 ページの『レプリケーション・キュー・マップの削除』
- 216 ページの『Q キャプチャー・スキーマまたは Q アプライ・スキーマのドロップ』

Q サブスクリプションの属性の変更

単一方向 Q サブスクリプションのいくつかの属性を変更できます。これらの属性の変更は、レプリケーションを停止しなくても行えます。この手順は、単一の Q サブスクリプションの属性を変更するか、あるいは複数の Q サブスクリプションの属性を変更するかによって異なります。

以下のリストでは、変更できる属性について説明しています。

検索条件

ソース表からどの行が複製されるかを決定するために使用される WHERE 文節。

ソース表削除

ソース表からの行の削除操作を複製するかどうかに関するオプション。

変更されたすべての行 (All changed rows)

いずれかの列が変更された時に、変更された列が Q サブスクリプションの一部ではなくても、ソース表の行を複製するかどうかに関するオプション。

変更を行ってから Q サブスクリプションを再初期化することにより、1 つの Q サブスクリプションの属性を変更できます。

変更を行ってから Q キャプチャー・プログラムを再初期化することにより、Q キャプチャー・スキーマ内で定義されている、複数の Q サブスクリプションの属性を変更できます。

重要: Q キャプチャー・プログラムを再初期化する場合、このアクションにより、Q キャプチャー・スキーマ内で定義されている、すべての XML 発行の再初期化も行われます。Q キャプチャー・プログラムの再初期化には、システム・リソースもさらに必要になります。属性を変更しなければならない Q サブスクリプションが 1

つか 2 つだけの場合、一度に 1 つの Q サブスクリプションを再初期化することにより、リソースの面でコストを減らすことができます。

制約事項:

Q サブスクリプションの他の属性を変更する場合、その Q サブスクリプションを削除し、別の属性を設定して再作成します。Q サブスクリプションの削除および再作成を行わずに、双方向またはピアツーピア・レプリケーションで使用される Q サブスクリプションの属性を変更することはできません。

手順:

レプリケーションを停止せずに、単一の Q サブスクリプションでの属性を変更するには、以下のようにします。

1. Q サブスクリプションの属性を変更する。

レプリケーション・センターで「Q サブスクリプションのプロパティ」ノートブックを使用し、一度に 1 つの Q サブスクリプションの属性を変更します。このノートブックをオープンするには、Q サブスクリプションを右マウス・ボタンでクリックして、「プロパティ」を選択します。詳細については、オンライン・ヘルプを参照してください。

2. Q サブスクリプションを再初期化する。

レプリケーション・センター

「Q サブスクリプションの管理」ウィンドウを使用して、Q サブスクリプションを再初期化します。このウィンドウをオープンするには、Q サブスクリプションのソース表が置かれている Q キャプチャー・サーバーを右マウス・ボタンでクリックして、「管理」→「Q サブスクリプション」を選択します。詳細については、オンライン・ヘルプを参照してください。

SQL 以下のようにコマンド・プロンプトまたはいずれかの DB2 コマンド行ツールを使用して、REINIT_SUB シグナルを Q キャプチャー・サーバーの IBMQREP_SIGNAL 表に挿入します。

```
insert into schema.IBMQREP_SIGNAL(  
    SIGNAL_TIME,  
    SIGNAL_TYPE,  
    SIGNAL_SUBTYPE,  
    SIGNAL_INPUT_IN,  
    SIGNAL_STATE  
) values (  
    CURRENT_TIMESTAMP,  
    'CMD',  
    'REINIT_SUB',  
    'subname',  
    'P' );
```

ここで、*schema* は Q サブスクリプションを処理する Q キャプチャー・プログラムを示し、*subname* は再初期化する Q サブスクリプションの名前です。

レプリケーションを静止せずに、複数の Q サブスクリプションの属性を変更するには、以下のようにします。

1. Q サブスクリプションの属性を変更する。

レプリケーション・センターで「Q サブスクリプションのプロパティ」ノートブックを使用し、一度に 1 つの Q サブスクリプションの属性を変更します。このノートブックをオープンするには、Q サブスクリプションのソース表が置かれている Q キャプチャー・サーバーを右マウス・ボタンでクリックして、「管理」→「Q サブスクリプション」を選択します。詳細については、オンライン・ヘルプを参照してください。

2. Q キャプチャー・プログラムを再初期化する。

レプリケーション・センター

「Q キャプチャーの再初期化」ウィンドウを使用して、1 つの Q キャプチャー・プログラムのすべての Q サブスクリプションを再初期化します。このウィンドウをオープンするには、再初期化する Q キャプチャー・プログラムが含まれている Q キャプチャー・サーバーを右マウス・ボタンでクリックして、「Q キャプチャー・プログラムの再初期化」を選択します。詳細については、オンライン・ヘルプを参照してください。

asnlccmd システム・コマンド

以下のように、**asnlccmd reinit** コマンドを使用して、1 つの Q キャプチャー・スキーマのすべての Q サブスクリプションを再初期化します。

```
asnlccmd capture_server=server_name capture_schema=schema
reinit
```

ここで、*server_name* は Q キャプチャー・プログラムが実行されているデータベースまたはサブシステムの名前であり、*schema* はすべての Q サブスクリプションを再初期化する Q キャプチャー・プログラムを示しています。

関連概念:

- 205 ページの『Q レプリケーション環境の変更 -- 概要』
- 15 ページの『Q キャプチャー・プログラム』
- 5 ページの『Q サブスクリプション』
- 98 ページの『単一方向レプリケーションの Q サブスクリプションの作成 -- 概要』
- 108 ページの『行のフィルタリングのための検索条件 (単一方向レプリケーション)』

関連タスク:

- 219 ページの『XML 発行の属性の変更』

関連資料:

- 368 ページの『asnlccmd: 実行中の Q キャプチャー・プログラムの操作』
- 437 ページの『IBMQREP_SIGNAL 表』

既存の Q サブスクリプションへの列の追加

ソース表からの変更のレプリケーションを停止せずに、単一方向レプリケーションでのアクティブな Q サブスクリプションに動的に列を追加できます。

列がソース表にすでに存在していても構いませんし、その表に列を追加してから、その列を同じトランザクション内の Q サブスクリプションに追加することもできます。

新しい列を追加するには、Q キャプチャー・サーバーで SQL シグナルを挿入します。SQL シグナルには、その列に関する詳細が含まれています。シグナルを挿入する場合、ターゲット表へまだ列を追加していなければ、それが自動的に行われます。複数の列を Q サブスクリプションに追加する場合、新しい列ごとに 1 つのシグナルを挿入します。単一トランザクションに複数の列を追加できます。

推奨: 必ず新しい列がソース表の列と一致するように、レプリケーション・プログラムが自動的に新しい列をターゲット表に追加するようにしてください。ソース表内の対応する列と同じ名前、データ・タイプ、ヌル特性、およびデフォルト値で、列がターゲット表に追加されます。

前提条件:

- 列が追加される Q サブスクリプションは、シグナルが挿入される時点で A (アクティブ) でなければなりません。
- ソース表に列が存在していなければなりません。シグナルの挿入として、同一のトランザクションでソース表に対する ALTER TABLE ADD COLUMN 操作を実行する場合、シグナルを挿入する前に ALTER 操作が行われなければなりません。
- 列のデータ・タイプが LONG VARCHAR または GRAPHIC である場合、ソース・データベースまたはサブシステムは、DATA CAPTURE CHANGES INCLUDE VARCHAR COLUMNS で構成されなければなりません。

制約事項:

- 双方向またはピアツーピア・レプリケーションでは、ADDCOL シグナルを使用して、列を既存の Q サブスクリプションに追加することはできません。
- 追加する列は NULL 可能か、または NOT NULL WITH DEFAULT として定義されなければなりません。
- 1 回の WebSphere MQ コミット・インターバル中に追加できる列は 20 までです。これは、Q Capture COMMIT_INTERVAL パラメーターで設定されます。

手順:

既存の Q サブスクリプションに列を追加するには、以下のようにします。

以下のようにコマンド・プロンプトまたはいずれかの DB2 コマンド行ツールを使用して、ADDCOL シグナルを Q キャプチャー・サーバーの IBMQREP_SIGNAL 表に挿入します。

```
insert into schema.IBMQREP_SIGNAL(
    SIGNAL_TIME,
    SIGNAL_TYPE,
    SIGNAL_SUBTYPE,
    SIGNAL_INPUT_IN,
    SIGNAL_STATE
) values (
    CURRENT_TIMESTAMP,
    'CMD',
    'ADDCOL',
    'subname;column_name',
    'P' );
```

ここで、

schema

列を追加する Q サブスクリプションを処理する、Q キャプチャー・プログラムを示します。

subname;column_name

列を追加する Q サブスクリプションの名前および追加する列の名前 (セミコロンで区切ります)。これらの名前には大文字小文字の区別があり、大文字小文字をそのまま保持するための二重引用符は必要ありません。

シグナルが処理されてから、Q キャプチャー・プログラムは新しい列を含むログ・データを読み取った時点で、その列への変更のキャプチャーを開始します。ADDCOL シグナルの挿入のコミット後にコミットされた列への変更が、ターゲット表の新しい列へ複製されます。新しい列が追加されるよりも前からターゲット表に存在していた行には、新しい列に対して NULL またはデフォルト値が入れられません。

ヒント:

Q キャプチャー・プログラムを停止して再始動しなくても、ソース表へ列を追加できます。ソース・データベースまたはインスタンスを静止してから列を追加するように計画している場合、レプリケーション・センターまたはコマンドを使用して、Q キャプチャー・プログラムの TERM パラメーターを N (いいえ) に設定できます。TERM=N を設定すると、データベースまたはインスタンスが静止モードの間も、プログラムの実行は継続します。DB2 UDB が静止モードではなくなった時に、Q キャプチャー・プログラムはログ内の最新の再始動点からの変更のキャプチャーに戻ります。この場合、プログラムの再始動は必要ありません。

レプリケーション・センターで「パラメーターの変更」 - 「Q キャプチャー・プログラムの実行」ウィンドウを使用するか、または **asnlccmd chgparms** コマンドを使用して、Q キャプチャー・プログラムの実行中に TERM パラメーターを設定できます。

関連概念:

- 205 ページの『Q レプリケーション環境の変更 -- 概要』
- 5 ページの『Q サブスクリプション』

関連タスク:

- 205 ページの『Q サブスクリプションの属性の変更』

関連資料:

- 235 ページの『Q キャプチャー・パラメーターの説明』
- 437 ページの『IBMQREP_SIGNAL 表』

Q サブスクリプションでの許容可能な SQL 状態の変更

単一方向 Q サブスクリプションでは、許容可能と定義された SQL 状態を変更できます。トランザクションをアプライする間に、Q アプライ・プログラムがこれらの SQL 状態のいずれかを見つけた場合、エラーの原因となった行はアプライされず、Q アプライ・プログラムはトランザクション内の次の行から続行します。

レプリケーション・センターを使用して、許容可能な SQL 状態を変更できます。定義する許容可能な SQL 状態は、IBMQREP_TARGETS コントロール表の OKSQLSTATES 列に保管されます。

許容可能な SQL 状態を変更するには、Q サブスクリプションで使用される受信キューでのメッセージ処理を停止し、属性を変更してから、受信キューでのメッセージ処理を開始しなければなりません。

制約事項:

双方向またはピアツーピア・レプリケーションで使用される Q サブスクリプションでは、許容可能な SQL 状態を変更できません。

手順:

Q サブスクリプションの許容可能な SQL 状態を変更するには、以下のようになります。

1. Q サブスクリプションで使用される受信キューでのメッセージ処理を停止する。

レプリケーション・センター

「受信キューの管理」ウィンドウを使用して、受信キューでのメッセージ処理を停止します。このウィンドウをオープンするには、受信キューが置かれている Q アプライ・サーバーを右マウス・ボタンでクリックして、「管理」→「受信キュー」を選択します。詳細については、オンライン・ヘルプを参照してください。

asnlqcmd コマンド

asnlqcmd stopq コマンドを使用して、受信キューでのメッセージ処理を停止します。

```
asnlqcmd apply_server=server_name apply_schema=schema  
stopq=receive_queue_name
```

ここで、*server_name* は Q アプライ・サーバーの名前であり、*schema* は受信キューからのメッセージを処理する Q アプライ・プログラムを示しています。さらに、*receive_queue_name* はメッセージ処理を停止する受信キューの名前です。

2. Q サブスクリプションの OKSQLSTATES 属性を変更する。

レプリケーション・センターで「Q サブスクリプションのプロパティ」ノートブックを使用し、Q サブスクリプションの OKSQLSTATES 属性を変更します。このノートブックをオープンするには、Q サブスクリプションを右マウス・ボタンでクリックして、「プロパティ」を選択します。詳細については、オンライン・ヘルプを参照してください。

3. Q サブスクリプションで使用される受信キューでのメッセージ処理を開始する。

レプリケーション・センター

「受信キューの管理」ウィンドウを使用して、受信キューでのメッセージ処理を開始します。このウィンドウをオープンするには、受信キューが置かれている Q アプライ・サーバーを右マウス・ボタンでクリックして、「管理」→「受信キュー」を選択します。詳細については、オンライン・ヘルプを参照してください。

asnlqcmd コマンド

asnlqcmd startq コマンドを使用して、受信キューでのメッセージ処理を開始します。

```
asnlqcmd apply_server=server_name apply_schema=schema  
startq=receive_queue_name
```

ここで、*server_name* は Q アプライ・サーバーの名前であり、*schema* は Q アプライ・プログラムを示しています。さらに、*receive_queue_name* はメッセージ処理を開始する受信キューの名前です。

関連概念:

- 205 ページの『Q レプリケーション環境の変更 -- 概要』
- 5 ページの『Q サブスクリプション』
- 98 ページの『単方向レプリケーションの Q サブスクリプションの作成 -- 概要』

関連タスク:

- 275 ページの『受信キュー上のメッセージ処理の停止』
- 273 ページの『受信キュー上のメッセージ処理の開始』

関連資料:

- 376 ページの『asnlqcmd: 実行中の Q アプライ・プログラムの操作』

レプリケーション・キュー・マップの属性の変更

レプリケーション・キュー・マップを更新して、特定の送信キューおよび受信キューを介して伝送されるトランザクションを、Q キャプチャー・プログラムまたは Q アプライ・プログラム、あるいはその両方が処理する方法を変更できます。

レプリケーション・キュー・マップには、ペアになっている送信キューと受信キューを使用する Q サブスクリプションが、どのように処理されるかに関する設定が含まれています。キュー・マップを更新してから送信キューまたは受信キュー、あるいはその両方を再初期化することにより、レプリケーションを停止しなくても、これらの設定のいくつかを変更できます。

レプリケーション・キュー・マップのプロパティは、Q キャプチャーおよび Q アプライ・コントロール表に保管されます。送信キューを再初期化する際に、Q キャプチャー・プログラムは `IBMQREP_SENDQUEUES` 表から最新の設定を入手します。受信キューを再初期化する際に、Q アプライ・プログラムは `IBMQREP_RECVQUEUES` 表から最新の設定を入手します。この新しい設定は、レプリケーション・キュー・マップを使用するすべての Q サブスクリプションに影響を与えます。

制約事項:

レプリケーション・キュー・マップの以下の属性は更新できません。

- キュー・マップの名前
- キュー・マップが指示している送信キュー、受信キュー、または管理キュー

- Q キャプチャー・サーバーおよび Q キャプチャー・スキーマ

手順:

レプリケーション・キュー・マップを更新して、Q キャプチャー・プログラムまたは Q アプライ・プログラムにその変更を認識させるには、以下のようにします。

1. キュー・マップのプロパティを変更する。

レプリケーション・センターで「レプリケーション・キュー・マップ・プロパティ」ウィンドウを使用して、以下のプロパティのいずれかを変更します。

送信キュー

- エラー・アクション。
- キューに入れるメッセージのバッファとして、Q キャプチャー・プログラムが割り当てるメモリの最大量。
- キューに入れるメッセージがない時に、Q キャプチャー・プログラムがまだ実行中であることを示すために送信する、ハートビート・メッセージ間のインターバル。

受信キュー

- キューからのトランザクションを処理するために、Q アプライ・プログラムが使用するエージェント・スレッドの数。
- キューから取り出すメッセージのバッファとして、Q アプライ・プログラムが割り当てるメモリの最大量。

「レプリケーション・キュー・マップ・プロパティ」ウィンドウをオープンするには、レプリケーション・キュー・マップを右マウス・ボタンでクリックして、「プロパティ」を選択します。詳細については、オンライン・ヘルプを参照してください。

2. 変更するプロパティに応じて、送信キューまたは受信キュー、あるいはその両方を再初期化する。両方のキューを再初期化しなければならない場合、任意の順序で行うことができます。次の方法のいずれかを使用します。

レプリケーション・センター

- 「送信キューの管理」ウィンドウを使用して、レプリケーション・キュー・マップでプロパティを変更した送信キューを再初期化する。このウィンドウをオープンするには、送信キューが置かれている Q キャプチャー・サーバーを右マウス・ボタンでクリックして、「管理」→「送信キュー」を選択します。
- 「受信キューの管理」ウィンドウを使用して、レプリケーション・キュー・マップでプロパティを変更した受信キューを再初期化する。このウィンドウをオープンするには、受信キューが置かれている Q アプライ・サーバーを右マウス・ボタンでクリックして、「管理」→「受信キュー」を選択します。

詳細については、オンライン・ヘルプを参照してください。

asnqccmd システム・コマンド (送信キュー)

以下のように、`asnqccmd reinitq` コマンドを使用して、レプリケーション・キュー・マップでプロパティを変更した送信キューを再初期化します。

```
asnlccmd capture_server=server_name capture_schema=schema
reinitq=queue_name
```

ここで、*server_name* は Q キャプチャー・プログラムが実行されているデータベースまたはサブシステムの名前であり、*schema* は送信キューを使用する Q キャプチャー・プログラムを示しています。さらに、*queue_name* は再初期化する送信キューの名前です。

asnlccmd システム・コマンド (受信キュー)

以下のように、**asnlccmd reinitq** コマンドを使用して、レプリケーション・キュー・マップでプロパティを変更した送信キューを再初期化します。

```
asnlccmd apply_server=server_name apply_schema=schema
reinitq=queue_name
```

ここで、*server_name* は Q アプライ・プログラムが実行されているデータベースまたはサブシステムの名前であり、*schema* は受信キューを使用する Q アプライ・プログラムを示しています。さらに、*queue_name* は再初期化する受信キューの名前です。

関連概念:

- 205 ページの『Q レプリケーション環境の変更 -- 概要』
- 6 ページの『レプリケーション・キュー・マップ』

関連タスク:

- 223 ページの『発行キュー・マップの属性の変更』
- 95 ページの『レプリケーション・キュー・マップの作成』

関連資料:

- 368 ページの『asnlccmd: 実行中の Q キャプチャー・プログラムの操作』
- 376 ページの『asnlccmd: 実行中の Q アプライ・プログラムの操作』
- 459 ページの『IBMQREP_RECVQUEUES 表』
- 434 ページの『IBMQREP_SENDQUEUES 表』

Q サブスクリプションの削除

Q キャプチャー・プログラムがアクティブに処理していない Q サブスクリプションを削除できます。Q サブスクリプションがアクティブの場合、非アクティブにしてからでなければ削除できません。

レプリケーション・センターで Q サブスクリプションを削除する場合、参照するターゲット表をドロップするかどうかを選択できます。ターゲット表の表スペースをドロップすることもできます。

注: Q サブスクリプションの削除によって、それが使用しているレプリケーション・キュー・マップが削除されることはありません。

前提条件:

- 削除する Q サブスクリプションは I (非アクティブ) または N (新規) 状態でなければなりません。

手順:

1 つ以上の Q サブスクリプションを削除するには、以下のようにします。

1. Q サブスクリプションが A (アクティブ) 状態の場合、非アクティブにする。次の方法のいずれかを使用します。

レプリケーション・センター

「Q サブスクリプションの管理」ウィンドウを使用して、1 つ以上の Q サブスクリプションを非アクティブにします。このウィンドウをオープンするには、Q サブスクリプションのソース表が置かれている Q キャプチャー・サーバーを右マウス・ボタンでクリックして、「管理」→「Q サブスクリプション」を選択します。このウィンドウを使用して、Q サブスクリプションの状態が A (アクティブ) から I (非アクティブ) に変更されたかどうかを検査します。非アクティブ・プロセスを完了してからでなければ、Q サブスクリプションは削除できません。詳細については、オンライン・ヘルプを参照してください。

SQL 以下のようにコマンド・プロンプトまたはいずれかの DB2 コマンド行ツールを使用して、Q サブスクリプションが定義されている Q キャプチャー・サーバーの IBMQREP_SIGNAL 表に CAPSTOP シグナルを挿入します。以下のように、非アクティブにする Q サブスクリプションごとにシグナルを挿入します。

```
insert into schema.IBMQREP_SIGNAL (
    SIGNAL_TIME,
    SIGNAL_TYPE,
    SIGNAL_SUBTYPE,
    SIGNAL_INPUT_IN,
    SIGNAL_STATE )
values (
    CURRENT_TIMESTAMP,
    'CMD',
    'CAPSTOP',
    'subname',
    'P' );
```

ここで、*schema* は Q キャプチャー・プログラムを示し、*subname* は非アクティブにする Q サブスクリプションの名前です。

2. Q サブスクリプションを削除する。

レプリケーション・センターで「Q サブスクリプションの削除」ウィンドウを使用し、1 つ以上の Q サブスクリプションを削除します。このウィンドウをオープンするには、Q サブスクリプションを右マウス・ボタンでクリックして、「削除」を選択します。このウィンドウを使用して、削除する Q サブスクリプションのターゲット表をドロップすることもできます。ターゲット表をドロップする場合、このウィンドウの使用を選択して、関連する表スペースをドロップすることもできます。ただし、他の表がその表スペースを使用していない場合に限りです。詳細については、オンライン・ヘルプを参照してください。

関連概念:

- 205 ページの『Q レプリケーション環境の変更 -- 概要』

関連タスク:

- 225 ページの『XML 発行の削除』
- 254 ページの『Q サブスクリプションまたは XML 発行の非アクティブ化』

関連資料:

- 437 ページの『IBMQREP_SIGNAL 表』

レプリケーション・キュー・マップの削除

どの Q サブスクリプションでも必要とされなくなったレプリケーション・キュー・マップを削除できます。

前提条件:

そのレプリケーション・キュー・マップを使用している Q サブスクリプションがないことを確認してください。もしあれば、スクリプトを生成してそのキュー・マップを削除しようとした時に、エラー・メッセージが出されます。

手順:

レプリケーション・キュー・マップを削除するには、レプリケーション・センターを使用します。

1. 「関連表示」ウィンドウをオープンし、いずれかの Q サブスクリプションによってそのレプリケーション・キュー・マップが使用されていないかを調べる。このウィンドウをオープンするには、レプリケーション・キュー・マップを右マウス・ボタンでクリックして、「**関連表示**」を選択します。
2. いずれかの Q サブスクリプションがそのキュー・マップを使用している場合、「Q サブスクリプションの削除」ウィンドウを使用して、その Q サブスクリプションを削除する。このウィンドウをオープンするには、Q サブスクリプションを右マウス・ボタンでクリックして、「**削除**」を選択します。
3. 「レプリケーション・キュー・マップの削除」ウィンドウを使用して、そのキュー・マップを削除する。このウィンドウをオープンするには、レプリケーション・キュー・マップを右マウス・ボタンでクリックして、「**削除**」を選択します。

詳細については、オンライン・ヘルプを参照してください。

関連概念:

- 205 ページの『Q レプリケーション環境の変更 -- 概要』
- 6 ページの『レプリケーション・キュー・マップ』

関連タスク:

- 226 ページの『発行キュー・マップの削除』

Q キャプチャー・スキーマまたは Q アプライ・スキーマのドロップ

スキーマ・オブジェクトは、Q キャプチャーまたは Q アプライ・プログラムのインスタンスと、それに関連するコントロール表を表します。Q キャプチャーまたは Q アプライ・スキーマをドロップすると、スキーマ・オブジェクトがレプリケーション・センターのオブジェクト・ツリーから削除され、コントロール表がドロップされます。

スキーマをドロップすると、以下のオブジェクトも削除されます。

Q キャプチャー・サーバーで

Q サブスクリプションと XML 発行の両方が除去されます。これは、それらの 2 つのオブジェクトの定義が同じコントロール表に保管されているためです。スキーマをドロップすると、レプリケーション・キュー・マップと発行キュー・マップの両方も除去されます。

Q アプライ・サーバーで

Q サブスクリプションおよびレプリケーション・キュー・マップが除去されます。

重要: Q サブスクリプションおよびレプリケーション・キュー・マップは、Q キャプチャー・サーバーと Q アプライ・サーバーの両方で定義されます。Q キャプチャー・スキーマだけまたは Q アプライ・スキーマだけをドロップする場合、そのスキーマをドロップする前に、もう一方のサーバーのコントロール表でも定義されている、すべての Q サブスクリプションまたはレプリケーション・キュー・マップを除去しなければなりません。

前提条件:

ドロップするスキーマによって識別される、Q キャプチャー・プログラムまたは Q アプライ・プログラムを停止しなければなりません。

手順:

Q キャプチャーまたは Q アプライ・スキーマをドロップするには、以下のようになります。

1. ドロップするスキーマによって識別される Q キャプチャー・プログラムまたは Q アプライ・プログラムを停止する。次の方法のいずれかを使用します。

レプリケーション・センター

「Q キャプチャーの停止」ウィンドウまたは「Q アプライの停止」ウィンドウを使用して、Q キャプチャー・プログラムまたは Q アプライ・プログラムを停止します。

- 「Q キャプチャーの停止」ウィンドウをオープンするには、停止する Q キャプチャー・プログラムが含まれている Q キャプチャー・サーバーを右マウス・ボタンでクリックして、「**Q キャプチャー・プログラムの停止**」を選択する。
- 「Q アプライの停止」ウィンドウをオープンするには、停止する Q アプライ・プログラムが含まれている Q アプライ・サーバーを右マウス・ボタンでクリックして、「**Q アプライ・プログラムの停止**」を選択する。

詳細については、オンライン・ヘルプを参照してください。

asnlccmd システム・コマンド (Q キャプチャー・プログラム)

以下のように、**asnlccmd stop** コマンドを使用して Q キャプチャー・プログラムを停止します。

```
asnlccmd capture_server=server_name capture_schema=schema stop
```

ここで、*server_name* は Q キャプチャー・プログラムが実行されているデータベースまたはサブシステムの名前であり、*schema* は停止する Q キャプチャー・プログラムを示しています。

SQL (Q キャプチャー・プログラム)

以下のようにコマンド・プロンプトまたはいずれかの DB2 コマンド行ツールを使用して、STOP シグナルを Q キャプチャー・サーバーの IBMQREP_SIGNAL 表に挿入します。

```
insert into schema.IBMQREP_SIGNAL(  
    SIGNAL_TIME,  
    SIGNAL_TYPE,  
    SIGNAL_SUBTYPE,  
    SIGNAL_INPUT_IN,  
    SIGNAL_STATE)  
values (  
    CURRENT_TIMESTAMP,  
    'CMD',  
    'STOP',  
    'NULL',  
    'P');
```

ここで、*schema* は停止する Q キャプチャー・プログラムを示しています。

asnlqcmd システム・コマンド (Q アプライ・プログラム)

以下のように、**asnlqcmd stop** コマンドを使用して、Q アプライ・プログラムを停止します。

```
asnlqcmd apply_server=server_name apply_schema=schema stop
```

ここで、*server_name* は Q アプライ・プログラムが実行されているデータベースまたはサブシステムの名前であり、*schema* は停止する Q アプライ・プログラムを示しています。

- オプション: Q キャプチャー・スキーマだけまたは Q アプライ・スキーマだけをドロップする場合、以下のアクションを行う。
 - もう一方のサーバーでも定義されている、Q サブスクリプションを非アクティブにする。詳細については、254 ページの『Q サブスクリプションまたは XML 発行の非アクティブ化』を参照してください。
 - Q サブスクリプションを削除する。詳細については、213 ページの『Q サブスクリプションの削除』を参照してください。
 - もう一方のサーバーでも定義されている、すべてのレプリケーション・キュー・マップを削除する。詳細については、215 ページの『レプリケーション・キュー・マップの削除』を参照してください。
- レプリケーション・センターで「Q キャプチャー・スキーマのドロップ」または「Q アプライ・スキーマのドロップ」ウィンドウを使用し、Q キャプチャーまたは Q アプライ・スキーマをドロップする。これらのウィンドウをオープンす

るには、スキーマを右マウス・ボタンでクリックして、「ドロップ」を選択します。詳細については、オンライン・ヘルプを参照してください。

Q キャプチャー・サーバーまたは Q アプライ・サーバーにある最新のスキーマをドロップすると、そのサーバーにはコントロール表のセットが含まれないことになり、レプリケーション・センターのオブジェクト・ツリーから除去されます。

関連概念:

- 205 ページの『Q レプリケーション環境の変更 -- 概要』
- 21 ページの『Q アプライ・プログラムと Q キャプチャー・プログラムのスキーマ』

関連資料:

- 368 ページの『asnqccmd: 実行中の Q キャプチャー・プログラムの操作』
- 437 ページの『IBMQREP_SIGNAL 表』

第 16 章 イベント発行環境の変更

イベント発行環境の変更 -- 概要

以下のトピックでは、イベント発行環境への日常的な変更に関する問題と手順について説明します。

- 『XML 発行の属性の変更』
- 221 ページの『既存の XML 発行への列の追加』
- 223 ページの『発行キュー・マップの属性の変更』
- 225 ページの『XML 発行の削除』
- 226 ページの『発行キュー・マップの削除』
- 227 ページの『Q キャプチャー・スキーマのドロップ』

XML 発行の属性の変更

XML 発行のいくつかの属性を変更できます。これらの属性の変更は、発行を停止しなくても行うことができます。この手順は、単一の XML 発行の属性を変更するか、あるいは複数の XML 発行の属性を変更するかによって異なります。

以下のリストでは、変更できる属性について説明しています。

検索条件

ソース表からどの行が複製されるかを決定するために使用される WHERE 文節。

ソース表削除

ソース表からの行の削除操作を複製するかどうかに関するオプション。

変更されたすべての行 (All changed rows)

何らかの列が変更された時に、変更された列が XML 発行の一部ではなくても、ソース表の行を複製するかどうかに関するオプション。

変更を行ってから XML 発行を再初期化することにより、1 つの XML 発行での属性を変更できます。

変更を行ってから Q キャプチャー・プログラムを再初期化することにより、Q キャプチャー・スキーマ内で定義されている、複数の XML 発行の属性を変更できます。

重要: Q キャプチャー・プログラムを再初期化する場合、このアクションにより、Q キャプチャー・スキーマ内で定義されている、すべての Q サブスクリプションの再初期化も行われます。Q キャプチャー・プログラムの再初期化には、システム・リソースもさらに必要になります。属性を変更しなければならない XML 発行が 1 つか 2 つだけの場合、一度に 1 つの XML 発行を再初期化することにより、リソースの点ではコストを減らすことができます。

制約事項:

XML 発行の他の属性を変更する場合、その XML 発行を削除し、別の属性を設定して再作成します。

手順:

発行を停止せずに、単一の XML 発行での属性を変更するには、以下を行います。

1. XML 発行の属性を変更する。

レプリケーション・センターで「XML 発行のプロパティ」ノートブックを使用し、一度に 1 つの XML 発行の属性を変更します。このノートブックをオープンするには、XML 発行を右マウス・ボタンでクリックして、「プロパティ」を選択します。詳細については、オンライン・ヘルプを参照してください。

2. XML 発行を再初期化する。

レプリケーション・センター

「XML 発行の管理」ウィンドウを使用して、XML 発行を再初期化します。このウィンドウをオープンするには、XML 発行のソース表が置かれている Q キャプチャー・サーバーを右マウス・ボタンでクリックして、「管理」→「XML 発行」を選択します。詳細については、オンライン・ヘルプを参照してください。

SQL 以下のようにコマンド・プロンプトまたはいずれかの DB2 コマンド行ツールを使用して、REINIT_SUB シグナルを Q キャプチャー・サーバーの IBMQREP_SIGNAL 表に挿入します。

```
insert into schema.IBMQREP_SIGNAL(  
    SIGNAL_TIME,  
    SIGNAL_TYPE,  
    SIGNAL_SUBTYPE,  
    SIGNAL_INPUT_IN,  
    SIGNAL_STATE  
) values (  
    CURRENT_TIMESTAMP,  
    'CMD',  
    'REINIT_SUB',  
    'subname',  
    'P' );
```

ここで、*schema* は XML 発行を処理する Q キャプチャー・プログラムを示し、*subname* は再初期化する XML 発行の名前です。

レプリケーションを静止せずに、複数の XML 発行の属性を変更するには、以下のようになります。

1. XML 発行の属性を変更する。

レプリケーション・センターで「XML 発行のプロパティ」ノートブックを使用し、一度に 1 つの XML 発行の属性を変更します。このノートブックをオープンするには、XML 発行を右マウス・ボタンでクリックして、「プロパティ」を選択します。詳細については、オンライン・ヘルプを参照してください。

2. Q キャプチャー・プログラムを再初期化する。

レプリケーション・センター

「Q キャプチャーの再初期化」ウィンドウを使用して、1 つの Q キャ

プチャー・プログラムの XML 発行をすべて再初期化します。このウィンドウをオープンするには、再初期化する Q キャプチャー・プログラムが含まれている Q キャプチャー・サーバーを右マウス・ボタンでクリックして、「**Q キャプチャー・プログラムの再初期化**」を選択します。詳細については、オンライン・ヘルプを参照してください。

asnqccmd システム・コマンド

以下のように、**asnqccmd reinit** コマンドを使用して、1 つの Q キャプチャー・スキーマの XML 発行をすべて再初期化します。

```
asnqccmd capture_server=server_name capture_schema=schema
reinit
```

ここで、*server_name* は Q キャプチャー・プログラムが実行されているデータベースまたはサブシステムの名前であり、*schema* はすべての XML 発行を再初期化する Q キャプチャー・プログラムを示しています。

関連概念:

- 219 ページの『イベント発行環境の変更 -- 概要』
- 189 ページの『XML 発行の行をフィルター処理する検索条件』
- 188 ページの『Q キャプチャー・プログラムが XML 発行用のメッセージを発行する時期』

関連タスク:

- 225 ページの『XML 発行の削除』
- 205 ページの『Q サブスクリプションの属性の変更』
- 185 ページの『複数の XML 発行の作成』
- 183 ページの『単一 XML 発行の作成』

関連資料:

- 368 ページの『asnqccmd: 実行中の Q キャプチャー・プログラムの操作』

既存の XML 発行への列の追加

ソース表からの変更の発行を停止せずに、既存の XML 発行に列を動的に追加できます。

列はソース表にすでに存在していても構いませんし、表に列を追加してから、その列を同じトランザクション内の XML 発行に追加することもできます。

新しい列を追加するには、Q キャプチャー・サーバーで SQL シグナルを挿入します。SQL シグナルには、その列に関する詳細が含まれています。複数の列を XML 発行に追加する場合、新しい列ごとに 1 つのシグナルを挿入します。単一トランザクションに複数の列を追加できます。

前提条件:

- 列が追加される XML 発行は、シグナルが挿入される時点で A (アクティブ) でなければなりません。

- ソース表に列が存在していなければなりません。シグナルの挿入として、同一のトランザクションでソース表に対する ALTER TABLE ADD COLUMN 操作を実行する場合、シグナルを挿入する前に ALTER 操作が行われなければなりません。
- 列のデータ・タイプが LONG VARCHAR または GRAPHIC である場合、ソース・データベースまたはサブシステムは、DATA CAPTURE CHANGES INCLUDE VARCHAR COLUMNS で構成されなければなりません。

制約事項:

- 追加する列は NULL 可能か、または NOT NULL WITH DEFAULT として定義されなければなりません。
- 1 回の WebSphere MQ コミット・インターバル中に追加できる列は 20 までです。これは、Q Capture COMMIT_INTERVAL パラメーターで設定されます。

手順:

既存の XML 発行に列を追加するには、以下のようにします。

以下のようにコマンド・プロンプトまたはいずれかの DB2 コマンド行ツールを使用して、ADDCOL シグナルを Q キャプチャー・サーバーの IBMQREP_SIGNAL 表に挿入します。

```
insert into schema.IBMQREP_SIGNAL(
    SIGNAL_TIME,
    SIGNAL_TYPE,
    SIGNAL_SUBTYPE,
    SIGNAL_INPUT_IN,
    SIGNAL_STATE
) values (
    CURRENT_TIMESTAMP,
    'CMD',
    'ADDCOL',
    'pubname;column_name',
    'P' );
```

ここで、

schema

列を追加する XML 発行を処理する Q キャプチャー・プログラムを示します。

pubname;column_name

列を追加する XML 発行の名前、および追加する列の名前 (セミコロンで区切ります)。これらの名前には大文字小文字の区別があり、大文字小文字をそのまま保持するための二重引用符は必要ありません。

シグナルが処理されてから、Q キャプチャー・プログラムは add column XML メッセージをユーザー・アプリケーションに送信し、Q キャプチャー・プログラムが新しい列を含むログ・データを読み取った時点で、その列への変更のキャプチャーを開始します。ADDCOL シグナルの挿入のコミット後にコミットされた列への変更が発行されます。

ヒント:

Q キャプチャー・プログラムを停止して再始動しなくても、ソース表へ列を追加できます。ソース・データベースまたはインスタンスを静止してから列を追加するように計画している場合、レプリケーション・センターまたはコマンドを使用して、Q

キャプチャー・プログラムの TERM パラメーターを N (いいえ) に設定できます。TERM=N を設定すると、データベースまたはインスタンスが静止モードの間も、プログラムの実行が継続されます。DB2 UDB が静止モードではなくなった時に、Q キャプチャー・プログラムはログ内の最新の再始動点からの変更のキャプチャーに戻ります。この場合、プログラムの再始動は必要ありません。

レプリケーション・センターで「パラメーターの変更」 - 「Q キャプチャー・プログラムの実行」ウィンドウを使用するか、または **asnqccmd chgparms** コマンドを使用して、Q キャプチャー・プログラムの実行中に TERM パラメーターを設定できます。

関連概念:

- 219 ページの『イベント発行環境の変更 -- 概要』
- 12 ページの『XML 発行』

関連タスク:

- 219 ページの『XML 発行の属性の変更』

関連資料:

- 235 ページの『Q キャプチャー・パラメーターの説明』
- 437 ページの『IBMQREP_SIGNAL 表』

発行キュー・マップの属性の変更

発行キュー・マップを更新して、特定の送信キューを使用して伝送されるトランザクションまたは行レベルの変更を、Q キャプチャー・プログラムが処理する方法を変更できます。

発行キュー・マップは 1 つの送信キューを指示し、その送信キューを使用する XML 発行を Q キャプチャー・プログラムが処理する方法に関する設定を示します。キュー・マップを更新してから送信キューを再初期化することにより、発行を停止しなくても、これらの設定のいくつかを変更できます。

発行キュー・マップのプロパティは、IBMQREP_SENDQUEUES コントロール表に保管されます。送信キューを再初期化する際に、Q キャプチャー・プログラムはこの表から最新の設定を入手します。新しい設定は、送信キューを使用するすべての XML 発行に影響を与えます。

制約事項:

発行キュー・マップの以下の属性は変更できません。

- キュー・マップの名前
- キュー・マップが指示している送信キュー
- レプリケーション・キュー・マップを使用する XML メッセージのコンテンツ・タイプ (メッセージに含まれているのが単一の行か、またはトランザクション全体か)

手順:

発行を停止せずに、発行キュー・マップを更新するには、以下を行います。

1. レプリケーション・センターで「発行キュー・マップ・プロパティ」ウィンドウを使用して、以下のプロパティのいずれかを変更する。
 - 送信キューに関するエラー・アクション。
 - 送信キューに入れるメッセージのバッファとして、Q キャプチャー・プログラムが割り当てるメモリの最大量。
 - 処理されるトランザクションがない時に、Q キャプチャー・プログラムがまだ実行中であることを示すために送信する、ハートビート・メッセージ間のインターバル。

「発行キュー・マップ・プロパティ」ウィンドウをオープンするには、発行キュー・マップを右マウス・ボタンでクリックし、「プロパティ」を選択します。詳細については、オンライン・ヘルプを参照してください。

2. 以下の方法のいずれかを使用して、Q キャプチャー・プログラムを停止させずに、加えた変更を Q キャプチャー・プログラムに認識させる。

レプリケーション・センター

「送信キューの管理」ウィンドウを使用して、プロパティを変更した送信キューを再初期化します。このウィンドウをオープンするには、再初期化する送信キューが置かれている Q キャプチャー・サーバーを右マウス・ボタンでクリックし、「管理」→「送信キュー」を選択します。詳細については、オンライン・ヘルプを参照してください。

asnlccmd システム・コマンド

以下のように、**asnlccmd reinitq** コマンドを使用して、プロパティを変更した送信キューを再初期化します。

```
asnlccmd capture_server=server_name capture_schema=schema
reinitq=queue_name
```

ここで、*server_name* は Q キャプチャー・プログラムが実行されているデータベースまたはサブシステムの名前であり、*schema* は Q キャプチャー・プログラムを示しています。さらに、*queue_name* は再初期化する送信キューの名前です。

関連概念:

- 219 ページの『イベント発行環境の変更 -- 概要』
- 26 ページの『Q キャプチャー・プログラムによって使用されるメモリー』
- 13 ページの『発行キュー・マップ』
- 9 ページの『イベント発行』

関連タスク:

- 211 ページの『レプリケーション・キュー・マップの属性の変更』
- 180 ページの『発行キュー・マップの作成』

関連資料:

- 368 ページの『asnlccmd: 実行中の Q キャプチャー・プログラムの操作』
- 434 ページの『IBMQREP_SENDQUEUES 表』

XML 発行の削除

Q キャプチャー・プログラムがアクティブに処理していない XML 発行を削除できます。XML 発行がアクティブの場合、非アクティブにしてからでなければ削除できません。

注: XML 発行の削除によって、それが使用している発行キュー・マップが削除されることはありません。

前提条件:

削除する XML 発行は I (非アクティブ) または N (新規) 状態でなければなりません。

手順:

1 つ以上の XML 発行を削除するには、以下を行います。

1. XML 発行が A (アクティブ) 状態の場合、非アクティブにする。次の方法のいずれかを使用します。

レプリケーション・センター

「XML 発行の管理」ウィンドウを使用して、1 つ以上の XML 発行を非アクティブにします。このウィンドウをオープンするには、XML 発行のソース表が置かれている Q キャプチャー・サーバーを右マウス・ボタンでクリックし、「管理」→「XML 発行」を選択します。このウィンドウを使用して、XML 発行の状態が A (アクティブ) から I (非アクティブ) に変更されたかどうかを検査します。非アクティブ・プロセスを完了してからでなければ、XML 発行は削除できません。詳細については、オンライン・ヘルプを参照してください。

SQL

コマンド・プロンプトまたはいずれかの DB2 コマンド行ツールを使用して、XML 発行が定義されている Q キャプチャー・サーバーの IBMQREP_SIGNAL 表に CAPSTOP シグナルを挿入します。以下のように、非アクティブにする XML 発行ごとにシグナルを挿入します。

```
insert into schema.IBMQREP_SIGNAL (
    SIGNAL_TIME,
    SIGNAL_TYPE,
    SIGNAL_SUBTYPE,
    SIGNAL_INPUT_IN,
    SIGNAL_STATE )
values (
    CURRENT_TIMESTAMP,
    'CMD',
    'CAPSTOP',
    'xml_publication_name',
    'P' );
```

ここで、*schema* は Q キャプチャー・プログラムを示し、*xml_publication_name* は非アクティブにする XML 発行の名前です。

2. XML 発行を削除する。

レプリケーション・センターで「XML 発行の削除」ウィンドウを使用し、1 つ以上の XML 発行を削除します。このウィンドウをオープンするには、XML 発

行を右マウス・ボタンでクリックし、「削除」を選択します。詳細については、オンライン・ヘルプを参照してください。

関連概念:

- 219 ページの『イベント発行環境の変更 -- 概要』
- 9 ページの『イベント発行』

関連タスク:

- 213 ページの『Q サブスクリプションの削除』
- 254 ページの『Q サブスクリプションまたは XML 発行の非アクティブ化』
- 185 ページの『複数の XML 発行の作成』
- 183 ページの『単一 XML 発行の作成』

関連資料:

- 437 ページの『IBMQREP_SIGNAL 表』

発行キュー・マップの削除

どの XML 発行でも必要とされなくなった発行キュー・マップを削除できます。

前提条件:

その発行キュー・マップを使用している XML 発行がないことを確認してください。それがあれば、スクリプトを生成してそのキュー・マップを削除しようとした時に、エラー・メッセージを受け取ります。

手順:

発行キュー・マップを削除するには、レプリケーション・センターを使用します。

1. 「関連表示」ウィンドウをオープンし、いずれかの XML 発行によってその発行キュー・マップが使用されていないかを調べる。このウィンドウをオープンするには、発行キュー・マップを右マウス・ボタンでクリックし、「関連表示」を選択します。
2. いずれかの XML 発行がそのキュー・マップを使用している場合、「XML 発行の削除」ウィンドウを使用して、その XML 発行を削除する。このウィンドウをオープンするには、XML 発行を右マウス・ボタンでクリックし、「削除」を選択します。
3. 「発行キュー・マップの削除」ウィンドウを使用して、そのキュー・マップを削除する。このウィンドウをオープンするには、発行キュー・マップを右マウス・ボタンでクリックし、「削除」を選択します。

詳細については、オンライン・ヘルプを参照してください。

関連概念:

- 219 ページの『イベント発行環境の変更 -- 概要』
- 13 ページの『発行キュー・マップ』
- 9 ページの『イベント発行』

関連タスク:

- 215 ページの『レプリケーション・キュー・マップの削除』
- 180 ページの『発行キュー・マップの作成』

Q キャプチャー・スキーマのドロップ

スキーマ・オブジェクトは、Q キャプチャー・プログラムのインスタンスと、それに関連するコントロール表を表します。Q キャプチャー・スキーマをドロップすると、スキーマ・オブジェクトがレプリケーション・センターのオブジェクト・ツリーから削除され、コントロール表がドロップされます。

スキーマをドロップすると、そのコントロール表で定義されているすべての XML 発行、Q サブスクリプション、発行キュー・マップ、またはレプリケーション・キュー・マップも削除されます。

前提条件:

ドロップするスキーマによって識別される Q キャプチャー・プログラムを停止しなければなりません。

手順:

Q キャプチャー・スキーマをドロップするには、以下のようになります。

1. ドロップするスキーマによって識別される Q キャプチャー・プログラムを停止する。次の方法のいずれかを使用します。

レプリケーション・センター

「Q キャプチャーの停止」ウィンドウを使用して、Q キャプチャー・プログラムを停止します。このウィンドウをオープンするには、停止する Q キャプチャー・プログラムが含まれている Q キャプチャー・サーバーを右マウス・ボタンでクリックし、「**Q キャプチャー・プログラムの停止**」を選択します。詳細については、オンライン・ヘルプを参照してください。

asnqccmd システム・コマンド

以下のように、**asnqccmd stop** コマンドを使用して Q キャプチャー・プログラムを停止します。

```
asnqccmd capture_server=server_name capture_schema=schema stop
```

ここで、*server_name* は Q キャプチャー・プログラムが実行されているデータベースまたはサブシステムの名前であり、*schema* は停止する Q キャプチャー・プログラムを示しています。

SQL 以下のようにコマンド・プロンプトまたはいずれかの DB2 コマンド行ツールを使用して、STOP シグナルを Q キャプチャー・サーバーの `IBMQREP_SIGNAL` 表に挿入します。

```
insert into ASN.IBMQREP_SIGNAL(  
    SIGNAL_TIME,  
    SIGNAL_TYPE,  
    SIGNAL_SUBTYPE,  
    SIGNAL_INPUT_IN,  
    SIGNAL_STATE)  
values (  
    CURRENT_TIMESTAMP,
```

```
'CMD',  
'STOP',  
'NULL',  
'P' );
```

- レプリケーション・センターで「Q キャプチャー・スキーマのドロップ」ウィンドウを使用し、スキーマをドロップする。このウィンドウをオープンするには、スキーマを右マウス・ボタンでクリックし、「**ドロップ**」を選択します。詳細については、オンライン・ヘルプを参照してください。

関連概念:

- 219 ページの『イベント発行環境の変更 -- 概要』
- 13 ページの『発行キュー・マップ』
- 5 ページの『Q サブスクリプション』
- 6 ページの『レプリケーション・キュー・マップ』
- 21 ページの『Q アプライ・プログラムと Q キャプチャー・プログラムのスキーマ』
- 12 ページの『XML 発行』

関連タスク:

- 216 ページの『Q キャプチャー・スキーマまたは Q アプライ・スキーマのドロップ』
- 256 ページの『Q キャプチャー・プログラムの停止』

関連資料:

- 368 ページの『asnqccmd: 実行中の Q キャプチャー・プログラムの操作』
- 437 ページの『IBMQREP_SIGNAL 表』

第 17 章 Q キャプチャー・プログラムの操作

Q キャプチャー・プログラムの操作 -- 概要

Q キャプチャー・プログラムは、Q サブスクリプションまたは XML 発行の一部であるソース表から、トランザクションまたは行レベルの変更をキャプチャーし、このトランザクション・データをメッセージとして WebSphere® MQ キューを介して送信します。Q キャプチャー・プログラムを操作するには、レプリケーション・センター、システム・コマンド、およびシステム・サービスを使用します。また、何通りかの方法で Q キャプチャー稼働パラメーターを変更できます。

以下のトピックには、Q キャプチャー・プログラムの操作方法について説明されています。

- 『Q キャプチャー・プログラムの開始』
- 232 ページの『Q キャプチャー・プログラムのコールド・スタートに関する考慮事項』
- 233 ページの『Q キャプチャー・プログラムのパラメーター -- 概要』
- 248 ページの『Q キャプチャー・パラメーターの変更 -- 概要』
- 253 ページの『Q サブスクリプションまたは XML 発行のアクティブ化』
- 254 ページの『Q サブスクリプションまたは XML 発行の非アクティブ化』
- 256 ページの『Q キャプチャー・プログラムの停止』

関連概念:

- 15 ページの『Q キャプチャー・プログラム』
- 21 ページの『Q アプライ・プログラムと Q キャプチャー・プログラムのスキーマ』

Q キャプチャー・プログラムの開始

Q キャプチャー・プログラムを開始して、アクティブまたは新規の Q サブスクリプションまたは XML 発行の DB2 リカバリー・ログから、トランザクションまたは行レベルの変更をキャプチャーし、そのトランザクション・データをメッセージとして WebSphere MQ キューに送信することを開始します。

開始モードを指定しないで Q キャプチャー・プログラムを初めて開始すると、デフォルトの開始モード `warmsi` が使用されます。このモードでは、プログラムは、読み取りを中止した点からログの読み取りを開始しようとします。プログラムを開始するのは今回が初めてなので、Q キャプチャーはコールド・スタート・モードに切り替わり、N (新規) または A (アクティブ) 状態になっている Q サブスクリプションまたは XML 発行の処理が開始されます。プログラムが変更のキャプチャーを開始するには、I (非アクティブ) 状態になっている Q サブスクリプションまたは XML 発行をアクティブ化する必要があります。

A (アクティブ) 状態になっている Q サブスクリプションまたは XML 発行がないとしても、Q キャプチャー・プログラムを開始できます。Q サブスクリプションまたは XML 発行をアクティブ化すると、Q キャプチャー・プログラムは変更のキャプチャーを開始します。

Q キャプチャー・プログラムを開始すると、開始パラメーター値を指定でき、次のいずれかの処置を行うまで、プログラムはその新しい値を使用します。

- プログラムの実行中にパラメーター値を変更する。
- プログラムを停止して再始動する。これにより、プログラムは IBMQREP_CAPPARMS 表を読み取り、そこに保管されている値を使用するように指示されます。

前提条件:

- Q キャプチャー・プログラムをリモート・ワークステーションから開始する場合は、Q キャプチャー・サーバーへの接続を構成する必要がある。詳細については、75 ページの『Q レプリケーションおよびイベント発行の接続要件』を参照してください。
- WebSphere MQ キュー・マネージャー、キュー、および他の必要なオブジェクトを、Q キャプチャー・サーバーと同じシステムで作成および構成する必要がある。詳細については、47 ページの『Q レプリケーションおよびイベント発行に必要な WebSphere MQ オブジェクト -- 概要』を参照してください。
- Q レプリケーション・オブジェクトとイベント発行オブジェクト、および WebSphere MQ オブジェクト用の適切な許可が必要。詳細については、39 ページの『Q キャプチャー・プログラムの許可要件』 および 70 ページの『WebSphere MQ オブジェクトの接続および許可に関する要件』を参照してください。
- 適切な Q キャプチャー・スキーマ用のコントロール表を作成する必要がある。詳細については、87 ページの『Q キャプチャー・プログラムと Q アプライ・プログラムのコントロール表の作成』を参照してください。
- Q キャプチャー・プログラムで機能するよう、ソース・データベースまたはサブシステムを構成する必要がある。詳細については、78 ページの『Q キャプチャー・プログラムが動作するソース・データベースの構成 (Linux、UNIX、Windows)』 および 85 ページの『Q レプリケーションおよびイベント発行用のデータベースの構成 (z/OS)』を参照してください。
- EXPORT ユーティリティを使用する自動ロードを指定している Q サブスクリプションが、N (新規) または A (アクティブ) 状態になっている場合は、ユーティリティが Q キャプチャー・サーバーに接続できるようにするために、Q アプライ・サーバーでパスワード・ファイルを作成する必要があります。詳細については、43 ページの『リモート・サーバー用のユーザー ID およびパスワードの保管』を参照してください。

手順:

次の方法のいずれかを使用して、Q キャプチャー・プログラムを開始します。

レプリケーション・センター

「Q キャプチャーの開始」ウィンドウを使用して、Q キャプチャー・プログラムを開始する。このウィンドウをオープンするには、開始したい Q キャプチャー・プログラムが含まれている Q キャプチャー・サーバーを右マ

ウス・ボタンをクリックして、「**Q キャプチャー・プログラムの開始**」を選択します。 `IBMQREP_CAPPARMS` コントロール表に保管されているパラメーター値を使用してプログラムを開始するか、開始する前に新しいランタイム値を指定できます。詳細については、オンライン・ヘルプを参照してください。

asnlcap システム・コマンド

asnlcap コマンドを使用して、Q キャプチャー・プログラムを開始し、次の開始パラメーターを指定する。

```
asnlcap capture_server=server_name capture_schema=schema parameters
```

ここで、*server_name* は Q キャプチャー・コントロール表が含まれているデータベースまたはサブシステムの名前であり、*schema* は開始したい Q キャプチャー・プログラムを示し、*parameters* は開始時に指定できる 1 つ以上のパラメーターです。

z/OS コンソールまたは TSO

z/OS では、Q キャプチャー・プログラムは、JCL を使用して開始するか、システム開始タスクとして開始できます。JCL で Q キャプチャー・プログラムを開始する場合は、新しい呼び出しパラメーターを指定できます。JCL を使用する場合に呼び出しパラメーターを指定する最良の方法は、そのパラメーターを `IBMQREP_CAPPARMS` 表に保管するという方法です。EXEC ステートメントの PARM パラメーターでは、100 文字を超えるサブパラメーターは指定できません。

Windows サービス

Windows オペレーティング・システムで DB2 レプリケーション・サービスを作成して、システムの始動時に Q キャプチャー・プログラムを自動的に開始できます。

Q キャプチャー・プログラムが開始したかどうかを確認するには、次の方法のいずれかを使用します。

- レプリケーション・センターの「**Q キャプチャー・メッセージ**」ウィンドウを使用して、プログラムが開始したことを示すメッセージを確認する。このウィンドウをオープンするには、メッセージを表示したい Q キャプチャー・プログラムが含まれている Q キャプチャー・サーバーを右マウス・ボタンをクリックして、「**レポート**」→「**Q キャプチャー・メッセージ**」を選択します。詳細については、オンライン・ヘルプを参照してください。
- レプリケーション・センターの「**状況のチェック**」ウィンドウを使用して、すべての Q キャプチャー・スレッドの状況を表示する。このウィンドウをオープンするには、チェックしたい Q キャプチャー・プログラムが置かれている Q キャプチャー・サーバーを右マウス・ボタンをクリックして、「**状況のチェック**」を選択します。詳細については、オンライン・ヘルプを参照してください。
- Q キャプチャー診断ログ・ファイル (Linux、UNIX、および Windows の場合は `db2instance.capture_server.capture_schema.QCAP.log`、z/OS の場合は `capture_server.capture_schema.QCAP.log`) の中に、プログラムが変更をキャプチャーしていることを示すメッセージがないかどうかを調べる。
- `IBMQREP_CAPTRACE` 表の中に、プログラムが変更をキャプチャーしていることを示すメッセージがないかどうかを調べる。

- **z/OS:** バッチ・モードで実行している場合は、z/OS コンソールまたは z/OS ジョブ・ログの中に、プログラムが開始したことを示すメッセージがないかどうかを調べる。

関連概念:

- 229 ページの『Q キャプチャー・プログラムの操作 -- 概要』
- 233 ページの『Q キャプチャー・プログラムのパラメーター -- 概要』
- 283 ページの『Q レプリケーションおよびイベント発行プログラムの履歴およびパフォーマンス・データ』
- 15 ページの『Q キャプチャー・プログラム』

関連タスク:

- 253 ページの『Q サブスクリプションまたは XML 発行のアクティブ化』
- 342 ページの『Windows Service Control Manager (SCM) を使用してレプリケーション・サービスを管理する方法 -- 概要』
- 336 ページの『JCL を使用した Q キャプチャー・プログラムの開始』

関連資料:

- 361 ページの『asnqcap: Q キャプチャー・プログラムの開始』
- 428 ページの『IBMQREP_CAPPARMS 表』
- 433 ページの『IBMQREP_CAPTRACE 表』

Q キャプチャー・プログラムのコールド・スタートに関する考慮事項

コールド・スタート・モードを使って Q キャプチャー・プログラムを開始する前に、このトピックを読んで Q キャプチャー・プログラムのコールド・スタートに関する問題を理解してください。

Q キャプチャー・プログラムをコールド・スタートする時。

- Q キャプチャー・プログラムを初めて始動する時 (必須)。
- 最後の再始動点ではなく、アクティブ・ログの最後から変更をキャプチャーする時。

Q キャプチャー・プログラムのコールド・スタートの結果として生じうる問題。

Q キャプチャー・プログラムが最初に始動した後の任意の時点で Q キャプチャー・プログラムをコールド・スタートすると、Q キャプチャー・プログラムは、最後の再始動点からではなく、最後から DB2 ログの読み取りを開始します。これにより以下ようになります。

- ソースおよびターゲットが同期しなくなり、ターゲットのロードが必要になることがあります。Q キャプチャー・プログラムが、Q アプライ・プログラムに渡されるはずだったデータのログ・レコードをスキップする可能性があります。これらのレコードにソース表への更新または挿入が含まれる場合、ソース表とターゲット表を同期化する唯一の方法は、ターゲット表をロードすることです (フル・リフレッシュとも呼ばれる)。
- ターゲットをロードすると、履歴データが失われることがあります。ソースからの削除の抑制を選択する場合に保持される履歴データは、ソース中の削除された行のレコードです。履歴データは、ターゲット表がロードされると失われます。

- ロードに時間がかかることがあります。たくさんの表または大量のデータを含む環境の場合、ロードにはかなりの時間と労力が必要です。これらの環境の場合、特に実動システム上では、フル・リフレッシュはコストが高くつく停止の原因となる可能性があります。Q キャプチャー・プログラムを最初に始動する場合以外は、コールド・スタート・オプションは最後の手段として使用してください。

Q キャプチャー・プログラムの予期しないコールド・スタートを防止する方法

- コールド・スタート・モードが、**IBMQREP_CAPPARMS** 表で指定されていないことを確認する。 **startmode** パラメーターの値が *cold* でないようにする必要があります。
- 可能なら常に **warmsa** スタート・モードではなく、**warmns** または **warmsi** スタート・モードを指定して Q キャプチャー・プログラムを再始動します。この場合、ウォーム・スタート情報が使用可能でなければ、Q キャプチャー・プログラムはコールド・スタートしません。
- レプリケーション・アラート・モニターで、Q キャプチャー・プログラムの状況をモニターします。たとえば、モニターが Q キャプチャー・プログラムがダウンしたことを検出する場合、**QCAPTURE_STATUS** アラート条件を使用して E メールを送信できます。Q キャプチャー・プログラムがダウンすると、プログラムのログ・レコードの読み取りが滞る可能性があります。プログラムが長い時間ダウンしていると、Q キャプチャー・プログラムではその時点でログ・レコードの読み取りが滞っているので、ソースおよびターゲット表を同期化するために環境をコールド・スタートすることが必要になる場合があります。
- システム上に十分な **DB2** ログ・データが保存され、このデータが Q レプリケーションおよびイベント発行に使用可能であることを確認する。ログ・ファイルが Q キャプチャー・プログラムに対して使用可能でない場合、プログラムはソース表に加えられる変更のキャプチャーを継続できず、コールド・スタートを必要とすることがあります。

関連概念:

- 173 ページの『さまざまなタイプの Q レプリケーションのロード・オプション』

関連タスク:

- 229 ページの『Q キャプチャー・プログラムの開始』

Q キャプチャー・プログラムのパラメーター

Q キャプチャー・プログラムのパラメーター -- 概要

Q キャプチャー・プログラムの稼働パラメーターは、とりわけプログラムの開始方法、使用するメモリーの量、接続するキュー・マネージャー、およびキューにメッセージをコミットする頻度などを管理します。以下のトピックでは、デフォルトの稼働パラメーターをリストし、各パラメーターの詳細とデフォルト設定を変更する理由について説明します。

- 234 ページの『Q キャプチャー稼働パラメーターのデフォルト値』

- 235 ページの『Q キャプチャー・パラメーターの説明』

関連概念:

- 229 ページの『Q キャプチャー・プログラムの操作 -- 概要』

Q キャプチャー稼働パラメーターのデフォルト値

レプリケーション・センターを使用してコントロール表を作成すると、Q キャプチャー・プログラムの稼働パラメーターのデフォルト値を含む単一行がある IBMQREP_CAPPARMS 表が作成されます。表 13 にこれらの値を示します。

デフォルトのパラメーター値をレプリケーション環境に合わせて変更するには、IBMQREP_CAPPARMS 表を更新するか、Q キャプチャー・プログラムの開始時またはプログラムの実行中に、保管された値を一時的にオーバーライドします。

注: コントロール表の作成時に、**adminq** (管理キュー)、**qmgr** (キュー・マネージャー)、および **restartq** (再始動キュー) パラメーターの値を指定します。

表 13. Q キャプチャー稼働パラメーターのデフォルト値

稼働パラメーター	デフォルト値	IBMQREP_CAPPARMS 表の列名
add_partition	N ¹	該当せず
adminq	なし ²	ADMINQ
autostop	N ¹	AUTOSTOP
capture_path	Q キャプチャーが開始されたディレクトリー ³	CAPTURE_PATH
capture_schema	ASN ⁴	該当せず
capture_server	DB2DBDFT ⁵	該当せず
commit_interval	500 ⁶	COMMIT_INTERVAL
logreuse	N ¹	LOGREUSE
logstdout	N ¹	LOGSTDOUT
memory_limit	32 ⁷ ; 16 ⁷ (z/OS の場合)	MEMORY_LIMIT
monitor_interval	300 ⁸	MONITOR_INTERVAL
monitor_limit	10080 ⁹	MONITOR_LIMIT
prune_interval	300 ⁸	PRUNE_INTERVAL
qmgr	なし ²	QMGR
restartq	なし ²	RESTARTQ
signal_limit	10080 ⁹	SIGNAL_LIMIT
sleep_interval	5000 ⁶	SLEEP_INTERVAL
startmode	warmsi ¹⁰	STARTMODE
term	Y ¹¹	TERM
trace_limit	10080 ⁹	TRACE_LIMIT

表 13. Q キャプチャー稼働パラメーターのデフォルト値 (続き)

稼働パラメーター	デフォルト値	IBMQREP_CAPPARMS 表の列名
----------	--------	-----------------------

注:

1. No (いいえ)。
2. Q キャプチャー・コントロール表の作成時には、この値を指定しなければならない。これはコマンド・パラメーターではありません。Q キャプチャー・プログラムは、この値を IBMQREP_CAPPARMS 表から読み取ります。
3. Q キャプチャー・プログラムが Windows サービスとして開始する場合は、**capture_path** は %sqllib%bin。Q キャプチャーを z/OS 上の JCL を使用して開始する場合は、デフォルトは、開始済みタスクまたはジョブと関連したユーザー ID です。
4. デフォルト・スキーマは変更できない。別のスキーマで Q キャプチャー・プログラムを使用するには、Q キャプチャー・プログラムの開始時に **capture_schema** 開始パラメーターを指定する。
5. Linux、UNIX、および Windows の場合は、Q キャプチャー・プログラム・サーバーは DB2DBDFT 環境変数の値 (指定されている場合)。z/OS の場合は、デフォルトの Q キャプチャー・サーバーはありません。
6. ミリ秒。
7. MB。
8. 秒。
9. 分。
10. Q キャプチャー・プログラムはウォーム・スタートする。これがこのプログラムの初回の開始である場合にのみ、コールド・スタートに切り替わります。
11. Yes (はい)。

関連概念:

- 248 ページの『Q キャプチャー・パラメーターの変更 -- 概要』
- 233 ページの『Q キャプチャー・プログラムのパラメーター -- 概要』
- 15 ページの『Q キャプチャー・プログラム』

関連資料:

- 235 ページの『Q キャプチャー・パラメーターの説明』
- 428 ページの『IBMQREP_CAPPARMS 表』

Q キャプチャー・パラメーターの説明

以下のセクションでは、Q キャプチャー・プログラムの稼働パラメーターについて説明し、必要に基づいてデフォルト値を変更する理由についても取り上げます。

Q キャプチャー・プログラムは、開始時に IBMQREP_CAPPARMS 表からパラメーターを読み取ります。プログラムの開始時およびプログラムの実行中に、一時ランタイム値を指定できるパラメーターもあります。詳細については、以下の各パラメーターの説明を参照してください。

- 236 ページの『add_partition (Linux、UNIX、Windows)』
- 236 ページの『adminq』
- 237 ページの『autostop』

- 237 ページの『capture_path』
- 238 ページの『capture_schema』
- 238 ページの『capture_server』
- 238 ページの『commit_interval』
- 240 ページの『logreuse』
- 241 ページの『logstdout』
- 241 ページの『memory_limit』
- 242 ページの『monitor_interval』
- 243 ページの『monitor_limit』
- 243 ページの『prune_interval』
- 244 ページの『qmgr』
- 244 ページの『restartq』
- 245 ページの『signal_limit』
- 245 ページの『sleep_interval』
- 246 ページの『startmode』
- 247 ページの『term』
- 248 ページの『trace_limit』

add_partition (Linux、UNIX、Windows)

デフォルト: `add_partition=N`

add_partition パラメーターは、Q キャプチャー・プログラムが最後に再始動されてから追加されたパーティションの DB2 リカバリー・ログの読み取りを、Q キャプチャー・プログラムが開始するかどうかを指定します。

Q キャプチャー・プログラムがログを読み取るようにするには、Q キャプチャー・プログラムの開始時に `add_partition=Y` を指定します。各新規パーティションにおいて、ウォーム・スタート・モードで Q キャプチャー・プログラムが開始されると、Q キャプチャーは、ログ・ファイルを、DB2 インスタンスに対する最初のデータベース CONNECT ステートメントの発行後に DB2 が使用した最初のログ・シーケンス番号 (LSN) から読み取ります。

adminq

adminq パラメーターは、Q キャプチャー管理キューを定義します。これは、Q キャプチャー・プログラムが、Q アプライ・プログラム、ユーザー・アプリケーション、または WebSphere MQ メッセージ・チャンネル・エージェントから、コントロール・メッセージを受信するために使用するキューです。これは、Q キャプチャー・プログラムおよびキュー・マネージャーが実行されているのと同じシステム上で定義されるローカル・キューです。管理キューの名前は Q キャプチャー・コントロール表の作成時に指定する必要があり、Q キャプチャー・プログラムは、このキューに接続できなければならず、接続できない場合は実行されません。

管理キューは永続キューでなければなりません。つまり、ロギングが使用可能になっていなければなりません。このキューには、循環ロギングをお勧めします。レプ

リケーションまたは発行環境に基づいて、最大項目数（許可されるメッセージ数）などの、このキューの他の属性を選択します。

Q キャプチャー・コントロール表の作成時に、管理キューの名前を指定します。この値は `IBMQREP_CAPPARMS` コントロール表に保管されます。表を更新することによって、この値を変更できます。Q キャプチャー・プログラムの開始時またはプログラムの実行中に、値を変更することはできません。

autostop

デフォルト: `autostop=N`

autostop パラメーターは、アクティブ DB2 ログの最後に達した場合に Q キャプチャー・プログラムが終了するかどうかを制御します。デフォルトでは、Q キャプチャー・プログラムは、ログの最後に達しても終了しません。

一般に、Q キャプチャー・プログラムは、ソース・データベースがアクティブの間は連続プロセスとして実行されるため、ほとんどの場合デフォルト (`autostop=N`) のままで使用します。`autostop=Y` を設定するのは、Q キャプチャー・プログラムを一定のインターバルで実行するシナリオ（接続頻度の低いシステムを同期させる場合など）や、テスト・シナリオの場合だけにしてください。

`autostop=Y` に設定すると、キャプチャー・プログラムはすべての適格トランザクションを検索し、ログの最後に達した時点で停止します。それ以外のトランザクションも検索する場合は、再度 Q キャプチャー・プログラムを開始する必要があります。

autostop パラメーターは、Q キャプチャー・プログラムの開始時またはプログラムの実行中に設定できます。`IBMQREP_CAPPARMS` 表に保管されているパラメーターの値を変更することもできます。

capture_path

capture_path パラメーターは、Q キャプチャー・プログラムが作業ファイルとログ・ファイルを保管するディレクトリーを指定します。デフォルトのパスはプログラムを開始するディレクトリーです。このパスは変更できます。

Windows

Q キャプチャー・プログラムを Windows サービスとして開始する場合、デフォルトではプログラムは `¥sqllib¥bin` ディレクトリーで開始します。

z/OS Q キャプチャー・プログラムは POSIX アプリケーションであるため、デフォルトのパスは、次に示す方法のどちらでこのプログラムを開始するかによって決まります。

- Q キャプチャー・プログラムを USS コマンド行プロンプトから開始する場合。パスは、このプログラムが開始されたディレクトリーです。
- Q キャプチャー・プログラムを開始済みのタスクまたは JCL を使用して開始する場合。デフォルト・パスは、その開始済みタスクまたはジョブと関連したユーザー ID のホーム・ディレクトリーです。

パスを変更するには、パス名または //QCAPV8 などの上位修飾子 (HLQ) のいずれかを指定できます。HLQ を使用すると、z/OS 順次データ・セット・ファイル名のファイル命名規則に準拠する順次ファイルが作成されません。

Q キャプチャー・プログラムの開始時に **capture_path** パラメーターを設定するか、IBMQREP_CAPPARMS 表に保管されているパラメーターの値を変更できます。このパラメーターは、Q キャプチャー・プログラムの実行中は変更できません。

capture_schema

デフォルト: **capture_schema=ASN**

capture_schema パラメーターを使用すれば、Q キャプチャー・サーバー上の Q キャプチャー・プログラムの複数のインスタンスを区別できます。

スキーマによって、1 つの Q キャプチャー・プログラムとそのコントロール表を識別できます。1 つのサーバー上で、スキーマが同じである 2 つの Q キャプチャー・プログラムを実行することはできません。

1 つの Q キャプチャー・サーバー上に複数の Q キャプチャー・プログラムのコピーを作成すると、データ・フローを複数の並行ストリームに分割することによってスループットを改善したり、同じソースを使用してさまざまなレプリケーション要件を満たすことができます。

capture_server

デフォルト (Linux、UNIX、Windows): **capture_server= DB2DBDFT** 環境変数の値 (設定されている場合)

デフォルト (z/OS): **capture_server= None**

capture_server パラメーターは、Q キャプチャー・プログラムが実行され、そのコントロール表が保管されているデータベースまたはサブシステムを示します。コントロール表には、ソース、Q サブスクリプション、WebSphere MQ キュー、およびユーザー設定についての情報が含まれています。Q キャプチャー・プログラムは DB2 ログを読み取るため、Q キャプチャー・プログラムがソース・データベースまたはサブシステムで実行されている必要があります。

z/OS: データ共有の場合、グループ・アタッチ名を使用しないでください。その代わりに、メンバー・サブシステム名を指定してください。

commit_interval

デフォルト: **commit_interval=500** ミリ秒

commit_interval パラメーターは、Q キャプチャー・プログラムが WebSphere MQ に対してトランザクションをコミットする頻度をミリ秒単位で指定します。デフォルトでは、Q キャプチャー・プログラムがコミット間で待機する時間は 500 ミリ秒 (0.5 秒) です。インターバルごとに、Q キャプチャー・プログラムは MQCMIT 呼び出しを発行します。これによって、送信キューに置かれたメッセージを Q アプリケーションまたは他のユーザー・アプリケーションで使用可能にするよう、WebSphere MQ キュー・マネージャーにシグナルが出されます。

MQCMIT 呼び出しにまとめられたすべての DB2 トランザクションは、WebSphere MQ 作業単位、またはトランザクションと見なされます。一般に、WebSphere MQ トランザクションごとに、複数の DB2 トランザクションが含まれています。大規模な DB2 トランザクションの場合は、コミット・インターバルに達しても Q キャプチャー・プログラムは MQCMIT 呼び出しを発行しません。大規模な DB2 トランザクション全体が送信キューに書き込まれない限り、Q キャプチャー・プログラムはコミットしません。

Q キャプチャー・プログラムによって読み取られたコミット済み DB2 トランザクションの数が 128 に達すると、**commit_interval** の設定に関係なく、プログラムは MQCMIT 呼び出しを発行します。

最適なコミット・インターバルを見つけるには、待ち時間 (トランザクションがソース・データベースでコミットされてから、ターゲット・データベースでコミットされるまでの遅延時間) と、コミット・プロセスに関連した CPU オーバーヘッドとの妥協点を探する必要があります。

- **待ち時間を削減するには、コミット・インターバルを短縮する。**

トランザクションは、より短い遅延で実行されます。これは、ソース・データベースに対する変更を使用して、イベントを起動する場合は特に重要です。コミット・インターバルを短縮することによって待ち時間を削減できるかどうかを判断するには、すべてのトランザクションを処理した後の Q キャプチャー・プログラムのアイドル時間が長いかどうかを判別します (IBMQREP_CAPMON コントロール表の CAPTURE_IDLE 値をチェックする)。また、コミット・インターバルごとに発行されるトランザクションの数が多い場合は (IBMQREP_CAPQMON 表の TRANS_PUBLISHED 値をチェックする)、Q キャプチャー・プログラムが WebSphere MQ に対して一度にコミットするトランザクションの数を少なくすることができます (『コミット・インターバルごとに発行されるトランザクション数を判別するための手順』を参照)。

- **CPU オーバーヘッドを削減するには、コミット・インターバルを延長する。**

コミット・インターバルを長くすると、WebSphere MQ トランザクションごとに、可能な限り多くの DB2 トランザクションを送信できます。また、コミット・インターバルを長くすると、メッセージのロギングに起因する I/O も削減できます。コミット・インターバルを長くする場合は、Q キャプチャー・プログラムに割り振られるメモリ、送信キューの最大項目数 (メッセージ数)、およびキュー・マネージャーの最大非コミット・メッセージ数 (MAXUMSGS) 属性により制限される場合があります。コミット間で Q キャプチャー・プログラムが待機する時間が長くなると、一部のトランザクションの発行が遅れる可能性があり、待ち時間が長くなる場合があります。

コミット・インターバルごとに発行されるトランザクション数を判別するための手順: コミット・インターバルごとに発行されるトランザクションの数を計算するには、次の公式を使用します。

$$\text{TRANS_PUBLISHED} / (\text{MONITOR_INTERVAL} / \text{COMMIT_INTERVAL})$$

以下のステップを実行します。

1. モニター・インターバル (行を IBMQREP_CAPQMON 表に挿入する秒数の間隔) を判別する。
2. この数をコミット・インターバルで除算する。

コミット・インターバルおよびモニター・インターバルを表示するには、レプリケーション・センターの以下のウィンドウを使用できます。

- 「パラメーターの変更 - Q キャプチャー・プログラムの実行中 (Change Parameters - Running Q Capture Program)」。
- 「パラメーターの変更 - 保管」(Q キャプチャー・プログラムが停止している場合)。

このウィンドウをオープンするには、チェックしたい Q キャプチャー・プログラムが含まれている Q キャプチャー・サーバーを右マウス・ボタンでクリックして、適切なメニュー項目を選択します。詳細については、オンライン・ヘルプを参照してください。このステップでは、一方または両方の値を秒に変換する必要があります。この結果はステップ 4 で使用するため、結果を保管しておいてください。

3. コミット・インターバルごとに発行するトランザクションは、1 つの送信キューについてか、この Q キャプチャー・プログラムが使用するすべての送信キューについてかを決定する。
 - 1 つの送信キューについての数を計算したい場合は、IBMQREP_CAPQMON 表の中でキューの最新の TRANS_PUBLISHED の値を探す。
 - この Q キャプチャー・プログラムについての数を計算したい場合は、この Q キャプチャー・プログラムが使用するすべての送信キューの TRANS_PUBLISHED の数を合計する。

1 つの送信キューまたはすべての送信キューについてこの値をチェックするには、レプリケーション・センターの「Q キャプチャー・スループット」ウィンドウを使用できます。このウィンドウをオープンするには、チェックしたい Q キャプチャー・プログラムが含まれている Q キャプチャー・サーバーを右マウス・ボタンでクリックして、「レポート」→「Q キャプチャー・スループット」を選択します。詳細については、オンライン・ヘルプを参照してください。

4. TRANS_PUBLISHED をステップ 2 の結果で除算する。結果は、Q キャプチャー・プログラムがコミット・インターバルごとに発行したトランザクションの数です。

commit_interval パラメーターは、Q キャプチャー・プログラムの開始時またはプログラムの実行中に設定できます。IBMQREP_CAPPARMS 表に保管されているパラメーターの値を変更することもできます。

logreuse

デフォルト: logreuse=N

各 Q キャプチャー・プログラムは、ワーク・ヒストリー (開始時刻および停止時刻、パラメーターの変更、エラー、整理、および DB2 ログの読み取りを中止したポイントなど) をトラッキングするログ・ファイルを保持します。

デフォルトでは、Q キャプチャー・プログラムは、プログラムが再始動すると既存のログ・ファイルに追加します。このデフォルトにより、プログラムの動作の履歴

を保持できます。この履歴が必要ないか、スペースを節約する場合には、**logreuse=Y** と設定します。Q キャプチャー・プログラムは、開始時にログ・ファイルをクリアし、ブランク・ファイルに書き込みます。

ログは、Q キャプチャー・プログラムを開始したディレクトリー (デフォルト)、または **capture_path** パラメーターを使用して設定した別の場所に保管されます。

Linux、UNIX、Windows: ログ・ファイル名は、*db2instance.capture_server.capture_schema.QCAP.log* です。たとえば、*DB2.SAMPLE.ASN.QCAP.log*。

z/OS: ログ・ファイルには DB2 インスタンス名は含まれません。たとえば、*SAMPLE.ASN.CAP.log*。また、高位修飾子 (HLQ) を使用するために、**capture_path** をスラッシュ (/) を使用して指定する場合は、z/OS 順次データ・セット・ファイルのファイル命名規則が適用され、**capture_schema** は 8 文字に切り捨てられます。

logreuse パラメーターは、Q キャプチャー・プログラムの開始時またはプログラムの実行中に設定できます。IBMQREP_CAPPARMS 表に保管されているパラメーターの値を変更することもできます。

logstdout

デフォルト: **logstdout=N**

デフォルトでは、Q キャプチャー・プログラムは、ワーク・ヒストリーをログにのみ書き込みます。プログラムの履歴をログの他に標準出力 (stdout) でも表示したい場合は、**logstdout** パラメーターを変更できます。

エラー・メッセージおよび一部のログ・メッセージ (初期化、停止、サブスクリプションのアクティブ化、およびサブスクリプションの非アクティブ化) は、このパラメーターの設定に関係なく、標準出力とログ・ファイルの両方に送られます。

logstdout パラメーターは、**asnlqcap** コマンドを使用した Q キャプチャー・プログラムの開始時、または **asnlqccmd** コマンドを使用したプログラムの実行中に設定できます。IBMQREP_CAPPARMS 表を更新して、保管されているパラメーターの値を変更することもできます。レプリケーション・センターを使用して Q キャプチャー・プログラムを開始する場合は、このパラメーターは適用されません。

memory_limit

デフォルト: **memory_limit=32 MB**

memory_limit パラメーターは、Q キャプチャー・プログラムが DB2 トランザクションをメモリー内で構築する場合に使用できるメモリーの量を指定します。デフォルトでは、Q キャプチャー・プログラムの最大使用量は 32 MB です。このパラメーターで割り振ったメモリー量が使用されると、Q キャプチャー・プログラムは、メモリー内のトランザクションを **capture_path** ディレクトリーに置かれているファイルにスピルします。z/OS では、Q キャプチャー・プログラムは VIO または CAPSPILL DD カードで指定されたファイルにスピルします。

必要に応じて、次のようにメモリー限度を調整することができます。

- **Q キャプチャー・プログラムのパフォーマンスを改善するには、メモリー限度を大きくする。**

スループットの向上を目標とする場合は、可能であればメモリー限度を最大化します。

- **システム・リソースを節約するには、メモリー限度を下げる。**

メモリー限度を減らすと、他のシステム操作との競合が減ります。ただし、メモリー限度の設定を低くしすぎると、スピル・ファイル用のスペースがシステム上で増え、さらに I/O が要求されるためにシステムの速度が低下する可能性があります。

IBMQREP_CAPMON 表の中のデータを使用して、ニーズに合った最適のメモリー限度を見付けることができます。たとえば、トランザクションをログから再構成するために Q キャプチャー・プログラムが使用しているメモリー量を確認するには、CURRENT_MEMORY の値をチェックします。または、メモリー限度を超過したときに、Q キャプチャー・プログラムがファイルにスピルしたトランザクションの数を調べるには、TRANS_SPILLED の値をチェックします。これらの値をチェックするには、レプリケーション・センターの「Q キャプチャー・スループット」ウィンドウを使用します。詳細については、レプリケーション・センターのオンライン・ヘルプを参照してください。

z/OS: JCL の REGION パラメーターも、Q キャプチャー・プログラムによるメモリー使用量を制限します。Q キャプチャー・プログラムは領域サイズを認識しないため、プログラムは memory_limit に達する前に、このオペレーティング・システム限度に達する場合があります。この状態を回避するには、memory_limit を領域サイズの半分程度に設定します。

memory_limit パラメーターは、Q キャプチャー・プログラムの開始時またはプログラムの実行中に設定できます。IBMQREP_CAPPARMS 表に保管されているパラメーターの値を変更することもできます。

monitor_interval

デフォルト: monitor_interval=300 秒 (5 分)

monitor_interval パラメーターは、2 つのコントロール表にパフォーマンス統計を挿入する頻度を Q キャプチャー・プログラムに通知します。IBMQREP_CAPMON 表は Q キャプチャー・プログラムの全体的なパフォーマンス統計を示し、IBMQREP_CAPQMON 表は送信キューごとの Q キャプチャー・プログラム統計を示します。

デフォルトでは、行はこれらの表に 300 秒 (5 分) ごとに挿入されます。一般に、Q キャプチャー・プログラムは、より短いインターバル (デフォルトのコミット・インターバルは 0.5 秒) で WebSphere MQ トランザクションをコミットします。したがって、モニター・インターバルおよびコミット・インターバルの出荷時デフォルトを使用する場合は、モニター表には一度に合計 600 のコミットが挿入されます。Q キャプチャー・プログラムのアクティビティーをより細分化されたレベルでモニターしたい場合は、コミット・インターバルに近いモニター・インターバルを使用してください。

monitor_interval パラメーターは、Q キャプチャー・プログラムの開始時またはプログラムの実行中に設定できます。 IBMQREP_CAPPARMS 表に保管されているパラメーターの値を変更することもできます。

monitor_limit

デフォルト: **monitor_limit**=10080 分 (7 日)

monitor_limit パラメーターは、整理の対象となるまでに、行が IBMQREP_CAPMON 表および IBMQREP_CAPQMON 表の中にとどまる時間を指定します。

デフォルトでは、10080 分 (7 日) より古い行が整理されます。

IBMQREP_CAPMON および IBMQREP_CAPQMON 表には、Q キャプチャー・プログラムのアクティビティーについての統計が含まれています。行はモニター・インターバルごとに挿入されます。必要に応じて、次のようにモニター限度を調整することができます。

- 統計を保持するには、モニター限度を大きくする。

Q キャプチャー・プログラムのアクティビティーを 1 週間より長く保持したい場合は、モニター限度を高く設定します。

- 統計を頻繁に確認する場合は、モニター限度を下げる。

Q キャプチャー・プログラムのアクティビティーを定期的にモニターする場合は、統計を 1 週間保持する必要はおそらくないため、モニター限度を低く設定できます。これで、整理がより頻繁に行われるようになります。

monitor_limit パラメーターは、Q キャプチャー・プログラムの開始時またはプログラムの実行中に設定できます。 IBMQREP_CAPPARMS 表に保管されているパラメーターの値を変更することもできます。

prune_interval

デフォルト: **prune_interval**=300 秒 (5 分)

prune_interval パラメーターは、Q キャプチャー・プログラムが、整理の対象となる行を IBMQREP_CAPMON、IBMQREP_CAPQMON、IBMQREP_SIGNAL、および IBMQREP_CAPTRACE 表から探す頻度を決定します。デフォルトでは、Q キャプチャー・プログラムは整理の対象となる行を 300 秒 (5 分) ごとに探します。

整理の頻度は、これらのコントロール表が増大する速度と、コントロール表の使用目的によって決まります。

- モニター表を管理するには、整理インターバルを短くする。

モニター・インターバルが短いため IBMQREP_CAPMON および IBMQREP_CAPQMON 表の増大速度が速い場合は、整理インターバルを短くしなければならない場合があります。これらのコントロール表および他のコントロール表の整理頻度が低い場合は、表スペース限度を超過する場合があります、Q キャプチャー・プログラムが強制的に停止します。ただし、表の整理があまりに頻繁にまたはピーク時に実行される場合は、そのような整理によって同じシステム上で実行されているアプリケーション・プログラムに支障が生じる可能性があります。

- レコードを保持する場合は、整理インターバルを長くする。

IBMQREP_CAPTRACE および他の表の整理頻度を低くして、Q キャプチャー・プログラムのパフォーマンスの履歴を長く保持することができます。

整理インターバルは、データが整理の対象となるまでの時間を決定する

trace_limit、**monitor_limit**、および **signal_limit** パラメーターと連係して機能します。たとえば、**prune_interval** が 300 秒、**trace_limit** が 10080 秒である場合は、Q キャプチャー・プログラムは 300 秒ごとに整理を試行します。Q キャプチャー・プログラムが IBMQREP_CAPTRACE 表で 10080 分 (7 日) より古い行を検出すると、それらを整理します。

prune_interval パラメーターは、Q キャプチャー・プログラムの開始時またはプログラムの実行中に設定できます。IBMQREP_CAPPARMS 表に保管されているパラメーターの値を変更することもできます。

qmgr

qmgr パラメーターは、Q キャプチャー・プログラムが使用する WebSphere MQ キュー・マネージャーの名前を指定します。キュー・マネージャーのジョブは、Q キャプチャー・プログラム用のキューおよびメッセージを管理することです。これは、Q キャプチャー・プログラムと同じシステム上で稼働している必要があります。キュー・マネージャーへのクライアント接続は WebSphere MQ でサポートされますが、この構成は Q キャプチャーではサポートされません。

キュー・マネージャーは、Q キャプチャー・プログラムがデータ・メッセージおよび通知メッセージを送信し、コントロール・メッセージを受信するために使用するキューを所有しています。Q レプリケーションおよびイベント発行プログラムと WebSphere MQ との間のすべての通信は、キュー・マネージャーを介して行われます。

Q キャプチャー・コントロール表の作成時に、キュー・マネージャーの名前を指定します。この値は IBMQREP_CAPPARMS コントロール表に保管されます。表を更新することによって、この値を変更できます。Q キャプチャー・プログラムの開始時またはプログラムの実行中に、値を変更することはできません。

restartq

restartq パラメーターで、Q キャプチャー・プログラムが使用する再始動キューを指定します。再始動キューには、Q キャプチャー・プログラムの再始動後に、DB2 ログの読み取りを開始する場所を Q キャプチャー・プログラムに通知する単一メッセージが含まれています。これは、Q キャプチャー・プログラムおよびキュー・マネージャーが実行されているのと同じシステム上で定義されるローカル・キューです。Q キャプチャー・コントロール表の作成時にこのキューの名前を指定する必要があります。Q キャプチャー・プログラムは、稼働するためにはこのキューに接続できなければなりません。

再始動キューは永続キューでなければなりません。つまり、ロギングが使用可能になっていなければなりません。このキューには、循環ロギングをお勧めします。レプリケーションまたは発行環境に基づいて、このキューの他の属性を選択します。

Q キャプチャー・コントロール表の作成時に、再始動キューの名前を指定します。この値は IBMQREP_CAPPARMS コントロール表に保管されます。表を更新することによって、この値を変更できます。Q キャプチャー・プログラムの開始時またはプログラムの実行中に、値を変更することはできません。

signal_limit

デフォルト: **signal_limit**=10080 分 (7 日)

signal_limit パラメーターは、整理の対象となる前に行が IBMQREP_SIGNAL 表にとどまる期間を指定します。

デフォルトでは、Q キャプチャー・プログラムによって、10080 分 (7 日) より古い行が整理インターバルごとに整理されます。

IBMQREP_SIGNAL 表には、ユーザーまたはユーザー・アプリケーションによって挿入されたシグナルが含まれています。また、Q アプライ・プログラムまたはユーザー・アプリケーションからコントロール・メッセージを受信した後に、Q キャプチャー・プログラムによって挿入された対応するシグナルも含まれています。Q キャプチャー・プログラムは、IBMQREP_SIGNAL 表に挿入するログ・レコードを読み取るときにシグナルを確認します。

これらのシグナルは、Q キャプチャー・プログラムに、実行の停止、Q サブスクリプションまたは XML 発行の非アクティブ化、ログ内の DB2 トランザクションの無視、または送信キューの無効化を通知します。さらに、LOADDONE シグナルは、ターゲット表がロードされたことを Q キャプチャー・プログラムに通知します。

シグナル限度は、環境に応じて次のように調整できます。

- **IBMQREP_SIGNAL** 表のサイズを管理するには、限度を短くする。

双方向 Q レプリケーションの場合は、Q アプライ・プログラムは受信するすべてのトランザクションについて、シグナルを IBMQREP_SIGNAL 表に挿入して適用し、Q キャプチャー・プログラムがトランザクションを再キャプチャーすることがないようにします。双方向 Q サブスクリプションの数が多い場合は、表が大きくなる可能性があるため、デフォルトのシグナル限度を下げて整理の頻度を高めることができます。

- レコードを保持する目的で表を使用する場合は、限度を長くします。

signal_limit パラメーターは、Q キャプチャー・プログラムの開始時またはプログラムの実行中に設定できます。IBMQREP_CAPPARMS 表に保管されているパラメーターの値を変更することもできます。

sleep_interval

デフォルト: **sleep_interval**=5000 ミリ秒 (5 秒)

sleep_interval パラメーターは、Q キャプチャー・プログラムがアクティブ・ログの最後に達し、メモリーに残っているトランザクションをアSEMBルした後に待機するミリ秒数を指定します。

デフォルトでは、Q キャプチャー・プログラムは 5000 ミリ秒 (5 秒) 間スリープします。このインターバルが経過すると、プログラムはログの読み取りを再開します。スリープ・インターバルは、環境に基づいて次のように調整できます。

- 待ち時間を削減するには、スリープ・インターバルを下げる。

スリープ・インターバルを小さくすると、待ち時間 (トランザクションがソースからターゲットに移動するのにかかる時間) の短縮、アイドル時間の削減、および大ボリューム・トランザクション環境でのスループットの増大により、パフォーマンスを改善できます。

- リソースを節約するには、スリープ・インターバルを大きくする。

スリープ・インターバルの値を大きくすると、ソース・データベースのトラフィックが少ない環境、またはターゲットを頻繁に更新する必要のない環境では、潜在的に CPU 使用量が節約されます。

sleep_interval パラメーターは、Q キャプチャー・プログラムの開始時またはプログラムの実行中に設定できます。IBMQREP_CAPPARMS 表に保管されているパラメーターの値を変更することもできます。

startmode

デフォルト: **startmode=warmsi**

startmode パラメーターは、Q キャプチャー・プログラムの開始時のモードを指定します。プログラムは、ウォーム・モードまたはコールド・モードで開始します。ウォーム・スタートを使用すると、Q キャプチャー・プログラムは、最後の実行後に中止した場所で変更のキャプチャーを継続します (ウォーム・スタートには 3 つのタイプがあります)。コールド・スタートを選択すると、プログラムはログの最後から読み取りを開始します。環境に応じて、以下の 4 つの開始モードの中から選択します。

cold Q キャプチャー・プログラムは、再始動キューおよび管理キューをクリアし、N (新規) または A (アクティブ) 状態になっているすべての Q サブスクリプションまたは XML 発行の処理を開始します。コールド・スタートの場合、Q キャプチャー・プログラムは DB2 リカバリー・ログの読み取りを、その末尾から開始します。

通常、コールド・スタートは、Q キャプチャー・プログラムを初めて開始する場合にのみ使用します。推奨される開始モードは **warmsi** です。Q キャプチャー・プログラムを初めて開始するのではない場合はコールド・スタートを使用できますが、最後の再始動点ではなくアクティブ・ログの最後から変更のキャプチャーを開始したい場合もあります。コールド・スタートを使用してターゲットのフル・リフレッシュ (新規ロード) を強制することはできません。フル・リフレッシュを強制する唯一の方法は、非アクティブにしてから、ロード・フェーズを持つ Q サブスクリプションまたは XML 発行をアクティブにするという方法です。

重要: 不必要なコールド・スタートを回避するには、この開始モードを IBMQREP_CAPPARMS 表に指定しないでください。

warmsi (ウォーム・スタート、初回はコールド・スタートに切り替える)

Q キャプチャー・プログラムは、初めて開始する場合でない限り、読み取

りを中止した点からログの読み取りを開始します。Q キャプチャー・プログラムは、プログラムの初回始動時にコールド・スタートに切り替わりません。warmnsi 始動モードでは、初めて開始する場合にのみ、Q キャプチャー・プログラムがコールド・スタートします。

warmns (ウォーム・スタート、決してコールド・スタートに切り替えない)

Q キャプチャー・プログラムは、読み取りを中止した点からログの読み取りを開始します。ウォーム・スタートできない場合、コールド・スタートに切り替わりません。この開始モードを使用して、Q キャプチャー・プログラムが予期せずにコールド・スタートすることを防いでください。この開始モードを使用すると、ウォーム・スタートを妨げている問題 (データベースや表スペースが使用不能など) を修復できます。warmns を使用すると、Q キャプチャー・プログラムがウォーム・スタートできない場合は、シャットダウンし、すべての表がそのまま残ります。

warmsa (可能な場合はウォーム・スタート、可能でない場合はコールド・スタート)

ウォーム・スタート情報が使用可能な場合、Q キャプチャー・プログラムは、読み取りを中止した点からログの読み取りを開始します。Q キャプチャー・プログラムは、ウォーム・スタートできない場合、コールド・スタートに切り替わりません。

ウォーム・スタート中、Q キャプチャー・プログラムは、I (非アクティブ) 状態になっていない Q サブスクリプションまたは XML 発行のみをロードします。

Q キャプチャー・プログラムの開始時に **startmode** パラメーターを設定するか、IBMQREP_CAPPARMS 表に保管されているパラメーターの値を変更できます。このパラメーターは、Q キャプチャー・プログラムの実行中は変更できません。

term

デフォルト: **term=Y**

term パラメーターは、DB2 が静止した場合に Q キャプチャー・プログラムが実行し続けるかどうかを制御します。

デフォルトでは、DB2 が静止した場合に Q キャプチャー・プログラムは実行を終了します。DB2 が静止モードになり、すべてのアプリケーション (Q キャプチャー・プログラムを含む) が強制的に切断される場合に、Q キャプチャー・プログラムが実行し続けるようにする場合は、デフォルトを変更できます。この場合、DB2 が静止モードから復帰すると、プログラムを再始動しなくても、Q キャプチャー・プログラムは、ログの最後の再始動点から変更のキャプチャーを再開します。

DB2 のシャットダウンに対する Q キャプチャー・プログラムの反応は変更できません。term パラメーターの設定に関係なく、DB2 がシャットダウンすると、Q キャプチャー・プログラムは停止します。DB2 が再び開始すると、最後の再始動点からのキャプチャーを開始するよう、Q キャプチャー・プログラムを warmns モードで開始する必要があります。

DB2 の静止またはシャットダウンが原因で、Q キャプチャー・プログラムが変更のキャプチャーを強制的に停止させられるかどうかに関係なく、データベースへの再接続後にプログラムが開始するログ点は、Q キャプチャー・プログラムが切断した時点と必ずしも一致している必要はありません。これは、Q キャプチャー・プログラ

ラムが、強制的に切断されたときにメモリー内にあったトランザクションを再キャプチャーする必要があるためです。再始動キューの中の最後の再始動メッセージが、再開する場所をプログラムに通知します。

重要: デフォルトを変更して Q キャプチャー・プログラムが実行を停止しないようにした場合で、DB2 が制限付きアクセス (たとえば、ACCESS MAINT) で再始動する場合は、Q キャプチャー・プログラムは接続できず、結果として停止します。

term パラメーターは、Q キャプチャー・プログラムの開始時またはプログラムの実行中に設定できます。 IBMQREP_CAPPARMS 表に保管されているパラメーターの値を変更することもできます。

trace_limit

デフォルト: **trace_limit=10080** 分 (7 日)

trace_limit パラメーターは、整理の対象となる前に行が IBMQREP_CAPTRACE 表にとどまる期間を指定します。

Q キャプチャー・プログラムは、すべての通知、警告、およびエラー・メッセージを IBMQREP_CAPTRACE 表に挿入します。デフォルトでは、10080 分 (7 日) より古い行が整理インターバルごとに整理されます。監査情報に関するニーズに合わせてトレース限度を変更してください。

trace_limit パラメーターは、Q キャプチャー・プログラムの開始時またはプログラムの実行中に設定できます。 IBMQREP_CAPPARMS 表に保管されているパラメーターの値を変更することもできます。

関連概念:

- 62 ページの『WebSphere MQ オブジェクトの設定要件』
- 233 ページの『Q キャプチャー・プログラムのパラメーター -- 概要』
- 15 ページの『Q キャプチャー・プログラム』
- 425 ページの『Q キャプチャー・コントロール表の詳細な構造 -- 概説』

Q キャプチャー・パラメーターの変更

Q キャプチャー・パラメーターの変更 -- 概要

Q キャプチャー・プログラムの稼働パラメーターの変更は、プログラムの開始時、プログラムの実行中、または IBMQREP_CAPPARMS コントロール表の更新により実行できます。詳細については、以下のトピックで説明します。

- 249 ページの『Q キャプチャー稼働パラメーターの変更方法』
- 250 ページの『IBMQREP_CAPPARMS 表に保管された Q キャプチャー・パラメーターの変更』
- 251 ページの『Q キャプチャー・プログラムの実行中に行うパラメーターの動的変更』

関連概念:

- 229 ページの『Q キャプチャー・プログラムの操作 -- 概要』

Q キャプチャー稼働パラメーターの変更方法

このトピックでは、パラメーターを変更する 3 つの方法の要旨について説明し、続いてそれぞれの相違点を具体的に示した例を取り上げます。

IBMQREP_CAPPARMS 表に保管されたパラメーターの変更

Q キャプチャー・プログラムの稼働パラメーターは、IBMQREP_CAPPARMS コントロール表に保管されています。インストール後は、この表には、プログラムの出荷時デフォルト値が入っています。Q キャプチャー・プログラムは、開始時にこの表を読み取ります。Q キャプチャー・プログラムの開始時または実行中に、他の方法でパラメーター値を変更できますが、変更はメモリー内のみにとどまります。Q キャプチャー・プログラムを停止して再始動する際には、IBMQREP_CAPPARMS 表に保管されている値が使用されます。この表は、レプリケーション・センターの「パラメーターの変更 - 保管」ウィンドウ、または SQL を使用して更新できます。

パラメーター値の開始時の設定

Q キャプチャー・プログラムを開始する際には、IBMQREP_CAPPARMS 表に保管されているパラメーター値をオーバーライドできます。レプリケーション・センターの「Q キャプチャーの開始」ウィンドウ、または **asnlqcap** システム・コマンドを使用して、稼働パラメーターの値を設定できます。変更はプログラムの開始時に有効になりますが、有効なのはプログラムの実行中だけです。

Q キャプチャー・プログラムの実行中に行うパラメーターの動的変更

ソースからの変更のキャプチャーを停止することなく、Q キャプチャー・プログラムのパラメーター値を動的に変更できます。Q キャプチャー・プログラムの実行中に値を変更するには、レプリケーション・センターの「パラメーターの変更 - Q キャプチャー・プログラムの実行中 (Change Parameters - Running Q Capture Program)」ウィンドウ、または **asnlqccmd chgparms** コマンドを使用します。変更はプログラムの実行が停止するまで、または次のパラメーター変更要求まで持続します。

例: Q キャプチャー稼働パラメーターの 3 つの変更方法

スキーマ ASN1 で識別される Q キャプチャー・プログラムで、500 ミリ秒 (0.5 秒) のコミット・インターバルのデフォルト設定を増やすとします。

1. Q キャプチャー・スキーマ ASN1 の IBMQREP_CAPPARMS 表を更新します。コミット・インターバルを 1000 ミリ秒 (1 秒) に設定する。レプリケーション・センターの「パラメーターの変更 - 保管」ウィンドウ、または次の SQL を使用します。

```
update asnl.ibmqrep_capparms set commit_interval=1000
```

今後のキャプチャー・プログラムの開始時には、デフォルト・コミット・インターバルが 1000 ミリ秒に設定されます。

2. コミット・インターバルをさらに長くして、レプリケーション・スループット (特定の期間で発行されたトランザクションの数) への影響を確認するとします。コントロール表に保管されている値を変更するのではなく、コミット・イン

ターバルを 2000 ミリ秒 (2 秒) に設定して Q キャプチャー・プログラムを開始します。レプリケーション・センターの「Q キャプチャーの開始」ウィンドウ、または次の **asnlcap** コマンドを使用します。

```
asnlcap capture_server=srcdb1 capture_schema="ASN1" commit_interval=2000
```

プログラムが 2 秒のコミット・インターバルを使用して稼働している間、このプログラムのパフォーマンスをモニターします。

3. パフォーマンスに基づいて、`commit_interval` を低くするとします。Q キャプチャー・プログラムを停止するのではなく、プログラムの実行中にパラメーターを 1500 ミリ秒 (1.5 秒) に動的に変更し、変更をモニターします。レプリケーション・センターの「パラメーターの変更 - Q キャプチャー・プログラムの実行中 (Change Parameters - Running Q Capture Program)」ウィンドウ、または **asnlccmd chgparms** コマンドを使用します。

```
asnlccmd capture_server=srcdb1 capture_schema="ASN1" chgparms  
commit_interval=1500
```

スループットおよび待ち時間統計のモニターを継続し、レプリケーション・センターまたは **asnlccmd chgparms** コマンドを使用してコミット・インターバル・パラメーターを調整できます。要件を満たす値を見つけたら、`IBMQREP_CAPPARMS` 表を更新します (ステップ 1 で説明)。Q キャプチャー・プログラムの次の開始時に、新しい値がデフォルトのコミット・インターバルとして使用されます。

関連概念:

- 248 ページの『Q キャプチャー・パラメーターの変更 -- 概要』
- 15 ページの『Q キャプチャー・プログラム』

関連タスク:

- 250 ページの『`IBMQREP_CAPPARMS` 表に保管された Q キャプチャー・パラメーターの変更』
- 251 ページの『Q キャプチャー・プログラムの実行中に行うパラメーターの動的変更』
- 229 ページの『Q キャプチャー・プログラムの開始』

IBMQREP_CAPPARMS 表に保管された Q キャプチャー・パラメーターの変更

Q キャプチャー・プログラムは、稼働パラメーターを `IBMQREP_CAPPARMS` コントロール表に保管します。プログラムの開始時または実行中に、これらの保管されたパラメーターをオーバーライドする場合は、変更はメモリー内のみにとどまります。Q キャプチャー・プログラムの次の開始時に、コントロール表に保管された値が使用されます。保管されたパラメーター値を変更するには、コントロール表を更新する必要があります。

`IBMQREP_CAPPARMS` 表には、単一行が含まれています。この表に行がない場合、または複数の行が含まれている場合は、Q キャプチャー・プログラムは実行されません。

1 つ以上の保管されたパラメーター値を変更したい場合は、IBMQREP_CAPPARMS 表を更新できます。この表は、Q キャプチャー・プログラムが開始時に読み取るため、更新を有効にするには、プログラムを停止してから再始動する必要があります。Q キャプチャー・プログラムを再初期化しても、プログラムは IBMQREP_CAPPARMS 表の新しい値を読み取るように指示されません。

手順:

次の方法のいずれかを使用して、IBMQREP_CAPPARMS 表内の Q キャプチャー・プログラムの保管された稼働パラメーターを変更します。

レプリケーション・センター

「パラメーターの変更 - 保管」ウィンドウを使用して、IBMQREP_CAPPARMS 表の値を表示または変更します。このウィンドウをオープンするには、保管されたパラメーターを表示および変更したい Q キャプチャー・プログラムが含まれている Q キャプチャー・サーバーを右マウス・ボタンでクリックして、「パラメーターの変更」→「保管済み (Saved)」を選択します。詳細については、オンライン・ヘルプを参照してください。

SQL コマンド・プロンプトまたは DB2 コマンド行ツールのいずれかを使用して、IBMQREP_CAPPARMS 表に対して SQL UPDATE ステートメントを発行します。たとえば、**monitor_interval** および **logstdout** のデフォルトを変更するには、次のようにします。

```
update schema.ibmqrep_capparms set monitor_interval=600, logstdout=Y
```

ここで、*schema* は、保管されたパラメーター値を変更したい Q キャプチャー・プログラムを示します。

関連概念:

- 248 ページの『Q キャプチャー・パラメーターの変更 -- 概要』
- 15 ページの『Q キャプチャー・プログラム』

関連資料:

- 428 ページの『IBMQREP_CAPPARMS 表』

Q キャプチャー・プログラムの実行中に行うパラメーターの動的変更

ソースからの変更のキャプチャーを停止することなく、Q キャプチャー・プログラムの動作を変更できます。Q キャプチャー・プログラムは、新しい設定をほぼ即時に使用し始めますが、変更は IBMQREP_CAPPARMS コントロール表に保管されません。Q キャプチャー・プログラムを停止してから再始動すると、コントロール表に保管された値が使用されます。

プログラムの実行中に、以下の Q キャプチャー・パラメーターを変更できます。

- **autostop**
- **commit_interval**
- **logreuse**
- **logstdout**

- **memory_limit**
- **monitor_interval**
- **monitor_limit**
- **prune_interval**
- **term**
- **trace_limit**

値を変更しても、その影響が即時に反映されない場合があります。レプリケーション・センターが **asnlccmd chgparms** コマンドを発行してから、Q キャプチャー・プログラムの操作が変更されるまでに、1 秒から 2 秒の遅延が発生する場合があります

前提条件:

パラメーターを変更したい Q キャプチャー・プログラムが実行されている必要があります。

手順:

以下の方法のいずれかを使用して、現行セッションの Q キャプチャー・プログラムの稼働パラメーターの値を動的に変更してください。

レプリケーション・センター

「パラメーターの変更 - Q キャプチャー・プログラムの実行中 (Change Parameters - Running Q Capture Program)」ウィンドウを使用します。このウィンドウをオープンするには、パラメーターを変更したい Q キャプチャー・プログラムが含まれている Q キャプチャー・サーバーを右マウス・ボタンでクリックして、「パラメーターの変更」→「Q キャプチャー・プログラムの実行」を選択します。詳細については、オンライン・ヘルプを参照してください。

asnlccmd システム・コマンド

asnlccmd chgparms コマンドを使用して、実行中の Q キャプチャー・プログラムのパラメーターを変更します。

```
asnlccmd capture_server=server capture_schema=schema chgparms parameters
```

ここで、*server* は Q キャプチャー・プログラムの名前であり、*schema* は実行中の Q キャプチャー・プログラムを示し、*parameters* は変更したい 1 つ以上のパラメーターです。

z/OS コンソールまたは TSO

MODIFY コマンドを使用して、Q キャプチャー・プログラムの実行中にパラメーター値を変更します。

関連概念:

- 248 ページの『Q キャプチャー・パラメーターの変更 -- 概要』
- 15 ページの『Q キャプチャー・プログラム』

関連タスク:

- 339 ページの『JCL を使用した、開始済みの Q レプリケーションおよびイベント発行プログラムの変更』

関連資料:

- 368 ページの『asnqccmd: 実行中の Q キャプチャー・プログラムの操作』

Q サブスクリプションまたは XML 発行のアクティブ化

Q キャプチャー・プログラムがソース表からデータを複製または発行するには、その前に、ソース表を指定する Q サブスクリプションまたは XML 発行の状態が、A (アクティブ) または N (新規) になっている必要があります。

デフォルトでは、新しく作成された Q サブスクリプションまたは XML 発行の状態は N (新規) になっていますが、Q キャプチャー・プログラムの開始または再初期化時に自動的にアクティブにされます。このデフォルトを変更すると、Q サブスクリプションまたは XML 発行を作成後にアクティブにする必要があります。

Q サブスクリプションまたは XML 発行がアクティブにされると、Q キャプチャー・プログラムはソースの変更のキャプチャーを開始し、レプリケーション・キュー・マップまたは発行キュー・マップで定義した WebSphere MQ キューを介した変更の送信も開始します。

前提条件:

- CAPSTART シグナルを読み取るか、サブスクリプション・メッセージをアクティブにするには、Q キャプチャー・プログラムが実行されている必要があります。Q サブスクリプションまたは XML 発行をアクティブにするときに Q キャプチャー・プログラムが停止している場合は、プログラムをウォーム・スタートする場合にのみ、シグナルまたはメッセージが処理されます。コールド・スタートする場合は、シグナルまたはメッセージは失われます。
- アクティブにしたい Q サブスクリプションが、EXPORT ユーティリティを使用する自動ロードを指定している場合は、ユーティリティが Q キャプチャー・サーバーに接続できるようにするために、Q アプライ・サーバー上でパスワード・ファイルを作成する必要があります。詳細については、43 ページの『リモート・サーバー用のユーザー ID およびパスワードの保管』を参照してください。

手順:

レプリケーション・センターまたは SQL を使用して、Q サブスクリプションまたは XML 発行をアクティブにできます。

レプリケーション・センター

次のいずれかのウィンドウを使用します。

- 「Q サブスクリプションの管理」ウィンドウを使用して、Q サブスクリプションをアクティブにする。このウィンドウをオープンするには、Q サブスクリプションのソース表が置かれている Q キャプチャー・サーバーを右マウス・ボタンでクリックして、「管理」→「Q サブスクリプション」を選択します。
- 「XML 発行の管理」ウィンドウを使用して、XML 発行をアクティブにする。このウィンドウをオープンするには、XML 発行のソース表が置かれている Q キャプチャー・サーバーを右マウス・ボタンでクリックして、「管理」→「XML 発行」を選択します。

詳細については、オンライン・ヘルプを参照してください。

SQL コマンド・プロンプトまたは DB2 コマンド行ツールのいずれかを使用して、次のように CAPSTART シグナルを Q キャプチャー・サーバーの IBMQREP_SIGNAL 表に挿入します。

```
insert into schema.IBMQREP_SIGNAL(  
    SIGNAL_TIME,  
    SIGNAL_TYPE,  
    SIGNAL_SUBTYPE,  
    SIGNAL_INPUT_IN,  
    SIGNAL_STATE  
) values (  
    CURRENT_TIMESTAMP,  
    'CMD',  
    'CAPSTART',  
    'subname',  
    'P' );
```

ここで、*schema* は Q キャプチャー・プログラムを示し、*subname* はアクティブにしたい Q サブスクリプションまたは XML 発行の名前です。

関連概念:

- 133 ページの『ソースからターゲットへのレプリケーションのセットアップ (複数方向) -- 概要』
- 229 ページの『Q キャプチャー・プログラムの操作 -- 概要』
- 179 ページの『ソースからの発行 (イベント発行) の設定 -- 概要』
- 91 ページの『ソースからターゲットへの (単一方向) レプリケーションのセットアップ -- 概要』

関連タスク:

- 254 ページの『Q サブスクリプションまたは XML 発行の非アクティブ化』

関連資料:

- 437 ページの『IBMQREP_SIGNAL 表』
- 525 ページの『サブスクリプション・メッセージのアクティブ化』

Q サブスクリプションまたは XML 発行の非アクティブ化

Q サブスクリプションまたは XML 発行を非アクティブにして、Q サブスクリプションまたは XML 発行の変更のキャプチャーを停止するように Q キャプチャー・プログラムに指示します。非アクティブにすると、Q キャプチャー・プログラムを停止することなく、Q サブスクリプションまたは XML 発行のアクティビティを削除または中断できます。

レプリケーション・センターを使用するか、SQL シグナルを IBMQREP_SIGNAL 表に挿入することによって、Q サブスクリプションまたは XML 発行を非アクティブにできます。Q キャプチャー・プログラムは、Q サブスクリプションまたは XML 発行の変更のキャプチャーを停止し、IBMQREP_SUBS 表の中で状態を I (非アクティブ) に変更します。

前提条件:

CAPSTOP シグナルを読み取るには、Q キャプチャー・プログラムが実行されていなければなりません。Q サブスクリプションまたは XML 発行を非アクティブにするときに Q キャプチャー・プログラムが停止している場合は、プログラムをウォーム・スタートする場合にのみ、シグナルが処理されます。コールド・スタートする場合は、シグナルが失われます。

手順:

レプリケーション・センターまたは SQL を使用して、Q サブスクリプションまたは XML 発行を非アクティブにできます。

レプリケーション・センター

次のいずれかのウィンドウを使用します。

- 「Q サブスクリプションの管理」ウィンドウを使用して、Q サブスクリプションを非アクティブにする。このウィンドウをオープンするには、Q サブスクリプションのソース表が置かれている Q キャプチャー・サーバーを右マウス・ボタンでクリックして、「管理」→「Q サブスクリプション」を選択します。
- 「XML 発行の管理」ウィンドウを使用して、XML 発行を非アクティブにする。このウィンドウをオープンするには、XML 発行のソース表が置かれている Q キャプチャー・サーバーを右マウス・ボタンでクリックして、「管理」→「XML 発行」を選択します。

詳細については、オンライン・ヘルプを参照してください。

SQL コマンド・プロンプトまたは DB2 コマンド行ツールのいずれかを使用して、次のように CAPSTOP シグナルを Q キャプチャー・サーバーの IBMQREP_SIGNAL 表に挿入します。

```
insert into schema.IBMQREP_SIGNAL (
    SIGNAL_TIME,
    SIGNAL_TYPE,
    SIGNAL_SUBTYPE,
    SIGNAL_INPUT_IN,
    SIGNAL_STATE)
values (
    CURRENT_TIMESTAMP,
    'CMD',
    'CAPSTOP',
    'subname',
    'P');
```

ここで、*schema* は Q キャプチャー・プログラムを示し、*subname* は非アクティブにしたい Q サブスクリプションまたは XML 発行です。

関連概念:

- 229 ページの『Q キャプチャー・プログラムの操作 -- 概要』

関連タスク:

- 253 ページの『Q サブスクリプションまたは XML 発行のアクティブ化』

関連資料:

- 437 ページの『IBMQREP_SIGNAL 表』
- 526 ページの『サブスクリプション・メッセージの非アクティブ化』
- 511 ページの『サブスクリプション非アクティブ化メッセージ』

Q キャプチャー・プログラムの停止

Q キャプチャー・プログラムを停止すると、DB2 ログからの読み取りと、メモリー内でのトランザクションの構築が停止します。キューに書き込まれたメッセージは、Q キャプチャー・プログラムが停止する前に WebSphere MQ にコミットされます。プログラムを停止したときにメモリー内にあった非コミット WebSphere MQ トランザクションまたは行の変更は、再始動メッセージに保管された再始動ポイントに基づいて、Q キャプチャー・プログラムの再始動時に再キャプチャーされます。

ヒント: 以下のようにすれば、Q サブスクリプションまたは XML 発行を追加または削除するために、Q キャプチャー・プログラムを停止する必要はありません。

- プログラムの実行中に 1 つまたは 2 つの Q サブスクリプションまたは XML 発行を追加したい場合は、Q サブスクリプションまたは XML 発行を自動的に開始しないように作成してから、それらをアクティブにする。
- 大量の Q サブスクリプションまたは XML 発行を追加したい場合は、それらを自動的に開始するように作成してから、Q キャプチャー・プログラムを再初期化する。
- Q サブスクリプションまたは XML 発行を非アクティブにしてから、それを削除することによって、Q キャプチャー・プログラムを停止することなく Q サブスクリプションまたは XML 発行を削除できる。

前提条件:

停止したい Q キャプチャー・プログラムが実行されていないなければならない。

手順:

次の方法のいずれかを使用して、Q キャプチャー・プログラムを停止します。

レプリケーション・センター

「Q キャプチャーの停止」ウィンドウを使用して、Q キャプチャー・プログラムを停止する。このウィンドウをオープンするには、停止したい Q キャプチャー・プログラムが含まれている Q アプライ・サーバーを右マウス・ボタンでクリックして、「**Q キャプチャー・プログラムの停止**」を選択します。詳細については、オンライン・ヘルプを参照してください。

asnlccmd システム・コマンド

以下のように、**asnlccmd stop** コマンドを使用して Q キャプチャー・プログラムを停止します。

```
asnlccmd capture_server=server_name capture_schema=schema stop
```

ここで、*server_name* は Q キャプチャー・プログラムが実行されているデータベースまたはサブシステムの名前であり、*schema* は停止する Q キャプチャー・プログラムを示しています。

SQL 以下のようにコマンド・プロンプトまたはいずれかの DB2 コマンド行ツールを使用して、STOP シグナルを Q キャプチャー・サーバーの `IBMQREP_SIGNAL` 表に挿入します。

```
insert into schema.IBMQREP_SIGNAL(  
    SIGNAL_TIME,  
    SIGNAL_TYPE,  
    SIGNAL_SUBTYPE,
```

```

        SIGNAL_INPUT_IN,
        SIGNAL_STATE)
values (
    CURRENT_TIMESTAMP,
    'CMD',
    'STOP',
    'NULL',
    'P' );

```

ここで、*schema* は停止したい Q キャプチャー・プログラムを示します。

Windows サービス

Windows オペレーティング・システムで DB2 レプリケーション・サービスを作成し、Windows Service Control Manager または **net stop** コマンドを使用して、Q キャプチャー・プログラムを停止できます。

z/OS コンソールまたは TSO

MODIFY コマンドを使用して、Q キャプチャー・プログラムを停止します。

関連概念:

- 229 ページの『Q キャプチャー・プログラムの操作 -- 概要』
- 15 ページの『Q キャプチャー・プログラム』

関連タスク:

- 225 ページの『XML 発行の削除』
- 213 ページの『Q サブスクリプションの削除』
- 161 ページの『3 つ以上のサーバーを使用したピアツーピア・グループでのレプリケーションの停止』
- 159 ページの『2 つのサーバーを使用した双方向およびピアツーピア・レプリケーションの停止』
- 276 ページの『Q アプライ・プログラムの停止』
- 253 ページの『Q サブスクリプションまたは XML 発行のアクティブ化』
- 254 ページの『Q サブスクリプションまたは XML 発行の非アクティブ化』
- 229 ページの『Q キャプチャー・プログラムの開始』
- 345 ページの『レプリケーション・サービスの停止』
- 339 ページの『JCL を使用した、開始済みの Q レプリケーションおよびイベント発行プログラムの変更』

関連資料:

- 368 ページの『asnlccmd: 実行中の Q キャプチャー・プログラムの操作』
- 437 ページの『IBMQREP_SIGNAL 表』

第 18 章 Q アプライ・プログラムの操作

Q アプライ・プログラムの操作 -- 概要

Q アプライ・プログラムは、ソース表からトランザクションを含むメッセージを読み取り、それらのメッセージを Q サブスクリプションが定義するターゲットに適用します。レプリケーション・センター、システム・コマンド、およびシステム・サービスを使用して Q アプライ・プログラムを操作し、いくつかの方法で Q アプライ稼働パラメーターを変更することができます。

以下のトピックでは、Q アプライ・プログラムの操作方法について説明しています。

- 『Q アプライ・プログラムの開始』
- 262 ページの『Q アプライ・プログラムのパラメーター -- 概要』
- 271 ページの『Q アプライ・パラメーターの変更 -- 概要』
- 273 ページの『受信キュー上のメッセージ処理の開始』
- 275 ページの『受信キュー上のメッセージ処理の停止』
- 276 ページの『Q アプライ・プログラムの停止』

関連概念:

- 19 ページの『Q アプライ・プログラム』
- 21 ページの『Q アプライ・プログラムと Q キャプチャー・プログラムのスキーマ』

Q アプライ・プログラムの開始

Q アプライ・プログラムを開始すると、トランザクション・メッセージのキューからの読み取りと、ターゲット表またはストアード・プロシージャーへのトランザクションの適用が開始されます。

Q アプライ・プログラムを開始すると、開始パラメーターを指定でき、次のいずれかの処置を取るまで、プログラムは新しい値を使用します。

- プログラムの実行中にパラメーター値を変更する。
- プログラムを停止してから再始動する。これにより、プログラムは IBMQREP_APPLYPARMS 表を読み取り、そこに保管されている値を使用するように指示されます。

前提条件:

- Q アプライ・プログラムをリモート・ワークステーションから開始する場合は、Q アプライ・サーバーに対する接続を構成する必要がある。詳細については、75 ページの『Q レプリケーションおよびイベント発行の接続要件』を参照してください。
- WebSphere MQ キュー・マネージャー、キュー、および他の必要なオブジェクトを、Q アプライ・サーバーと同じシステム上で作成して構成する必要がある。詳

細については、47ページの『Q レプリケーションおよびイベント発行に必要な WebSphere MQ オブジェクト -- 概要』を参照してください。

- Q レプリケーション・オブジェクトおよび WebSphere MQ オブジェクト用の適切な許可を持っている必要がある。詳細については、40ページの『Q アプライ・プログラムの許可要件』 および 70ページの『WebSphere MQ オブジェクトの接続および許可に関する要件』を参照してください。
- 適切な Q アプライ・スキーマ用のコントロール表を作成する必要がある。詳細については、87ページの『Q キャプチャー・プログラムと Q アプライ・プログラムのコントロール表の作成』を参照してください。
- Q アプライ・プログラムと連動するように、ターゲット・データベースまたはサブシステムを構成する必要がある。詳細については、80ページの『Q アプライ・プログラムが動作するターゲット・データベースの構成 (Linux、UNIX、Windows)』 および 85ページの『Q レプリケーションおよびイベント発行用のデータベースの構成 (z/OS)』を参照してください。
- EXPORT/IMPORT または EXPORT/LOAD ユーティリティーを使用するターゲットのために自動ロードを指定する Q サブスクリプションがある場合は、Q キャプチャー・サーバーでの Q アプライ認証用のパスワード・ファイルが存在していなければならない。詳細については、396ページの『asnpwd: パスワード・ファイルの作成および保守』を参照してください。

手順:

次の方法のいずれかを使用して、Q アプライ・プログラムを開始します。

レプリケーション・センター

「Q アプライの開始」ウィンドウを使用して、Q アプライ・プログラムを開始する。このウィンドウをオープンするには、開始したい Q アプライ・プログラムが含まれている Q アプライ・サーバーを右マウス・ボタンでクリックして、「**Q アプライ・プログラムの開始**」を選択します。

IBMQREP_APPLYPARMS コントロール表に保管されているパラメーター値を使用してプログラムを開始するか、開始する前に新しいランタイム値を指定できます。詳細については、オンライン・ヘルプを参照してください。

asnlqapp システム・コマンド

asnlqapp コマンドを使用して、Q アプライ・プログラムを開始する。オプションで、次の開始パラメーターを指定できます。

```
asnlqapp apply_server=server_name apply_schema=schema parameters
```

ここで、*server_name* は Q アプライ・コントロール表が定義されており、Q アプライ・プログラムがターゲットに変更を適用するデータベースまたはサブシステムの名前であり、*schema* は開始したい Q アプライ・プログラムを示し、*parameters* は開始時に指定できる 1 つ以上のパラメーターです。

z/OS コンソールまたは TSO

z/OS では、Q アプライ・プログラムは、JCL を使用して開始するか、システム開始タスクとして開始できます。JCL で Q アプライ・プログラムを開始する場合は、新しい呼び出しパラメーターを指定できます。これらの値は、IBMQREP_APPLYPARMS 表に保管されているパラメーター値をオーバーライドします。EXEC ステートメントの PARM パラメーターでは、100文字を超えるサブパラメーターを指定できません。

Windows サービス

Windows オペレーティング・システム上で DB2 レプリケーション・サービスを作成して、システムの始動時に Q アプライ・プログラムを自動的に開始できます。

Q アプライ・プログラムが開始したかどうかを確認するには、次の方法のいずれかを使用します。

- レプリケーション・センターの「Q アプライ・メッセージ」ウィンドウを使用して、プログラムが開始したことを示すメッセージを確認する。このウィンドウをオープンするには、メッセージを表示したい Q アプライ・プログラムが含まれている Q アプライ・サーバーを右マウス・ボタンでクリックして、「レポート」→「Q アプライ・メッセージ」を選択します。詳細については、オンライン・ヘルプを参照してください。
- レプリケーション・センターの「状況のチェック」ウィンドウを使用して、すべての Q アプライ・スレッドの状況を表示する。このウィンドウをオープンするには、チェックしたい Q アプライ・プログラムが置かれている Q アプライ・サーバーを右マウス・ボタンでクリックして、「状況のチェック」を選択します。詳細については、オンライン・ヘルプを参照してください。
- Q アプライ・ログ・ファイル (Linux、UNIX、および Windows の場合は `db2instance.apply_server.apply_schema.QAPP.log`、z/OS の場合は `apply_server.apply_schema.QAPP.log`) の中に、プログラムが変更を適用していることを示すメッセージがあるかを調べます。
- IBMQREP_APPLYTRACE 表の中に、プログラムが変更を適用していることを示すメッセージがあるかを調べます。

関連概念:

- 259 ページの『Q アプライ・プログラムの操作 -- 概要』
- 262 ページの『Q アプライ・プログラムのパラメーター -- 概要』
- 283 ページの『Q レプリケーションおよびイベント発行プログラムの履歴およびパフォーマンス・データ』
- 19 ページの『Q アプライ・プログラム』

関連タスク:

- 253 ページの『Q サブスクリプションまたは XML 発行のアクティブ化』
- 342 ページの『Windows Service Control Manager (SCM) を使用してレプリケーション・サービスを管理する方法 -- 概要』
- 337 ページの『JCL を使用した Q アプライ・プログラムの開始』

関連資料:

- 372 ページの『asqapp: Q アプライ・プログラムの開始』
- 452 ページの『IBMQREP_APPLYPARMS 表』
- 454 ページの『IBMQREP_APPLYTRACE 表』

Q アプライ・プログラムのパラメーター

Q アプライ・プログラムのパラメーター -- 概要

Q アプライ・プログラムの稼働パラメーターは、接続先のキュー・マネージャー、コントロール表を整理する頻度、および何も作業がない場合に停止するかどうか、などを指定します。以下のトピックでは、デフォルトの稼働パラメーターをリストし、各パラメーターの詳細とデフォルト設定を変更する理由について説明します。

- 『Q アプライ稼働パラメーターのデフォルト値』
- 263 ページの『Q アプライ・パラメーターの説明』

関連概念:

- 259 ページの『Q アプライ・プログラムの操作 -- 概要』

Q アプライ稼働パラメーターのデフォルト値

レプリケーション・センターを使用してコントロール表を作成する場合、IBMQREP_APPLYPARMS 表は、Q アプライ・プログラムのオペレーティング・システムのデフォルト値を含む単一行を持つように作成されます。表 14 はこれらのデフォルト値を示しています。

デフォルト・パラメーター値は、IBMQREP_APPLYPARMS 表を更新することによって、または Q アプライ・プログラムの始動時またはプログラムの実行中に、保管された値を一時的にオーバーライドすることによって、レプリケーション環境に合うように変更できます。

注: コントロール表の作成時に、**qmgr** (キュー・マネージャー) パラメーターに値を提供します。

表 14. Q アプライ稼働パラメーターのデフォルト値

稼働パラメーター	デフォルト値	IBMQREP_APPLYPARMS 表の列名
apply_path	Q アプライが始動されたディレクトリー ¹	APPLY_PATH
apply_schema	ASN ²	該当せず
apply_server	DB2DBDFT ³	該当せず
autostop	N ⁴	AUTOSTOP
deadlock_retries	3	DEADLOCK_RETRIES
logreuse	N ⁴	LOGREUSE
logstdout	N ⁴	LOGSTDOUT
monitor_interval	300 ⁵	MONITOR_INTERVAL
monitor_limit	10080 ⁶	MONITOR_LIMIT
prune_interval	300 ⁵	PRUNE_INTERVAL
pwdfile	asnpwd.aut	PWDFILE
qmgr	なし ⁷	QMGR
term	Y ⁸	TERM

表 14. Q アプライ稼働パラメーターのデフォルト値 (続き)

稼働パラメーター	デフォルト値	IBMQREP_APPLYPARMS 表の列名
trace_limit	10080 ⁶	TRACE_LIMIT

注:

1. Q アプライ・プログラムが Windows サービスとして開始する場合、**apply_path** は %sqllib%\bin です。Q アプライが z/OS で JCL を使用して開始する場合、デフォルトは、開始済みのタスクまたはジョブと関連したユーザー ID です。
2. デフォルトのスキーマは変更できません。Q アプライ・プログラムを別のスキーマで使用するには、Q アプライ・プログラム始動時に **apply_schema** パラメーターを指定する。
3. Linux、UNIX、および Windows の場合、デフォルトの Q アプライ・サーバーは、DB2DBDFT 環境変数の値です (指定されている場合)。z/OS の場合は、デフォルトの Q アプライ・サーバーはありません。
4. No (いいえ)。
5. 秒。
6. 分。
7. Q アプライ・コントロール表の作成時には、この値を指定しなければなりません。これは コマンド・パラメーターではありません。Q アプライ・プログラムは、この値を IBMQREP_APPLYPARMS 表から読み取ります。
8. Yes (はい)。

関連概念:

- 271 ページの『Q アプライ・パラメーターの変更 -- 概要』
- 262 ページの『Q アプライ・プログラムのパラメーター -- 概要』
- 19 ページの『Q アプライ・プログラム』

関連資料:

- 263 ページの『Q アプライ・パラメーターの説明』
- 452 ページの『IBMQREP_APPLYPARMS 表』

Q アプライ・パラメーターの説明

以下のトピックは、Q アプライ・プログラムの稼働パラメーターを説明し、さらに必要に基づいてデフォルト値を変更する必要がある場合の理由を説明します。

Q アプライ・プログラムは、プログラム始動時に、パラメーターを IBMQREP_APPLYPARMS 表から読み取ります。プログラムの開始時およびプログラムの実行中に、一時ランタイム値を指定できるパラメーターもあります。詳細については、各パラメーターの説明を参照してください。

- 264 ページの『autostop』
- 264 ページの『apply_path』
- 265 ページの『apply_schema』
- 265 ページの『apply_server』
- 265 ページの『deadlock_retries』

- 266 ページの『logreuse』
- 267 ページの『logstdout』
- 267 ページの『monitor_interval』
- 268 ページの『monitor_limit』
- 268 ページの『pwdfile』
- 269 ページの『prune_interval』
- 269 ページの『qmgr』
- 270 ページの『term』
- 270 ページの『trace_limit』

autostop

デフォルト: **autostop=N**

autostop パラメーターにより、アプライするトランザクションがない場合に、Q アプライ・プログラムを自動的に停止させることができます。デフォルト (**autostop=N**) では、Q アプライ・プログラムは、キューが空でトランザクションの到着を待機する場合に実行を継続します。

通常、Q アプライ・プログラムは、ターゲット・データベースがアクティブの時はいつでも、連続するプロセスとして実行されるので、ほとんどの場合デフォルトのままにしておきます (**autostop=N**)。 **autostop=Y** に設定するのは、接続されたシステムを低い頻度で同期化するときや、テスト・シナリオなどの、設定された間隔で Q アプライ・プログラムが実行されるシナリオの場合のみです。

autostop=Y に設定すると、Q アプライ・プログラムは、すべての受信キューが一度空になった後シャットダウンします。各受信キューごとのブラウザー・スレッドがキューにメッセージがないことを検出すると、スレッドはキューからの読み取りを停止します。すべてのスレッドが停止した後、Q アプライ・プログラムが停止します。ブラウザー・スレッドが停止したキューにメッセージは着信し続ける可能性があります。Q アプライ・プログラムを再び始動するまで収集されます。

Q アプライ・プログラムの始動時、またはプログラムの実行中に、**autostop** パラメーターを設定できます。また、IBMQREP_APPLYPARMS 表に保管された値を変更することもできます。

apply_path

apply_path パラメーターは、Q アプライ・プログラムが自分用の作業ファイルとログ・ファイルを保管するディレクトリーを指定します。デフォルトのパスはプログラムを開始するディレクトリーです。このパスは変更できます。

Windows

Q アプライ・プログラムを Windows サービスとして始動する場合、デフォルトではプログラムは %sqllib%\bin ディレクトリーで始動されます。

z/OS Q アプライ・プログラムは POSIX アプリケーションであるため、デフォルトのパスは、次に示す方法のどちらでこのプログラムを開始するかによって決まります。

- Q アプライ・プログラムを USS コマンド行プロンプトから始動する場合。パスは、このプログラムが始動されたディレクトリーです。

- Q アプライ・プログラムを開始済みのタスクまたは JCL を使用して始動する場合。デフォルト・パスは、その開始済みタスクまたはジョブと関連したユーザー ID のホーム・ディレクトリーです。

パスを変更するには、パス名または //QAPP などの上位修飾子 (HLQ) のいずれかを指定できます。HLQ を使用すると、z/OS 順次データ・セット・ファイル名のファイル命名規則に準拠する順次ファイルが作成されます。

Q アプライ・プログラムの始動時に、**apply_path** パラメーターを設定するか、または IBMQREP_APPLYPARMS 表に保管された値を変更できます。このパラメーターは、Q アプライ・プログラム実行中には変更できません。

apply_schema

デフォルト: **apply_schema=ASN**

apply_schema パラメーターを使用すると、Q アプライ・サーバー上の Q アプライ・プログラムの複数インスタンスを識別することができます。

スキーマは、1 つの Q アプライ・プログラムおよびそのコントロール表を識別します。同じスキーマを持つ 2 つの Q アプライ・プログラムを 1 つのサーバー上で実行することはできません。

単一の Q アプライ・プログラムは、複数のブラウザー・スレッドを作成できます。各ブラウザー・スレッドは、単一の受信キューからメッセージを読み取ります。このため、ターゲットにアプライされているデータの流れを分割するために、サーバー上の Q アプライ・プログラムの複数のインスタンスを作成する必要はありません。

apply_server

デフォルト (Linux、UNIX、Windows): **apply_server= DB2DBDFT** 環境変数の値 (設定されている場合)

デフォルト (z/OS): **apply_server= None**

apply_server パラメーターは、Q アプライ・プログラムが実行し、そのコントロール表が保管されるデータベースまたはサブシステムを識別します。コントロール表には、ターゲット、Q サブスクリプション、WebSphere MQ キュー、およびユーザー・プリファレンスについての情報が含まれます。Q アプライ・サーバーは、ターゲットを含むものと同じデータベースまたはサブシステムでなければなりません。

deadlock_retries

デフォルト: **deadlock_retries=3**

deadlock_retries パラメーターは、Q アプライ・プログラムが SQL デッドロックまたはロック・タイムアウトを検出したときに、ターゲット表に変更を再アプライしようとする回数を指定します。デフォルトは 3 回の試行です。Q アプライ・プログラムは、各試行間で 1 秒待機します。また、このパラメーターは、Q アプライ・プログラムが SQL デッドロック後、コントロール表に対して行の挿入、更新、または削除を試行する回数も制御します。

デッドロックを検出したときの Q アプライ・プログラムの動作は、何を実行していたかによって異なります。

- デッドロックがターゲット表で発生する場合、Q アプライ・プログラムは、設定された制限に達するまで試行を続けます。制限に達してもデッドロックが続く場合、Q アプライ・プログラムは Q サブスクリプションに指定されたエラー・アクションを取ります。
- デッドロックの発生時にブラウザー・スレッドがコントロール表にアクセスしている場合、スレッドは設定された制限に達するまで試行を続けます。制限に達してもデッドロックが続く場合、スレッドは正常に停止します。
- Q アプライ・メイン・スレッドがコントロール表にアクセスしている場合、スレッドは設定された制限に達するまで試行を続けます。制限に達してもデッドロックが続く場合、Q アプライ・プログラムは正常に停止します。
- Q アプライ・プログラムがモニターまたは整理の目的でコントロール表にアクセスしている場合、試行を続け、エラーをログ記録します。

アプリケーションがターゲット・データベースを頻繁に更新しており、高レベルの競合が起こっている場合、**deadlock_retries** にさらに高い値を設定することができます。または、大量の受信キューおよび対応するブラウザー・スレッドがある場合、**deadlock_retries** にさらに高い値を設定すると、IBMQREP_DONEMSG 表などのコントロール表だけでなく、ピアツーピアおよび他の複数方向レプリケーション環境で生じる可能性がある競合の解決にも役立つことがあります。

制限: deadlock_retries のデフォルト値を下げることはできません。しかし、値を上げることは可能です。

Q アプライ・プログラムの始動時、またはプログラムの実行中に、**deadlock_retries** パラメーターを設定できます。また、IBMQREP_APPLYPARMS 表に保管された値を変更することもできます。

logreuse

デフォルト: **logreuse=N**

各 Q アプライ・プログラムは、キューからの読み取りの開始および停止時、パラメーター値の変更時、コントロール表の整理時、またはエラーの検出時など、作業履歴を追跡するログ・ファイルを保持します。

デフォルトでは、Q アプライ・プログラムはプログラムの再始動時に既存のログ・ファイルに追加します。このデフォルトにより、プログラムの動作の履歴を保持することができます。この履歴が必要ないか、スペースを節約する場合には、**logreuse=Y** と設定します。Q アプライ・プログラムは、始動時にログ・ファイルをクリアし、次いでそのブランク・ファイルに書き込みます。

ログは、デフォルトでは Q アプライ・プログラムが始動するディレクトリー、または **apply_path** パラメーターを使用して設定された別の場所に保管されます。

Linux、UNIX、Windows: ログ・ファイル名は *db2instance.apply_server.apply_schema.QAPP.log* です。たとえば、DB2.SAMPLE.ASN.QAPP.log。

z/OS: ログ・ファイルには DB2 インスタンス名は含まれません。たとえば、SAMPLE.ASN.APP.log。また、**apply_path** が高位修飾子 (HLQ) を使用するように 2 つのスラッシュ (/) で指定されている場合は、z/OS 順次データ・セット・ファイルのファイル命名規則が適用され、**apply_schema** 名は、切り捨てられて 8 文字の長さになります。

Q アプライ・プログラムの始動時、またはプログラムの実行中に、**logreuse** パラメーターを設定できます。また、IBMQREP_APPLYPARMS 表に保管された値を変更することもできます。

logstdout

デフォルト: **logstdout=N**

デフォルトでは、Q アプライ・プログラムは作業履歴をログにのみ書き込みます。プログラムの履歴をログの他に標準出力 (stdout) でも表示したい場合は、**logstdout** パラメーターを変更できます。

エラー・メッセージおよび一部のログ・メッセージ (初期化、停止、サブスクリプションのアクティブ化、およびサブスクリプションの非アクティブ化) は、このパラメーターの設定に関係なく、標準出力とログ・ファイルの両方で見ることができます。

logstdout パラメーターは、Q アプライ・プログラムを **asnlqapp** コマンドで開始した場合に指定できます。レプリケーション・センターを使用して Q アプライ・プログラムを開始する場合、このパラメーターは適用されません。また、Q アプライ・プログラムの実行中に、**asnlqacmd** コマンドを使用して **logstdout** パラメーターの値を変更することもできます。SQL を使用して、IBMQREP_APPLYPARMS 表に保管された値を変更することができます。

monitor_interval

デフォルト: **monitor_interval=300** 秒 (5 分)

monitor_interval パラメーターは Q アプライ・プログラムに、IBMQREP_APPLYMON 表にパフォーマンス統計を挿入する頻度を通知します。これらの統計は、「Q アプライ・スループット (Q Apply Throughput)」および「待ち時間 (Latency)」ウィンドウを使用して表示できます。詳細については、レプリケーション・センターのオンライン・ヘルプを参照してください。

デフォルトでは、行は 300 秒 (5 分) ごとにこの表に挿入されます。必要に応じて、**monitor_interval** を調整することができます。

- **Q アプライ・プログラムのアクティビティをより詳細なレベルでモニターするには、モニター間隔を短くします。**

たとえば、5 分間隔ではなく、毎分ごとにキューを分割してメッセージ数の統計を表示できます。

- **より長い期間にわたる Q アプライ・パフォーマンス統計を表示するには、モニター間隔を長くします。**

たとえば、5 分間を何度も繰り返して待ち時間統計を表示する場合、結果を平均してパフォーマンスのより大まかなビューを入手できます。30 分または 1 時間ごとに平均した結果を見るほうが、ご使用のレプリケーション環境で役立つ場合もあります。

Q アプライ・プログラムの開始時、またはプログラムの実行中に、**monitor_interval** パラメーターを設定できます。また、IBMQREP_APPLYPARMS 表に保管された値を変更することもできます。

monitor_limit

デフォルト: **monitor_limit**=10080 分 (7 日)

monitor_limit パラメーターは、整理の対象にする前に、行がどのくらい古くなるまで IBMQREP_APPLYMON 表に残しておく必要があるのかを指定します。

デフォルトでは、10080 分 (7 日) より古い行は整理されます。

IBMQREP_APPLYMON 表は、Q アプライ・プログラムのアクティビティーについての統計を提供します。行はモニター・インターバルごとに挿入されます。必要に応じて、モニター限度を調整することができます。

- モニター限度を増やして統計を保持します。

Q アプライ・プログラムのアクティビティーを一週間を超えても保持する場合は、モニター限度を高くします。

- 統計を頻繁に確認する場合は、モニター限度を下げます。

Q アプライ・プログラムのアクティビティーを通常の基準でモニターする場合は、1 週間分の統計を保持する必要性はないと考えられるので、モニター限度を低めに設定します。

Q アプライ・プログラムの開始時、またはプログラムの実行中に、**monitor_limit** パラメーターを設定できます。また、IBMQREP_APPLYPARMS 表に保管された値を変更することもできます。

pwdfile

デフォルト: **pwdfile**=asnpwd.aut

pwdfile パラメーターは、Q アプライ・プログラムが Q キャプチャー・サーバーに接続するために使用する暗号化されたパスワード・ファイルの名前を指定します。この接続は、Q サブスクリプションが EXPORT/IMPORT または EXPORT/LOAD ユーティリティーを使用する自動ロードを指定する場合にのみ必要です。**asnpwd** コマンドを使用してパスワード・ファイルを作成する場合、デフォルトのファイル名は **asnpwd.aut** です。別の名前でパスワード・ファイルを作成するか、名前を変更する場合、**pwdfile** パラメーターを一致させるように変更する必要があります。Q アプライ・プログラムは、**apply_path** パラメーターで指定されたディレクトリー内でパスワード・ファイルを探します。

z/OS: パスワード・ファイルは不要です。

Q アプライ・プログラムの開始時に、**pwdfile** パラメーターを設定し、**IBMQREP_APPLYPARMS** 表に保管された値を変更できます。この値は、Q アプライ・プログラム実行中には変更できません。

prune_interval

デフォルト: **prune_interval=300** 秒 (5 分)

prune_interval パラメーターは、Q アプライ・プログラムが古い行を検索し、**IBMQREP_APPLYMON** および **IBMQREP_APPLYTRACE** 表から削除する頻度を決定します。デフォルトでは、Q アプライ・プログラムは 300 秒 (5 分) ごとに行を検索し、整理します。

整理の頻度は、これらのコントロール表が増大する速度と、それらの表の使用目的によって異なります。

- モニター表を管理するには、整理インターバルを短くします。

IBMQREP_APPLYMON 表が、短くされたモニター間隔のために増大の速度が速すぎる場合、整理インターバルも短くすることが必要な場合があります。この表が十分な頻度で整理されないと、表スペースの制限を超える場合があり、Q アプライ・プログラムが強制的に停止されます。しかし、表の整理があまりに頻繁にまたはピーク時に実行される場合は、そのような整理によって同じシステム上で実行するアプリケーション・プログラムに支障が生じる可能性があります。

- レコードを保持するには、整理インターバルを長くします。

IBMQREP_APPLYTRACE および **IBMQREP_APPLYMON** 表の整理の頻度を下げることによって、Q アプライ・プログラムのパフォーマンスの履歴をさらに長く保持することが必要な場合があります。

整理インターバルは **trace_limit** および **monitor_limit** パラメーターと関連して機能して、データが整理の対象となるのに十分な程度古くなったことを判別します。たとえば、**prune_interval** が 300 秒、**trace_limit** が 10080 秒である場合は、Q アプライ・プログラムは 300 秒ごとに整理を試行します。Q アプライ・プログラムは、**IBMQREP_APPLYTRACE** 表で 10080 分 (7 日) より古い行を検出すると、それらをすべて整理します。

Q アプライ・プログラムの始動時、またはプログラムの実行中に、**prune_interval** パラメーターを設定できます。また、**IBMQREP_APPLYPARMS** 表に保管された値を変更することもできます。

qmgr

qmgr パラメーターは、Q アプライ・プログラムが使用する WebSphere MQ キュー・マネージャーの名前を指定します。キュー・マネージャーのジョブは、Q アプライ・プログラムのキューとメッセージを管理することです。これは、Q アプライ・プログラムと同じシステム上で実行することが必要です。キュー・マネージャーへのクライアント接続は WebSphere MQ によりサポートされていますが、この構成は Q アプライによってサポートされていません。

キュー・マネージャーは、Q アプライ・プログラムがデータおよび情報メッセージの受信、およびコントロール・メッセージの送信に使用するキューを所有していま

す。Q レプリケーションおよびイベント発行プログラムと WebSphere MQ との間のすべての通信は、キュー・マネージャーを介して行われます。

Q アプライ・コントロール表の作成時には、キュー・マネージャーの名前を指定します。値は、IBMQREP_APPLYPARMS コントロール表に保管されます。この表を更新することによって値を変更できます。Q アプライ・プログラムの開始時、またはプログラムの実行中には、値を変更できません。

term

デフォルト: **term=Y**

term パラメーターは、Q アプライ・プログラムが DB2 静止時に実行し続けるかどうかを制御します。

デフォルト (**term=Y**) では、Q アプライ・プログラムは DB2 静止時に終了します。DB2 が静止モードで、すべてのアプリケーション (Q アプライ・プログラムを含む) を強制的に切断している間に Q アプライ・プログラムが実行し続けるようにするには、このデフォルトを変更します。この場合、DB2 が静止モードではなくなるときに、Q アプライ・プログラムは、プログラムを再始動しなくても、停止したところからトランザクションのアプライを開始します。

DB2 シャットダウンに対する Q アプライ・プログラムの反応を変更することはできません。 **term** の設定に関係なく、Q アプライ・プログラムは DB2 シャットダウン時に停止します。DB2 が再び始動するときに、Q アプライ・プログラムを開始して、メッセージの読み取りとトランザクションのアプライを開始することが必要です。

重要: Q アプライ・プログラムが実行を停止しないようにデフォルトを変更し、DB2 が制限されたアクセス (たとえば ACCESS MAINT) で再始動する場合、Q アプライ・プログラムは接続できず、その後停止します。

Q アプライ・プログラムの開始時、またはプログラムの実行中に、**term** パラメーターを設定できます。また、IBMQREP_APPLYPARMS 表に保管された値を変更することもできます。

trace_limit

デフォルト: **trace_limit=10080** 分 (7 日)

trace_limit パラメーターは、整理の対象となるまでに、行が IBMQREP_APPLYTRACE 表の中に留まる時間を指定します。

Q アプライ・プログラムは、すべての情報、警告、およびエラー・メッセージを IBMQREP_APPLYTRACE 表に挿入します。デフォルトでは、10080 分 (7 日) より古い行は整理インターバルごとに整理されます。監査情報に関するニーズに合わせてトレース限度を変更してください。

Q アプライ・プログラムの始動時、またはプログラムの実行中に、**trace_limit** パラメーターを設定できます。また、IBMQREP_APPLYPARMS 表に保管された値を変更することもできます。

関連概念:

- 62 ページの『WebSphere MQ オブジェクトの設定要件』
- 262 ページの『Q アプライ・プログラムのパラメーター -- 概要』
- 19 ページの『Q アプライ・プログラム』
- 448 ページの『Q アプライ・コントロール表の詳細な構造 -- 概説』

Q アプライ・パラメーターの変更

Q アプライ・パラメーターの変更 -- 概要

Q アプライ・プログラムの稼働パラメーターは、プログラム開始時、プログラムの実行中、または IBMQREP_APPLYPARMS コントロール表の更新によって変更できません。

各種のパラメーター変更方法の概略説明と各方法の違いを明確にする例については、249 ページの『Q キャプチャー稼働パラメーターの変更方法』を参照してください。方法は、Q アプライ・プログラムの場合と同じです。

Q アプライ・パラメーターの変更方法についての詳細は、以下のトピックを参照してください。

- 『IBMQREP_APPLYPARMS 表に保管された Q アプライ・パラメーターの変更』
- 272 ページの『Q アプライ・プログラムの実行中のパラメーターの動的変更』

関連概念:

- 259 ページの『Q アプライ・プログラムの操作 -- 概要』

IBMQREP_APPLYPARMS 表に保管された Q アプライ・パラメーターの変更

Q アプライ・プログラムは、その稼働パラメーターを IBMQREP_APPLYPARMS コントロール表に保管します。プログラムの開始時または実行中に、これらの保管されたパラメーターをオーバーライドする場合は、変更はメモリー内のみにとどまります。次に Q アプライ・プログラムを開始すると、コントロール表に保管された値が使用されます。保管済みパラメーター値を変更するには、コントロール表を更新する必要があります。

IBMQREP_APPLYPARMS 表には単一の行が含まれています。この表に行がない場合、または複数の行が入っている場合、Q アプライ・プログラムは実行しません。

1 つ以上の保管済みパラメーター値を変更する場合は、IBMQREP_APPLYPARMS 表の個々の列を更新することができます。Q アプライ・プログラムは始動時にこの表を読み取るので、変更を有効にするには、プログラムを停止してから再始動する必要があります。

手順:

次の方法のいずれかを使用して、IBMQREP_APPLYPARMS 表で Q アプライ・プログラム用の保管済みパラメーターを変更します。

レプリケーション・センター

IBMQREP_APPLYPARMS 表の値のいずれかを表示または変更したい場合は、「パラメーターの変更 - 保管済み (Change Parameters - Saved)」ウィンドウを開きます。ウィンドウをオープンするには、表示または変更する保管済みパラメーターを持つ Q アプライ・プログラムがある Q アプライ・サーバーを右マウス・ボタンでクリックし、「**パラメーターの変更 (Change Parameter)**」→「**保管済み (Saved)**」を選択します。詳細については、オンライン・ヘルプを参照してください。

SQL コマンド・プロンプトまたは DB2 コマンド行ツールの 1 つを使用して、IBMQREP_APPLYPARMS 表に SQL UPDATE ステートメントを出します。たとえば、**prune_interval** および **deadlock_retries** のデフォルトを変更するには、次のようにします。

```
update schema.ibmqrep_applyparms set prune_interval=600, deadlock_retries=10
```

schema は、変更する保管済みパラメーター値を持つ Q アプライ・プログラムを識別します。

関連概念:

- 271 ページの『Q アプライ・パラメーターの変更 -- 概要』
- 19 ページの『Q アプライ・プログラム』

関連資料:

- 452 ページの『IBMQREP_APPLYPARMS 表』

Q アプライ・プログラムの実行中のパラメーターの動的変更

Q アプライ・プログラムは、トランザクションをターゲットに適用している間にも、その振る舞いを変更することができます。Q アプライ・プログラムは、新しい設定をほとんど即時に使用し始めますが、変更は IBMQREP_APPLYPARMS コントロール表には保管されません。Q アプライ・プログラムを停止して再始動すると、コントロール表に保管された値が使用されます。

プログラムの実行中に、以下の Q アプライ・パラメーターを変更できます。

- **autostop**
- **logreuse**
- **logstdout**
- **monitor_interval**
- **monitor_limit**
- **prune_interval**
- **term**
- **trace_limit**
- **deadlock_retries**

値を変更しても、その影響が即時に反映されない場合があります。**asnlccmd chgparms** コマンドがユーザーまたはレプリケーション・センターによって発行された時刻と、Q アプライ・プログラムがその操作を変更するまでの間に、1、2 秒の遅延が発生する可能性があります。

前提条件:

変更するパラメーターを持つ Q アプライ・プログラムは、実行していなければなりません。

手順:

以下の方法のいずれかを使用して、現行セッションの Q アプライ・プログラムの稼働パラメーターの値を変更してください。

レプリケーション・センター

レプリケーション・センターで、「パラメーターの変更 - Q アプライ・プログラムの実行 (Change Parameters - Running Q Apply Program)」ウィンドウを使用します。ウィンドウをオープンするには、変更するパラメーターを持つ Q アプライ・プログラムを含む Q アプライ・サーバーを右マウス・ボタンでクリックし、「パラメーターの変更 (Change Parameter)」→「Q アプライ・プログラム実行 (Running Q Apply Program)」を選択します。詳細については、オンライン・ヘルプを参照してください。

asnlqcmd システム・コマンド

asnlqcmd chgparms コマンドを使用して、実行中の Q アプライ・プログラムのパラメーターを変更します。

```
asnlqcmd apply_server=server_name apply_schema=schema chgparms parameters
```

ここで、*server_name* は Q アプライ・サーバーの名前、*schema* は実行中の Q アプライ・プログラムを示し、*parameters* は変更する 1 つ以上のパラメーターです。

z/OS コンソールまたは TSO

MODIFY コマンドを使用して、Q アプライ・プログラムの実行中にパラメーター値を変更します。

関連概念:

- 271 ページの『Q アプライ・パラメーターの変更 -- 概要』
- 19 ページの『Q アプライ・プログラム』

関連タスク:

- 339 ページの『JCL を使用した、開始済みの Q レプリケーションおよびイベント発行プログラムの変更』

関連資料:

- 376 ページの『asnlqcmd: 実行中の Q アプライ・プログラムの操作』
- 452 ページの『IBMQREP_APPLYPARMS 表』

受信キュー上のメッセージ処理の開始

次のような理由で、受信キュー上のメッセージの処理を開始するように Q アプライ・プログラムに指示することが必要になる場合があります。

- ターゲットで競合、SQL エラー、または永続デッドロックが発生し、Q アプライ・プログラムの停止または受信キューからの読み取りの停止を求めるプロンプトが出された。

- 受信キューからのメッセージの処理を停止するように Q アプライ・プログラムに指示した。

前提条件:

- Q アプライ・プログラムが実行されていないなければならない。
- 受信キューが I (非アクティブ) 状態になっていないなければならない。

手順:

受信キュー上でのメッセージ処理を開始するには、次のようにしてレプリケーション・センターまたはコマンド行を使用します。

レプリケーション・センター

「受信キューの管理」ウィンドウを使用して、受信キューでのメッセージ処理を開始します。このウィンドウをオープンするには、受信キューが置かれている Q アプライ・サーバーを右マウス・ボタンでクリックして、「管理」→「受信キュー」を選択します。詳細については、オンライン・ヘルプを参照してください。

asnlqcmd command

asnlqcmd startq コマンドを次のように使用して、受信キューでのメッセージ処理を開始します。

```
asnlqcmd apply_server=server_name apply_schema=schema
startq=receive_queue_name
```

ここで、*server_name* は Q アプライ・サーバーの名前であり、*schema* は Q アプライ・プログラムを示し、*receive_queue_name* はメッセージ処理を開始したい受信キューの名前です。

Q アプライ・プログラムがキューからのメッセージの読み取りを開始したことを確認するには、次のいずれかを行います。

- 「受信キューの管理」ウィンドウを使用して、キューの状態が A (アクティブ) に変わっているかどうかを確認する。
- Q アプライ診断ログ・ファイルから、キューで処理が開始したことを示すメッセージがないかどうかチェックする。
- `IBMQREP_RECVQUEUES` コントロール表を調べて、`STATE` 列でキューの状態が A (アクティブ) に変わっているかどうかを確認する。

```
SELECT RECVQ, STATE FROM schema.IBMQREP_RECVQUEUES
WHERE RECVQ = 'receive_queue_name';
```

ここで、*schema* は Q アプライ・プログラムを示し、*receive_queue_name* はチェックする受信キューの名前です。

関連概念:

- 259 ページの『Q アプライ・プログラムの操作 -- 概要』

関連タスク:

- 275 ページの『受信キュー上のメッセージ処理の停止』

関連資料:

- 376 ページの『asnlqcmd: 実行中の Q アプライ・プログラムの操作』

受信キュー上のメッセージ処理の停止

レプリケーション・センターまたはコマンドを使用して、受信キュー上のメッセージの処理を停止するように Q アプライ・プログラムに指示できます。あるキューで処理を停止すると、次のようになります。

- Q アプライ・プログラムは、他の受信キューからのトランザクションの適用を継続する。
- このキューを使用する Q サブスクリプションがアクティブの場合は、処理を停止した受信キューへのメッセージの到着が継続する。

受信キューでメッセージの処理を再開することができ、メッセージが失われることはありません。

受信キュー上でのメッセージ処理を停止すると、Q アプライ・プログラムは IBMQREP_RECVQUEUES 表でキューの状態を I (非アクティブ) に変更します。Q アプライ・プログラムを停止してから再開しても、**asnlqcmd startq** コマンドを発行した場合にのみ、キューからの読み取りが再開されます。

受信キューでメッセージの処理を停止すると、Q アプライ・プログラムは、メモリー内のトランザクションをキューから適用することを終了します。Q アプライ・プログラムは、ターゲットに部分的に適用されたトランザクションをロールバックします。ロールバックされたトランザクションは、Q アプライ・プログラムの次の開始時に処理されます。データの消失はありません。

前提条件:

- Q アプライ・プログラムが実行されていないなければならない。
- 受信キューが A (アクティブ) 状態になっていないなければならない。

手順:

受信キュー上でのメッセージ処理を停止するには、次のようにしてレプリケーション・センターまたはコマンド行を使用します。

レプリケーション・センター

「受信キューの管理」ウィンドウを使用して、受信キューでのメッセージ処理を停止します。このウィンドウをオープンするには、受信キューが置かれている Q アプライ・サーバーを右マウス・ボタンでクリックして、「管理」→「受信キュー」を選択します。詳細については、オンライン・ヘルプを参照してください。

asnlqcmd command

asnlqcmd stopq コマンドを次のように使用して、受信キューでのメッセージ処理を停止します。

```
asnlqcmd apply_server=server_name apply_schema=schema
stopq=receive_queue_name
```

ここで、*server_name* は Q アプライ・サーバーの名前であり、*schema* は受信キューからのメッセージを処理する Q アプライ・プログラムを示し、*receive_queue_name* はメッセージ処理を停止したい受信キューの名前です。

Q アプライ・プログラムがキューからのメッセージの読み取りを停止したことを確認するには、次のいずれかを行います。

- 「受信キューの管理」ウィンドウを使用して、キューの状態が I (非アクティブ) に変わっているかどうかを確認する。
- Q アプライ診断ログ・ファイルから、キューの処理が停止したことを示すメッセージがないかどうかをチェックする。
- IBMQREP_RECVQUEUES コントロール表を調べて、STATE 列でキューの状態が I (非アクティブ) に変わっているかどうかを確認する。

```
SELECT RECVQ, STATE FROM schema.IBMQREP_RECVQUEUES  
WHERE RECVQ = 'receive_queue_name';
```

ここで、*schema* は Q アプライ・プログラムを示し、*receive_queue_name* はチェックする受信キューの名前です。

関連概念:

- 259 ページの『Q アプライ・プログラムの操作 -- 概要』

関連タスク:

- 273 ページの『受信キュー上のメッセージ処理の開始』

関連資料:

- 376 ページの『asnqacmd: 実行中の Q アプライ・プログラムの操作』
- 459 ページの『IBMQREP_RECVQUEUES 表』

Q アプライ・プログラムの停止

Q アプライ・プログラムを停止するには、レプリケーション・センターまたはシステム・コマンドを使用します。Q アプライ・プログラムを停止すると、次のアクションがとられます。

- すべての受信キューからのメッセージの読み取りを停止する
- ターゲットに部分的に適用されているがコミットされていないトランザクションをロールバックする
- 通常の方法でシャットダウンする

Q アプライ・プログラムが停止しても、受信キューでは、実行中の Q キャプチャー・プログラムからのメッセージの収集が継続します。Q アプライ・プログラムを再び開始すると、これらのメッセージの読み取りが開始され、ロールバックされたトランザクションを含むメッセージが再び読み取られます。プログラムは、ターゲットへのトランザクションの適用に戻ります。トランザクションは 1 回だけ適用され、複製されたデータが失われることはありません。

重要:

- ターゲット表のロード中に Q アプライ・プログラムを停止する場合は、Q アプライ・プログラムが再び開始して表がロードされるまで、ターゲット表を更新す

ることをどのアプリケーションにも許可しない。プログラムを再始動すると、ターゲット表の内容が削除され、再び表がロードされます。 Q アプライ・プログラムの停止中にターゲット表に加えられた更新は失われます。

- ロード・プロセス中に、Q アプライ・プログラムは、ターゲット表に対する参照保全制約を除去する。これらの制約は、プログラムが再び開始して表がロードされるまで再適用されません。 Q アプライ・プログラムの停止中ターゲット表に加えられた更新は、参照保全のチェックが行われません。

ヒント: Q アプライ・プログラムでの単一の受信キューからのメッセージの読み取りを停止したい場合は、**stopq** コマンドを使用します。

前提条件:

停止したい Q アプライ・プログラムが実行されていないなければならない。

手順:

次の方法のいずれかを使用して、Q アプライ・プログラムを停止します。

レプリケーション・センター

「Q アプライの停止」ウィンドウを使用して、Q アプライ・プログラムを停止する。このウィンドウをオープンするには、停止したい Q アプライ・プログラムが含まれている Q アプライ・サーバーを右マウス・ボタンでクリックして、「**Q アプライ・プログラムの停止**」を選択します。詳細については、オンライン・ヘルプを参照してください。

asnlqcmd システム・コマンド

次の **asnlqcmd stop** コマンドを使用して、Q アプライ・プログラムを停止できます。

```
asnlqcmd apply_server=server_name apply_schema=schema stop
```

ここで、*server_name* は Q アプライ・サーバーの名前であり、*schema* は実行中の Q アプライ・プログラムを示します。

Windows サービス

Windows オペレーティング・システム上で DB2 レプリケーション・サービスを作成し、Windows Service Control Manager または **net stop** コマンドを使用して、Q アプライ・プログラムを停止できます。

z/OS コンソールまたは TSO

MODIFY コマンドを使用して、Q アプライ・プログラムを停止します。

関連概念:

- 259 ページの『Q アプライ・プログラムの操作 -- 概要』
- 163 ページの『Q レプリケーションのターゲット表のロードのオプション -- 概要』

関連タスク:

- 259 ページの『Q アプライ・プログラムの開始』
- 256 ページの『Q キャプチャー・プログラムの停止』
- 345 ページの『レプリケーション・サービスの停止』

- 339 ページの『JCL を使用した、開始済みの Q レプリケーションおよびイベント発行プログラムの変更』

関連資料:

- 376 ページの『asnqacmd: 実行中の Q アプライ・プログラムの操作』

第 19 章 Q レプリケーションおよびイベント発行プログラムに関するレポートの表示

Q レプリケーションおよびイベント発行プログラムに関するレポートの表示 -- 概要

レプリケーション・センターおよびシステム・コマンドを使用して、Q キャプチャー・プログラム、Q アプライ・プログラム、およびレプリケーション・アラート・モニターの状況をチェックできます。レプリケーション・センターを使用して、プログラムが保管している、パフォーマンスに関連した履歴データも表示できます。

以下のトピックでは、表示できるレポートのタイプの詳細を示し、パフォーマンスの問題、待ち時間、および例外を理解するうえで重要な 2 つの概念について説明します。

- 『Q レプリケーションおよびイベント発行プログラムの状況のチェック』
- 281 ページの『Q キャプチャー・プログラム、Q アプライ・プログラム、およびレプリケーション・アラート・モニター・プログラムのスレッド』
- 283 ページの『Q レプリケーションおよびイベント発行プログラムの履歴およびパフォーマンス・データ』
- 285 ページの『待ち時間』
- 288 ページの『例外』

Q レプリケーションおよびイベント発行プログラムの状況のチェック

Q キャプチャー・プログラムおよび Q アプライ・プログラムの状況は、レプリケーション・センターまたはシステム・コマンドを使用してチェックできます。レプリケーション・アラート・モニターの現行の状況は、システム・コマンドを使用してチェックできます。

プログラム・スレッドの状況に関するメッセージは、レプリケーション・センターとコマンドのどちらでも表示できます。これらのメッセージは、プログラムが正しく作動しているかどうかを判別するのに役立てることができます。

手順:

Q レプリケーションおよびイベント発行プログラムの状況をチェックするには、次の方法のいずれかを使用します。

レプリケーション・センター

「状況のチェック」ウィンドウを使用して、Q キャプチャー・プログラムまたは Q アプライ・プログラムの現在の状況をチェックします。(レプリケーション・センターを使用してレプリケーション・アラート・モニターの状況をチェックすることはできません。)

- Q キャプチャーの状況を表示するには、チェックする Q キャプチャー・プログラムが入っている Q キャプチャー・サーバーを右マウス・ボタンでクリックし、「状況のチェック」を選択します。
- Q アプライの状況を表示するには、チェックする Q アプライ・プログラムが入っている Q アプライ・サーバーを右マウス・ボタンでクリックし、「状況のチェック」を選択します。

詳細については、オンライン・ヘルプを参照してください。

コマンド行

以下のコマンドを使用して、プログラムの状況をチェックします。

asnlccmd システム・コマンド (Q キャプチャー・プログラム)

各 Q キャプチャー・スレッドの状況を示すメッセージを表示するには、次のように **asnlccmd status** コマンドを使用します。

```
asnlccmd capture_server=server_name capture_schema=schema status
```

server_name は、Q キャプチャー・プログラムが実行しているデータベースまたはサブシステムの名前で、*schema* は、チェックする Q キャプチャー・プログラムを示します。

asnlacmd システム・コマンド (Q アプライ・プログラム)

各 Q アプライ・スレッドの状況を示すメッセージを表示するには、次のように **asnlacmd status** コマンドを使用します。

```
asnlacmd apply_server=server_name apply_schema=schema status
```

server_name は、Q アプライ・プログラムが実行しているデータベースまたはサブシステムの名前で、*schema* は、チェックする Q アプライ・プログラムを示します。

asnlmcmd システム・コマンド (レプリケーション・アラート・モニター)

各モニター・スレッドの状況を示すメッセージを表示するには、次のように **asnlmcmd status** コマンドを使用します。

```
asnlmcmd monitor_server=server_name monitor_qual=qualifier status
```

server_name は、モニターが実行しているデータベースまたはサブシステムの名前で、*qualifier* は、チェックするモニターを示します。

関連概念:

- 279 ページの『Q レプリケーションおよびイベント発行プログラムに関するレポートの表示 -- 概要』
- 281 ページの『Q キャプチャー・プログラム、Q アプライ・プログラム、およびレプリケーション・アラート・モニター・プログラムのスレッド』

関連資料:

- 384 ページの『asnlmcmd: 実行中のレプリケーション・アラート・モニターの操作』
- 368 ページの『asnlccmd: 実行中の Q キャプチャー・プログラムの操作』

Q キャプチャー・プログラム、Q アプライ・プログラム、およびレプリケーション・アラート・モニター・プログラムのスレッド

Q レプリケーションおよびイベント発行プログラムの状況をチェックする際は、プログラム・スレッドに関して受信するメッセージから、プログラムが正しく作動しているかどうかを判別できます。以下のリストでは、Q キャプチャー・プログラム、Q アプライ・プログラム、およびレプリケーション・アラート・モニター・プログラムのスレッドを示しています。

Q キャプチャー・プログラム

Q キャプチャー・プログラムには、以下のスレッドがあります。

ワーカー

DB2[®] リカバリー・ログを読み取り、サブスクライブする表の変更をキャプチャーし、メモリー内のトランザクションをトランザクション・メッセージとして再ビルドして、そのメッセージを送信キューに書き込みます。

管理 Q アプライ・プログラムやユーザー・アプリケーションによって管理キューに書き込まれる制御メッセージを処理するほか、エラー・ロギングやモニターにも使用されます。

整理 一部の Q キャプチャー・コントロール表から古いデータを削除します。

Hold1 1 つのサーバー上で同一のスキーマを持つ 2 つの Q キャプチャー・プログラムが実行することを防止し、Q キャプチャー・プログラムに送信されるシグナルを処理します。

ワーカー、管理、および整理スレッドは、通常は、実行中または休止の状態になります。Hold1 スレッドは、通常は待ち状態です。ワーカー・スレッドが実行中の状態であるのに、データがキャプチャーされていない場合は、レプリケーション・センターの「Q キャプチャー・メッセージ」ウィンドウか、IBMQREP_CAPTRACE 表で、データがキャプチャーされていない理由を説明しているメッセージを探してください。

Q アプライ・プログラム

Q アプライ・プログラムには、以下のスレッドがあります。

ブラウザー

受信キューからトランザクション・メッセージを読み取り、トランザクション間の依存関係を保守し、1 つ以上のエージェント・スレッドを立ち上げます。Q アプライ・プログラムは、各受信キューに対して 1 つのブラウザー・スレッドを立ち上げます。

エージェント

メモリー内のトランザクションを再ビルドし、それらをターゲットにアプライします。トランザクションの並行処理に使用するエージェント・スレッドの数は、レプリケーション・キュー・マップの作成時に設定します。

ハウスキーピング

データを保管および削除することによって、Q アプライのコントロール表を保守します。

ブラウザー・スレッド、エージェント・スレッド、およびハウスキーピング・スレッドは、通常は、実行中の状態になります。エージェント・スレッドが実行中の状態であるのに、データがアプライされていない場合は、「Q アプライ・メッセージ」ウィンドウまたは `IBMQREP_APPLYTRACE` 表を使用して、データがアプライされていない理由を説明しているメッセージを探してください。

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・プログラムには、以下のスレッドがあります。

ワーカー

コントロール表をモニターして、ユーザー定義アラート条件と合致しているかどうか調べ、アラート通知を送信します。

管理 内部メッセージおよびエラー・メッセージを処理します。

シリアライゼーション

モニター・プログラムのスレッド間のシグナルを処理します。

ワーカーおよび管理スレッドは、通常は作業中または休止の状態になります。シリアライゼーション・スレッドは、通常は待ち状態です。ワーカー・スレッドが作業中の状態であるのに、予想されるアラート通知が受信されない場合は、`IBMSNAP_MONTRACE` コントロール表および `IBMSNAP_MONTRAIL` コントロール表で、アラートが送信されない理由を説明しているメッセージおよびその他の情報を探してください。

Q キャプチャー・プログラム、Q アプライ・プログラム、およびレプリケーション・アラート・モニター・プログラムの状況をチェックしたとき、メッセージがスレッドの通常の状態を示していない場合は、表 15 で説明されているアクションを取ることが必要になる可能性があります。

表 15. Q レプリケーションおよびイベント発行プログラムのスレッドの状態の説明と、推奨アクション

スレッドの状態	説明	推奨アクション
実行中	スレッドがアクティブ状態で処理中です。	なし。
存在	スレッドは存在していますが開始できません。	IBM ソフトウェア・サポートに連絡してください。
開始済み	スレッドが開始していますが、初期化できません。	スレッドが多すぎる、または十分な処理能力がないなどのシステム・リソースの潜在的な問題を調査してください。
初期化	スレッドは初期化されていますが作業できません。	IBM ソフトウェア・サポートに連絡してください。
休止	スレッドはスリープしており、実行すべき作業ができれば起動します。	なし。

表 15. Q レプリケーションおよびイベント発行プログラムのスレッドの状態の説明と、推奨アクション (続き)

スレッドの状態	説明	推奨アクション
停止	スレッドは実行されていません。	レプリケーション・センターの「Q キャプチャー・メッセージ」ウィンドウ、「Q アプライ・メッセージ」ウィンドウ、または「モニター・メッセージ」ウィンドウを使用して、スレッドが停止した理由を説明しているメッセージを探してください。あるいは、IBMQREP_CAPTRACE、IBMQREP_APPLYTRACE、または IBMSNAP_MONTRACE コントロール表で同様のメッセージを探してください。

関連概念:

- 291 ページの『レプリケーション・アラート・モニター』
- 19 ページの『Q アプライ・プログラム』
- 15 ページの『Q キャプチャー・プログラム』
- 279 ページの『Q レプリケーションおよびイベント発行プログラムに関するレポートの表示 -- 概要』

関連タスク:

- 279 ページの『Q レプリケーションおよびイベント発行プログラムの状況のチェック』

Q レプリケーションおよびイベント発行プログラムの履歴およびパフォーマンス・データ

レプリケーション・センターを使用して、Q キャプチャー・プログラム、Q アプライ・プログラム、またはレプリケーション・アラート・モニターについての、さまざまな履歴およびパフォーマンス・データをチェックできます。

プログラムは、このデータをコントロール表に保管します。これには、IBMQREP_CAPMON、IBMQREP_CAPQMON、IBMQREP_CAPTRACE、IBMQREP_APPLYMON、IBMQREP_EXCEPTIONS、IBMQREP_APPLYTRACE、IBMSNAP_ALERTS、および IBMSNAP_MONTRACE の各コントロール表があります。各表に保管されるデータの量は、表が整理される頻度に基づいて決まります。

推奨: トラブルシューティングやパフォーマンス評価のためにデータを検査できるように、これらの表には少なくとも 1 週間分のデータを保存しておいてください。

レプリケーション・センターには、データを表示し、必要に合わせて調整するためのウィンドウがあります。スプレッドシート・プログラムで使用するために、データを印刷またはエクスポートすることもできます。

表 16 は、Q レプリケーションおよびイベント発行プログラムのパフォーマンスまたは履歴について尋ねられる代表的な質問と、その質問の答えを得るのに役立つデータを提供するレプリケーション・センターのウィンドウを示しています。

表 16. レプリケーション・センター内の履歴およびパフォーマンス情報の保管場所

質問	使用するウィンドウ
Q キャプチャー・プログラム、Q アプライ・プログラム、およびモニター・プログラムからの最新のメッセージはどのようなものか？	Q キャプチャー・メッセージ Q アプライ・メッセージ モニター・メッセージ
<ul style="list-style-type: none"> 送信キューに書き込まれた行およびトランザクション数は？ Q サブスクリプションまたは XML 発行の定義のために、キャプチャーされたが送信キューに書き込まれなかった行およびトランザクション数は？ ファイルにスピルされたトランザクション数は？ 処理された最大のトランザクションは？ Q キャプチャー・プログラムが使用するメモリー量は？ 	Q キャプチャー・スループット
<ul style="list-style-type: none"> アプライされた行およびトランザクション数は？ 競合またはエラーのためにアプライされなかった行数は？ ロック・タイムアウトおよびデッドロックのために行が再試行された回数は？ 競合のために並列にアプライされなかったトランザクション数は？ 参照保全制約に違反したトランザクション数は？ 参照保全制約のために行が再試行された回数は？ 一時予備キューに書き込まれた行数は？ アプライされた予備行数は？ 	Q アプライ・スループット
DB2 リカバリー・ログを読み取るまでの Q キャプチャー・プログラムのおおよその待ち時間は？	Q キャプチャー待ち時間
以下の経過時間	待ち時間
<ul style="list-style-type: none"> トランザクションがソース・データベースでコミットされてから、送信キューに書き込まれるまでの時間は？ トランザクションが送信キューに書き込まれてから、受信キューから取り出されるまでの時間は？ トランザクションが受信キューから取り出されてから、ターゲットでコミットされるまでの時間は？ トランザクションがソースでコミットされてから、ターゲットでコミットされるまでの時間は？ 	
一部の行がアプライされない原因となった競合や SQL エラーの種類は？ その詳細は？	例外
レプリケーション・アラート・モニターによって発行されたアラートは？	アラートの表示

レポート・ウィンドウをオープンするには、Q キャプチャー・サーバー、Q アプライ・サーバー、またはモニター修飾子を右マウス・ボタンでクリックして、以下のいずれかのメニュー項目を選択します。

Q キャプチャー・サーバー

レポート →

メッセージ

Q キャプチャー・スループット

Q キャプチャー待ち時間

Q アプライ・サーバー

レポート →

メッセージ

Q アプライ・スループット

待ち時間

例外

モニター修飾子

モニター・メッセージの表示

アラートの表示

各ウィンドウの使用法の詳細については、オンライン・ヘルプを参照してください。

関連概念:

- 288 ページの『例外』
- 285 ページの『待ち時間』
- 291 ページの『レプリケーション・アラート・モニターによるレプリケーションのモニター -- 概要』
- 19 ページの『Q アプライ・プログラム』
- 15 ページの『Q キャプチャー・プログラム』
- 279 ページの『Q レプリケーションおよびイベント発行プログラムに関するレポートの表示 -- 概要』
- 448 ページの『Q アプライ・コントロール表の詳細な構造 -- 概説』
- 425 ページの『Q キャプチャー・コントロール表の詳細な構造 -- 概説』

待ち時間

待ち時間とは、Q キャプチャー・サーバーから Q アプライ・サーバーに対して複製するため、トランザクションでかかる時間の尺度のことです。Q キャプチャーおよび Q アプライ・プログラムは、レプリケーション・プロセスのさまざまな段階間でかかる待ち時間を追跡するためのパフォーマンス・データを保管します。これらの統計は、問題を見極め、環境を調整するのに役立ちます。

レプリケーション・センターの「待ち時間 (Latency)」ウィンドウを使用して、DB2® リカバリー・ログと送信キュー、送信キューと受信キュー、および受信キューとターゲット表間でトランザクションが移動するのにかかる時間を確認できま

す。「Q キャプチャー待ち時間 (Q Capture Latency)」ウィンドウを使用すると、Q キャプチャー・プログラムがログへの変更にどの程度追いついているかを大まかな尺度で表示することができます。

注: リモート Q キャプチャーおよび Q アプライ・サーバー間で複製されるトランザクションが関係する待ち時間の尺度は、ソース表とターゲット表の間の刻時の差に影響を受けることがあります。正しい尺度とするために、刻時機構は確実に同期化してください。

以下のセクションでは、待ち時間の各尺度について詳述します。

Q キャプチャー待ち時間

Q キャプチャー待ち時間は、特定の時点と、最後にコミットされたトランザクションのタイム・スタンプとの間の差を測定します。この尺度は、IBMQREP_CAPMON コントロール表の MONITOR_TIME および CURRENT_LOG_TIME 列の値を使用します。集合として調査されると、これらの待ち時間測定は、Q キャプチャー・プログラムがデータベース・ログにどの程度追いついているかを判別するのに役立ちます。

たとえば、Q キャプチャー・プログラムが午前 10 時に IBMQREP_CAPMON 表 (MONITOR_TIME) にパフォーマンス・データの行を挿入し、最後にコミットされたトランザクション (CURRENT_LOG_TIME) のタイム・スタンプが午前 9 時 59 分だった場合は、Q キャプチャー待ち時間は 1 分になります。ログ読み取りパフォーマンスを改善するには、追加の Q キャプチャー・スキーマの作成、およびいくつかの Q サブスクリプションまたは XML 発行を新規スキーマに移動することを考慮してください。各追加スキーマは、別のワーカー・スレッドを追加してログを読み取ります。

Q キャプチャー・トランザクション待ち時間

Q キャプチャー・トランザクション待ち時間は、Q キャプチャー・プログラムが DB2 リカバリー・ログ中でトランザクションのコミット・ステートメントを読み取った時点から、そのトランザクションを含むメッセージが送信キュー上に挿入されるまでの時間を測定します。この統計により、Q キャプチャー・プログラムがメモリーにトランザクションを再アSEMBルし、Q サブスクリプションまたは XML 発行の設定に基づいて行および列をフィルターに掛けてから、トランザクション・メッセージをキューに入れるのにかかる時間を推定できます。Q キャプチャー・トランザクション待ち時間を削減するには、以下の調整の実施を考慮してください。

- Q キャプチャー・プログラムが割り振るメモリーの合計量を設定する、MEMORY_LIMIT パラメーターの値を増やす。
- レプリケーション・キュー・マップまたは発行キュー・マップの作成時に定義される、MAX_MSG_SIZE パラメーターを上げる。このパラメーターは、Q キャプチャー・プログラムが、各送信キューごとにトランザクション・メッセージを作成するために割り振るメモリーの量を設定する。最大メッセージ・サイズが小さすぎる場合、Q キャプチャー・プログラムはトランザクションを複数のメッセージに分割するので、より長い処理時間と待ち時間を増やすことが必要になります。

キュー待ち時間

Q 待ち時間は、Q キャプチャー・プログラムが送信キューにトランザクシ

オンを挿入してから、Q アプライ・プログラムが受信キューからトランザクションを受け取るまでの間の時間を測定します。この統計により、WebSphere® MQ パフォーマンスを推定できる可能性があります。分散環境ではネットワーク・パフォーマンスも考慮する要因となることがあります。また、コミット・インターバルが長いと、Q キャプチャー・プログラムに、送信キューに入れられているいくつかのトランザクションのコミット前に待機するようにプロンプトが出されます。この遅延が、キュー待ち時間の原因となることがあります。

Q アプライ待ち時間

Q アプライ待ち時間とは、Q アプライ・プログラムが受信キューからトランザクションを受け取った後に、トランザクションがターゲット表にアプライされる時間を測定したものです。受信キューに指定したエージェント・スレッドが多いほど、この数値は小さくなるはずですが、ただし、依存するすべての以前のトランザクションがアプライされるまで、Q アプライ・プログラムは、従属表と関連するトランザクションのアプライを遅らせることに注意してください。この遅延によって、Q アプライ待ち時間が増えることがあります。

エンドツーエンド待ち時間

エンドツーエンド待ち時間は、Q キャプチャー・プログラムがトランザクション・コミット・ステートメントのログ・レコードを読み取ってから、Q アプライ・プログラムがターゲットでトランザクションをコミットするまでの間の時間を測定します。この統計は、Q レプリケーション待ち時間の全体的な尺度になります。

「待ち時間 (Latency)」ウィンドウをオープンするには、統計を表示する Q アプライ・プログラムを含む Q アプライ・サーバーを右マウス・ボタンでクリックして、「レポート (Reports)」→「待ち時間 (Latency)」を選択します。

「Q キャプチャー待ち時間 (Q Capture Latency)」ウィンドウをオープンするには、統計を表示する Q キャプチャー・プログラムを含む Q キャプチャー・サーバーを右マウス・ボタンでクリックして、「レポート (Reports)」→「Q キャプチャー待ち時間 (Q Capture Latency)」を選択します。

詳細については、オンライン・ヘルプを参照してください。

関連概念:

- 19 ページの『Q アプライ・プログラム』
- 15 ページの『Q キャプチャー・プログラム』
- 279 ページの『Q レプリケーションおよびイベント発行プログラムに関するレポートの表示 -- 概要』

関連タスク:

- 95 ページの『レプリケーション・キュー・マップの作成』

関連資料:

- 449 ページの『IBMQREP_APPLYMON 表』
- 427 ページの『IBMQREP_CAPMON 表』
- 432 ページの『IBMQREP_CAPQMON 表』

例外

例外は、競合または SQL エラーが原因で、Q アプライ・プログラムがターゲットにアプライしなかった行です。Q アプライ・プログラムは、競合またはエラーに関する詳細とともにこれらの行を IBMQREP_EXCEPTIONS 表に保管します。

Q アプライ・プログラム以外のアプリケーションがターゲットを更新する場合、競合が発生する可能性があります。たとえば、Q アプライ・プログラムがすでに存在する行への挿入を試行した場合などです。

SQL エラーには、広範囲の起こりうる問題を対象にしています。たとえば、ターゲット・サーバーのリカバリー・ログがいっぱいになった場合に、Q アプライ・プログラムが行のアプライを試行すると、SQL エラーが生成されます。

レプリケーション・センターの「例外」ウィンドウを使用して、問題の行に関する詳細を表示できます。また、Q アプライ・プログラムがコントロール表に保管するデータを使用して、アプライされなかった行をリカバリーしたり、レプリケーション環境での起こりうる問題に関する詳細を入手できます。

以下のリストでは、Q アプライ・プログラムが行をアプライできなくなる問題のタイプを示しています。

予期しない SQL エラー

Q サブスクリプションで許容可能と定義されていない SQL エラーの原因となった行。

許容可能な SQL エラー

許容可能と定義されている SQL エラーの原因となった行。Q サブスクリプションの作成時に、許容可能な SQL エラーを定義します。

値に基づく競合

ターゲットに存在しない行の削除または更新、あるいはすでに存在する行の挿入といった、競合の原因となった行。Q アプライ・プログラムは、使用された競合検出のタイプ (キー値だけの検査、キー列と変更された非キー列の検査、キー列とすべての非キー列の検査) を記録します。また保管されているものの、その行にアプライが行われたかどうかについても記録します。Q サブスクリプションの作成時に、競合が発生した場合でも、Q アプライ・プログラムがいくつかの行をアプライするように指定します。

バージョンに基づく競合

古い行からの値で行を更新したり、同じキーを持つ新しい行がすでにターゲットに存在していた場合に行を挿入しようとするといった、ピアツーピア・レプリケーションにおける競合の原因となった行。

アプライできなかった行を Q アプライ・プログラムが保管する時に、エラーまたは競合が発生した時間、エラーの理由、および問題が生じた SQL 操作のタイプを記録します。Q アプライ・プログラムは、受信キューおよび Q サブスクリプションの名前、およびトランザクションに関するソース・コミット情報も保管します。

予期しない SQL エラーおよび許容可能な SQL エラーの場合、Q アプライ・プログラムは、トランザクションの実行のために使用される SQLCA 構造からのエラー・メッセージ・トークンとともに、そのトランザクションで DB2[®] によって戻さ

れる SQL コードおよび SQL 状態コードを保管します。値に基づく競合およびバージョンに基づく競合の場合、Q アプライ・プログラムは行をアプライできなかった理由を記録します。

アプライされない行は、Q キャプチャー・プログラムがイベント発行での XML データ・メッセージの作成に使用するのと同じタグおよびスキーマを使用し、XML 形式で保管されます。(スキーマとは、どのタグが XML 文書で正しいかを定義するファイルです。Q キャプチャー・プログラムからのメッセージの場合、スキーマ文書は mqcap.xsd です。) アプリケーションがデータをリカバリーできるように、さらには、タグ付けによって、列の名前と値、および行の操作のタイプ (挿入、更新、または削除) の表示を容易にするため、行は XML 形式で保管されます。

ピアツーピア・レプリケーションでは、競合または SQL エラーが発生したサーバーでアプライされなかった行を Q アプライ・プログラムが保管します。

「例外」ウィンドウをオープンするには、例外を表示する Q アプライ・プログラムが含まれている Q アプライ・サーバーを右マウス・ボタンでクリックして、「レポート」→「例外」を選択します。

関連概念:

- 19 ページの『Q アプライ・プログラム』
- 279 ページの『Q レプリケーションおよびイベント発行プログラムに関するレポートの表示 -- 概要』
- 118 ページの『Q レプリケーションのエラー・オプション』

関連資料:

- 457 ページの『IBMQREP_EXCEPTIONS 表』

第 20 章 レプリケーション・アラート・モニターによるレプリケーションのモニター

レプリケーション・アラート・モニターによるレプリケーションのモニター -- 概要

レプリケーション・アラート・モニターを使用して、SQL レプリケーション環境、Q レプリケーション環境、またはイベント発行環境をモニターできます。以下のトピックでは、レプリケーション・アラート・モニターの動作について説明しています。

- 『レプリケーション・アラート・モニター』
- 294 ページの『レプリケーション・アラート・モニターのアラート条件および通知 -- 概要』
- 300 ページの『レプリケーション・アラート・モニターのセットアップ』
- 303 ページの『レプリケーション・アラート・モニターの操作』

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、レプリケーション環境の状況をチェックするプログラムです。レプリケーション・アラート・モニターは、実行中にレプリケーションの状況を自動的にチェックし、レプリケーション環境で発生した特定の条件について通知します。例えば、SQL レプリケーションの場合、レプリケーション・アラート・モニターは、アプライ・プログラムの終了時に通知できます。同様に、Q レプリケーションの場合、レプリケーション・アラート・モニターは、Q キャプチャー・プログラムが Q サブスクリプションを非アクティブ化すると通知できます。

以下の方式を使用すると、レプリケーション環境の状況を手動でチェックできます。

- キャプチャー・プログラム、Q キャプチャー・プログラム、アプライ・プログラム、または Q アプライ・プログラムに関する統計を報告するレプリケーション・センターのウィンドウを表示できる。
- コントロール表に対して SELECT ステートメントを実行し、これらのプログラムの操作に関する統計を表示できる。

いずれの場合も、統計はいつでも使用できますが、手動で検索しなければなりません。多数のサーバー上で継続的にレプリケーション・アクティビティをモニターするには、レプリケーション環境に関する統計を手動でチェックする方法は効果的ではありません。レプリケーション・アラート・モニターは、ご使用のオペレーティング・システムすべてにわたるシステム環境を自動的にモニターします。レプリケーション・アラート・モニターは、サーバー上でレプリケーションをチェックし、注意が必要な条件を自動的にアラートします。

レプリケーション・アラート・モニターは以下のレプリケーション環境をモニターできます。

- SQL レプリケーション
- Q レプリケーション
- イベント発行

レプリケーション・アラート・モニターを使用して、以下のレプリケーション・プログラムをモニターできます。

- キャプチャー・プログラム (SQL レプリケーション)
- アプライ・プログラム (SQL レプリケーション)
- Q キャプチャー・プログラム (Q レプリケーションまたはイベント発行)
- Q アプライ・プログラム (Q レプリケーション)

レプリケーション・アラート・モニターを構成する方法は 2 つあります。1 つのモニターを実行することも、複数のモニターを実行することもできます。一般的には、少数のレプリケーション・プログラムをモニターする場合には、1 つのモニターを使用します。多数のレプリケーション・プログラムをモニターしたり、特定のプログラムのモニターを優先順位付けしたり、モニターの作業負荷を分割したりするには、追加のモニターを使用します。複数のモニターをセットアップすることは、レプリケーション・アラート・モニターをクローン作成することに似ています。モニターという、個々が固有で独立しているが似ているレプリケーション・アラート・モニターを作成し、システム中のサーバーをモニターします。これらのモニターは相互に通信しませんが、個々がモニターしているサーバーに関するアラートを送信します。したがって、これらのモニターは集合としてシステム中のすべてのサーバーに関するアラートを送信します。複数のモニターをセットアップすると、個々のモニターのコントロール情報は、モニターに割り当てられているサーバー上に保管されます。

1 つのモニターをセットアップすると、すべてのコントロール情報が 1 つのサーバー上に保管されます。個々のモニターは複数のレプリケーション・プログラムをモニターできますが、モニターが個々のサーバー上で一度にチェックするアラートは 1 つのみです。いずれか 1 つのサーバーに戻る前に、モニター対象の他のサーバーをすべてチェックしなければなりません。

多数のモニターをセットアップする場合と 1 つのモニターを構成する場合のどちらでも、コントロール情報が含まれるサーバーのことをモニター・コントロール・サーバーといいます。

レプリケーション・アラート・モニターのコンポーネントについて説明する用語を以下に示します。

モニター

モニターは、レプリケーション・アラート・モニターの 1 つのインスタンスまたはオカレンスです。モニターをセットアップして、1 つ以上のサーバー上で実行しているレプリケーション・プログラムの状況をチェックできます。個々のモニターは、割り当て先の 1 つ以上のサーバー上のレプリケーション・アクティビティをチェックします。

モニター修飾子

モニター修飾子は、ユーザー指定のモニター名です。すべてのモニターにユニークなモニター修飾子があります。

モニター・コントロール・サーバー

モニター・コントロール・サーバーとは、レプリケーション・アラート・モニターのコントロール情報が含まれるサーバーのことです。

アラート

アラートとは、レプリケーション環境中のイベントや条件に関して知らせる通知のことです。レプリケーション・アラート・モニターは、E メールかページャーを使用してアラートを送信します。

アラート条件

アラート条件とは、レプリケーション・アラート・モニターがアラートを送信するレプリケーション環境の条件のことです。アラート条件には 3 種類あり、それは状況によって起動されるアラート条件、イベントによって起動されるアラート条件、およびしきい値によって起動されるアラート条件です。

状況によって起動されるアラート条件

状況アラート条件は、レプリケーション・プログラムの状況について通知します。例えば、APPLY_STATUS アラート条件を指定すると、レプリケーション・アラート・モニターは、アプライ・プログラムが実行していない場合にアラートを送信します。

イベントによって起動されるアラート条件

イベント・アラート条件は、レプリケーション中に特定のイベントが発生した時点で通知します。例えば、QAPPLY_ERRORS アラート条件を指定すると、レプリケーション・アラート・モニターは、Q アプライ・プログラムが IBMQREP_APPLYTRACE 表にエラーを記録した時点でアラートを送信します。

しきい値によって起動されるアラート条件

しきい値アラート条件は、レプリケーション環境中でしきい値を超過した時点で通知します。例えば、QCAPTURE_MEMORY アラート条件を指定すると、レプリケーション・アラート・モニターは、Q キャプチャー・プログラムの使用メモリーがしきい値で許可されている量を超えた時点でアラートを送信します。

連絡先 連絡先とは、レプリケーション・アラート・モニターからアラートを受信する人物の E メール・アドレスかページャー・アドレスのことです。

連絡先グループ

連絡先グループとは、同じアラートを受信する連絡先の集合のことです。

レプリケーション・アラート・モニターは、DB2[®] UDB for Linux、UNIX[®]、Windows[®]、z/OS[™] オペレーティング・システム上のサーバーをモニターします。レプリケーション・アラート・モニターは、コントロール表が DB2 レプリケーションのバージョン 8 アーキテクチャー以上になっている DB2 UDB レプリケーション・プログラムをモニターするために使用できます。

制約事項:

- iSeries™ サーバーの場合、レプリケーション・アラート・モニターは Linux、UNIX、または Windows サーバー上で実行しなければならない。この場合、レプリケーション・アラート・モニターは、リモート側で iSeries サーバーをモニターしなければなりません。
- DB2 for iSeries サーバー上でモニター・コントロール・サーバーをセットアップできない。
- レプリケーション・アラート・モニターは、フェデレーテッド・データベース・システムでソースとして使用される、DB2 以外のリレーショナル・データベースに関連するトリガーはモニターしません。

関連概念:

- 291 ページの『レプリケーション・アラート・モニターによるレプリケーションのモニター -- 概要』

レプリケーション・アラート・モニターのアラート条件および通知

レプリケーション・アラート・モニターのアラート条件および通知 -- 概要

以下のトピックには、レプリケーション・アラート・モニターのアラート条件に関する情報が記載されています。

- 『レプリケーション・アラート・モニターのアラート条件』
- 298 ページの『レプリケーション・アラート条件の E メール通知』
- 299 ページの『レプリケーション中のアラートを送信する ASNMAIL exit ルーチン』

関連概念:

- 291 ページの『レプリケーション・アラート・モニターによるレプリケーションのモニター -- 概要』

レプリケーション・アラート・モニターのアラート条件

アラート条件とは、モニターがアラートを送信するレプリケーション環境の条件のことです。アラートとは、アラート条件を引き起こした状況、イベント、またはしきい値を記述したメッセージのことです。関連するパラメーター値を報告するアラートもあります。例えば、QCAPTURE_MEMORY アラート条件のメッセージは、Q キャプチャー・プログラムが使用しているメモリーの量と、超過したメモリーしきい値を報告します。

以下の表は、レプリケーション環境をモニターするために使用できるアラート条件を説明します。

- 295 ページの『Q キャプチャー・プログラムのアラート条件』
- 295 ページの『Q アプライ・プログラムのアラート条件』
- 296 ページの『キャプチャー・プログラムのアラート条件』
- 297 ページの『アプライ・プログラムのアラート条件』

Q キャプチャー・プログラムのアラート条件

表 17 は、Q キャプチャー・プログラムのアラート条件を説明しています。

表 17. Q キャプチャー・プログラムのアラート条件

アラート条件	説明
QCAPTURE_STATUS	レプリケーション・アラート・モニターは、Q キャプチャー・プログラムが実行していないとアラートを送信します。
QCAPTURE_ERRORS	レプリケーション・アラート・モニターは、IBMQREP_CAPTRACE 表の OPERATION 列中に値が 'ERROR' の行を検出するとアラートを送信します。
QCAPTURE_WARNINGS	レプリケーション・アラート・モニターは、IBMQREP_CAPTRACE 表の OPERATION 列中に値が 'WARNING' の行を検出するとアラートを送信します。
QCAPTURE_LATENCY	Q キャプチャー待ち時間とは、データがデータベースに書き込まれた時刻と、Q キャプチャー・プログラムがそのデータを渡した時刻との間の差を測定したものです。レプリケーション・アラート・モニターは、Q キャプチャー待ち時間がユーザー指定のしきい値を超えているとアラートを送信します。Q キャプチャー待ち時間は秒単位で測定されます。
QCAPTURE_MEMORY	レプリケーション・アラート・モニターは、Q キャプチャー・プログラムが使用するメモリーがユーザー指定のしきい値を超えているとアラートを送信します。メモリーは MB 単位で測定されます。
QCAPTURE_TRANSIZE	レプリケーション・アラート・モニターは、Q キャプチャー・プログラムが処理しているトランザクションの使用メモリーがユーザー指定のしきい値を超えているとアラートを送信します。メモリーは MB 単位で測定されます。
QCAPTURE_SUBSINACT	レプリケーション・アラート・モニターは、Q キャプチャー・プログラムが Q サブスクリプションを非アクティブ化するとアラートを送信します。

Q アプライ・プログラムのアラート条件

表 18 は、Q アプライ・プログラムのアラート条件を説明しています。

表 18. Q アプライ・プログラムのアラート条件

アラート条件	説明
QAPPLY_STATUS	レプリケーション・アラート・モニターは、Q アプライ・プログラムが実行していないとアラートを送信します。
QAPPLY_ERRORS	レプリケーション・アラート・モニターは、IBMQREP_APPLYTRACE 表の OPERATION 列中に値が 'ERROR' の行を検出するとアラートを送信します。
QAPPLY_WARNINGS	レプリケーション・アラート・モニターは、IBMQREP_APPLYTRACE 表の OPERATION 列中に値が 'WARNING' の行を検出するとアラートを送信します。
QAPPLY_LATENCY	Q アプライ待ち時間とは、Q アプライ・プログラムが受信キューからトランザクションを受け取った後に、トランザクションがターゲット表にアプライされる時間を測定したものです。レプリケーション・アラート・モニターは、Q アプライ待ち時間がユーザー指定のしきい値を超えているとアラートを送信します。Q アプライ待ち時間はミリ秒単位で測定されます。

表 18. Q アプライ・プログラムのアラート条件 (続き)

アラート条件	説明
QAPPLY_EELATENCY	Q アプライ・エンドツーエンド待ち時間とは、レプリケーションが変更をキャプチャーし、それをターゲット・データベースにアプライするために必要な合計時間を測定したものです。レプリケーション・アラート・モニターは、Q アプライ・エンドツーエンド待ち時間がユーザー指定のしきい値を超えているとアラートを送信します。Q アプライ・エンドツーエンド待ち時間は秒単位で測定されます。
QAPPLY_MEMORY	レプリケーション・アラート・モニターは、Q アプライ・プログラムが使用するメモリーがユーザー指定のしきい値を超えているとアラートを送信します。メモリーは MB 単位で測定されます。
QAPPLY_EXCEPTIONS	レプリケーション・アラート・モニターは、Q アプライ・プログラムの操作中に例外を検索するとアラートを送信します。
QAPPLY_SPILLQDEPTH	レプリケーション・アラート・モニターは、予備キューの満杯率がユーザー指定のしきい値を超えているとアラートを送信します。満杯率はパーセントで表されます。
QAPPLY_QDEPTH	レプリケーション・アラート・モニターは、いずれかのキューの満杯率がユーザー指定のしきい値を超えているとアラートを送信します。満杯率はパーセントで表されます。

キャプチャー・プログラムのアラート条件

表 19 は、キャプチャー・プログラムのアラート条件を説明しています。

表 19. キャプチャー・プログラムのアラート条件

アラート条件	説明
CAPTURE_STATUS	レプリケーション・アラート・モニターは、キャプチャー・プログラムが実行していないとアラートを送信します。
CAPTURE_ERRORS	レプリケーション・アラート・モニターは、IBMSNAP_CAPTRACE 表の OPERATION 列中に値が 'ERROR' の行を検出するとアラートを送信します。
CAPTURE_WARNINGS	レプリケーション・アラート・モニターは、IBMSNAP_CAPTRACE 表の OPERATION 列中に値が 'WARNING' の行を検出するとアラートを送信します。
CAPTURE_LASTCOMMIT	レプリケーション・アラート・モニターは、キャプチャー・プログラムの最後のコミットから経過した時間がユーザー指定のしきい値を超えているとアラートを送信します。経過時間は秒単位で測定されます。
CAPTURE_CLATENCY	現行キャプチャー待ち時間とは、データがデータベースに書き込まれた時刻と、Q キャプチャー・プログラムがそのデータを渡した時刻との間の差を測定したものです。レプリケーション・アラート・モニターは、現行キャプチャー待ち時間がユーザー指定のしきい値を超えているとアラートを送信します。
CAPTURE_HLATENCY	履歴キャプチャー待ち時間とは、最後にモニターがサーバーでアラート条件をチェックした後に測定した、すべてのキャプチャー待ち時間を複合したものです。レプリケーション・アラート・モニターは、履歴キャプチャー待ち時間がユーザー指定のしきい値を超えているとアラートを送信します。

表 19. キャプチャー・プログラムのアラート条件 (続き)

アラート条件	説明
CAPTURE_MEMORY	レプリケーション・アラート・モニターは、キャプチャー・プログラムが使用するメモリーがユーザー指定のしきい値を超えているとアラートを送信します。メモリーは MB 単位で測定されます。

アプライ・プログラムのアラート条件

表 20 は、アプライ・プログラムのアラート条件を説明しています。

表 20. アプライ・プログラムのアラート条件

アラート条件	説明
APPLY_STATUS	レプリケーション・アラート・モニターは、アプライ・プログラムが実行していないとアラートを送信します。
APPLY_SUBSFALING	レプリケーション・アラート・モニターは、サブスクリプションが失敗するとアラートを送信します。
APPLY_SUBSINACT	レプリケーション・アラート・モニターは、サブスクリプションが非アクティブ化されるとアラートを送信します。
APPLY_ERRORS	レプリケーション・アラート・モニターは、IBMSNAP_APPLYTRACE 表の OPERATION 列中に値が 'ERROR' の行を検出するとアラートを送信します。
APPLY_WARNINGS	レプリケーション・アラート・モニターは、IBMSNAP_APPLYTRACE 表の OPERATION 列中に値が 'WARNING' の行を検出するとアラートを送信します。
APPLY_FULLREFRESH	レプリケーション・アラート・モニターは、フル・リフレッシュがあるとアラートを送信します。
APPLY_REJTRANS	レプリケーション・アラート・モニターは、サブスクリプション・セット中でトランザクションがリジェクトされるとアラートを送信します。
APPLY_SUBSDELAY	レプリケーション・アラート・モニターは、サブスクリプション処理の遅延がユーザー指定のしきい値より長くなるとアラートを送信します。
APPLY_REWORKED	レプリケーション・アラート・モニターは、アプライ・プログラムが再処理するサブスクリプション・セット中の行が、ユーザー指定のしきい値より多くなるとアラートを送信します。
APPLY_LATENCY	アプライ・エンドツーエンド待ち時間とは、レプリケーションが変更をキャプチャーし、それをターゲット・データベースにアプライするために必要な合計時間を測定したものです。レプリケーション・アラート・モニターは、アプライ・エンドツーエンド待ち時間がユーザー指定のしきい値を超えているとアラートを送信します。アプライ・エンドツーエンド待ち時間は秒単位で測定されます。

関連概念:

- 291 ページの『レプリケーション・アラート・モニターによるレプリケーションのモニター -- 概要』
- 291 ページの『レプリケーション・アラート・モニター』

関連タスク:

- 305 ページの『レプリケーション・アラート・モニターのアラート条件の選択』

レプリケーション・アラート条件の E メール通知

レプリケーション・アラート・モニターは、アラート条件が発生した際に E メール通知できます。E メール通知の内容は、指定された E メール・アドレスがページャー用のものかどうかによって異なります。次の例は、1 セットのアラートについて、それぞれの場合に予想される情報のタイプを示しています。ページャー以外の装置宛てに送信される E メールは、特定のサーバーで各アラート条件が発生した時刻を示しています。また、各アラート条件が発生した回数、および関連するメッセージも示しています。レプリケーション・アラート・モニターからページャー宛てに送信される E メールには、完全なメッセージの代わりに、アラートを起動したパラメーターのサマリーが含まれます。アラート条件が何回も発生した場合、タイム・スタンプはアラート条件が最後に発生した時刻を反映しています。

ページャー以外の装置宛ての E メール通知の例 (SQL レプリケーション):

```
To:      repladmin@company.com
From:    replmon@server.com
Subject: Monitor: "MONQUAL" Alerts issued
```

```
ASN5129I MONITOR "MONQUAL". The Replication Alert Monitor on
server "WSDB" reports an e-mail alert
```

```
2002-01-20-10.00.00    1 ASN0552E Capture : "ASN" The program
encountered an SQL error. The server name is "CORP". The SQL
request is "PREPARE". The table name "PROD1.INVOICESCD".
The SQLCODE is "-204". The SQLSTATE is "42704". The SQLERRMC
is "PROD1.INVOICESCD". The SQLERRP is "readCD"
```

```
2002-01-20-10.05.00    2 ASN5152W Monitor "MONQUAL". The current
Capture latency exceeds the threshold value. The Capture control
server is "CORP". The schema is "ASN". The Capture
latency is "90" seconds. The threshold is "60" seconds
```

```
2002-01-20-10.05.00    4 ASN5154W Monitor "MONQUAL". The memory
used by the Capture program exceeds the threshold value. The
Capture control server is "CORP". The schema is "ASN".
The amount of memory used is "34" bytes. The threshold is
"30" megabytes.
```

ページャー宛ての E メール通知の例 (SQL レプリケーション):

```
To:      repladmin@company.com
From:    replmon@server.com
Subject: Monitor: "MONQUAL" Alerts issued
```

```
MONQUAL - MONDB
```

```
2002-01-20-10.00.00 ASN0552E 1 CAPTURE-ERRORS - CORP - ASN
2002-01-20-10.05.00 ASN5152W 2 CAPTURE_CLATENCY - CORP - ASN - 90 - 60
2002-01-20-10.05.00 ASN5154W 4 CAPTURE_MEMORY - CORP - ASN - 34 - 30
```

SQL レプリケーションでは、レプリケーション・アラート・モニターは通知を送信する際に、キャプチャー・コントロール・サーバーおよびアプライ・コントロール・サーバー別にアラートをグループ化します。1 つのサーバーがキャプチャー・コントロール・サーバーとアプライ・コントロール・サーバー両方である場合、レプリケーション・アラート・モニターはそのサーバーのすべてのアラートをともにグループ化します。

Q レプリケーションでは、レプリケーション・アラート・モニターは通知を送信する際に、Q キャプチャー・サーバーおよび Q アプライ・サーバー別にアラートをグループ化します。1 つのサーバーが Q キャプチャー・サーバーと Q アプライ・サーバー両方である場合、レプリケーション・アラート・モニターはそのサーバーのすべてのアラートをともにグループ化します。

E メール通知のサイズがその E メールのタイプの制限を超えた場合は、レプリケーション・アラート・モニターは複数の E メールに分けて通知を送信します。通常の E メール通知の最大サイズは 1024 文字です。ページャーの E メール・アドレスの場合の制限は 250 文字です。

ASNMAIL exit ルーチンは、レプリケーション・アラート・モニターに関する E メール通知を送信します。別のアラート処理を行うように、この exit ルーチンを変更できます。例えば、ASNMAIL ユーザー exit ルーチンに、問題管理システムにアラートを保管させることができます。詳細については、『レプリケーション中のアラートを送信する ASNMAIL exit ルーチン』を参照してください。

関連概念:

- 294 ページの『レプリケーション・アラート・モニターのアラート条件』

関連タスク:

- 303 ページの『レプリケーション・アラート・モニターの操作』
- 313 ページの『選択されたアラート条件の通知基準の指定』

関連資料:

- 299 ページの『レプリケーション中のアラートを送信する ASNMAIL exit ルーチン』

レプリケーション中のアラートを送信する ASNMAIL exit ルーチン

ASNMAIL exit ルーチンは、通知を処理します。この exit ルーチンは次の入力を受け取ります。

```
asnmail email_server to_address subject alert_message alert_message
```

表 21 では、ASNMAIL exit ルーチンの入力について説明しています。

表 21. ASNMAIL exit ルーチンの入力

入力	説明
<i>email_server</i>	これは、SMTP プロトコルを使用する E メール・サーバーのアドレスです。このサーバー・アドレスは、 asnmon コマンドの始動時に指定された email_server パラメーターから渡されます。
<i>to_address</i>	これは、通知を受ける連絡先の E メール・アドレスです。
<i>subject</i>	これは通知の件名です。
<i>alert_message</i>	これは、アラート・メッセージを含むストリングです。

アラートを E メールで送信する代わりに、ASNMAIL exit ルーチンに変更を加えて、問題管理システムなどにアラートを入力することもできます。

¥\$qlib¥\$samples¥rep¥ ディレクトリーには、ASNMAIL exit ルーチンのサンプルが含まれています。 asnmail.c サンプルには、サンプル・プログラムを使用する場合の入力パラメーターと指示が含まれています。

関連概念:

- 298 ページの『レプリケーション・アラート条件の E メール通知』
- 294 ページの『レプリケーション・アラート・モニターのアラート条件』

関連タスク:

- 303 ページの『レプリケーション・アラート・モニターの操作』

レプリケーション・アラート・モニターのセットアップ

レプリケーション・アラート・モニターのセットアップ

レプリケーション環境は、サーバー上で実行されるレプリケーション・プログラムと、それらのプログラムをサポートするコントロール表で構成されます。レプリケーション・アラート・モニターは、この環境をモニターします。

手順:

この手順は、レプリケーション・アラート・モニターのインストール時に 1 回使用してください。レプリケーション環境をモニターするようにレプリケーション・アラート・モニターをセットアップするには、次のようにします。

1. モニター負荷の評価。モニターする必要があるサーバーが多いほど、モニター負荷は大きくなります。
2. モニターの構成方法の選択。構成オプションが 2 つあります。1 つ以上のモニターを実行できます。以下のようにするには、複数のモニターを使用してください。
 - **レプリケーション・プログラムごとにモニター頻度を変える。** レプリケーション・プログラムでアラート条件をチェックする頻度を上げるには、monitor_interval を小さくしてモニターをセットアップします。例えば、あるモニターを、あるキャプチャー・サーバーで 15 分おきに CAPTURE_WARNINGS アラート条件をモニターするよう割り当てることができます。さらに、別のモニターを、別のキャプチャー・サーバーで 50 分おきに CAPTURE_WARNINGS アラート条件をモニターするよう割り当てることができます。
 - **さまざまなアプリケーションを別々にモニターする。** レプリケーション・アプリケーションごとにモニターをセットアップしてください。例えば、さまざまなグループに別々のモニターがアラートを送信したり、管理者が 2 種類のアプリケーション用にアラートを区別するのに役立てることができます。同様に、さまざまな条件をチェックするのに、別々のモニターを割り当てることができます。
 - **アラート条件の優先順位付けを行う。** 例えば、QAPPLY_STATUS アラート条件を使用して、10 分おきに Q アプライ・プログラムの状況をモニターする

こともできます。一方、QAPPLY_MEMORY アラート条件を使用して、300分おきに同じ Q アプライ・プログラムのメモリーをモニターすることもできます。

3. モニター・コントロール・サーバーとしてセットアップするサーバーの選択。セットアップしたいモニターごとに 1 つずつモニター・コントロール・サーバーを選択してください。
4. モニター・コントロール・サーバーごとのコントロール表の作成。詳細については、『レプリケーション・アラート・モニターのコントロール表の作成』を参照してください。
5. レプリケーション・アラート・モニターの連絡先情報の定義。詳細については、302 ページの『レプリケーション・アラート・モニターの連絡先情報の定義』を参照してください。

関連概念:

- 291 ページの『レプリケーション・アラート・モニターによるレプリケーションのモニター -- 概要』

レプリケーション・アラート・モニターのコントロール表の作成

レプリケーション・アラート・モニターを使用できるようにするには、その前にモニター管理テーブルを作成しなければなりません。このテーブルには、アラート条件、連絡先情報、ランタイム・パラメーター、およびモニター用の他のメタデータが保管されます。モニター管理テーブルを作成するサーバーは、モニター・コントロール・サーバーと呼ばれます。

DB2 UDB for Linux、DB2 UDB for UNIX、DB2 UDB for Windows、または DB2 UDB for z/OS をモニター・コントロール・サーバーにすることができます。たいていの場合、必要とされるモニター・コントロール・サーバーは 1 つだけですが、レプリケーション環境に応じて複数のサーバーを使用することができます。たとえば、モニターするレプリケーション・プログラムと同じシステム上でモニターを実行する場合は、モニターを実行するサーバー上のローカル・モニターごとにコントロール表のセットを作成してください。

手順:

レプリケーション・アラート・モニターのコントロール表を作成するには、レプリケーション・センターの「モニター・コントロール表の作成」ウィンドウを使用します。このウィンドウをオープンするには、「モニター・コントロール・サーバー」フォルダーを右マウス・ボタンでクリックして、「モニター・コントロール表の作成」を選択します。詳細については、オンライン・ヘルプを参照してください。

関連概念:

- 291 ページの『レプリケーション・アラート・モニターによるレプリケーションのモニター -- 概要』

関連資料:

- 424 ページの『モニター・コントロール・サーバーの表のリスト』

レプリケーション・アラート・モニターの連絡先情報の定義

初めてレプリケーション・アラート・モニターを使用する場合は、その前にアラート条件の通知先の個人またはグループに関する連絡先情報を定義する必要があります。モニターの実行後に連絡先情報を変更できます。

連絡先情報はモニター・コントロール・サーバーに保管されます。同じモニター・コントロール・サーバー上で実行するモニターは、連絡先を共有できます。複数のモニター・コントロール・サーバーがある場合には、それぞれのサーバーに連絡先を定義する必要があります。

各連絡先の E メール・アドレスと名前を指定して連絡先を定義した後で、連絡先をグループにできます。例えば、すべての DB2 管理者に関する連絡先情報を含む DB2 administrators という連絡先グループをセットアップすることもできます。サーバー間で連絡先やグループの情報をコピーすることもできます。

手順:

レプリケーション・アラート・モニターの連絡先情報を定義するには、次のようにします。

1. モニター・コントロール・サーバー上で、モニターの連絡先と連絡先グループを作成する。
 - a. レプリケーション・センターで、「連絡先の作成」ウィンドウを使用して連絡先情報を定義する。
 - b. オプション: レプリケーション・センターの「連絡先グループの作成」ウィンドウを使用して、連絡先グループを作成する。

このウィンドウをオープンするには、連絡先または連絡先グループに追加したいモニター・コントロール・サーバーを展開し、「連絡先」フォルダーを右マウス・ボタンでクリックして、「連絡先の作成」→「担当者」または「連絡先の作成」→「グループ」を選択します。詳細については、オンライン・ヘルプを参照してください。

2. オプション: レプリケーション・センターの「連絡先およびグループのコピー」ウィンドウを使用して、モニター・コントロール・サーバー間で連絡先情報をコピーする。

このウィンドウをオープンするには、連絡先または連絡先グループが置かれているモニター・コントロール・サーバーを展開する。「連絡先」フォルダーを選択します。目次ペインで、コピーしたい連絡先または連絡先グループを右クリックして、「コピー」を選択します。詳細については、オンライン・ヘルプを参照してください。

レプリケーション・センターでレプリケーション・アラート・モニター用に作成した連絡先は、タスク・センターやヘルス・センターなどの他の DB2 UDB センターでは使用できません。他の DB2 UDB センターで作成された連絡先は、レプリケーション・アラート・モニターでは使用できません。

関連概念:

- 298 ページの『レプリケーション・アラート条件の E メール通知』

関連資料:

- 299 ページの『レプリケーション中のアラートを送信する ASNMAIL exit ルーチン』

レプリケーション・アラート・モニターの操作

レプリケーション・アラート・モニターの操作

レプリケーション・アラート・モニターを実行して、レプリケーション環境をモニターできます。

前提条件:

レプリケーション・アラート・モニターを操作できるようにするには、その前にレプリケーション・アラート・モニターをセットアップしなければなりません。詳細については、300 ページの『レプリケーション・アラート・モニターのセットアップ』を参照してください。

手順:

モニターごとに以下のタスクを実行しなければなりません。以下のトピックでは、レプリケーション・アラート・モニターの操作方法について説明しています。

1. モニターの作成。詳細については、304 ページの『レプリケーションまたは発行用のモニターの作成』を参照してください。
2. モニターのアラート条件の選択。詳細については、305 ページの『レプリケーション・アラート・モニターのアラート条件の選択』を参照してください。
3. オプション: モニターのパラメーターの設定。詳細については、311 ページの『レプリケーション・アラート・モニターのパラメーターの設定 -- 概要』を参照してください。
 - 308 ページの『レプリケーション・アラート・モニターの操作に使用するパラメーターの説明』
 - 307 ページの『レプリケーション・アラート・モニターの操作に使用するパラメーターのデフォルト値』
4. レプリケーション・アラート・モニターの開始。詳細については、306 ページの『モニターの開始』を参照してください。

パラメーターの設定は、モニターの開始前、モニターの開始時、またはモニターの実行中に行えます。実行中のモニターのパラメーター設定をリフレッシュするには、モニターを再初期化しなければなりません。詳細については、307 ページの『モニターの再初期化』を参照してください。

実行中のモニターを停止することもできます。詳細については、314 ページの『モニターの停止』を参照してください。

関連概念:

- 291 ページの『レプリケーション・アラート・モニターによるレプリケーションのモニター -- 概要』

レプリケーションまたは発行用のモニターの作成

モニター管理テーブルを作成した後に、レプリケーション・センターの「モニターの作成ウィザード」を使用してモニターを作成し、レプリケーションまたは発行環境のモニターに使うアラート条件を選択します。

前提条件:

モニターを作成するには、その前にレプリケーション・アラート・モニターをセットアップしなければなりません。詳細については、300 ページの『レプリケーション・アラート・モニターのセットアップ』を参照してください。

手順:

モニターを作成するには、次のようにします。

1. レプリケーション・センターで、「モニターの作成ウィザード」をオープンし、モニターの名前と、モニターがアラート条件をチェックするレプリケーション・プログラムまたは発行プログラムを指定する。

ウィザードをオープンするには、モニターを作成するモニター・コントロール・サーバーを展開し、「モニター」フォルダーを右マウス・ボタンでクリックして、「作成」を選択します。

- a. 「開始」ページで、モニター修飾子を指定する。このモニターがアラート条件をチェックするプログラムを指定する。SQL レプリケーションで使用するサブスクリプション・セットをモニターすることもできます。

ウィザードでは以下の 1 つ以上のページが表示されるので、このモニターにアラート条件をチェックさせたいレプリケーション・プログラムに応じてアラート条件を選択できます。

- Q キャプチャー・プログラムのアラート条件の選択。
- Q アプライ・プログラムのアラート条件の選択。
- キャプチャー・プログラムのアラート条件の選択。
- アプライ・プログラムのアラート条件の選択。
- サブスクリプション・セットのアラート条件の選択。

たとえば、Q キャプチャー・プログラムと Q アプライ・プログラムをモニターするように指定する場合、「モニターの作成」ウィザードには、「Q アプライ・プログラムのアラート条件の選択 (Select alert conditions for the Q Capture program)」ページと、「Q キャプチャー・プログラムのアラート条件の選択 (Select alert conditions for the Q Apply program)」ページが表示されます。

2. 上記のページの 1 つから、2 次ダイアログを開く。そこでは以下を実行できます。
 - a. モニターしたいプログラムまたはサブスクリプション・セットを指定する。
 - b. チェックしたいアラート条件、および適切なアラート条件のパラメーターを指定する。例えば、毎分モニターにアラート条件をチェックさせるには、**monitor_interval** パラメーター値を 60 に設定できます。
3. 「サマリー」ページで、「完了」をクリックする。

詳細については、オンライン・ヘルプを参照してください。

関連概念:

- 291 ページの『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・モニターのアラート条件の選択

レプリケーション・アラート・モニターは、以下の時点でレプリケーションおよび発行プログラムのアクティビティをモニターします。

- 開始時に、個々のモニターが即時にアラート条件をチェックする。
- 指定した時間間隔で、個々のモニターが定期的アラート条件をチェックする。

モニターの作成時に、モニターにアラートの送信を求めるアラート条件を選択します。モニター対象の各 Q キャプチャー・プログラム、Q アプライ・プログラム、キャプチャー・プログラム、アプライ・プログラム、またはサブスクリプション・セットごとにアラート条件を選択することができます。

レプリケーション・センターを使用して、モニターの実行中にアラート条件を変更することもできます。これは既存のモニターを開き、アラート条件を変更し、次いでモニターを再初期化することによって実行します。

手順:

レプリケーション・アラート・モニターのアラート条件を選択するには、次のようにします。

1. モニター対象として選択したプログラムに応じて、レプリケーション・センターの「モニターの作成」ウィザードの以下のページのいずれか 1 つを使用する。
 - Q キャプチャー・プログラムのアラート条件の選択。
 - Q アプライ・プログラムのアラート条件の選択。
 - キャプチャー・プログラムのアラート条件の選択。
 - アプライ・プログラムのアラート条件の選択。
 - サブスクリプション・セットのアラート条件の選択。
2. ユーザーの環境と互換性のあるしきい値を選択する。

たとえば、キャプチャー・プログラムが 30 秒のコミット・インターバルで実行されている場合は、30 秒より長いキャプチャー待ち時間のしきい値を指定します。または、アプライ・プログラムが 10 分ごとにサブスクリプション・セットを処理するようにスケジュールする場合、APPLY_SUBSDELAY アラート条件のしきい値を 10 分より長い値に設定します。

レプリケーション・アラート・モニターのアラート条件を変更するには、次のようにします。

1. レプリケーション・センターで、Q キャプチャー・プログラム、Q アプライ・プログラム、キャプチャー・プログラム、アプライ・プログラム、またはサブスクリプション・セットの「アラート条件 (Alert Conditions)」ウィンドウを開きます。このウィンドウをオープンするには次のようにします。
 - a. 適正なモニター・コントロール・サーバー内の「**モニター**」フォルダーを展開する。
 - b. モニターを選択する。

- c. 目次ペインで、アラート条件を変更したい Q キャプチャー・スキーマ、Q アプライ・スキーマ、キャプチャー・スキーマ、アプライ・スキーマ、またはサブスクリプション・セットを右クリックする。
 - d. 「変更」を選択する。
2. アラート条件を変更する。
 3. モニターを再初期化する。詳細については、307 ページの『モニターの再初期化』を参照してください。

詳細については、オンライン・ヘルプを参照してください。

関連概念:

- 294 ページの『レプリケーション・アラート・モニターのアラート条件』
- 291 ページの『レプリケーション・アラート・モニター』

モニターの開始

レプリケーション・センターを使用して、モニターを開始します。モニターは、継続的に実行するか、1 モニター・サイクルだけ実行するかを決めることができます。パラメーターの値を設定して、モニターが実行中にエラーを検出した場合に連絡する人物の E メール・アドレスを入力することもできます。

前提条件:

- モニターの作成。詳細については、304 ページの『レプリケーションまたは発行用のモニターの作成』を参照してください。
- パスワード・ファイルの作成詳細については、396 ページの『asnpwd: パスワード・ファイルの作成および保守』を参照してください。
- モニター管理テーブルと、モニターするプログラムが実行しているサーバーに対する許可を持っていることを確認してください。

手順:

モニターを開始するには、次の方法のいずれかを使用します。

レプリケーション・センター

「モニターの開始」ウィンドウを使用します。このウィンドウをオープンするには、開始するモニターを示すモニター修飾子を右マウス・ボタンでクリックし、「**モニターの開始**」を選択します。詳細については、オンライン・ヘルプを参照してください。

asnmon システム・コマンド

asnmon コマンドを使用して、モニターを開始し、オプションで始動パラメーターを指定できます。詳細については、379 ページの『asnmon: レプリケーション・アラート・モニターの始動』を参照してください。

Windows Service Control Manager

モニターを実行するよう、Windows Service Control Manager をセットアップします。

z/OS コンソールまたは TSO

モニターを開始するよう、自動リスタート・マネージャー (ARM) のリカバリー・システムをセットアップできます。

関連概念:

- 291 ページの『レプリケーション・アラート・モニター』
- 340 ページの『自動リスタート・マネージャー (ARM) のリカバリー・システム』

関連タスク:

- 307 ページの『モニターの再初期化』
- 314 ページの『モニターの停止』
- 348 ページの『レプリケーション・プログラムのスケジューリング (Windows)』

モニターの再初期化

実行中のモニターを再初期化できます。モニターを再初期化すると、連絡先、アラート条件、およびパラメーター値に加えたすべての更新が認識されます。たとえば、モニターの実行時に連絡先の新規 E メール・アドレスを追加した場合は、モニターを再初期化します。

手順:

モニターを再初期化するには、次の方法のいずれかを使用します。

レプリケーション・センター

「モニターの再初期化 (Reinitialize Monitor)」ウィンドウを使用して、モニターを再初期化します。このウィンドウをオープンするには、再初期化するモニターを示すモニター修飾子を右マウス・ボタンでクリックし、「**モニターの再初期化 (Reinitialize Monitor)**」を選択します。詳細については、オンライン・ヘルプを参照してください。

asnmcmd システム・コマンド

実行中のモニターを再初期化するには、**asnmcmd reinit** コマンドを使用できます。詳細については、384 ページの『asnmcmd: 実行中のレプリケーション・アラート・モニターの操作』を参照してください。

関連タスク:

- 306 ページの『モニターの開始』
- 314 ページの『モニターの停止』

レプリケーション・アラート・モニターの操作に使用するパラメーターのデフォルト値

レプリケーション・アラート・モニターのパラメーターの値を設定して、レプリケーション・アラート・モニターの動作を変更できます。表 22 には、パラメーターごとのデフォルト値が示されています。

表 22. レプリケーション・アラート・モニター稼働パラメーターのデフォルト値

稼働パラメーター	デフォルト値
alert_prune_limit	10080 分
autopruner	Y
email_server	デフォルト値なし

表 22. レプリケーション・アラート・モニター稼働パラメーターのデフォルト値 (続き)

稼働パラメーター	デフォルト値
max_notification_minutes	60 分
max_notifications_per_alert	3
monitor_errors	デフォルト値なし
monitor_interval	300 秒
monitor_limit	10080 分
monitor_path	asnmon コマンドが呼び出されたディレクトリーです。
runonce	N
trace_limit	10080 分

関連概念:

- 308 ページの『レプリケーション・アラート・モニターの操作に使用するパラメーターの説明』
- 291 ページの『レプリケーション・アラート・モニター』

関連タスク:

- 303 ページの『レプリケーション・アラート・モニターの操作』

レプリケーション・アラート・モニターの操作に使用するパラメーターの説明

このトピックでは、レプリケーション・アラート・モニターの操作に使用できる以下のパラメーターについて説明します。

- 『alert_prune_limit』
- 309 ページの 『autoprune』
- 309 ページの 『email_server』
- 309 ページの 『max_notification_minutes』
- 309 ページの 『max_notifications_per_alert』
- 309 ページの 『monitor_errors』
- 310 ページの 『monitor_interval』
- 310 ページの 『monitor_limit』
- 310 ページの 『monitor_path』
- 310 ページの 『runonce』
- 311 ページの 『trace_limit』

alert_prune_limit

デフォルト: **alert_prune_limit=10080 分 (7 日)**

新規のモニター・サイクルの開始時に、レプリケーション・アラート・モニターは IBMSNAP_ALERTS 表から整理の対象となる行を整理します。デフォルトでは、レプリケーション・アラート・モニターは、10080 分 (7 日) よりも古い行を削除します。**alert_prune_limit** パラメーターは、レプリケーション・アラート・モニターが

表に保管する古いデータの量を制御します。このパラメーターは、データがどの程度古くなったらレプリケーション・アラート・モニターによって整理されるかを指定します。

システム上の `IBMSNAP_ALERTS` 表のストレージ・スペースが小さいときには、**alert_prune_limit** パラメーターの値を小さくできます。除去の制限を下げるとスペースは節約できますが、処理コストが増加します。また、**alert_prune_limit** パラメーターの値を大きくして、すべてのアラート・アクティビティの履歴を保持しておくことも考えられます。SQL レプリケーションに限り、整理の制限が大きいほど、大きな変更データ (CD) 表と UOW 表のスペースが必要になりますが、処理コストは小さくなります。

autoprun

デフォルト: `autoprun=y`

autoprun パラメーターは、オート・プルーンを制御します。レプリケーション・アラート・モニターは、モニター・コントロール表にコピー済みの行は、`IBMSNAP_ALERTS` 表から自動的に整理します。

email_server

email_server パラメーターは、`ASNMAIL exit` ルーチンを使用できるようにします。デフォルトの `ASNMAIL` ルーチンを使用すると、レプリケーション・アラート・モニターが E メールを使用してアラートを送信できます。Simple Mail Transfer Protocol (SMTP) を使用するよう設定されている E メール・サーバーのアドレスに、このパラメーターの値を設定してください。

max_notification_minutes

デフォルト: `max_notifications_minutes=60`

max_notifications_minutes パラメーターは、アラート条件が複数回発生したかどうかをモニターがトラッキングする期間を指定します。デフォルトでは、60 分の間にアラート条件が複数回発生した場合、レプリケーション・アラート・モニターは、60 分の間に最大 3 つのアラートを送信します。**max_notifications_per_alert** パラメーターは、**max_notifications_minutes** パラメーターによって指定された時間内に送信するアラート条件の通知の数を、モニターに指示します。

max_notifications_per_alert

デフォルト: `max_notifications_per_alert=3`

max_notifications_per_alert パラメーターは、1 つのアラートに関する通知の最大送信数を、レプリケーション・アラート・モニターに指示します。デフォルトでは、レプリケーション・アラート・モニターがアラート条件を複数回受信した場合、60 分の間にそのアラート条件に関して最大 3 つの通知を送信します。

monitor_errors

レプリケーション・アラート・モニターは、モニター処理で発生するすべてのエラーを保管します。操作エラーの一例として、レプリケーション・アラート・モニターがモニター・コントロール・サーバーに接続できない場合があります。操作エラーの通知を受信したい場合は、**monitor_errors** パラメーターに E メール・アドレ

スを指定しなければなりません。 E メール・アドレスを指定しないと、レプリケーション・アラート・モニターは操作エラーをログに記録しますが、エラーの通知を送信しません。

email_server パラメーターに有効な E メール・サーバーが記述されていない場合は、レプリケーション・アラート・モニターは **monitor_errors** パラメーターを無視します。

monitor_interval

デフォルト: **monitor_interval=300** 秒 (5 分)

monitor_interval パラメーターは、アラート条件をチェックする頻度をレプリケーション・アラート・モニターに指示します。デフォルトでは、レプリケーション・アラート・モニターは 300 秒ごとに、サーバー上の特定のモニターについて、すべてのアラート条件をチェックします。

monitor_limit

デフォルト: **monitor_limit=10080** 分 (7 日)

Q レプリケーションの場合、**monitor_limit** パラメーターは、Q キャプチャー・プログラムが行を整理する前に、IBMQREP_CAPMON 表と IBMQREP_CAPQMON 表に行を保持する期間を指定します。SQL レプリケーションの場合、**monitor_limit** パラメーターは、Q キャプチャー・プログラムが行を整理する前に、IBMSNAP_CAPMON 表に行を保持する期間を指定します。整理インターバルのたびに、これらの表中の行が現行タイム・スタンプに基づいてこの制限より古い場合、キャプチャー・プログラムと Q キャプチャー・プログラムはそれらの行を整理する。

monitor_path

デフォルト: **monitor_path=asnmon** コマンドが呼び出されたディレクトリー

monitor_path パラメーターは、レプリケーション・アラート・モニターが使用するログ・ファイルのロケーションを指定します。

runonce

デフォルト: **runonce=n**

レプリケーション・アラート・モニターは開始されると、デフォルトとして、ユーザーから選択されたアラート条件をモニターするインターバルで実行されます。レプリケーション・アラート・モニターを 1 時間おきに実行したり、その他の時間間隔で実行したり、一度だけ実行するようにスケジュールすることさえできます。

runonce=y が指定されると、レプリケーション・アラート・モニターはユーザーから選択されたすべてのアラート条件を一度チェックし、**monitor_interval** パラメーターを無視します。バッチ処理中でレプリケーション・アラート・モニターを実行するときに **runonce** を使用できます。たとえば、アプライ・プログラムが完了した後で、**runonce=y** を使用して、失敗したサブスクリプション・セットがあるかどうかを確認できます。失敗したサブスクリプション・セットがある場合には、レプリケーション・アラート・モニターは連絡先の個人またはグループに通知を送信します。

デフォルトでは **monitor_interval** は 300 秒 (5 分) です。レプリケーション・アラート・モニターは 300 秒ごとに、サーバー上の特定のモニターごとに、すべてのアラート条件をチェックします。レプリケーション・アラート・モニターはアラート条件を検出すると、通知を送信します。

trace_limit

デフォルト: **trace_limit**=10080 分 (7 日)

trace_limit パラメーターは、IBMSNAP_MONTRACE 表と IBMSNAP_MONTRAIL 表を整理する頻度を、レプリケーション・アラート・モニターに指示します。レプリケーション・アラート・モニターは、10080 分 (7 日) 間これらの表に行を保管します。レプリケーション・アラート・モニターは、**trace_limit** パラメーターに指定された値より古い行を整理します。

関連概念:

- 307 ページの『レプリケーション・アラート・モニターの操作に使用するパラメーターのデフォルト値』
- 291 ページの『レプリケーション・アラート・モニター』

関連タスク:

- 303 ページの『レプリケーション・アラート・モニターの操作』

レプリケーション・アラート・モニターのパラメーターの設定

レプリケーション・アラート・モニターのパラメーターの設定 -- 概要

さまざまなパラメーターの値を設定して、レプリケーション・アラート・モニターの動作を決定できます。モニターを作成するつどパラメーターを設定できます。

手順:

レプリケーション・アラート・モニターのパラメーターを設定するには、次のようにします。

1. レプリケーション・アラート・モニターの実行頻度の指定。詳細については、312 ページの『レプリケーション・アラート・モニターの実行頻度の指定』を参照してください。
2. レプリケーション・アラート・モニターからのデータの整理インターバルの指定。詳細については、312 ページの『レプリケーション・アラート・モニターからのデータの整理インターバルの指定』を参照してください。
3. 選択されたアラート条件の通知基準の指定。詳細については、313 ページの『選択されたアラート条件の通知基準の指定』を参照してください。
4. 操作エラーの通知基準の指定。詳細については、313 ページの『操作エラーの通知基準の指定』を参照してください。

298 ページの『レプリケーション・アラート条件の E メール通知』には、E メール通知、グループ化、および連絡先に関する詳細情報が説明されています。

関連タスク:

- 300 ページの『レプリケーション・アラート・モニターのセットアップ』

レプリケーション・アラート・モニターの実行頻度の指定

レプリケーション・アラート・モニターがレプリケーション環境のアラート条件をチェックする頻度を決定しなければなりません。

手順:

レプリケーション・アラート・モニターの実行頻度を指定するには、次の方法を使用します。

- **asnmon** コマンドの **runonce** パラメーターを使用して、レプリケーション・アラート・モニターを繰り返し実行するか、それとも 1 回のみ実行するかを指定する。
- **asnmon** コマンドの **monitor_interval** パラメーターを使用して、**runonce=n** の場合のレプリケーション・アラート・モニターの実行頻度を指定する。
- レプリケーション・センターを使用して、レプリケーション・アラート・モニターを開始したときに実行時間を指定することもできる。

関連概念:

- 307 ページの『レプリケーション・アラート・モニターの操作に使用するパラメーターのデフォルト値』
- 308 ページの『レプリケーション・アラート・モニターの操作に使用するパラメーターの説明』
- 291 ページの『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・モニターからのデータの整理インターバルの指定

レプリケーション・アラート・モニターは、モニター表を自動的に整理できます。モニターがモニター表を自動的に整理するかどうか、および整理する場合はモニターが表を整理する方法を決定しなければなりません。

手順:

モニター表を整理する頻度を指定するには、次の方法を使用します。

- **autoprun** パラメーターを使用して、レプリケーション・アラート・モニターにコントロール表を自動的に整理させたいかどうかを指定する。
- **alert_prune_limit** パラメーターの値を変更して、レプリケーション・アラート・モニターが表に保管する履歴データの量を制御する。データがどの程度古くなったらレプリケーション・アラート・モニターによって **IBMSNAP_ALERTS** 表から整理されるかを指定する。
- **trace_limit** パラメーターの値を変更して、レプリケーション・アラート・モニターが行をモニター表に保管する期間を制御する。

関連概念:

- 307 ページの『レプリケーション・アラート・モニターの操作に使用するパラメーターのデフォルト値』

- 308 ページの『レプリケーション・アラート・モニターの操作に使用するパラメーターの説明』
- 291 ページの『レプリケーション・アラート・モニター』

選択されたアラート条件の通知基準の指定

選択したアラート条件は、レプリケーション・アラート・モニターによって保管されます。電子メール (E メール) によって自動的にアラート条件を連絡先に通知するように、通知パラメーターをセットアップできます。

手順:

アラート条件の通知基準を指定するには、次の方法を使用します。

1. **max_notifications_per_alert** パラメーターを設定して、特定の時間内の最大通知数を制御する。 **max_notifications_minutes** パラメーターで指定された時間内の特定のアラート条件に関する、受け取りたい通知の最大数を指定する。
2. **email_server** パラメーターを設定して、アラート条件が生じた時点で DB2 が E メール通知できるようにする。このパラメーターの値は、SMTP プロトコルを使用する E メール・サーバーのアドレスに設定してください。
3. オプション: ASNMAIL exit ルーチンに独自の拡張機能を作成して、アラート条件の処理方法をカスタマイズできる。このオプションは、問題管理やその他のシステムと統合するのに便利です。

関連概念:

- 298 ページの『レプリケーション・アラート条件の E メール通知』
- 307 ページの『レプリケーション・アラート・モニターの操作に使用するパラメーターのデフォルト値』
- 308 ページの『レプリケーション・アラート・モニターの操作に使用するパラメーターの説明』
- 294 ページの『レプリケーション・アラート・モニターのアラート条件』
- 291 ページの『レプリケーション・アラート・モニター』

関連タスク:

- 302 ページの『レプリケーション・アラート・モニターの連絡先情報の定義』

操作エラーの通知基準の指定

操作中にエラーが発生すると、レプリケーション・アラート・モニターは通知を送信します。

手順:

操作エラーの通知基準を指定するには、次の方法を使用します。

monitor_errors パラメーターの値を E メール・アドレスに設定します。モニターは、発生した操作エラーの通知をこのアドレスに送信します。 Simple Mail Transfer Protocol (SMTP) プロトコルを使用して、E メール・アドレスを入力してください。

関連概念:

- 307 ページの『レプリケーション・アラート・モニターの操作に使用するパラメーターのデフォルト値』
- 308 ページの『レプリケーション・アラート・モニターの操作に使用するパラメーターの説明』

モニターの停止

モニターを停止すると、アラート条件の対象のレプリケーション・プログラムまたは発行プログラムのチェックは停止します。モニターを停止するには、レプリケーション・センター、システム・コマンド、または DB2 レプリケーション・サービスを使用することができます。

手順:

モニターを停止するには、次の方法のいずれかを使用します。

レプリケーション・センター

「モニターの停止 (Stop Monitor)」ウィンドウを使用して、モニターを停止します。このウィンドウをオープンするには、停止するモニターを示すモニター修飾子を右マウス・ボタンでクリックし、「**モニターの停止 (Stop Monitor)**」を選択します。詳細については、オンライン・ヘルプを参照してください。

asnmcmd システム・コマンド

モニターを停止するには、**asnmcmd stop** コマンドを使用できます。詳細については、384 ページの『asnmcmd: 実行中のレプリケーション・アラート・モニターの操作』を参照してください。

Windows Service Control Manager

DB2 レプリケーション・サービスを停止する場合、レプリケーション・サービスの停止時にモニターは自動的に停止します。

キャプチャー、アプライ、Q キャプチャー、または Q アプライ・プログラムの実行中にモニターが停止した場合、次の開始時にモニターは以下のアクションを実行します。

- モニターが停止中に合致していたアラート条件をチェックする。
- 合致していたすべての条件に対してアラートを発行する。

関連タスク:

- 306 ページの『モニターの開始』
- 307 ページの『モニターの再初期化』

第 21 章 Q レプリケーションおよび発行環境の保守

Q レプリケーションおよびイベント発行環境の保守 -- 概要

Q レプリケーションおよびイベント発行はデータベース・システムと共同で処理を行うため、既存のデータベース・アクティビティの変更は最小限で済みます。しかし、システム全体の円滑な実行を保証し、潜在的な問題を回避するためには、レプリケーション環境の処理要件を判別し、これらの要件がデータベース・システムに影響を与える可能性を判別する必要があります。

以下のセクションでは、Q レプリケーションおよびイベント発行環境にある、ソース・システム、コントロール表、およびターゲット表の保守の方法を説明します。

- 『Q レプリケーションおよびイベント発行のソース・システムの保守に関する考慮事項 -- 概要』
- 322 ページの『Q レプリケーションおよびイベント発行でのコントロール表の保守 -- 概要』
- 325 ページの『Q レプリケーションおよびイベント発行でのターゲット表の保守』
- 326 ページの『Q レプリケーションおよびイベント発行のパッケージおよびプランの再バインドに関する考慮事項』

Q レプリケーションおよびイベント発行のソース・システムの保守に関する考慮事項

Q レプリケーションおよびイベント発行のソース・システムの保守に関する考慮事項 -- 概要

レプリケーション・ソース・システムには次のオブジェクトが含まれます。

- データの複製、または発行元にするソース表。
- Q キャプチャー・プログラムが、ソース表に加えられる変更を取り込むために読み取るログ・データ。

Q キャプチャー・プログラムおよび Q アプライ・プログラムが常に処理を進められるように、Q レプリケーションおよびイベント発行に対するソース表およびログ・データの可用性を考慮する必要があります。以下のトピックには、Q レプリケーションおよびイベント発行環境に対して、ソース・システムを保守する際に考慮すべき情報が含まれています。

- 316 ページの『Q レプリケーションおよびイベント発行環境でのソース表の保守』
- 316 ページの『Q レプリケーションおよびイベント発行のログ・ファイルの保存 -- 概要』
- 321 ページの『Q レプリケーションおよびイベント発行環境でのコンプレッション・ディクショナリーの管理に関する考慮事項 (z/OS)』

関連概念:

- 315 ページの『Q レプリケーションおよびイベント発行環境の保守 -- 概要』

Q レプリケーションおよびイベント発行環境でのソース表の保守

レプリケーション・ソースはデータベース表です。Q レプリケーションおよびイベント発行のソース表には、システム上の他のデータベース表と同じ保守が必要です。

Q レプリケーションおよびイベント発行は、ほとんどの処理においては、ソース表に直接アクセスする必要はありません。しかし、Q レプリケーションおよびイベント発行は、以下の状況ではソース表に直接アクセスしなければなりません。

- Q アプライ・プログラムがロードを実行するとき。
- ログ・マネージャーが圧縮されたログ・レコードを読み取ろうとしたとき (z/OS)。
- Q キャプチャー・プログラムが LOB データを取り込むとき。

手順:

Q レプリケーションおよびイベント発行のソース表を保守するには、以下のようになります。

- これらの表に関して、既存のユーティリティおよび保守ルーチンを引き続き実行する。
- ロード時に Q アプライ・プログラムが中断されないように、ソース表に対して読み取りアクセスが可能であることを確認する。
- ソース表が圧縮される場合は、共用モード接続を使用して、z/OS™ ユーティリティがアプリケーション・サーバーに接続することを確認する。ユーティリティおよび保守ルーチンが、排他モード接続を使用してアプリケーション・サーバーに接続する場合、ソース表は Q レプリケーションおよびイベント発行では使用不可能です。排他モードでは、DB2® UDB がデータベースおよび z/OS 圧縮表スペースをオフラインにすることが必要です。

関連概念:

- 315 ページの『Q レプリケーションおよびイベント発行環境の保守 -- 概要』

関連タスク:

- 325 ページの『Q レプリケーションおよびイベント発行でのターゲット表の保守』

Q レプリケーションおよびイベント発行のログ・ファイルの保存

Q レプリケーションおよびイベント発行のログ・ファイルの保存 -- 概要

Q キャプチャー・プログラムが必要とするログ・ファイルを知る必要がある場合は、Q キャプチャー・プログラムからのデータと、DB2 UDB からのデータの組み合わせを使用して、Q キャプチャー・プログラムが必要とする最も古いログ・ファイルを判別することが必要です。以下のトピックでは、ログ・データを保存すべき

理由、およびプラットフォームに基づき、Q キャプチャー・プログラムが必要とする最も古いログ・ファイルの判別方法を説明します。

- 『Q レプリケーションおよびイベント発行のログ・データを保存する理由』
- 318 ページの『Q キャプチャー・プログラムが必要とする最も古いログ・ファイルの判別 (Linux、UNIX、Windows)』
- 319 ページの『Q キャプチャー・プログラムが必要とする最も古いログ・ファイルの判別 (z/OS)』

関連概念:

- 316 ページの『Q レプリケーションおよびイベント発行環境でのソース表の保守』
- 15 ページの『Q キャプチャー・プログラム』

Q レプリケーションおよびイベント発行のログ・データを保存する理由

DB2[®] リカバリー・ログは、

- DB2[®] リカバリー機能を提供します。
- 実行中の Q キャプチャー・プログラムに情報を提供します。

DB2[®] リカバリーおよび Q レプリケーションまたはイベント発行の両方について、ログ・データを保存する必要があります。また、このデータを削除する前に、Q キャプチャー・プログラムおよび DB2[®] が、ログのセットの処理を完全に終了していることを確実に確認する必要があります。

ログ・データはログ・バッファー、アクティブ・ログ、またはアーカイブ・ログに入っています。Q キャプチャー・プログラムはウォーム・スタートのたびに、プログラムの停止後に作成されたすべての DB2[®] ログと、処理が完全に終了していないすべての DB2[®] ログを要求します。

Q キャプチャー・プログラムを継続して実行すると、通常は DB2[®] リカバリー・ログが最新になります。また、ログ・ファイルを 1 週間以上保存するのであれば、既存のログ保存手順を使用し続けることができます。しかし、次の場合には、Q レプリケーションおよびイベント発行に合わせて、ログ保存手順を変更する必要があります。

- DB2[®] がバックアップを完了したら即時にログ・レコードを削除する場合、そしてこれらのログ・レコードが順方向リカバリーに必要とされない場合。
- ストレージの制約があるため、アーカイブしたりカバリー・ログを頻繁に削除する必要がある場合。
- Q キャプチャー・プログラムを連続的ではなく周期的に実行する場合。

関連概念:

- 316 ページの『Q レプリケーションおよびイベント発行のログ・ファイルの保存 -- 概要』

関連タスク:

- 318 ページの『Q キャプチャー・プログラムが必要とする最も古いログ・ファイルの判別 (Linux、UNIX、Windows)』

- 319 ページの『Q キャプチャー・プログラムが必要とする最も古いログ・ファイルの判別 (z/OS)』

Q キャプチャー・プログラムが必要とする最も古いログ・ファイルの判別 (Linux、UNIX、Windows)

Q キャプチャー・プログラムは、DB2 UDB for Linux、UNIX、Windows からログ・レコードを要求し、受け取るログ・レコードについての情報を保守します。しかし、DB2 UDB はこれらのログ・レコードを含むログ・ファイルを管理し、Q キャプチャー・プログラムにはログ・ファイル情報を提供しません。したがって、Q キャプチャー・コントロール表を読み取って、Q キャプチャー・プログラムが必要とする最も古いログ・レコードのログ・シーケンス番号を判別することが必要です。DB2 を使用して、最も古いログ・レコードを含むログ・ファイルで、ログ・シーケンス番号を参照してください。Q キャプチャー・プログラムでは、このログ・ファイルおよびこれより新しいログ・ファイルが必要です。

手順:

Q キャプチャー・プログラムが必要とする最も古いログ・ファイルを判別するには、次のようにします。

1. 以下の SQL ステートメントを実行して、Q キャプチャー・プログラムがそのコントロール表で履歴を確認、処理、および記録した最新のトランザクションのログ・レコード・シーケンス番号を取得する。

```
SELECT max(RESTART_SEQ)
FROM capschema.IBMQREP_CAPMON
WITH UR;
```

RESTART_SEQ 値は、CHAR(10) FOR BIT DATA 列であり、16 進の 20 文字のように見えます。以下に例を示します。

```
00000000123456123456
```

RESTART_SEQ 値の最後の 12 文字に注目してください。この番号がログ・シーケンス番号です。Q キャプチャー・プログラムには、これより古いログ・レコードは必要ありません。上記の例では、ログ・シーケンス番号は次のとおりです。

```
123456123456
```

2. DB2® UDB コマンド行から、データベース構成のアクティブ・ログ・ファイルのパスを取得する。以下に例を示します。

```
db2 get db cfg for yourdbname
```

この *yourdbname* はデータベース名です。画面に表示された出力からアクティブ・ログ・ファイルのパスを確認します。以下に例を示します。

```
Path to log files =C:\DB2\NODE0000\SQL00001\SQLLOGDIR\
```

3. 現行作業ディレクトリーを、アクティブ・ログ・ファイルへの上記のパスに変更する。以下に例を示します。

```
C:¥>cd c:¥DB2¥NODE0000¥SQL00001
```

4. DB2 UDB コマンド行から、**db2flsn** コマンド、および最初のステップのログ・シーケンス番号を使用して、**Q** キャプチャー・プログラムが必要とする最も古いログ・ファイルの名前を入力する。以下に例を示します。

```
C:¥DB2¥NODE0000¥SQL00001¥>db2flsn 123456123456
```

db2flsn コマンドを実行するには、SQLLOGCTL.LFH ファイルに読み取りアクセスできる必要があります。このファイルは、アクティブ・ログ・ファイルのパスの 1 つ上のディレクトリー (C:¥DB2¥NODE0000¥SQL00001¥) にあります。

システムは、ログ・シーケンス番号により識別されるログ・レコードを含むファイルの名前を検索して表示します。以下に例を示します。

```
Given LSN is contained in the log file S000123.LOG
```

5. 検索されたログ・ファイルの経過日数を確認する。

Q キャプチャー・プログラムは、任意の時点から再始動を実行するために、このログ・ファイル、およびこれより新しいログ・ファイルを必要とします。**Q** キャプチャー・プログラムの連続稼働を保証するために、このログ・ファイル、およびこれより新しいログ・ファイルは保存する必要があります。これより古いログは削除できます。

推奨: **Q** キャプチャー・プログラムは DB2 がアクティブになっている間は常に実行してください。これにより、**Q** キャプチャー・プログラムは、最新のログ・レコードがある DB2 ログの末尾に可能な限り近いところを読み取り続けることができます。ログの最後に向かって読み取りを行うと、**Q** キャプチャー・プログラムが必要とする最も古いログ・ファイルの数を最小限にできます。

関連概念:

- 316 ページの『**Q** レプリケーションおよびイベント発行のログ・ファイルの保存 -- 概要』

関連タスク:

- 319 ページの『**Q** キャプチャー・プログラムが必要とする最も古いログ・ファイルの判別 (z/OS)』

Q キャプチャー・プログラムが必要とする最も古いログ・ファイルの判別 (z/OS)

Q キャプチャー・プログラムは、DB2 UDB for z/OS からログ・レコードを要求し、受け取るログ・レコードについての情報を保守します。しかし、DB2 UDB はこれらのログ・レコードを含むログ・ファイルを管理し、**Q** キャプチャー・プログラムにはログ・ファイル情報を提供しません。したがって、**Q** キャプチャー・コントロール表を読み取って、**Q** キャプチャー・プログラムが必要とする最も古いログ・レコードのログ・シーケンス番号を判別することが必要です。DB2 UDB を使用して、最も古いログ・レコードを含むログ・ファイルで、ログ・シーケンス番号

を参照してください。 Q キャプチャー・プログラムでは、このログ・ファイルおよびこれより新しいログ・ファイルが必要です。

手順:

Q キャプチャー・プログラムが必要とする最も古いログ・ファイルを判別するには、次のようにします。

1. 以下の SQL ステートメントを実行して、Q キャプチャー・プログラムがそのコントロール表で履歴を確認し、処理し、記録した最新のトランザクションのログ・シーケンス番号を取得する。

```
SELECT max(RESTART_SEQ)
FROM capschema.IBMQREP_CAPMON
WITH UR;
```

以下はログ・シーケンス番号の例です。

000055551F031230000

常に 0000 である、ログ・シーケンス番号の最初の 4 文字は無視してください。以下の 12 文字は、アクティブ・ログ・シーケンス番号に相当します。(この 12 文字の値は、非データ共有環境では相対バイト・アドレス (RBA) であり、データ共有環境ではログ・レコード・シーケンス番号 (LRSN) です。)最後の 4 文字は、非データ共有環境では 0000 になります。これら最後の 4 文字は、データ共有環境ではメンバー ID に相当します。

2. DSNJU004 ユーティリティを使用してログ・マップ印刷ユーティリティを呼び出す。このユーティリティは、ブートストラップ・データ・セット (BSDS) に関する情報を表示します。

以下に例を示します。

```
# ACTIVE LOG COPY 1 DATA SETS
# START RBA/TIME      END RBA/TIME          DATE  LTIME  DATA SET INFORMATION
#-----
# 555551F03000          555551F05FFF          1998.321 12:48  DSN=DSNC710.LOGCOPY1.DS02
#2001.57 15:46:32.2    2001.057 15:47:03.9    PASSWORD=(NULL)STATUS=TRUNCATED,REUSABLE
# 555551F06000          555551F09FFF          1998.321 12:49  DSN=DSNC710.LOGCOPY1.DS03
#2001.57 15:47:32.2    2001.057 15:48:12.9    PASSWORD=(NULL)STATUS=TRUNCATED,REUSABLE
```

3. RESTART_SEQ 値の 12 文字のアクティブ・ログ番号を、表示された各行の開始 RBA および対応する終了 RBA の範囲と比較する。
4. 開始 RBA と終了 RBA 内にある、IBMQREP_CAPMON 表からの 12 文字のアクティブ・ログ番号の行を見つける。この例では次のようになっています。

```
# 555551F03000          555551F05FFF          1998.321 12:48  DSN=DSNC710.LOGCOPY1.DS02
#2001.57 15:46:32.2    2001.057 15:47:03.9    PASSWORD=(NULL)STATUS=TRUNCATED,REUSABLE
```

5. このアクティブ・ログ番号に対応するデータ・セット情報を確認する。この例では次のようになっています。

DSNC710.LOGCOPY1.DS02

6. このデータ・セットの日時を確認する。

Q キャプチャー・プログラムは、任意の時点から再始動を実行するために、このデータ・セット、およびこれより新しいデータ・セットを必要とします。

関連概念:

- 316 ページの『Q レプリケーションおよびイベント発行のログ・ファイルの保存 -- 概要』

関連タスク:

- 318 ページの『Q キャプチャー・プログラムが必要とする最も古いログ・ファイルの判別 (Linux、UNIX、Windows)』

Q レプリケーションおよびイベント発行環境でのコンプレッション・ディクショナリーの管理に関する考慮事項 (z/OS)

DB2® のコンプレッション・ディクショナリーを使用する場合は、Q キャプチャー・プログラムとの間でユーティリティーの使用を調整する必要があります。

DB2® コンプレッション・ディクショナリーの更新

Q キャプチャー・プログラムがログ・レコードを要求した場合、DB2® は圧縮された表スペースに保管されている表のログ・レコードを圧縮解除する必要があります。DB2® は現行のコンプレッション・ディクショナリーを使用して圧縮を解除します。コンプレッション・ディクショナリーが一時的に使用できない場合、DB2® は Q キャプチャー・プログラムにエラーを戻します。Q キャプチャー・プログラムは何度か処理の続行を試みます。しかし、ディクショナリーが引き続き使用できない状態である場合は、Q キャプチャー・プログラムは ASN0011E メッセージを発行し、Q サブスクリプションを非アクティブにし、終了します。これらの状況を防止するために、表のコンプレッション・ディクショナリーに影響を与えるアクティビティーを実行する前に、Q キャプチャー・プログラムが表のすべてのログ・レコードを処理するようにしてください。これらのアクティビティーには以下のものが含まれます。

- 圧縮設定を変更する表スペースの変更
- データ共有環境から非データ共有環境へのコピーを含めて、DSN1COPY を使用したサブシステム間での圧縮した表スペースのコピー
- 表スペースに対する REORG ユーティリティーの実行

新しいコンプレッション・ディクショナリーを生成する場合は、現在実行中のアプリケーション、および Q キャプチャー・プログラムに対して、REORG ユーティリティーを以下のように同期させてください。

1. 表を更新するすべてのアプリケーション・プログラムを静止させる。
2. Q キャプチャー・プログラムで、ログに保管されている表の更新をすべてキャプチャーする。
3. 圧縮表に対して REORG ユーティリティーを使用し、新しいコンプレッション・ディクショナリーを作成する。
4. アプリケーション・プログラムを再始動する。

推奨: 再編成時にコンプレッション・ディクショナリーの現行バージョンを維持するには、KEEPDICTIONARY=YES オプションを使用します。

KEEPDICTIONARY=YES オプションを使用すると、既存のログ・レコードに対してディクショナリーの互換性を維持できます。

DB2® コンプレッション・ディクショナリーのラッチ

コンプレッション・ディクショナリーの可用性についても考慮する必要があります。Q キャプチャー・プログラムが圧縮されたログ・レコードを読み取る際には、DB2® はディクショナリーにアクセスするために、ソースの圧縮された表スペースのラッチを取ります。DB2® のログ読み取りインターフェースがこのラッチを必要としたときに、ソース・システム上の圧縮された表スペースが STOPPED 状態にある場合は、Q キャプチャー・プログラムは停止します。その逆に、ソースの表スペースへの完全アクセスを必要とする、または表スペースが STOPPED 状態であることを必要とするユーティリティーは、Q キャプチャー・プログラムがディクショナリーの読み取り中にラッチを保持しているために締め出されることがあります。

ラッチが使用できないために一時的なロックアウトが発生しないように、ソースの圧縮された表スペースを DB2® (またはベンダー) のユーティリティーで排他的に使用する必要があるときには、Q キャプチャー・プログラムを中断してください。

関連概念:

- 315 ページの『Q レプリケーションおよびイベント発行のソース・システムの保守に関する考慮事項 -- 概要』
- 15 ページの『Q キャプチャー・プログラム』

Q レプリケーションおよびイベント発行でのコントロール表の保守

Q レプリケーションおよびイベント発行でのコントロール表の保守 -- 概要

Q レプリケーションは、ソース定義、Q サブスクリプション定義、およびその他のレプリケーション固有のコントロール情報を保管するためにコントロール表を使用します。コントロール表のサイズは静的なものもあれば、データベースおよびレプリケーションのサイズの要件に従って拡大または縮小するものもあります。

以下のトピックでは、コントロール表で実行する保守アクティビティーについて説明します。

- 323 ページの『Q レプリケーションおよびイベント発行でのコントロール表の整理』
- 324 ページの『Q レプリケーションおよびイベント発行におけるコントロール表での RUNSTATS ユーティリティーの使用に関する考慮事項』
- 324 ページの『Q レプリケーションおよびイベント発行コントロール表の再編成』

レプリケーション・プログラムが DB2 サーバーに接続できないという状況についても考慮する必要があります。詳細については、325 ページの『レプリケーション・プログラムが DB2 UDB サーバーに接続できない場合』を参照してください。

関連概念:

- 315 ページの『Q レプリケーションおよびイベント発行環境の保守 -- 概要』
- 415 ページの『Q レプリケーションおよびイベント発行のコントロール表 -- 概要』

Q レプリケーションおよびイベント発行でのコントロール表の整理

Q レプリケーションとイベント発行のほとんどのコントロール表のサイズは、頻繁には変わりません。しかし、定期的にサイズが大きくなるコントロール表もあります。表 23 は、定期的にサイズが大きくなる可能性があり、Q キャプチャー・プログラムが整理するコントロール表をリストしています。表 24 は、定期的にサイズが大きくなる可能性があり、Q アプライ・プログラムが整理するコントロール表をリストしています。レプリケーション・アラート・モニターは、定期的にサイズが大きくなる可能性のある `IBMSNAP_ALERTS` 表を整理します。`alert_prune_limit` パラメーターは、表に保持されるデータの量を指定します。サイズが大きくなる割合は、レプリケーション構成およびパラメーターに依存しています。以下のコントロール表のサイズの拡張をモニターし、使用可能な各種の整理方式を考慮してください。

手順:

手動でコントロール表を整理するには、整理オプションで `asnlcmd` コマンドを使用します。

テーブルが自動的に整理される方法を制御するには、表 23 または表 24 で示されているパラメーターの値を変更します。

Q キャプチャー・プログラムは、リストされるパラメーターを使用し、整理の対象になる行をこれらの表から判別することによって、以下の表を整理します。

`prune_interval` パラメーターは、Q キャプチャー・プログラムが整理できる行を検査する頻度を指定します。

表 23. Q キャプチャー・プログラムが整理するコントロール表

コントロール表	整理対象になる行を指定するパラメーター
<code>IBMQREP_CAPMON</code>	<code>monitor_limit</code>
<code>IBMQREP_CAPQMON</code>	<code>monitor_limit</code>
<code>IBMQREP_CAPTRACE</code>	<code>trace_limit</code>
<code>IBMQREP_SIGNAL</code>	<code>signal_limit</code>

Q アプライ・プログラムは、リストされるパラメーターを使用し、整理対象になる行をこれらの表から判別することによって、以下の表を整理します。

`prune_interval` パラメーターは、Q アプライ・プログラムが整理できる行を検査する頻度を指定します。

表 24. Q アプライ・プログラムが整理するコントロール表

コントロール表	整理対象になる行を指定するパラメーター
<code>IBMQREP_APPLYMON</code>	<code>monitor_limit</code>

表 24. Q アプライ・プログラムが整理するコントロール表 (続き)

コントロール表	整理対象になる行を指定するパラメーター
IBMQREP_APPLYTRACE	trace_limit

IBMSNAP_ALERTS 表の整理を制御するには、**alert_prune_limit** パラメーターの値を変更します。

関連概念:

- 322 ページの『Q レプリケーションおよびイベント発行でのコントロール表の保守 -- 概要』

Q レプリケーションおよびイベント発行におけるコントロール表での RUNSTATS ユーティリティの使用に関する考慮事項

RUNSTATS ユーティリティは、表および関連する索引の物理的特性に関する統計を更新します。以前に Q レプリケーションで使用していたのと同じ頻度で、既存の表に対して引き続き RUNSTATS ユーティリティを実行します。しかし、表に含まれるデータの量が大量である場合には、定期的に拡大 (および定期的に整理) されるコントロール表に対しては RUNSTATS ユーティリティを一度だけ実行してください。RUNSTATS が動的な表に関して有用な情報を報告するのは、これらの表が実動レベルで最大のサイズにある場合です。オブティマイザーは必要な統計を獲得して、データへのアクセスの最良のストラテジーを判断します。

関連概念:

- 322 ページの『Q レプリケーションおよびイベント発行でのコントロール表の保守 -- 概要』
- 415 ページの『Q レプリケーションおよびイベント発行のコントロール表 -- 概要』

Q レプリケーションおよびイベント発行コントロール表の再編成

頻繁にサイズを変更するコントロール表があれば、定期的に再編成してフラグメント化されたデータを削除し、スペースを再利用してください。

手順:

コントロール表を再編成するには、次のようにします。

REORG コマンド (UNIX、Windows)

PREFORMAT オプションを指定した REORG ユーティリティ (z/OS)

このユーティリティの PREFORMAT オプションは、Q キャプチャー・プログラムの挿入処理を迅速化します。

推奨: 以下のコントロール表は週に一度は再編成してください。

- IBMQREP_APPLYTRACE
- IBMQREP_CAPMON
- IBMQREP_CAPQMON
- IBMQREP_CAPTRACE

- IBMSNAP_MONTRAIL
- IBMSNAP_MONTRACE
- IBMQREP_SIGNAL

IBMSNAP_ALERTS および IBMQREP_EXCEPTIONS 表も、レプリケーション環境によっては、頻繁に大きくなったりサイズが変わることがあります。

関連概念:

- 322 ページの『Q レプリケーションおよびイベント発行でのコントロール表の保守 -- 概要』

関連タスク:

- 323 ページの『Q レプリケーションおよびイベント発行でのコントロール表の整理』

レプリケーション・プログラムが DB2 UDB サーバーに接続できない場合

正しく実行するには、Q キャプチャー・プログラム、Q アプライ・プログラム、およびレプリケーション・アラート・モニターが、コントロール表を含む DB2 UDB サーバーに接続できなければなりません。これらの各プログラムは、これらのコントロール表に対して読み取りおよび書き込み可能であることも必要です。レプリケーション・プログラムがそのコントロール表にアクセスできないと、適切なエラー・メッセージが出され、プログラムはシャットダウンします。

接続性の問題では、通常は接続性が元に戻ったときにプログラムを再始動する必要があります。たとえば、コントロール表を含む DB2 UDB サーバーがシャットダウンまたは静止しているために Q キャプチャー・プログラムがシャットダウンする場合、DB2 UDB サーバーの実行時に Q キャプチャー・プログラムを再始動するだけで済みます。

プログラムが DB2 UDB サーバーに接続できるのに、プログラムがコントロール表にアクセスしようとする SQL エラーを受け取る場合、その SQL エラーに適切な訂正アクションを実行してから、プログラムを再始動してください。たとえば SQL エラーが、コントロール表のリカバリーが必要であることを示す場合、標準 DB2 リカバリー手順を使用して、表のリカバリーを行ってからプログラムを再始動します。

関連概念:

- 75 ページの『Q レプリケーションおよびイベント発行の接続要件』
- 118 ページの『Q レプリケーションのエラー・オプション』

Q レプリケーションおよびイベント発行でのターゲット表の保守

ターゲット表の保守は、他の表の保守と類似しています。しかし、Q アプライ・プログラムはターゲット表の保守方法に影響を与えません。Q アプライ・プログラムによる更新および削除ステートメントのパフォーマンスは、オプティマイザーが使用する統計の現行性に大きく影響されます。

手順:

ターゲット表を保守するには、次のようにします。

- ターゲット・サーバー上の表は、データベース・システムの他の表を保守するのと同じ方法で保守してください。
- ターゲット表が既存のデータベース表である場合も、Q レプリケーションにより自動的に生成されるように指定された表である場合も、これらの表に対して現在のバックアップおよび保守のルーチンを使用してください。
- ユーティリティを実行するためにターゲット表をオフラインにする前に、Q アプライ・プログラムを停止してください。

関連概念:

- 315 ページの『Q レプリケーションおよびイベント発行環境の保守 -- 概要』
- 316 ページの『Q レプリケーションおよびイベント発行環境でのソース表の保守』

Q レプリケーションおよびイベント発行のパッケージおよびプランの再バインドに関する考慮事項

Q レプリケーションおよびイベント発行のパッケージおよびプランの多くは、非コミット読み取り (UR) 分離性を使用してバインドされます。パッケージおよびプランの自動再バインドに使用される内部の保守プログラムは、Q キャプチャー・プログラムと Q アプライ・プログラムとの間で競合の問題の原因になることがあります。内部の保守プログラムは、カーソル固定 (CS) 分離性標準オプションでレプリケーション・パッケージを再バインドする場合、Q キャプチャー・プログラムおよび Q アプライ・プログラムに干渉します。Q レプリケーションおよびイベント発行のパッケージは、最適なシステム・パフォーマンスを維持するために、分離 UR にバインドしておく必要があります。

Linux、UNIX[®]、Windows[®] の場合

Q キャプチャー・プログラム、Q アプライ・プログラム、およびレプリケーション・アラート・モニターのパッケージは、プログラムがそのコントロール表に最初に接続するときに自動的にバインドされます。

z/OS の場合

Q キャプチャー・プログラム、Q アプライ・プログラム、および共通パッケージが自動的にバインドされます。z/OS サンプル ASNQBNDL を使用して、DB2 サブシステムで ASNCOMMON、ASNQCAPTURE、ASNQAPPLY、および ASNMON パッケージをバインドすることができます。

関連概念:

- 81 ページの『オプション : プログラム・パッケージのバインド (Linux、UNIX、Windows) -- 概要』

関連タスク:

- 82 ページの『オプション : Q キャプチャー・プログラム・パッケージのバインド (Linux、UNIX、Windows)』

- 83 ページの『オプション : Q アプリ・プログラム・パッケージのバインド (Linux、UNIX、Windows)』
- 84 ページの『オプション : レプリケーション・アラート・モニター・パッケージのバインド (Linux、UNIX、Windows)』

第 22 章 ソース表とターゲット表間の相違検出および修復

ソース表とターゲット表間の相違検出および修復

ターゲット表にユーザーまたはアプリケーションが予期しない変更を行った場合、または、システムが拡張ネットワークや、ターゲット・システムの障害を起こしている場合などは、ソース表とターゲット表の同期ができません。

`tdiff` および `trepair` ユーティリティーを使用すると、手動でソース表とターゲット表を比較したり、ターゲットのロード (フル・リフレッシュ) を行ったりしなくても、Q レプリケーションおよび SQL レプリケーションで、この 2 つの表の相違を検出し、修復することができます。

2 つのユーティリティーは、Q キャプチャー、Q アプライ、キャプチャー、およびアプライ・プログラムとは別に実行されます。DB2 SQL を使用してソース表とターゲット表からデータを取り出しますが、WebSphere MQ キューは使用しません。このユーティリティーは、ログ、トリガー、または分離レベルには依存していません。

手順:

以下のようにして、ソース表とターゲット表間の相違を検出し、修復します。

1. `tdiff` ユーティリティーを実行する。詳細については、『`Tdiff: 表相違検出ユーティリティー`』を参照してください。

このユーティリティーは、ソース表とターゲット表の相違検出のリストを生成します。

2. `trepair` ユーティリティーを実行する。詳細については、332 ページの『`Trepair: 表修復ユーティリティー`』を参照してください。

このユーティリティーは、ソース表で一致していない行を削除し、ターゲット表では一致していないが、ソース表にはある行を挿入することにより、ターゲット表を修復します。

Tdiff: 表相違検出ユーティリティー

`tdiff` ユーティリティーは、ソース表にあるすべての列を、ターゲット表の対応する列と比較し、DB2[®] UDB 表の形式の 2 つの表間の相違リストを生成します。

`tdiff` ユーティリティーを使用する場合、`asntdiff` コマンドを実行し、比較するソースおよびターゲット表を含む Q サブスクリプション (Q レプリケーション) の名前またはサブスクリプション・セット・メンバー (SQL レプリケーション) の名前を指定します。

`asntdiff` コマンドは、Linux、UNIX[®]、Windows[®]、および z/OS[™] オペレーティング・システム上で実行可能です。コマンドは、Linux、UNIX、Windows、z/OS、また

は iSeries™ オペレーティング・システム上で表を比較します。 **asntdiff** コマンドはフェデレーテッド・ソースとターゲットで使用できます。

Q レプリケーションの場合、ターゲットは表でなければならず、ストアード・プロシージャにすることはできません。 **SQL** レプリケーションの場合、ターゲットはユーザー表でなければならず、ポイント・イン・タイム表、レプリカ表、ユーザーがコピーした表にすることはできません。

コマンドを実行して、**Q** サブスクリプションまたはサブスクリプション・セット・メンバーを一意的に識別する **SQL WHERE** 文節を以下のように指定します。

Q レプリケーション

WHERE 文節は、**Q** キャプチャー・サーバーのある **IBMQREP_SUBS** コントロール表の行を、**SUBNAME** 列の値に基づいて識別します。例:

```
where="where subname = 'my_qsub'"
```

SQL レプリケーション

WHERE 文節は、アプライ・コントロール・サーバーにある **IBMSNAP_SUBS_MEMBR** 表の行を、**SET_NAME** 列の値に基づいて識別します。例:

```
where="where set_name = 'my_set' and source_table='EMPLOYEE'"
```

WHERE 文節でさらに多くの述部を使用して、一意的にサブスクリプション・セット・メンバーを識別することが必要になる場合があります。たとえば、**IBMSNAP_SUBS_MEMBR** 表からの **APPLY_QUAL**、**SOURCE_OWNER**、**TARGET_OWNER**、または **TARGET_TABLE** 列を、文節に追加することが必要になる場合があります。

相違検出表

asntdiff コマンドを使用して、ソース・データベースまたはサブシステムに相違検出表を作成します。相違検出表は *schema.ASNTDIFF* という名前が付けられます。ここで *schema* は、比較するソース表およびターゲット表に関する情報の入った **Q** キャプチャー・コントロール表またはアプライ・コントロール表のスキーマです。

相違検出表には 2 列以上の列があります。1 つの列にある値 (末尾に空白・スペースがある「**DIFF**」という名前) は、どの表に相違のある行が入っているかを示します。他の列には、レプリケーション・キー列の値が入っています。相違検出表には、ソース表およびターゲット表の一致していない行ごとに 1 行があります。

DIFF 列に + 2 の値がある場合、以下のいずれかの理由で、ターゲット表に一致していない行があることを示しています。

- ソースに一致するキーがない。
- ソースに一致するキーはあるが、等しいデータではない。
- 行は、ソースよりもターゲットでさらに重複している。

DIFF 列に - 2 の値がある場合、以下のいずれかの理由で、ソース表に一致していない行があることを示しています。

- ターゲットに一致するキーがない。

- ターゲットに一致するキーはあるが、等しいデータではない。
- 行は、ターゲットよりもソースでさらに重複している。

値 ? 1 は、1 列以上のソース・キー列に無効な文字があることを示しています。

値 ? 2 は、1 列以上のターゲット・キー列に無効な文字があることを示しています。

例:

以下のリストは、ソースの EMPLOYEE 表を同じ表のターゲット・コピーと比較することによって戻される値です。レプリケーションのキー列は、以下のように従業員番号 EMPNO です。

```
DIFF EMPNO
+ 2 000010
+ 2 000020
- 2 000999
- 2 000010
- 2 000020
- 2 000030
- 2 000050
- 2 000070
- 2 000090
- 2 000100
```

太字の 2 行はいずれも、EMPNO 値は **000020** であるが、DIFF 値は + 2 および - 2 と異なっていることを示しています。つまり、この行は、ソースおよびターゲット表の両方にありますが、非キー・データは 2 つの表の間では異なっていることを示しています。

値 ? 1 および ? 2 はこの例にはありません。

注: Q レプリケーションで、比較するソースおよびターゲット表が、ソース表から削除を抑制するオプションをもつ Q サブスクリプションの一部である場合、**asntdiff** コマンドを実行して、ターゲット表から削除されない余分の行が相違として表示されます。

tdiff ユーティリティの使用時期

tdiff ユーティリティは、ソースおよびターゲット表が安定しているときに使用するのが最適です。このユーティリティは、Q キャプチャー・プログラムおよび Q アプライ・プログラムまたはキャプチャー・プログラムおよびアプライ・プログラムがアイドル状態の時に実行することができます。たとえば、Q キャプチャー・プログラムが、DB2 リカバリー・ログの末尾に到達し、すべての変更がターゲットにアプライされると、ユーティリティの実行が可能になります。アプリケーションがソースを更新中の場合、比較は正確ではない場合があります。

レプリケーション・プログラムが実行中であると、**asntdiff** コマンドを 2 回以上実行して、ソースおよびターゲット表との間の進行中の相違の全体像を取得しなければなりません。

関連タスク:

- 329 ページの『ソース表とターゲット表間の相違検出および修復』

関連資料:

- 「*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*」の『ASN.IBMSNAP_SUBS_MEMBR』
- 408 ページの『asntdiff: ソース表とターゲット表とのデータの比較』
- 410 ページの『asntrep: ソース表とターゲット表の間の違いの修復』
- 442 ページの『IBMQREP_SUBS 表』

Trepair: 表修復ユーティリティー

trepair ユーティリティーは、ターゲットから一致しない行を削除し、ターゲットから失われている行を挿入することにより、すべての DB2 サーバー上のソース表とターゲット表の相違を修復します。Trepair は、Linux、UNIX、または Windows オペレーティング・システム上で実行されます。

Q レプリケーションの場合、ターゲットは表でなければならず、ストアド・プロシージャにすることはできません。SQL レプリケーションの場合、ターゲットはユーザー表でなければならず、ポイント・イン・タイム表、レプリカ表、ユーザーがコピーした表にすることはできません。Q サブスクリプションでピアツーピア・レプリケーションを行うときに、trepair を使用する場合、論理表のすべてのコピーを一度に 2 コピーずつ修復しなければなりません。

trepair ユーティリティーを使用する場合、**asntdiff** コマンドの実行後、**asntrep** を実行します。**asntrep** コマンドは、ソース・データベースまたはサブシステムからターゲットへ相違検出表をコピーし、この表を使用して、ターゲット表を修復します。

asntrep コマンドは、ターゲット・データベースまたはサブシステムから相違検出表をドロップしません。ユーザーが手動で、表をドロップしなければなりません。

asntrep コマンドを使用する場合、**asntdiff** コマンドに使用したものと同一 WHERE 文節を使用して、同期させるソース表とターゲット表のある Q サブスクリプションまたはサブスクリプション・セットのメンバーを識別します。詳細については、329 ページの『Tdiff: 表相違検出ユーティリティー』を参照してください。

推奨: trepair ユーティリティーを使用する前に、ターゲット表の属性を考慮してください。修復プロセス中に、ターゲット表の参照保全 (RI) 制約はドロップされません。つまり、これは、挿入または削除が、RI 制約に違反しているため、ターゲット表からの行の挿入または削除が失敗する可能性があるということです。また、重複したソース行があると、そのユニーク索引が原因でターゲット表の修復が不可能になる可能性があります。

関連タスク:

- 329 ページの『ソース表とターゲット表間の相違検出および修復』

関連資料:

- 「*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*」の『ASN.IBMSNAP_SUBS_MEMBR』
- 408 ページの『asntdiff: ソース表とターゲット表とのデータの比較』

- 410 ページの『asntrep: ソース表とターゲット表の間の違いの修復』
- 442 ページの『IBMQREP_SUBS 表』

第 23 章 システム・サービスを使用してレプリケーション・プログラムを操作する方法

システム・サービスを使用して Q レプリケーションおよびイベント発行プログラムを操作する方法 -- 概要

それぞれのオペレーティング・システムごとに設計されているシステム・サービスを使用することによって、Q レプリケーションおよびイベント発行のレプリケーション・プログラムを操作できます。z/OS™ オペレーティング・システムは、レプリケーション・プログラムの操作に、ジョブ制御言語 (JCL)、システム開始タスク、または自動リスタート・マネージャー (ARM) を使用できます。Windows® オペレーティング・システムは、システム・サービスを使用してレプリケーション・プログラムを操作できます。レプリケーション・プログラムは、Linux オペレーティング・システム、UNIX® オペレーティング・システム、Windows オペレーティング・システム、および z/OS オペレーティング・システム上でスケジュールすることができます。

以下のトピックでは、システム・サービスについて、またそれを使用してレプリケーション・プログラムを操作する方法について説明します。

- 『システム・サービスを使用した Q レプリケーションおよびイベント発行プログラムの操作 (z/OS) -- 概要』
- 342 ページの『Windows Service Control Manager (SCM) を使用してレプリケーション・サービスを管理する方法 -- 概要』
- 347 ページの『レプリケーション・プログラムのスケジューリング -- 概要』

システム・サービスを使用したレプリケーション・プログラムの操作 (z/OS)

システム・サービスを使用した Q レプリケーションおよびイベント発行プログラムの操作 (z/OS) -- 概要

ジョブ制御言語 (JCL) またはシステム開始タスクを使用することにより、z/OS™ オペレーティング・システムで Q レプリケーションおよびイベント発行のレプリケーション・プログラムを開始することができます。自動リスタート・マネージャー (ARM) を使用して、失敗したレプリケーション・プログラムを再始動できます。

以下のトピックでは、これらのタスクとツールについて説明しています。

- 336 ページの『JCL を使用した Q レプリケーションおよびイベント発行プログラムの実行 -- 概要』
- 339 ページの『システム開始タスクを使用した Q レプリケーションおよびイベント発行プログラムの実行』
- 340 ページの『自動リスタート・マネージャー (ARM) のリカバリー・システム』

- 341 ページの『Q レプリケーションおよびイベント発行プログラムを再始動するために自動リスタート・マネージャー (ARM) をセットアップする方法』

関連概念:

- 335 ページの『システム・サービスを使用して Q レプリケーションおよびイベント発行プログラムを操作する方法 -- 概要』

JCL を使用したレプリケーション・プログラムの実行

JCL を使用した Q レプリケーションおよびイベント発行プログラムの実行 -- 概要

z/OS オペレーティング・システムは、レプリケーション・プログラムの操作に、ジョブ制御言語 (JCL) を使用できます。

以下のトピックでは、JCL を使用してレプリケーション・プログラムを操作する方法について説明します。

- 『JCL を使用した Q キャプチャー・プログラムの開始』
- 337 ページの『JCL を使用した Q アプライ・プログラムの開始』
- 338 ページの『JCL を使用したレプリケーション・アラート・モニターの開始』
- 338 ページの『JCL を使用したバッチ・モードでの Q レプリケーションおよびイベント発行プログラムの実行』
- 339 ページの『JCL を使用した、開始済みの Q レプリケーションおよびイベント発行プログラムの変更』

関連概念:

- 335 ページの『システム・サービスを使用した Q レプリケーションおよびイベント発行プログラムの操作 (z/OS) -- 概要』

関連タスク:

- 339 ページの『システム開始タスクを使用した Q レプリケーションおよびイベント発行プログラムの実行』

JCL を使用した Q キャプチャー・プログラムの開始

JCL を使用して Q キャプチャー・プログラムを実行する際、Q キャプチャー・プログラムはバッチ・ジョブとして実行します。たとえば、Q キャプチャー・ジョブは、オペレーティング・システムが Q キャプチャー・プログラムを実行するために使用するバッチ・ジョブです。言い換えると、JCL は z/OS オペレーティング・システム上で Q キャプチャー・プログラムを実行します。

手順:

JCL を使用して Q キャプチャー・プログラムを開始するには、次のようにします。

1. Q キャプチャー・ジョブの PARM フィールドに、適切なオプションの呼び出しパラメーターを指定することにより、z/OS 用の JCL を準備する。
 - 以下の条件のいずれかが当てはまる場合、JCL で TZ 環境変数および言語環境変数を設定する必要がある。
 - システム規模の /etc/profile ファイルで TZ 環境変数を設定しなかった。

- 実行中のレプリケーション・プログラムのホーム・ディレクトリーの profile ファイルで TZ 環境変数を設定しなかった。

TZ 変数の設定の詳細については、「*z/OS UNIX System Services User's Guide*」を参照してください。

次に示す呼び出し JCL の行の例では、TZ および LANG 変数を設定していません。

```
//CAPJFA EXEC PGM=ASNQCAP,  
// PARM='ENVAR('TZ=PST8PDT','LANG=en_US')/CAPTURE_SERVER=DQRG  
// capture_schema=JFA'
```

- DD ステートメント MSGS が指定されていない場合、LANG および NLSPATH 環境変数を指定する。
- レプリケーション・プログラムが /tmp ディレクトリー以外のディレクトリーに一時ファイルを作成しないようにする場合、TMPDIR 環境変数を指定する。TMPDIR 環境変数は、レプリケーション・プログラムが一時ファイルを作成するディレクトリーのパス名を指定します。

2. z/OS コンソールまたは TSO から JCL をサブミットする。

関連タスク:

- 337 ページの『JCL を使用した Q アプライ・プログラムの開始』
- 338 ページの『JCL を使用したレプリケーション・アラート・モニターの開始』

JCL を使用した Q アプライ・プログラムの開始

JCL を使用して Q アプライ・プログラムを実行する際、Q アプライ・プログラムはバッチ・ジョブとして実行します。たとえば、Q アプライ・ジョブは、オペレーティング・システムが Q アプライ・プログラムを実行するために使用するバッチ・ジョブです。言い換えると、JCL は z/OS オペレーティング・システム上で Q アプライ・プログラムを実行します。

手順:

JCL を使用して Q アプライ・プログラムを開始するには、以下の手順に従います。

Q アプライ・ジョブの PARM フィールドに適切な呼び出しパラメーターを指定することにより、z/OS 用の JCL を準備します。サイトの要件に合うように、JCL を調整します。

以下は、z/OS オペレーティング・システム上の Q アプライ・プログラムの JCL での呼び出しの例です。

```
//PLS EXEC PGM=ASNQAPP, // PARM='APPLY_SERVER=DQRG APPLY_SCHEMA=JAY'
```

APPLY_SERVER は、Q アプライ・ターゲット表が置かれている DB2 サブシステムです。

関連タスク:

- 336 ページの『JCL を使用した Q キャプチャー・プログラムの開始』
- 338 ページの『JCL を使用したレプリケーション・アラート・モニターの開始』

JCL を使用したレプリケーション・アラート・モニターの開始

JCL を使用してレプリケーション・アラート・モニターを実行する際、レプリケーション・アラート・モニターはバッチ・ジョブとして実行します。モニター・ジョブとは、オペレーティング・システムがレプリケーション・アラート・モニターを実行するために使用するバッチ・ジョブのことです。JCL は z/OS オペレーティング・システム上でレプリケーション・アラート・モニターを実行します。

手順:

レプリケーション・アラート・モニター・ジョブの PARM フィールドで適切な呼び出しパラメーターを指定して、z/OS 用の JCL を準備します。サイトの要件に合うように、JCL を調整します。ライブラリー SASNSAMP(ASNMON#) にある呼び出し JCL のサンプルは、レプリケーション・アラート・モニター (z/OS 版) に含まれています。

以下は、呼び出し JCL でのこの行の例です。

```
//monasn EXEC PGM=ASNMON,PARM='monitor_server=DSN
                                monitor_qual=monqual'
```

関連タスク:

- 336 ページの『JCL を使用した Q キャプチャー・プログラムの開始』
- 337 ページの『JCL を使用した Q アプライ・プログラムの開始』

JCL を使用したバッチ・モードでの Q レプリケーションおよびイベント発行プログラムの実行

手順:

JCL を使用することにより、バッチ・モードでレプリケーション・プログラムを実行することができます。

バッチ・モードでレプリケーション・プログラムを実行するには、次のようにします。

1. 該当するプログラムのライブラリー SASNSAMP で JCL をカスタマイズする。
表 25 は、各プログラムを開始するのに使用されるサンプル・ジョブを示します。

表 25.

サンプル	プログラム
ASNQSTRA	Q アプライ
ASNQSTRC	Q キャプチャー
ASNSTRM	レプリケーション・アラート・モニター
ASNQTON	トレース (Q キャプチャー・プログラムまたは Q アプライ・プログラムの場合)

2. DPROPR ジョブ (Q キャプチャー、Q アプライ、レプリケーション・アラート・モニター、および Asntrc) の PARM フィールドに、適切なオプションの呼び出しパラメーターを指定することにより、z/OS 版の JCL を準備する。z/OS コンソールまたは TSO から JCL をサブミットします。

JCL を使用した、開始済みの Q レプリケーションおよびイベント発行プログラムの変更

JCL を使用して、すでに開始済みのプログラムを変更することができます。

手順:

プログラムを変更するには、MODIFY コマンドを使用します。たとえば、MODIFY コマンドを使用して、すでに実行中の Q アプライ・プログラムを停止します。MODIFY コマンドは z/OS コンソールから使用する必要があります。以下の構文例は、省略形 F の使い方を示しています。

```
►► F jobname, | パラメーター | ◀◀
```

jobname は、プログラムが開始されたときに指定されるジョブ名です。たとえば、キャプチャー・プログラムを停止するには、以下のコマンドを使用します。

```
F capjfa,stop
```

MODIFY の詳細については、「z/OS MVS システム・コマンド」を参照してください。

システム開始タスクを使用した Q レプリケーションおよびイベント発行プログラムの実行

システム開始タスクを使用して、以下のレプリケーション・プログラムを操作できます。

- Q キャプチャー・プログラム
- Q アプライ・プログラム
- レプリケーション・アラート・モニター

手順:

プログラムを z/OS オペレーティング・システムのシステム開始タスクとして開始するには、次のようにします。

1. PROCLIB にプロシージャー (*procname*) を作成する。
2. *procname* について、RACF STARTED クラス内に項目を作成する。この項目は、*procname* を、Q キャプチャー・プログラムを開始するために使用される RACF ユーザー ID と関連付けます。プログラムを開始する前に、このユーザー ID に必要な DB2 許可が付与されるようにしてください。
3. z/OS コンソールから、**start *procname*** を実行する。

以下のサンプル・プロシージャーは Q キャプチャー・プログラム用です。

```
// PARM='CAPTURE_SERVER=DSN7 capture_schema=ASN startmode=cold'  
//STEPLIB DD DSN=qrh1qua1.SASNLOAD,DISP=SHR  
// DD DSN=dsnh1qua1.SDSNLOAD,DISP=SHR  
//* DD DSN=mqh1qua1.SCSQANLE,DISP=SHR  
//* DD DSN=mqh1qua1.SCSQLOAD,DISP=SHR  
//* DD DSN=xm1h1qua1.SIXMMOD1,DISP=SHR  
//CAPSPILL DD DSN=&&CAPSPILL,DISP=(NEW,DELETE,DELETE),  
// UNIT=VIO,SPACE=(CYL,(50,70)),  
// DCB=(RECFM=VB,BLKSIZE=6404)  
//MSGSD DD PATH='/usr/lpp/db2rep1_08_02/msg/En_US/db2asn.cat'  
//CEEDUMP DD SYSOUT=* //SYSRINT DD SYSOUT=* //SYSUDUMP DD DUMMY
```

'qrhlqual' は、Q レプリケーション・ターゲット・ライブラリーの高位修飾子、'dsnhlqual' は、DB2 ターゲット・ライブラリーの高位修飾子、'mqhlqual' は MQSeries ターゲット・ライブラリーの高位修飾子、'xmlhlqual' は XML Toolkit ライブラリーの高位修飾子です。Q キャプチャー・プログラムを実行する JCL は、MQSeries および XML Toolkit ライブラリーが LNKLST にインストールされていない場合、それらを STEPLIB に入れなければなりません。

関連概念:

- 335 ページの『システム・サービスを使用した Q レプリケーションおよびイベント発行プログラムの操作 (z/OS) -- 概要』

関連タスク:

- 336 ページの『JCL を使用した Q レプリケーションおよびイベント発行プログラムの実行 -- 概要』

自動リスタート・マネージャー (ARM) のリカバリー・システム

ARM を使用して、以下の Q レプリケーションおよびイベント発行プログラムを再始動することができます。

- Q キャプチャー・プログラム。
- Q アプライ・プログラム。
- レプリケーション・アラート・モニター。

ARM とは、バッチ・ジョブまたは開始タスクの可用性を向上させることができる、MVS™ オペレーティング・システム用のリカバリー・システムです。ARM は、失敗したジョブまたはタスクをオペレーターの介入なしで再始動することができます。ARM はまた、障害の発生したシステム上で実行されているジョブまたはタスクを再始動することもできます。ARM は、アプリケーションを識別するためにエレメント名を使用します。MVS ARM を使用するようにセットアップされている各アプリケーションは、それ自身について固有のエレメント名を生成し、ARM とのすべての連絡にこの名前を使用します。ARM はエレメント名をトラッキングし、エレメント名に対して再始動ポリシーを定義します。ARM の設定の詳細については、「z/OS MVS プログラミング: シスプレックス・サービス・ガイド」を参照してください。

表 26 は、ARM の構成時に各レプリケーション・プログラムで使用するエレメント名を示しています。

表 26. レプリケーション・プログラムのエレメント名

レプリケーション・プログラム	エレメント名
Q キャプチャー	ASNQCxxxxyyyy
Q アプライ	ASNQAxxxxyyyy
レプリケーション・アラート・モニター	ASNAMxxxxyyyy

表 26 で、xxxx は DB2® サブシステム名を表し、yyyy はデータ共用メンバー名を表しています。ARM は、データ共用構成の場合にのみ、データ共用メンバー名を必要とします。データ共用を使用しない構成に関しては、yyyy をブランクで置き換えてください。エレメント名の長さは、16 文字にする必要があります。エレメント名

は、シスプレックス全体を通じてユニークのものにする必要があります。このため、ARM を使用するには、1 つのサブシステムにつき、特定のプログラムのインスタンスは 1 つしか実行できません。

レプリケーション・プログラムは初期化時にエレメント名を使用して ARM に登録されます。登録時にイベント出口を ARM に提供することはありません。レプリケーション・プログラムは MVS サブシステムとして実行されるのではないため、ARM はイベント出口を必要としません。ARM は、登録されたプログラムが異常終了した場合、それを再始動します。たとえば、操作中にセグメント違反が発生する場合、ARM は Q キャプチャー・プログラムを再始動します。登録されたレプリケーション・プログラムは、通常終了した場合、または無効な登録を検出した場合は、登録解除されます。たとえば、Q アプライ・プログラムは、STOP コマンドによって終了した場合、登録解除されます。

関連概念:

- 335 ページの『システム・サービスを使用した Q レプリケーションおよびイベント発行プログラムの操作 (z/OS) -- 概要』

関連タスク:

- 341 ページの『Q レプリケーションおよびイベント発行プログラムを再始動するために自動リスタート・マネージャー (ARM) をセットアップする方法』

Q レプリケーションおよびイベント発行プログラムを再始動するために自動リスタート・マネージャー (ARM) をセットアップする方法

Q アプライ・プログラム、Q キャプチャー・プログラム、およびレプリケーション・アラート・モニターなどのレプリケーション・プログラムを再始動するように自動リスタート・マネージャー (ARM) をセットアップすることができます。

手順:

ARM をセットアップするには、次のようにします。

1. ARM をインストールする。詳細については、「*z/OS MVS プログラミング: シスプレックス・サービス・ガイド*」を参照してください。
2. レプリケーション・プログラムをセットアップする。
3. 該当するロード・モジュールを APF 許可ライブラリーにコピーする。Q キャプチャー・プログラムは、ARM を使用していない場合でも、APF によって許可されていなければなりません。

ヒント: パラメーター **NOTERM=Y** を指定して Q キャプチャー・プログラムまたは Q アプライ・プログラムを開始すると、DB2 UDB が静止してもプログラムは停止しません。この場合、プログラムは ARM から登録解除されません。プログラムは実行を続けますが、DB2 UDB が再始動されるまではデータをキャプチャーしません。

関連概念:

- 335 ページの『システム・サービスを使用した Q レプリケーションおよびイベント発行プログラムの操作 (z/OS) -- 概要』

- 340 ページの『自動リスタート・マネージャー (ARM) のリカバリー・システム』

レプリケーション・サービスの管理によるレプリケーション・プログラムの操作 (Windows)

Windows Service Control Manager (SCM) を使用してレプリケーション・サービスを管理する方法 -- 概要

Windows Service Control Manager を使用することにより、各 Q キャプチャー・サーバー、Q アプライ・コントロール・サーバー、およびモニター・コントロール・サーバーのサービスのリストを作成、実行、表示したり、サービスをドロップすることができます。DB2 UDB は他の DB2 サービスを使用してサービスをグループ化します。

まずレプリケーション・サービスを作成し、その後レプリケーション・サービスを実行します。

現行のレプリケーション・サービスのリストを表示したり、レプリケーション・サービスをドロップしたりすることもできます。サービスを開始した後で、プログラムのパラメーターを変更する場合は、サービスをドロップして、新しいサービスを作成する必要があります。

以下のトピックでは、レプリケーション・サービスについて、またそれを使用して実行できるタスクについて説明します。

- 『Q レプリケーションおよびイベント発行のレプリケーション・サービス (Windows)』
- 344 ページの『レプリケーション・サービスの作成』
- 345 ページの『レプリケーション・サービスの開始』
- 345 ページの『レプリケーション・サービスの停止』
- 346 ページの『レプリケーション・サービスのリストの表示』
- 346 ページの『レプリケーション・サービスのドロップ』

関連概念:

- 335 ページの『システム・サービスを使用して Q レプリケーションおよびイベント発行プログラムを操作する方法 -- 概要』

Q レプリケーションおよびイベント発行のレプリケーション・サービス (Windows)

レプリケーション・サービスとは、Windows® オペレーティング・システム上でレプリケーション・プログラムを開始するプログラムのことです。レプリケーション・サービスを使用して、以下のレプリケーション・プログラムを開始できます。

- Q キャプチャー・プログラム。
- Q アプライ・プログラム。
- レプリケーション・アラート・モニター。

レプリケーション・サービスを作成する際、DB2® UDB はこれを Windows Service Control Manager (SCM) に自動モードで追加し、SCM はサービスを開始します。Windows は固有のサービス名および表示名を使用してサービスを登録します。

以下の用語は、レプリケーション・サービスの命名規則を説明しています。

レプリケーション・サービス名

レプリケーション・サービス名は、各サービスを一意的に識別し、サービスを停止または開始するときにはこれを使用します。名前は以下のようなフォーマットになります。

`DB2.instance.alias.program.qualifier_or_schema`

表 27 では、レプリケーション・サービス名の入力について説明しています。

表 27. レプリケーション・サービス名の入力

入力	説明
<i>instance</i>	DB2 インスタンスの名前。
<i>alias</i>	Q キャプチャー・サーバー、Q アプライ・サーバー、またはモニター・コントロール・サーバーのデータベース別名。
<i>program</i>	以下の値のいずれか: QCAP (Q キャプチャー・プログラムの場合)、QAPP (Q アプライ・プログラムの場合)、MON (レプリケーション・アラート・モニターの場合)。
<i>qualifier_or_schema</i>	以下の ID のいずれか: Q アプライ・スキーマ、Q キャプチャー・スキーマ、またはモニター修飾子。

例: 以下のサービス名は、スキーマ ASN を持ち、INST1 という名前のインスタンスの下でデータベース DB1 を処理している Q キャプチャー・プログラムのものです。

`DB2.INST1.DB1.QCAP.ASN`

例: 以下のサービス名は、スキーマ ASN を持ち、INST1 という名前のインスタンスの下でデータベース DB1 を処理している Q アプライ・プログラムのものです。

`DB2.INST1.DB1.QAPP.ASN`

レプリケーション・サービスの表示名

表示名は、「サービス」ウィンドウで表示されるテキスト・ストリングで、読みやすい形式のサービス名です。たとえば、以下の表示名は、スキーマ ASN を持ち、INST1 という名前のインスタンスの下でデータベース DB1 を処理している Q キャプチャー・プログラムのものです。

`DB2 - INST1 DB1 QCAPTURE ASN`

サービスの記述を追加する場合は、レプリケーション・サービスを作成した後で、SCM を使用してください。また、SCM を使用し、サービスのユーザー名およびパスワードを指定することもできます。

関連タスク:

- 342 ページの『Windows Service Control Manager (SCM) を使用してレプリケーション・サービスを管理する方法 -- 概要』
- 344 ページの『レプリケーション・サービスの作成』
- 346 ページの『レプリケーション・サービスのリストの表示』
- 346 ページの『レプリケーション・サービスのドロップ』

レプリケーション・サービスの作成

Windows オペレーション・システム上で Q キャプチャー・プログラム、Q アプライ・プログラム、およびレプリケーション・アラート・モニター・プログラムを開始するために、DB2 レプリケーション・サービスを作成できます。

システムには複数のレプリケーション・サービスを追加できます。すべての Q キャプチャー・サーバーのスキーマごとにサービスを追加し、すべての Q アプライ・サーバーおよびモニター・コントロール・サーバーの修飾子ごとにサービスを追加できます。たとえば、5 つのデータベースがあり、各データベースが Q アプライ・サーバー、Q キャプチャー・サーバー、およびモニター・コントロール・サーバーである場合は、15 のレプリケーション・サービスを作成できます。各サーバーに複数のスキーマまたは修飾子がある場合は、さらに多くのサービスを作成できます。

前提条件:

レプリケーション・サービスを作成する前に、DB2 インスタンス・サービスが実行されていることを確認してください。レプリケーション・サービスを作成するときに DB2 インスタンス・サービスが実行されていないと、レプリケーション・サービスは作成されますが、自動的に開始されなくなります。

手順:

レプリケーション・サービスを作成するには、次のようにします。

asnscri コマンドを使用します。コマンド構文およびパラメーター記述の詳細については、390 ページの『asnscri: レプリケーション・プログラムを開始する DB2 レプリケーション・サービスの作成』を参照してください。

サービスを作成するときには、Windows にログオンするために使用するアカウント名と、そのアカウント名のパスワードを指定する必要があります。

ヒント: レプリケーション・サービスが正しくセットアップされた場合は、サービスが正常に開始された後、サービス名が **STDOUT** に送信されます。サービスが開始されない場合は、開始しようとしているプログラムの診断ログ・ファイルを調べてください。デフォルトでは、診断ログ・ファイルは **DB2PATH** 環境変数で指定されたディレクトリー内にあります。サービスとして開始するプログラムのパス・パラメーターを指定して (**capture_path**、**apply_path**、**monitor_path**)、このデフォルトをオーバーライドできます。また、Windows Service Control Manager (SCM) を使用して、サービスの状況を表示できます。

関連概念:

- 342 ページの『Q レプリケーションおよびイベント発行のレプリケーション・サービス (Windows)』

関連タスク:

- 346 ページの『レプリケーション・サービスのリストの表示』
- 346 ページの『レプリケーション・サービスのドロップ』

レプリケーション・サービスの開始

レプリケーション・サービスを作成したら、サービスを停止して、再度開始できます。

重要: サービスからレプリケーション・プログラムを開始した後で、同じスキーマまたは修飾子を使用して別のレプリケーション・プログラムを開始すると、エラーが発生します。

手順:

レプリケーション・サービスを開始するには、次の方法のいずれかを使用します。

- Windows Service Control Manager (SCM)
- **net start** コマンド

関連概念:

- 342 ページの『Q レプリケーションおよびイベント発行のレプリケーション・サービス (Windows)』

関連タスク:

- 345 ページの『レプリケーション・サービスの停止』

レプリケーション・サービスの停止

レプリケーション・サービスを作成したら、サービスを停止してから、再度開始できます。

重要: レプリケーション・サービスを停止すると、サービスに関連するプログラムは自動的に停止します。しかし、レプリケーション・システム・コマンド (**asnqacmd**、**asnqccmd**、または **asnmcmd**) を使用してプログラムを停止する場合、プログラムを開始したサービスは実行を続けます。これは明示的に停止しなければなりません。

手順:

以下の方法のいずれかを使用して、サービスを停止します。

- Windows Service Control Manager (SCM)
- **net stop** コマンド

関連概念:

- 342 ページの『Q レプリケーションおよびイベント発行のレプリケーション・サービス (Windows)』

関連タスク:

- 345 ページの『レプリケーション・サービスの開始』

レプリケーション・サービスのリストの表示

すべてのレプリケーション・サービスとそのプロパティのリストを表示することができます。

手順:

レプリケーション・サービスのリストを表示するには、**asnlist** コマンドを使用します。

レプリケーション・サービスのリストと各サービスの説明を表示するには、**asnlist** コマンドに *details* パラメーターを指定して使用します。

関連概念:

- 342 ページの『Q レプリケーションおよびイベント発行のレプリケーション・サービス (Windows)』

関連タスク:

- 344 ページの『レプリケーション・サービスの作成』
- 346 ページの『レプリケーション・サービスのドロップ』

関連資料:

- 394 ページの『asnslist: DB2 レプリケーション・サービスのリスト』

レプリケーション・サービスのドロップ

レプリケーション・サービスがなくなった場合は、Windows Service Control Manager (SCM) からドロップされるように、サービスをドロップできます。サービスによって開始されるプログラムの開始パラメーターを変更する場合は、サービスをドロップします。その後、新しい開始パラメーターを使用して新規サービスを作成することができます。

手順:

DB2 レプリケーション・サービスをドロップするには、**asnsdrop** コマンドを使用します。

関連概念:

- 342 ページの『Q レプリケーションおよびイベント発行のレプリケーション・サービス (Windows)』

関連タスク:

- 344 ページの『レプリケーション・サービスの作成』
- 346 ページの『レプリケーション・サービスのリストの表示』

関連資料:

- 395 ページの『asnsdrop: DB2 レプリケーション・サービスのドロップ』

レプリケーション・プログラムのスケジューリング

レプリケーション・プログラムのスケジューリング -- 概要

Q キャプチャー・プログラム、Q アプライ・プログラム、またはレプリケーション・アラート・モニター・プログラムは、指定の時間に開始するようにスケジュールすることができます。以下のトピックでは、さまざまなオペレーティング・システムでのプログラムのスケジューリングについて説明します。

- 『レプリケーション・プログラムのスケジューリング (Linux、UNIX)』
- 348 ページの『レプリケーション・プログラムのスケジューリング (Windows)』
- 349 ページの『レプリケーション・プログラムのスケジューリング (z/OS)』

関連概念:

- 335 ページの『システム・サービスを使用して Q レプリケーションおよびイベント発行プログラムを操作する方法 -- 概要』

レプリケーション・プログラムのスケジューリング (Linux、UNIX)

レプリケーション・プログラムは開始時刻をスケジュールすることができます。

手順:

Linux または UNIX オペレーティング・システム上で特定の時刻にレプリケーション・プログラムを開始するには、**at** コマンドを使用します。

表 28 は、金曜日の午後 3 時にレプリケーション・プログラムを開始するために使用されるコマンドを示しています。

表 28. レプリケーション・プログラムのコマンドのスケジューリング (Linux、UNIX)。

レプリケーション・プログラム	Linux または UNIX コマンド
Q キャプチャー	at 3pm Friday asnqcap autoprune=n
Q アプライ	at 3pm Friday asnqapply applyqual=myqual
レプリケーション・アラート・モニター	at 3pm Friday asnmon monitor_server=db2srv1 monitor_qualifier=mymon

関連概念:

- 347 ページの『レプリケーション・プログラムのスケジューリング -- 概要』

関連タスク:

- 348 ページの『レプリケーション・プログラムのスケジューリング (Windows)』
- 349 ページの『レプリケーション・プログラムのスケジューリング (z/OS)』

レプリケーション・プログラムのスケジューリング (Windows)

レプリケーション・プログラムは開始時刻をスケジュールすることができます。

前提条件:

1. AT コマンドを使用してレプリケーション・プログラムをスケジュールする前に、Windows スケジュール・サービスを開始します。
2. レプリケーション・プログラムのディレクトリー (CAPTURE_PATH、APPLY_PATH、または MONITOR_PATH) にパスワード・ファイルを作成します。パスワード・ファイルには、開始するレプリケーション・プログラムが実行されているサーバーの項目が含まれていなければなりません。
3. 出力をファイルに送信し、エラーをチェックします。ファイルへの転送時に "^\n" という文字をメモしてください。

手順:

Windows オペレーション・システム上で特定の時刻にレプリケーション・プログラムを開始するには、以下のいずれかの方法を使用します。

- Windows Service Control Manager を使用する。
- AT コマンドを使用してプログラムを特定の時刻に開始する。

表 29 は、金曜日の午後 3 時にレプリケーション・プログラムを開始するために使用されるコマンドを示しています。

表 29. レプリケーション・プログラムのコマンドのスケジューリング (Windows)。

レプリケーション・プログラム	Windows コマンド
Q キャプチャー	<pre>c:¥>at 15:00 db2cmd asnqcap capture_server=qcapdb capture_schema=schema capture_path=c:¥capture ^> c:¥capture¥asnqcap.out</pre>
Q アプライ	<pre>c:¥>at 15:00 db2cmd asnqapp apply_server=qappdb apply_schema=applyqual apply_path=c:¥apply ^> c:¥apply¥asnqapp.out</pre>
レプリケーション・アラート・モニター	<pre>c:¥>at 15:00 db2cmd asnmon monitor_server=mondb monitor_qual=monqual monitor_path=c:¥monitor ^> c:¥monitor¥asnmon.out</pre>

関連概念:

- 347 ページの『レプリケーション・プログラムのスケジューリング -- 概要』

関連タスク:

- 347 ページの『レプリケーション・プログラムのスケジューリング (Linux、UNIX)』
- 349 ページの『レプリケーション・プログラムのスケジューリング (z/OS)』

レプリケーション・プログラムのスケジューリング (z/OS)

z/OS オペレーティング・システム上でレプリケーション・プログラムの開始時刻をスケジュールするには、2 つの異なるコマンドを使用します。

手順:

z/OS オペレーティング・システム上でプログラムをスケジュールするには、以下のメソッドを使用します。

1. z/OS 用のプログラムを呼び出すプロシージャを **PROCLIB** に作成します。
2. リソース・アクセス管理機能 (RACF) モジュール (または、MVS セキュリティー・パッケージの該当する定義) を修正し、プロシージャをユーザー ID に関連付けます。
3. **SYS1.LPALIB** でモジュールをリンク・エディットします。
4. 指定した時点で Q キャプチャー・プログラムまたは Q アプライ・プログラムを開始するには、**\$TA JES2** コマンドまたは **AT NetView** コマンドのいずれかを使用します。

\$TA JES2 コマンドの使用については、「*MVS/ESA JES2* コマンド」を参照してください。**AT NetView** コマンドの使用については、「*NetView (MVS) コマンド・リファレンス*」を参照してください。

関連概念:

- 347 ページの『レプリケーション・プログラムのスケジューリング -- 概要』

関連タスク:

- 347 ページの『レプリケーション・プログラムのスケジューリング (Linux、UNIX)』
- 348 ページの『レプリケーション・プログラムのスケジューリング (Windows)』

第 5 部 Q レプリケーションおよびイベント発行の参照情報

本書の第 5 部は、以下の章で構成されています。

- 353 ページの『第 24 章 命名規則およびガイドライン』では、Q レプリケーションおよびイベント発行オブジェクトに有効な名前を説明しています。
- 359 ページの『第 25 章 Q レプリケーションおよびイベント発行用のコマンド』では、レプリケーション・センターに代わる、上級ユーザーが使用できるコマンドを説明しています。このコマンドを使用して、Q レプリケーションまたはイベント発行用のプログラムを開始、操作、変更、およびモニターすることができます。
- 415 ページの『第 26 章 Q レプリケーションおよびイベント発行のコントロール表』では、Q キャプチャー・プログラム、Q アプライ・プログラム、およびレプリケーション・アラート・モニターの制御情報を保管する表の構造を説明しています。
- 491 ページの『第 27 章 イベント発行の XML メッセージの構造』では、イベント発行時に Q キャプチャー・プログラムがアプリケーションに送信する XML メッセージのフォーマットを説明しています。

第 24 章 命名規則およびガイドライン

Q レプリケーションおよびイベント発行の命名規則とガイドライン -- 概要

Q レプリケーションおよびイベント発行についてオブジェクトを作成する場合、文字のタイプ、および各オブジェクト名の長さに関する特定の制限を守る必要があります。また、大文字と小文字が処理される方法にも注意を払う必要があります。次のトピックでは、命名規則とガイドラインを説明します。

- 『Q レプリケーションおよびイベント発行オブジェクトの命名規則』
- 356 ページの『Q レプリケーションおよびイベント発行で小文字のオブジェクト名が処理される方法』

Q レプリケーションおよびイベント発行オブジェクトの命名規則

以下の表は、Q レプリケーションおよびイベント発行でのオブジェクトの名前の制限を示しています。

表 30. Q レプリケーションおよびイベント発行でのオブジェクトの名前の制限

オブジェクト	名前の制限	長さの制限
ソース表とターゲット表	ソース表およびターゲット表の名前は、DB2 UDB 表名の命名規則に従う必要があります。	
ソース列とターゲット列	ソースおよびターゲット列の名前は、DB2 UDB 列名の命名規則に従う必要があります。	重要 (z/OS): DB2 UDB for z/OS では、表の列で短いスキーマ名と長いスキーマ名の両方がサポートされます。表の列の名前には、最大で以下のバイト数を含めることができます。 <ul style="list-style-type: none">• 18 バイト、DB2 UDB for z/OS バージョン 8 またはそれ以前の互換モードで実行しているサブシステムの場合• 128 バイト、DB2 UDB for z/OS バージョン 8 新機能モードで実行しているサブシステムの場合

表 30. Q レプリケーションおよびイベント発行でのオブジェクトの名前の制限 (続き)

オブジェクト	名前の制限	長さの制限
表の所有者 (z/OS)	DB2 UDB for z/OS では、表の所有者に短いスキーマ名と長いスキーマ名の両方がサポートされます。	表の所有者の名前には、最大で以下のバイト数を含めることができます。 <ul style="list-style-type: none"> • 30 バイト、DB2 UDB for z/OS バージョン 8 以前の互換モードで実行しているサブシステムの場合 • 128 バイト、DB2 UDB for z/OS バージョン 8 新機能モードで実行しているサブシステムの場合
送信キュー	送信キューの名前には、DB2 UDB および WebSphere MQ が VARCHAR データ・タイプに許可する文字をすべて含めることができます。送信キュー名にはスペースを入れることはできません。	48 文字以下
受信キュー	受信キューの名前には、DB2 UDB および WebSphere MQ が VARCHAR データ・タイプに許可する文字をすべて含めることができます。受信キュー名にはスペースを入れることはできません。	48 文字以下
再始動キュー	再始動キューの名前には、DB2 UDB および WebSphere MQ が VARCHAR データ・タイプに許可する文字をすべて含めることができます。 再始動キュー名にはスペースを入れることはできません。	48 文字以下
Q サブスクリプション	Q サブスクリプションの名前には、DB2 UDB が VARCHAR データ・タイプ列に許可する文字をすべて含めることができます。Q キャプチャー・プログラムの場合、すべての Q サブスクリプション名は固有でなければなりません。Q サブスクリプションの名前はソース・サーバーとターゲット・サーバー両方に保管されるので、名前がソース・サーバーとターゲット・サーバー両方のコード・ページで互換性があることを確認してください。 Q サブスクリプション名には、スペースまたはセミコロン (;) を入れることはできません。	30 文字以下

表 30. Q レプリケーションおよびイベント発行でのオブジェクトの名前の制限 (続き)

オブジェクト	名前の制限	長さの制限
Q サブスクリプション・グループ	<p>Q サブスクリプション・グループの名前には、DB2 UDB が VARCHAR データ・タイプ列に許可する文字をすべて含めることができます。</p> <p>推奨: 各論理表には固有のグループ名を使用してください。</p>	30 文字以下
XML 発行	<p>XML 発行の名前には、DB2 UDB が VARCHAR データ・タイプ列に許可する文字をすべて含めることができます。各 Q キャプチャー・プログラムについて、すべての XML 発行は固有でなければなりません。XML 発行の名前が、サブスクライブ・アプリケーションのコード・ページと互換性があることを確認してください。</p> <p>XML 発行には、スペースまたはセミコロン (;) を入れることはできません。</p>	30 文字以下
Q キャプチャー・スキーマ	<p>Q キャプチャー・スキーマの名前に入れられるのは、以下の有効な文字だけです。</p> <ul style="list-style-type: none"> • A ~ Z (英大文字) • a ~ z (英小文字) • 数表示 (0 ~ 9) • 下線文字 (_) 	<p>Q キャプチャー・スキーマの名前は、以下のストリングにすることができます。</p> <ul style="list-style-type: none"> • Linux、UNIX、Windows: 30 以下の文字 • z/OS バージョン 8 以前の互換モードで実行しているサブシステム: 18 以下の文字 • z/OS バージョン 8 の新機能モードで実行しているサブシステム: 128 以下の文字
Q アプライ・スキーマ	<p>Q アプライ・スキーマの名前に入れられるのは、以下の有効な文字だけです。</p> <ul style="list-style-type: none"> • A ~ Z (英大文字) • a ~ z (英小文字) • 数表示 (0 ~ 9) • 下線文字 (_) 	<p>Q アプライ・スキーマの名前は、以下のストリングにすることができます。</p> <ul style="list-style-type: none"> • Linux、UNIX、Windows: 30 以下の文字 • z/OS バージョン 8 以前の互換モードで実行しているサブシステム: 18 以下の文字 • z/OS バージョン 8 の新機能モードで実行しているサブシステム: 128 以下の文字

表 30. Q レプリケーションおよびイベント発行でのオブジェクトの名前の制限 (続き)

オブジェクト	名前の制限	長さの制限
モニター修飾子	<p>モニター修飾子の名前に入れられるのは、以下の有効な文字だけです。</p> <ul style="list-style-type: none"> • A ~ Z (英大文字) • a ~ z (英小文字) • 数表示 (0 ~ 9) • 下線文字 (_) 	<p>モニター修飾子の名前は、18文字以下のストリングにすることができます。</p>

関連概念:

- 291 ページの『レプリケーション・アラート・モニター』
- 5 ページの『Q サブスクリプション』
- 21 ページの『Q アプライ・プログラムと Q キャプチャー・プログラムのスキーマ』
- 12 ページの『XML 発行』

関連資料:

- 356 ページの『Q レプリケーションおよびイベント発行で小文字のオブジェクト名が処理される方法』

Q レプリケーションおよびイベント発行で小文字のオブジェクト名が処理される方法

Q レプリケーションおよびイベント発行のシステム・コマンドおよびレプリケーション・センターは、デフォルトとして、ユーザーから指定されたすべての名前を大文字に変換します。入力された名前のおりに正確に大文字と小文字を維持するには、大文字小文字混合文字の名前を二重引用符 (またはターゲット・システムで使用できるように構成されている他の文字) で囲んでください。たとえば、myqual または MyQual または MYQUAL と入力すると、名前は MYQUAL として保管されます。これらの同じ名前を二重引用符で囲んで入力すると、myqual または MyQual または MYQUAL としてそれぞれ保管されます。オペレーティング・システムによっては二重引用符が認識されないことがあります。その場合は、エスケープ文字としてバックスラッシュまたは円記号 (¥) を使用する必要があります。

重要 (Windows の場合): Q キャプチャー・プログラム、Q アプライ・プログラム、またはレプリケーション・アラート・モニター用に Windows サービスをセットアップするときには、Q キャプチャー・スキーマ、Q アプライ・スキーマ、およびモニター修飾子にユニークな名前を使用する必要があります。大文字と小文字を使用してこれらの名前を区別することはできません。同じ名前を区別するために固有のパスを使用する必要があります。たとえば、3 つの Q アプライ・スキーマ、myschema、MySchema、および MYSCHEMA を使用するとします。3 つの名前は同じ文字を使用していますが、大文字と小文字が異なります。これら 3 つの修飾子が同じ Q アプライ・サーバーにあると、名前の競合が発生します。

WebSphere MQ オブジェクトの場合、すべての命名規則は、WebSphere MQ の指定と同じになります。

関連概念:

- 291 ページの『レプリケーション・アラート・モニター』
- 5 ページの『Q サブスクリプション』
- 21 ページの『Q アプライ・プログラムと Q キャプチャー・プログラムのスキーマ』
- 12 ページの『XML 発行』

関連資料:

- 353 ページの『Q レプリケーションおよびイベント発行オブジェクトの命名規則』

第 25 章 Q レプリケーションおよびイベント発行用のコマンド

Q レプリケーションおよびイベント発行用のシステム・コマンド -- 概要

このセクションでは、Q レプリケーションおよびイベント発行プログラムを開始、操作、変更、およびモニターするための、Linux、UNIX[®]、Windows[®]、および z/OS[™] 上の UNIX System Services (USS) のコマンドについて説明します。一部のコマンド (レプリケーション・アラート・モニター用、Windows サービス用、トレース機能用、およびパスワード・ファイル用) は、SQL レプリケーションと共有されます。

以下のトピックでは、システム・コマンドとその構文について説明します。

- 『ロードマップ: Q レプリケーションおよびイベント発行のシステム・コマンド』
- 361 ページの『asnqcap: Q キャプチャー・プログラムの開始』
- 368 ページの『asnqcmd: 実行中の Q キャプチャー・プログラムの操作』
- 372 ページの『asnqapp: Q アプライ・プログラムの開始』
- 376 ページの『asnqacmd: 実行中の Q アプライ・プログラムの操作』
- 379 ページの『asnmon: レプリケーション・アラート・モニターの始動』
- 384 ページの『asnmcmd: 実行中のレプリケーション・アラート・モニターの操作』
- 390 ページの『asnscrt: レプリケーション・プログラムを開始する DB2 レプリケーション・サービスの作成』
- 394 ページの『asnslst: DB2 レプリケーション・サービスのリスト』
- 395 ページの『asnscdrop: DB2 レプリケーション・サービスのドロップ』
- 396 ページの『asnpwd: パスワード・ファイルの作成および保守』
- 400 ページの『asntrc: レプリケーション・トレース機能の操作』
- 408 ページの『asntdiff: ソース表とターゲット表とのデータの比較』
- 410 ページの『asntrep: ソース表とターゲット表の間の違いの修復』
- 412 ページの『asnqfmt: Q レプリケーションとイベント発行のメッセージのフォーマットおよび表示』

ロードマップ: Q レプリケーションおよびイベント発行のシステム・コマンド

レプリケーション・プログラムを開始、捜査、および変更するために、Linux、UNIX、Windows、および z/OS 上の UNIX System Services (USS) 上でシステム・コマンドを使用することができます。

パラメーターは、**name=value** の対で、任意の順序で指定できます。パラメーターとその引き数は大文字小文字が区別されません。大文字小文字を区別したい場合は二重引用符 ("") を使用してください。

引き数なしで **yes/no** (ブール) パラメーターを指定することもサポートされていますが、推奨しません。たとえば、**logreuse** と **logreuse=y** は同じことです。しかし、**logreuse** なしを指定するには、**logreuse=n** を使用しなければなりません。

疑問符を付けてコマンドを呼び出すと (たとえば **asnqcap "?"**)、コマンド構文を示すヘルプ・メッセージが表示されます。

表 31 は、システム・コマンドとの共通タスクの突き合わせに役立ちます。

表 31. Q レプリケーションおよびイベント発行のタスクとそれに対応するシステム・コマンド

目的	このコマンドを使用します。	参照ページ
Q キャプチャー・プログラムを開始し、開始パラメーターを指定する (Linux、UNIX、Windows、z/OS)	asnqcap	361
実行中の Q キャプチャー・プログラムを操作する (Linux、UNIX、Windows、z/OS)	asnqccmd	368
<ul style="list-style-type: none"> • パラメーター値を検査する • パラメーターを変更する • コントロール表を整理する • Q キャプチャー状況を検査する • Q キャプチャーを停止する • 1 つまたはすべての Q サブスクリプションまたは XML 発行を再初期化する • 1 つの送信キューを再初期化する 		
Q アプライ・プログラムを開始し、開始パラメーターを指定する (Linux、UNIX、Windows、z/OS)	asnqapp	372
実行中の Q アプライ・プログラムを操作する (Linux、UNIX、Windows、z/OS)	asnqacmd	376
<ul style="list-style-type: none"> • パラメーター値を検査する • パラメーターを変更する • Q アプライ状況を検査する • コントロール表を整理する • Q アプライを停止する • Q アプライのキューからの読み取りを停止する • Q アプライのキューからの読み取りを開始する • 1 つの受信キューを再初期化する 		
レプリケーション・アラート・モニターを開始し、開始パラメーターを指定する (Linux、UNIX、Windows、z/OS)	asnmon	379

表 31. Q レプリケーションおよびイベント発行のタスクとそれに対応するシステム・コマンド (続き)

目的	このコマンドを使用します。	参照ページ
実行中のレプリケーション・アラート・モニターを操作する (Linux、UNIX、Windows、z/OS) <ul style="list-style-type: none"> • パラメーター値を検査する • パラメーターを変更する • モニター状況を検査する • モニターを停止する • モニターを再初期化する 	asnmcmd	384
Q レプリケーション・プログラムまたは SQL レプリケーション・プログラムを開始する DB2 レプリケーション・サービスを作成する (Windows のみ)	asnscri	390
DB2 レプリケーション・サービスをリストする (Windows のみ)	asnslist	394
DB2 レプリケーション・サービスをドロップする (Windows のみ)	asnsdrop	395
Q レプリケーション・アナライザーを操作する (Linux、UNIX、Windows)	asnqanalyze	386
パスワード・ファイルを作成および保守する (Linux、UNIX、Windows)	asnpwd	396
レプリケーション・トレース機能を操作する (Linux、UNIX、Windows、z/OS)	asntrc	400
ソース表とターゲット表のデータを比較する (Linux、UNIX、Windows、z/OS)	asntdiff	408
ソース表とターゲット表の間の相違を修復する (Linux、UNIX、Windows)	asntrep	410
Q レプリケーションで使用される WebSphere MQ メッセージをフォーマットおよび表示する (Linux、UNIX、Windows、z/OS)	asnqfmt	412

関連概念:

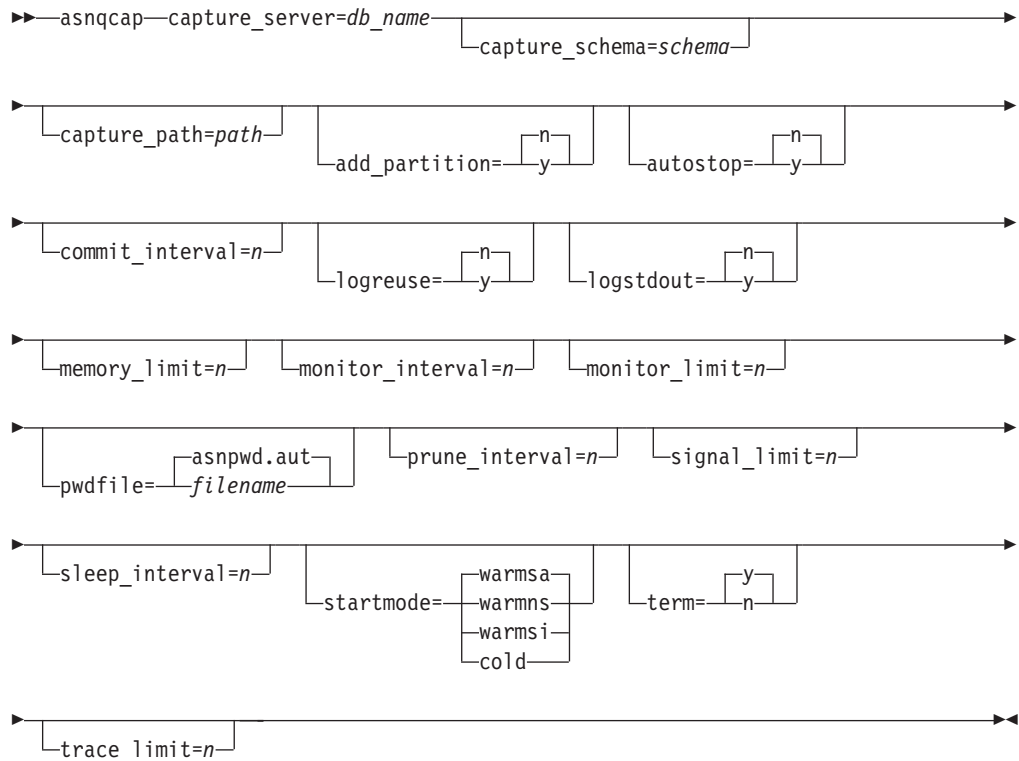
- 359 ページの『Q レプリケーションおよびイベント発行用のシステム・コマンド -- 概要』

asnqcap: Q キャプチャー・プログラムの開始

Linux、UNIX、Windows、および z/OS 上の UNIX System Services (USS) で Q キャプチャー・プログラムを開始するには、**asnqcap** コマンドを使用します。このコマンドは、オペレーティング・システムのプロンプトまたはシェル・スクリプト内で実行します。指定する開始パラメーターはいずれもこのセッションにのみ適用されます。

Q キャプチャー・プログラムを開始すると、停止されるかまたはリカバリー不能エラーが検出されるまで実行を続けます。

構文



パラメーター

表 32 は、**asnqcap** コマンドの呼び出しパラメーターを定義します。

表 32. *asnqcap* 呼び出しパラメーターの定義

パラメーター	定義
capture_server=db_name	Q キャプチャー・コントロール表を含むデータベースまたはサブシステムの名前を指定します。 Linux、UNIX、Windows: Q キャプチャー・サーバーを指定しない場合、このパラメーターはデフォルトで DB2DBDFT 環境変数の値になります。 z/OS: Q キャプチャー・プログラムを実行する DB2 サブシステムの名前を指定します。データ共有の場合、グループ・アタッチ名を使用しないでください。その代わりに、メンバー・サブシステム名を指定してください。
capture_schema=schema	開始したい Q キャプチャー・プログラムを識別する名前を指定します。
capture_path=path	Q キャプチャー・プログラムのログおよび作業ファイルの作成先となるロケーションを指定します。デフォルトは、 asnqcap コマンドを呼び出したディレクトリーです。これは絶対パス名であり、大文字小文字を区別する場合は二重引用符 (") を使用します。

表 32. *asncap* 呼び出しパラメーターの定義 (続き)

パラメーター	定義
add_partition=y/n	<p>Linux、UNIX、Windows: Q キャプチャー・プログラムが最後に再始動されてから追加されたパーティションのログ・ファイルの読み取りを Q キャプチャー・プログラムが開始するかどうかを指定します。</p> <p>y Q キャプチャー・プログラムは、1 つ以上の新規パーティション上でログ・ファイルの読み取りを開始します。各パーティション上で、Q キャプチャー・プログラムはデータベースを最後に始動したときに最初に使用されたログ・シーケンス番号 (LSN) からログの読み取りを開始します。</p> <p>n (デフォルト) Q キャプチャー・プログラムが最後に再始動されてから、新規パーティションは追加されていません。</p>
autostop=y/n	<p>アクティブ DB2 ログの最後に達した後、Q キャプチャー・プログラムが停止するかどうかを指定します。</p> <p>y アクティブ DB2 ログの最後に達すると、Q キャプチャー・プログラムは停止します。</p> <p>n (デフォルト) アクティブ DB2 ログの最後に達した後、Q キャプチャー・プログラムは停止しません。</p>
commit_interval=n	<p>Q キャプチャー・プログラムが MQCMIT 呼び出しを発行する頻度をミリ秒で指定します。この呼び出しは、キューに置かれたデータ・メッセージおよび通知メッセージを、Q アプリ・プログラムまたはサブスクライブ・アプリケーションで使用可能にするための信号を WebSphere MQ キュー・マネージャーに送ります。デフォルトは 500 ミリ秒 (0.5 秒) です。</p>
logreuse=y/n	<p>Q キャプチャー・プログラムがその診断ログ・ファイルを再利用するかどうか、またはこのファイルにメッセージを追加するかどうかを指定します (ログ・ファイル名は、Linux、UNIX、および Windows では <i>db2instance.capture_server.capture_schema.QCAP.log</i>、z/OS では <i>capture_server.capture_schema.QCAP.log</i> となります)。</p> <p>y Q キャプチャー・プログラムは再始動時にログ・ファイルをクリアして、そのブランク・ファイルに書き込むことで、そのログ・ファイルを再利用します。</p> <p>n (デフォルト) Q キャプチャー・プログラムは、Q キャプチャー・プログラムの再始動後であってもログ・ファイルにメッセージを付加します。</p>

表 32. *asncap* 呼び出しパラメーターの定義 (続き)

パラメーター	定義
logstdout=y/n	<p>Q キャプチャー・プログラムがその診断ログ・ファイルとコンソールの両方にログ・メッセージを送信するかどうかを指定します。</p> <p>y Q キャプチャー・プログラムは、そのログ・ファイルとコンソール (stdout) の両方にログ・メッセージを送信します。</p> <p>n (デフォルト) Q キャプチャー・プログラムは、ほとんどのログ・メッセージをログ・ファイルにのみ送ります。</p> <p>初期化、停止、およびサブスクリプションのアクティブ化および非アクティブ化に関するメッセージは、このパラメーターの設定に関係なく、コンソール (stdout) とログ・ファイルの両方に送信されます。</p>
memory_limit=n	<p>Q キャプチャー・プログラムがトランザクションの作成に使用できるメモリーの量を MB 単位で指定します。この割り振りがすべて使用されると、メモリー内のトランザクションはファイルに書き出されます。デフォルトは、Linux、UNIX、および Windows の場合 32 MB、z/OS の場合 16 MB です。</p>
monitor_interval=n	<p>Q キャプチャー・プログラムが行を IBMQREP_CAPMON および IBMQREP_CAPQMON 表に追加する頻度を秒単位で指定します。デフォルトは 300 秒 (5 分) です。</p>
monitor_limit=n	<p>整理の対象となるまでに行が IBMQREP_CAPMON 表および IBMQREP_CAPQMON 表の中に留まる分数を指定します。整理インターバルのたびに、これらの表中の行が現行タイム・スタンプに基づいてこの制限より古い場合、Q キャプチャー・プログラムはそれらの行を整理します。デフォルトは 10,080 分 (7 日) です。</p>
pwdfile=filename	<p>複数のパーティション・データベースとの接続に使用されるパスワード・ファイルの名前を指定します。パスワード・ファイルを指定しない場合、デフォルトは <i>asnpwd.aut</i> です。</p> <p>このコマンドは、capture_path パラメーターで指定されたディレクトリー内でパスワード・ファイルを探します。 capture_path パラメーターを指定しない場合、このコマンドは、コマンドを呼び出したディレクトリー内でパスワード・ファイルを探します。</p>
prune_interval=n	<p>古くなったために整理する IBMQREP_SIGNAL、IBMQREP_CAPTRACE、IBMQREP_CAPMON、および IBMQREP_CAPQMON 表内の行を Q キャプチャー・プログラムが検索する頻度を秒単位で指定します。デフォルトは 300 秒 (5 分) です。</p>

表 32. *asncap* 呼び出しパラメーターの定義 (続き)

パラメーター	定義
signal_limit=n	整理の対象となるまでに行が <code>IBMQREP_SIGNAL</code> 表の中に留まる分数を指定します。整理インターバルのたびに、 <code>IBMQREP_SIGNAL</code> 表中の行が現行タイム・スタンプに基づいて行がシグナル制限より古い場合、Q キャプチャー・プログラムはそれらの行を整理します。デフォルトは 10,080 分 (7 日) です。
sleep_interval=n	Q キャプチャー・プログラムが、アクティブ・ログと、メモリー内に留まっているトランザクションの処理の後アイドル状態になる秒数を指定します。デフォルトは 5000 ミリ秒 (5 秒) です。

表 32. *asncap* 呼び出しパラメーターの定義 (続き)

パラメーター	定義
startmode=mode	<p>Q キャプチャー・プログラムが始動時に取るアクションを指定します。</p> <p>warmsi (デフォルト) Q キャプチャー・プログラムは、プログラムの初回始動時を除いて、中止されたところからログの読み取りを開始します。Q キャプチャー・プログラムは、プログラムの初回始動時にコールド・スタートに切り替わります。warmsi 始動モードを使用すると、Q キャプチャー・プログラムは確実に、初回始動時のみコールド・スタートします。</p> <p>warmns Q キャプチャー・プログラムは、中止されたところからログの読み取りを開始します。ウォーム・スタートできない場合、コールド・スタートに切り替わりません。warmns 始動モードを使用すると、Q キャプチャー・プログラムの予期しないコールド・スタートを避けられます。ウォーム・スタートした場合、Q キャプチャー・プログラムは終了したところから処理を再開します。Q キャプチャー・プログラムの開始後にエラーが発生すると、プログラムは終了し、すべての表は未変更のままになります。</p> <p>warmsa ウォーム・スタート情報が使用可能な場合、Q キャプチャー・プログラムは、ログの読み取りを中止した位置から開始します。Q キャプチャー・プログラムは、ウォーム・スタートできない場合、コールド・スタートに切り替わります。</p> <p>cold Q キャプチャー・プログラムは再始動キューと管理キューをクリアし、N (新規) または A (アクティブ) 状態にあるすべての Q サブスクリプションまたは XML 発行の処理を開始します。コールド・スタートの場合、Q キャプチャー・プログラムは DB2 リカバリー・ログの読み取りを、その末尾から開始します。</p> <p>ウォーム・スタート中、Q キャプチャー・プログラムは、I (非アクティブ) 状態になっていない Q サブスクリプションまたは XML 発行のみをロードします。</p>

表 32. *asnqcap* 呼び出しパラメーターの定義 (続き)

パラメーター	定義
term=y/n	<p>DB2 が静止または終了した場合に Q キャプチャー・プログラムは終了するかどうかを指定します。</p> <p>y (デフォルト) DB2 が静止または終了した場合、Q キャプチャー・プログラムは終了します。</p> <p>n DB2 が静止した場合、Q キャプチャー・プログラムは実行を継続します。DB2 が静止モードではなくなると、Q キャプチャー・プログラムは warmns モードで始動し、IBMQREP_CAPPARMS 表を再び読み取り、DB2 が静止されて中止されたところからキャプチャーを開始します。</p> <p>DB2 が FORCE または異常終了で終了した場合は、このパラメーターを n にしても Q キャプチャー・プログラムは終了します。</p> <p>このパラメーターを n にし、制限付きアクセス (ACCESS MAINT) を使用して DB2 を始動すると、Q キャプチャー・プログラムは接続できないので、結果として終了します。</p>
trace_limit=n	<p>整理の対象となるまでに行が IBMQREP_CAPTRACE 表の中に留まる分数。整理インターバルのたびに、この表中の行が現行タイム・スタンプに基づいてトレース制限より古い場合は、Q アプライ・プログラムはそれらの行を整理する。デフォルトは 10,080 分 (7 日) です。</p>

asnqcap の例

次の例は、**asnqcap** コマンドの使用方法を示しています。

例 1

warmsi を使用するスキーマ alpha で、sourcedb という名前のサーバー上で Q キャプチャー・プログラムを開始するには、また logreuse および logstdout のデフォルト設定を一時的にオーバーライドするには、次のようにします。

```
asnqcap capture_server=sourcedb capture_schema="alpha" startmode=warmsi
logreuse=y logstdout=y
```

例 2

Q キャプチャー・プログラムを開始し、デフォルト以上の頻度でキュー上のメッセージをコミットするように指示するには、次のようにします。

```
asnqcap capture_server=sourcedb capture_schema="alpha" commit_interval=250
```

例 3

Q キャプチャー・プログラムを開始し、トランザクションの作成に使用するデフォルトのメモリー量を一時的に増やすには、次のようにします。

```
asncqcap capture_server=sourcedb capture_schema="alpha" memory_limit=64
```

例 4

Q キャプチャー・プログラムを開始し、その作業ファイルを /home/files/qcapture ディレクトリーに送信するには、次のようにします。

```
asncqcap capture_server=sourcedb capture_schema="alpha"  
capture_path="/home/files/qcapture"
```

関連概念:

- 229 ページの『Q キャプチャー・プログラムの操作 -- 概要』
- 359 ページの『Q レプリケーションおよびイベント発行用のシステム・コマンド -- 概要』

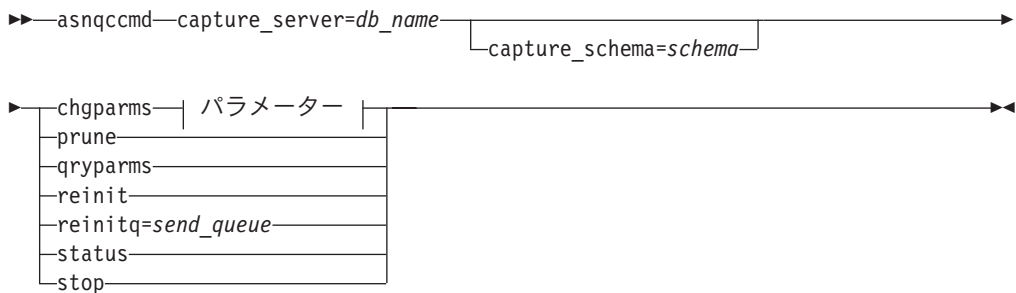
関連資料:

- 235 ページの『Q キャプチャー・パラメーターの説明』
- 359 ページの『ロードマップ: Q レプリケーションおよびイベント発行のシステム・コマンド』

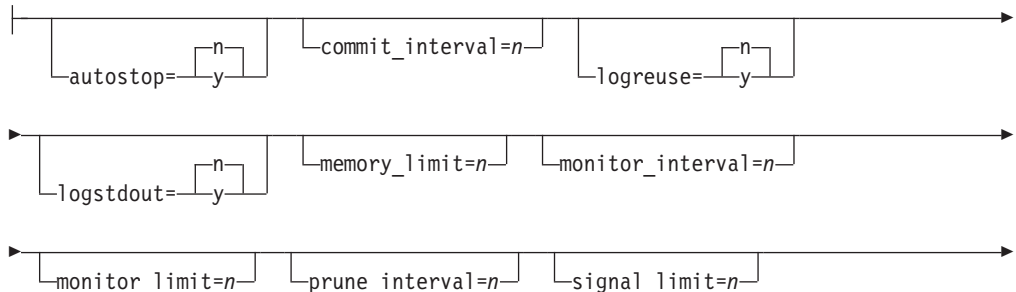
asncqcmd: 実行中の Q キャプチャー・プログラムの操作

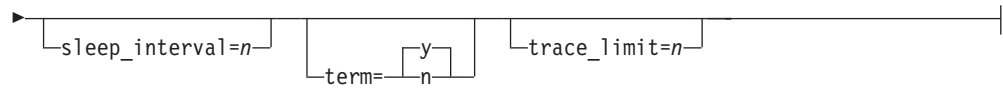
Linux、UNIX、Windows、および z/OS 上の UNIX System Services (USS) で実行中の Q キャプチャー・プログラムにコマンドを送信するには、**asncqcmd** を使用します。このコマンドは、オペレーティング・システムのプロンプトまたはシェル・スクリプト内で実行します。

構文



パラメーター:





パラメーター

表 33 は、**asnlccmd** コマンドの呼び出しパラメーターを定義します。

表 33. *asnlccmd* 呼び出しパラメーターの定義

パラメーター	定義
capture_server = <i>db_name</i>	<p>Q キャプチャー・コントロール表を含むデータベースまたはサブシステムの名前を指定します。</p> <p>Linux、UNIX、Windows: Q キャプチャー・サーバーを指定しない場合、このパラメーターはデフォルトで DB2DBDFT 環境変数の値になります。</p> <p>z/OS: Q キャプチャー・プログラムを実行する DB2 サブシステムの名前を指定します。データ共有の場合、グループ・アタッチ名を使用しないでください。その代わりに、メンバー・サブシステム名を指定してください。</p>
capture_schema = <i>schema</i>	<p>処理したい Q キャプチャー・プログラムを識別する名前を指定します。</p>

表 33. *asnlccmd* 呼び出しパラメーターの定義 (続き)

パラメーター	定義
chgpargs	<p>Q キャプチャー・プログラムの実行中に、以下の稼働パラメーターのうち 1 つ以上を変更することを指定します。</p> <ul style="list-style-type: none"> • autostop • commit_interval • logreuse • logstdout • memory_limit • monitor_interval • monitor_limit • prune_interval • signal_limit • sleep_interval • term • trace_limit <p>1 つの asnlccmd chgpargs コマンド中に複数のパラメーターを指定したり、必要に応じてこれらのパラメーター値を変更したりできます。変更内容は IBMQREP_CAPPARMS 表の値を一時的にオーバーライドしますが、この表に書き込まれません。Q キャプチャー・プログラムを停止して再始動する際には、IBMQREP_CAPPARMS 中の値が使用されます。</p> <p>asnlccap: Q キャプチャー・プログラムの開始には、このコマンドを使用してオーバーライドできるパラメーターの説明が記載されています。</p> <p>重要: 変更するパラメーターは、chgpargs パラメーターの直後に指定する必要があります。</p>
prune	<p>Q キャプチャー・プログラムに、IBMQREP_CAPMON、IBMQREP_CAPQMON、IBMQREP_CAPTRACE、および IBMQREP_SIGNAL 表の整理を 1 回行うように指示することを指定します。この整理は、prune_interval パラメーターによって指定される通常のスケジュールされた整理とは別のものです。</p>
qryparms	<p>Q キャプチャー・プログラムの現行の稼働パラメーター値を標準出力 (stdout) に書き込む場合に指定します。</p>
reinit	<p>Q キャプチャー・プログラムを非アクティブ化し、その後 IBMQREP_SUBS、IBMQREP_SRC_COLS、および IBMQREP_SENDQUEUES 表の最新の値を使用して Q サブスクリプションおよび XML 発行をすべてアクティブにすることを指定します。このコマンドを使用すると、Q キャプチャー・プログラムの実行中に、複数の Q サブスクリプションまたは XML 発行の属性の一部を変更することができます。このコマンドは、ターゲットの新規ロードを要求しません。</p>

表 33. *asnlccmd* 呼び出しパラメーターの定義 (続き)

パラメーター	定義
reinitq=send_queue	Q キャプチャー・プログラムが <code>IBMQREP_SENDQUEUES</code> 表からの最新の属性を使用して送信キュー上で Q キャプチャー・プログラムをリフレッシュすることを指定します。このコマンドは、この送信キューを使用するすべての Q サブスクリプションまたは XML 発行に影響を与えます。属性 <code>ERROR_ACTION</code> 、 <code>HEARTBEAT_INTERVAL</code> 、 <code>MAX_MESSAGE_SIZE</code> のみリフレッシュされます。
status	各 Q キャプチャー・スレッド (メイン、管理、整理、保留、およびワーカー) の状態を示すメッセージを受け取ることを指定します。
stop	Q キャプチャー・プログラムを正しい順序で停止し、プログラムがその時点までに処理したメッセージをコミットすることを指定します。

asnlccmd の例

次の例は、**asnlccmd** コマンドの使用方法を示しています。

例 1

Q キャプチャー・プログラムに、Q キャプチャー・コントロール表の最新の値を使用してすべての Q サブスクリプションおよび XML 発行をリフレッシュすることを指示します。

```
asnlccmd capture_server=sourcedb capture_schema="alpha" reinit
```

例 2

Q キャプチャー・プログラムに、Q1 という名前の送信キューを使用するすべての Q サブスクリプションおよび XML 発行の `ERROR_ACTION`、`HEARTBEAT_INTERVAL`、および `MAX_MESSAGE_SIZE` 属性をリフレッシュすることを指示します。

```
asnlccmd capture_server=sourcedb capture_schema="alpha" reinitq="Q1"
```

例 3

実行中の Q キャプチャー・プログラムのデフォルトの整理インターバルを一時的に 1 分に短縮し、Q サブスクリプションおよび XML 発行を処理した後 Q キャプチャー・プログラムがスリープになるデフォルトの時間を一時的に延長するには、次のようにします。

```
asnlccmd capture_server=sourcedb capture_schema="alpha" chgparms
  prune_interval=60 sleep_interval=10000
```

例 4

Q キャプチャー・プログラムのスレッドの状況に関するメッセージを受け取るには、次のようにします。


```
asnlccmd capture_server=sourcedb capture_schema="alpha" status
```

関連概念:

- 229 ページの『Q キャプチャー・プログラムの操作 -- 概要』
- 359 ページの『Q レプリケーションおよびイベント発行用のシステム・コマンド -- 概要』

関連資料:

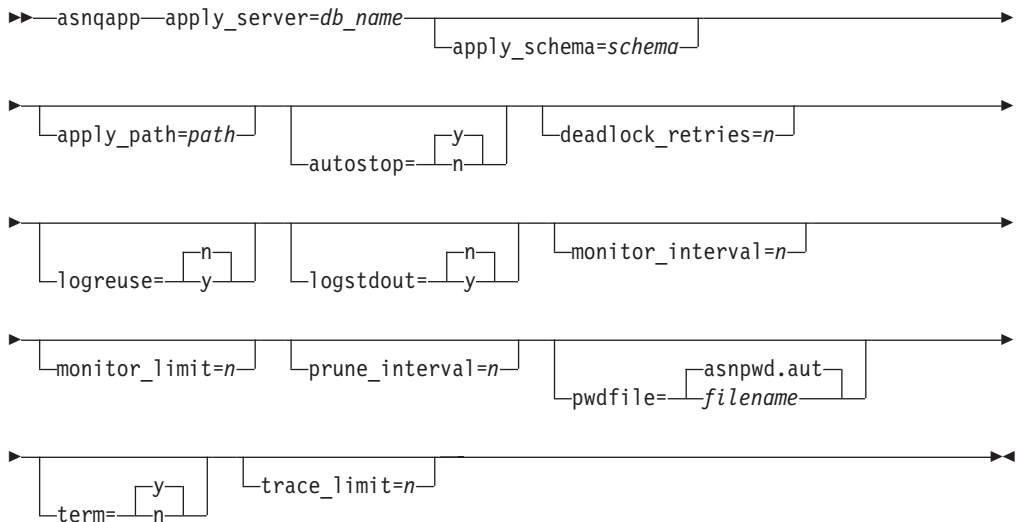
- 235 ページの『Q キャプチャー・パラメーターの説明』
- 359 ページの『ロードマップ: Q レプリケーションおよびイベント発行のシステム・コマンド』

asnlapp: Q アプリ・プログラムの開始

Linux、UNIX、Windows、および z/OS 上の UNIX System Services (USS) で Q アプリ・プログラムを開始するには、**asnlapp** コマンドを使用します。このコマンドは、オペレーティング・システムのプロンプトまたはシェル・スクリプト内で実行します。指定する開始パラメーターはいずれもこのセッションにのみ適用されます。

Q アプリ・プログラムを開始すると、停止されるかまたはリカバリー不能エラーが検出されるまで実行を続けます。

構文



パラメーター

表 34 は、**asnlapp** コマンドの呼び出しパラメーターを定義します。

表 34. *asnlapp* 呼び出しパラメーターの定義

パラメーター	定義
apply_server=db_name	<p>Q アプライ・コントロール表を含むデータベースまたはサブシステムの名前を指定します。</p> <p>Linux、UNIX、Windows: Q アプライ・サーバーを指定しない場合、このパラメーターはデフォルトで DB2DBDFT 環境変数の値になります。</p> <p>z/OS: Q アプライ・プログラムを実行する DB2 サブシステムの名前を指定します。データ共有の場合、グループ・アタッチ名を使用しないでください。その代わりに、メンバー・サブシステム名を指定してください。</p>
apply_schema=schema	開始したい Q アプライ・プログラムを識別する名前を指定します。
apply_path=path	Q アプライ・プログラムのログおよび作業ファイルの作成先となるロケーションを指定します。デフォルトは、 asnlapp コマンドを呼び出したディレクトリーです。これは絶対パス名です。大文字小文字を区別したい場合は二重引用符 (") を使用してください。
autostop=y/n	<p>一度すべての受信キューが空になった後、Q アプライ・プログラムが終了するかどうかを指定します。</p> <p>y 一度すべての受信キューが空になると Q アプライ・プログラムは停止します。ロード・フェーズ中、Q アプライ・プログラムは、Q サブスクリプションがロードされるまで autostop=y を無視します。</p> <p>n (デフォルト) 一度すべての受信キューが空になっても Q アプライ・プログラムは実行を継続します。</p>
deadlock_retries=n	SQL デッドロックまたはロック・タイムアウトの後、Q アプライ・プログラムが表に対して変更の再アプライを試みる回数。Q アプライ・プログラムは、次のデッドロックの再試行まで 1 秒待機します。デフォルトは 3 回の試行です。このデフォルト値よりも小さな値を指定することはできません。
logreuse=y/n	<p>Q アプライ・プログラムがその診断ログ・ファイルを再利用するかどうか、またはこのファイルに追加するかどうかを指定します (ログ・ファイル名は、Linux、UNIX、および Windows では <i>db2instance.apply_server.apply_schema.QAPP.log</i>、z/OS では <i>apply_server.apply_schema.QAPP.log</i> となります)。</p> <p>y Q アプライ・プログラムは再始動時にログ・ファイルをクリアして、そのブランク・ファイルに書き込むことで、そのログ・ファイルを再利用します。</p> <p>n (デフォルト) Q アプライ・プログラムは、再始動時に既存の Q アプライ・ログ・ファイルに新規情報を追加します。</p>

表 34. *asncapp* 呼び出しパラメーターの定義 (続き)

パラメーター	定義
logstdout=y/n	<p>Q アプライ・プログラムがそのファイルとコンソールの両方にログ・メッセージを送信するかどうかを指定します。</p> <p>y Q アプライ・プログラムは、ログ・ファイルとコンソール (stdout) の両方にメッセージを送信します。</p> <p>n (デフォルト) Q アプライ・プログラムは、ほとんどのログ・メッセージをログ・ファイルにのみ送ります。</p> <p>初期化、停止、およびサブスクリプションのアクティブ化および非アクティブ化に関するメッセージは、このパラメーターの設定に関係なく、コンソール (stdout) とログ・ファイルの両方に送信されます。</p>
monitor_interval=n	<p>Q アプライ・プログラムが行を IBMQREP_APPLYMON 表に追加する頻度を秒単位で指定します。デフォルトは 300 秒 (5 分) です。</p>
monitor_limit=n	<p>整理の対象となるまでに行が IBMQREP_APPLYMON 表の中に留まる分数を指定します。整理インターバルのたびに、この表中の行が現行タイム・スタンプに基づいて monitor_limit より古い場合、Q アプライ・プログラムはそれらの行を整理します。デフォルトは 10,080 分 (7 日) です。</p>
prune_interval=n	<p>古くなったために整理する IBMQREP_APPLYMON および IBMQREP_APPLYTRACE 表内の行を Q アプライ・プログラムが検索する頻度を秒単位で指定します。デフォルトは 300 秒 (5 分) です。</p>
pwdfile=file name	<p>暗号化パスワード・ファイルの名前を指定します。EXPORT/IMPORT または EXPORT/LOAD ユーティリティーを使用する自動ロードを指定した場合、Q アプライ・プログラムはパスワード・ファイルを使用して、Q キャプチャー・サーバーに接続します。 asnpwd コマンドを使用してパスワード・ファイルを作成すると、デフォルト・ファイル名は asnpwd.aut になります。Q アプライ・プログラムは、apply_path パラメーターで指定されたディレクトリー内でパスワード・ファイルを探します。 apply_path パラメーターを指定しない場合、Q アプライ・プログラムは asncapp コマンドを呼び出したディレクトリー内でパスワード・ファイルを探します。</p>

表 34. *asnlqpp* 呼び出しパラメーターの定義 (続き)

パラメーター	定義
term=y/n	<p>DB2 が静止した場合に Q アプライ・プログラムが終了するかどうかを指定します。</p> <p>y (デフォルト) DB2 が静止した場合、Q アプライ・プログラムは終了します。</p> <p>n DB2 が静止した場合、Q アプライ・プログラムは実行を継続します。DB2 が静止モードではなくなると、Q アプライ・プログラムは、DB2 が静止されて中止されたところからトランザクションのアプライを開始します。</p> <p>DB2 が FORCE または異常終了で終了した場合は、このパラメーターを n にしても Q アプライ・プログラムは終了しません。</p> <p>このパラメーターを n にし、制限付きアクセス (ACCESS MAINT) を使用して DB2 を始動すると、Q アプライ・プログラムは接続できないので、結果として終了します。</p>
trace_limit=n	<p>整理の対象となるまでに行が IBMQREP_APPLYTRACE 表の中に留まる分数。整理インターバルのたびに、この表中の行が現行タイム・スタンプに基づいて monitor_limit より古い場合、Q アプライ・プログラムはそれらの行を整理する。デフォルトは 10,080 分 (7 日) です。</p>

asnlqpp の例

次の例は、**asnlqpp** コマンドの使用方法を示しています。

例 1

スキーマ **alpha**、**/home/files/qapply** ディレクトリーにある作業ファイル、**pass1.txt** という名前のパスワード・ファイルを使用して、サーバー **targetdb** 上で Q アプライ・プログラムを開始するには、次のようにします。

```
asnlqpp apply_server=targetdb apply_schema="alpha"
apply_path="/home/files/qapply" pwordfile="pass1.txt"
```

例 2

Q アプライ・プログラムを開始し、IBMQREP_APPLYMON 表に行を挿入するためのデフォルト・インターバルを一時的に短縮するには、次のようにします。

```
asnlqpp apply_server=targetdb apply_schema="alpha" monitor_interval=100
```

例 3

Q アプライ・プログラムを開始し、一度すべてのキューが空になった後に実行を停止するように指示するには、次のようにします。

```
asnlqapp apply_server=targetdb apply_schema="alpha" autostop=y
```

関連概念:

- 259 ページの『Q アプライ・プログラムの操作 -- 概要』
- 359 ページの『Q レプリケーションおよびイベント発行用のシステム・コマンド -- 概要』

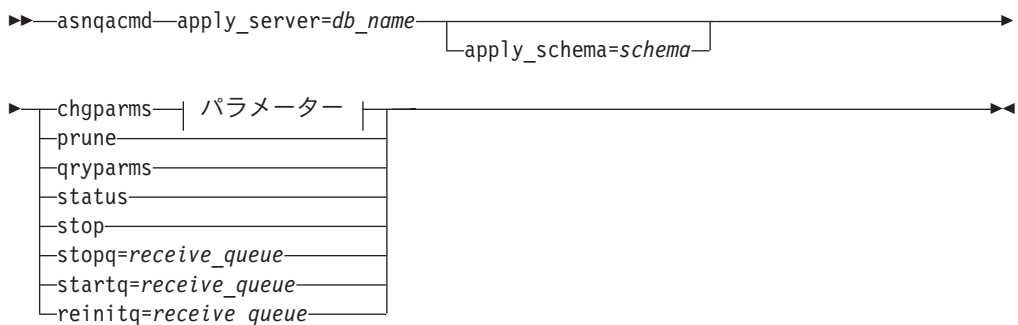
関連資料:

- 263 ページの『Q アプライ・パラメーターの説明』
- 359 ページの『ロードマップ: Q レプリケーションおよびイベント発行のシステム・コマンド』

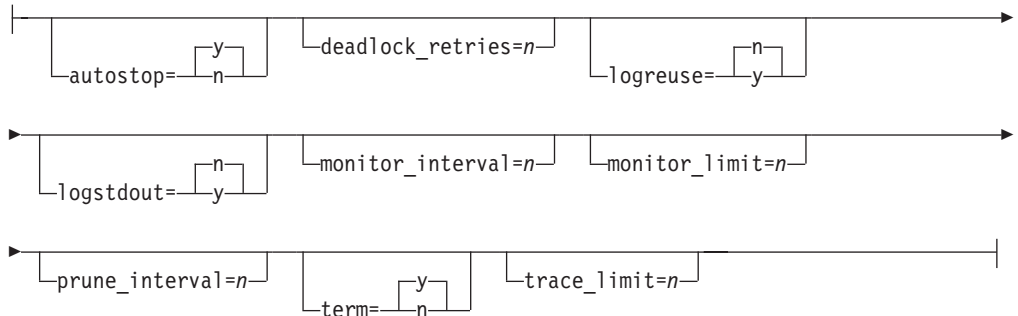
asnlqcmd: 実行中の Q アプライ・プログラムの操作

Linux、UNIX、Windows、および z/OS 上の UNIX System Services (USS) で実行中の Q アプライ・プログラムにコマンドを送信するには、**asnlqcmd** コマンドを使用します。このコマンドは、オペレーティング・システムのプロンプトまたはシェル・スクリプト内で実行します。

構文



パラメーター:



パラメーター

377 ページの表 35 は、**asnlqcmd** コマンドの呼び出しパラメーターを定義します。

表 35. *asnlqcmd* 呼び出しパラメーターの定義

パラメーター	定義
apply_server=db_name	<p>Q アプライ・コントロール表を含むデータベースまたはサブシステムの名前を指定します。</p> <p>Linux、UNIX、Windows: Q アプライ・サーバーを指定しない場合、このパラメーターはデフォルトで DB2DBDFT 環境変数の値になります。</p> <p>z/OS: Q アプライ・プログラムを実行する DB2 サブシステムの名前を指定します。データ共有の場合、グループ・アタッチ名を使用しないでください。その代わりに、メンバー・サブシステム名を指定してください。</p>
apply_schema=schema	<p>処理したい実行中の Q アプライ・プログラムを識別する名前を指定します。</p>
chgparms	<p>Q アプライ・プログラムの実行中に、以下の稼働パラメーターのうち 1 つ以上を変更することを指定します。</p> <ul style="list-style-type: none"> • autostop • logreuse • logstdout • monitor_interval • monitor_limit • prune_interval • term • trace_limit • deadlock_retries <p>1 つの asnlqcmd chgparms コマンド中に複数のパラメーターを指定したり、必要に応じてこれらのパラメーター値を変更したりできます。変更内容は IBMQREP_APPLYPARMS 表の値を一時的にオーバーライドしますが、この表に書き込まれません。Q アプライ・プログラムを停止して再始動する際には、IBMQREP_APPLYPARMS 中の値が使用されます。</p> <p>asnlqapp: Q アプライ・プログラムの開始には、このコマンドを使用してオーバーライドできるパラメーターの説明が記載されています。</p> <p>重要: 変更するパラメーターは、chgparms パラメーターの直後に指定する必要があります。</p>
prune	<p>Q アプライ・プログラムに、一度 IBMQREP_APPLYMON および IBMQREP_APPLYTRACE 表を整理することを指示します。この整理は、prune_interval パラメーターによって指定される通常のスケジュールされた整理とは別のものです。</p>
qryparms	<p>Q アプライ・プログラムの現行の稼働パラメーター値を標準出力 (stdout) に書き込む場合に指定します。</p>
status	<p>各アプライ・スレッド (メイン、ハウスキーピング、ブラウザー、およびエージェント) の状態に関するメッセージを確認することを指定します。</p>

表 35. *asnlqcmd* 呼び出しパラメーターの定義 (続き)

パラメーター	定義
stop	Q アプライ・プログラムを正しい順序で停止し、プログラムがその時点までに処理したメッセージをコミットすることを指定します。
stopq=receive_queue	Q アプライ・プログラムに、受信キューのメッセージの処理を停止するように指示します。メモリー内のトランザクションはすべて処理されます。
startq=receive_queue	Q アプライ・プログラムに、受信キューからのメッセージの処理を開始することを指示します (まだ処理されていない場合)。
reinitq=receive_queue	Q アプライ・プログラムが、特定の受信キューを使用するすべての Q サブスクリプションに関する <code>IBMQREP_RECVQUEUES</code> 表からの <code>NUM_APPLY_AGENTS</code> および <code>MEMORY_LIMIT</code> 属性を更新することを指定します。このコマンドは、Q アプライ・プログラムが、コマンドを発行するときに <code>IBMQREP_RECVQUEUES</code> 表内で命名される受信キューから読み取りを行っている場合にのみ機能します。

asnlqcmd の例

次の例は、**asnlqcmd** コマンドの使用方法を示しています。

例 1

Q1 という名前の受信キューを使用するすべての Q サブスクリプションを、Q アプライ・エージェントの数および `IBMQREP_RECVQUEUES` 表からのメモリー制限についての最新の値で更新するには、次のようにします。

```
asnlqcmd apply_server=targetdb apply_schema="alpha" reinitq=Q1
```

例 2

実行中の Q アプライ・プログラムに、すべてのキューが一度空になった後に停止し、モニター・インターバルを延長してトレース制限を短縮することを指示するには、次のようにします。

```
asnlqcmd apply_server=targetdb apply_schema="alpha" chgparms autostop=y
monitor_interval=500 trace_limit=5000
```

例 3

それぞれの Q アプライ・スレッドの状態についてメッセージを受け取るには、次のようにします。

```
asnlqcmd apply_server=targetdb apply_schema="alpha" status
```

例 4

Q アプライ・プログラムの現行の稼働パラメーター値を標準出力に送信するには、次のようにします。


```
asnqacmd apply_server=targetdb apply_schema="alpha" qryparms
```

例 5

一度 IBMQREP_APPLYMON および IBMQREP_APPLYTRACE 表を整理するには、次のようにします。

```
asnqacmd apply_server=targetdb apply_schema="alpha" prune
```

関連概念:

- 259 ページの『Q アプライ・プログラムの操作 -- 概要』
- 359 ページの『Q レプリケーションおよびイベント発行用のシステム・コマンド -- 概要』

関連資料:

- 263 ページの『Q アプライ・パラメーターの説明』
- 359 ページの『ロードマップ: Q レプリケーションおよびイベント発行のシステム・コマンド』

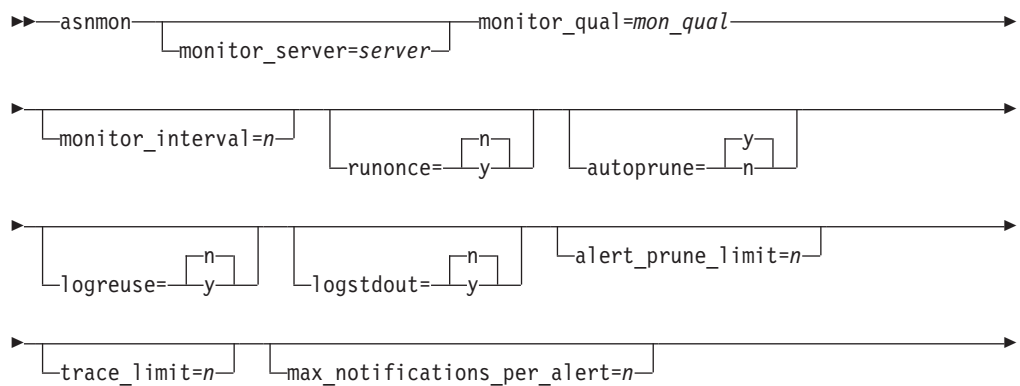
asnmon: レプリケーション・アラート・モニターの始動

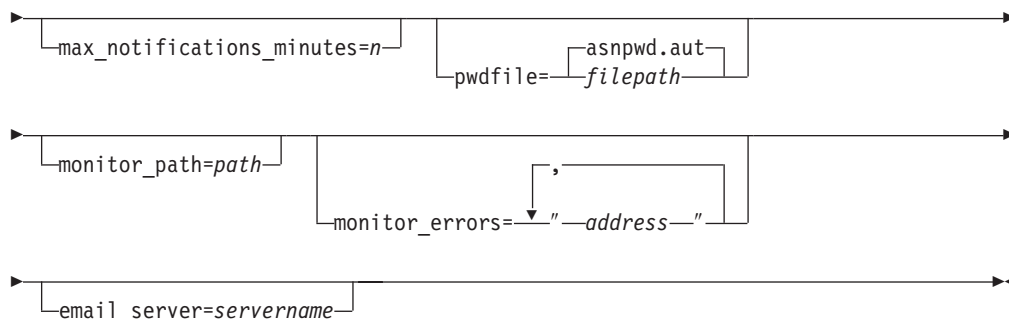
Linux、UNIX、Windows、および z/OS 上の UNIX System Services (USS) でレプリケーション・アラート・モニターを操作するには、**asnmon** コマンドを使用します。このコマンドは、オペレーティング・システムのプロンプトまたはシェル・スクリプト内で実行します。

レプリケーション・アラート・モニターは次の情報を記録します。

- Q キャプチャー・プログラムと Q アプライ・プログラム、およびキャプチャー・プログラムとアプライ・プログラムの状況
- コントロール表に書き込まれたエラー・メッセージ
- しきい値

構文





パラメーター

表 36 は、**asnmon** コマンドの呼び出しパラメーターを定義します。

表 36. Linux、UNIX、Windows、および z/OS オペレーティング・システム用 *asnmon* 呼び出しパラメーター定義

パラメーター	定義
monitor_server=server	レプリケーション・アラート・モニター・プログラムを実行し、モニター・コントロール表が存在する、モニター・コントロール・サーバーの名前を指定します。これを入力する場合は、最初のパラメーターでなければなりません。 Linux、UNIX、Windows: モニター・コントロール・サーバーを指定しない場合、このパラメーターはデフォルトで DB2DBDFT 環境変数の値になります。 z/OS: デフォルトは DSN です。
monitor_qual=mon_qual	レプリケーション・アラート・モニター・プログラムが使用するモニター修飾子を指定します。モニター修飾子は、モニター対象のサーバーおよび関連するモニター条件を識別します。 モニター修飾子は指定しなければなりません。モニター修飾子名には大文字小文字の区別があり、最大 18 文字です。
monitor_interval=n	このモニター修飾子について、どのような頻度 (秒数) で、レプリケーション・アラート・モニター・プログラムを実行するかを指定します。デフォルトは 300 秒 (5 分) です。 runonce パラメーターを <i>y</i> に設定すると、レプリケーション・アラート・モニターはこのパラメーターを無視します。 重要: この monitor_interval パラメーターは、レプリケーション・アラート・モニター・プログラムにのみ影響を与えます。このパラメーターは、Q キャプチャー、Q アプライ、キャプチャー、およびアプライ・プログラムには影響を与えません。

表 36. Linux、UNIX、Windows、および z/OS オペレーティング・システム用 *asnmon* 呼び出しパラメーター定義 (続き)

パラメーター	定義
runonce=y/n	<p>このモニター修飾子について、レプリケーション・アラート・モニター・プログラムを 1 回だけ実行するかどうかを指定します。</p> <p>n (デフォルト) レプリケーション・アラート・モニター・プログラムは、monitor_interval パラメーターに指定された頻度で実行されます。</p> <p>y レプリケーション・アラート・モニター・プログラムはモニター・サイクルを 1 回だけ実行します。</p> <p>runonce パラメーターを y に設定すると、レプリケーション・アラート・モニターは monitor_interval パラメーターを無視します。</p>
autoprun=y/n	<p>レプリケーション・アラート・モニターのアラート表 (IBMSNAP_ALERTS) 内の行のオート・プルーンを使用可能にするかどうかを指定します。</p> <p>y (デフォルト) レプリケーション・アラート・モニター・プログラムは、alert_prune_limit パラメーターの値よりも古い、IBMSNAP_ALERTS 表内の行を自動的に整理します。</p> <p>n オート・プルーンは使用不可になります。</p>
logreuse=y/n	<p>レプリケーション・アラート・モニター・プログラムが、診断ログ・ファイル (<i>db2instance.monitor_server.mon_qual.MON.log</i>) を再利用するか、またはメッセージを付加するかを指定します。</p> <p>n (デフォルト) レプリケーション・アラート・モニター・プログラムはログ・ファイルにメッセージを付加します。</p> <p>y レプリケーション・アラート・モニター・プログラムは、ログ・ファイルを削除し、レプリケーション・アラート・モニター・プログラムの再始動時にそれを再作成することにより、ログ・ファイルを再利用します。</p>
logstdout=y/n	<p>レプリケーション・アラート・モニター・プログラムがメッセージをどこに送信するかを指定します。</p> <p>n (デフォルト) レプリケーション・アラート・モニター・プログラムはログ・ファイルにのみメッセージを送信します。</p> <p>y レプリケーション・アラート・モニター・プログラムは、メッセージをログ・ファイルと標準出力 (stdout) の両方に送信します。</p>

表 36. Linux、UNIX、Windows、および z/OS オペレーティング・システム用 *asnmon* 呼び出しパラメーター定義 (続き)

パラメーター	定義
alert_prune_limit = <i>n</i>	レプリケーション・アラート・モニターのアラート表 (IBMSNAP_ALERTS) に、行を保存しておく期間 (分) を指定します。この値よりも古い行はすべて整理されます。デフォルトは 10,080 分 (7 日) です。
trace_limit = <i>n</i>	レプリケーション・アラート・モニターのトレース表 (IBMSNAP_MONTRACE) 表内に何分留まれば、その行が整理の対象として適格になるかを示す分数を指定します。 trace_limit パラメーターの値よりも古い、すべての IBMSNAP_MONTRACE 行が、次の整理サイクルで削除されます。デフォルトは 10,080 分 (7 日) です。
max_notifications_per_alert = <i>n</i>	max_notifications_minutes パラメーター値で指定された期間中にアラートが起こった場合、同じアラートをユーザーに送信する最大回数を指定します。このパラメーターは、同じアラートをユーザーに何回も再送するのを避けるために使用します。デフォルトは 3 です。
max_notifications_minutes = <i>n</i>	このパラメーターは max_notifications_per_alert パラメーターと一緒に働き、アラート条件が起こる期間を示します。デフォルトは 60 分です。
pwdfile = <i>filepath</i>	パスワード・ファイルの完全修飾名を指定します。このファイルは asnpwd コマンドを使用して定義します。デフォルトのファイル名は、 asnpwd.aut です。
monitor_path = <i>path</i>	レプリケーション・アラート・モニター・プログラムが使用するログ・ファイルのロケーションを指定します。デフォルトは、 asnmon コマンドが呼び出されたディレクトリーです。
monitor_errors = <i>address</i>	アラート・モニターがモニター・コントロール・サーバーに接続する前に致命的エラーが検出された場合、その通知を送信する先の E メール・アドレスを指定します。このパラメーターを使用して、無効な開始パラメーター、誤りのモニター修飾子、ダウンしているデータベース、またはその他のエラーにより、モニター・コントロール・サーバー接続が失敗したという通知を送信します。 E メール・アドレスのテキストは二重引用符で囲みます。 複数の E メール・アドレスを入力することができます。 E メール・アドレスをコンマで区切ってください。コンマの前後にはスペースを入力できます。
email_server = <i>servername</i>	Eメールのサーバー・アドレスを指定します。このパラメーターは、SMTP (Simple Mail Transfer Protocol) を指定して ASNMAIL 出口ルーチンを使用する場合にのみ入力します。

戻りコード

asnmon コマンドは、正常終了したときにゼロの戻りコードを戻します。コマンドが失敗する場合、ゼロ以外の戻りコードが戻されます。

asnmon の例

次の例は、**asnmon** コマンドの使用法を示しています。

例 1

デフォルトのパラメーターを使用して、レプリケーション・アラート・モニターを開始します。

```
asnmon monitor_server=wsdb monitor_qual=monqual
```

例 2

指定したモニター修飾子について、120 秒 (2 分) おきに実行するレプリケーション・アラート・モニターを開始します。

```
asnmon monitor_server=wsdb monitor_qual=monqual monitor_interval=120
```

例 3

レプリケーション・アラート・モニターを開始し、指定したモニター修飾子について、1 回だけ実行することを指定します。

```
asnmon monitor_server=wsdb monitor_qual=monqual runonce=y
```

例 4

レプリケーション・アラート・モニターを開始し、モニター・エラーが検出された場合は E メールで通知を送信します。

```
asnmon monitor_server=wsdb monitor_qual=monqual  
monitor_errors="repladm@company.com, dbadmin@company.com"
```

例 5

120 秒 (2 分) おきに実行するレプリケーション・アラート・モニターを開始し、1440 分 (24 時間) 待ってからアラートを送信します。

```
asnmon monitor_server=wsdb monitor_qual=monqual monitor_interval=120  
max_notifications_per_alert=2 max_notifications_minutes=1440
```

max_notifications_minutes パラメーター値 (1440 分) で指定された期間中にアラートが起こった場合、このレプリケーション・アラート・モニター・プログラムは最大 2 つのアラートを送信します。

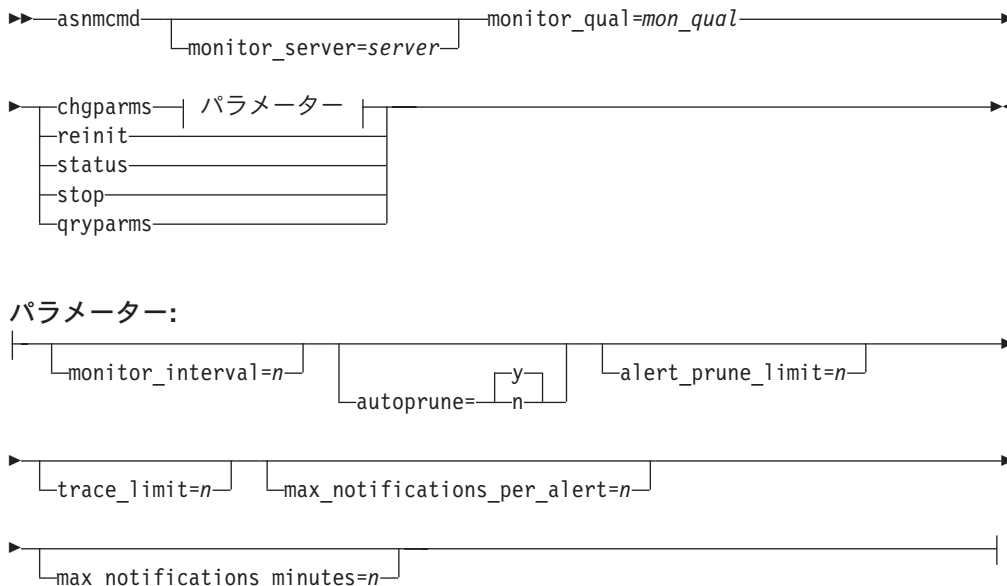
関連資料:

- 384 ページの『asnmcmd: 実行中のレプリケーション・アラート・モニターの操作』

asnmcmd: 実行中のレプリケーション・アラート・モニターの操作

Linux、UNIX、Windows、および z/OS 上の UNIX System Services (USS) で実行中のレプリケーション・アラート・モニターにコマンドを送信するには、**asnmcmd** を使用します。このコマンドは、オペレーティング・システムのプロンプトまたはシェル・スクリプト内で実行します。

構文



パラメーター

表 37 は、**asnmcmd** コマンドの呼び出しパラメーターを定義します。

表 37. Linux、UNIX、Windows、および z/OS オペレーティング・システム用 *asnmcmd* 呼び出しパラメーター定義

パラメーター	定義
monitor_server=server	レプリケーション・アラート・モニター・プログラムを実行し、モニター・コントロール表が存在する、モニター・コントロール・サーバーの名前を指定します。これを入力する場合は、最初のパラメーターでなければなりません。 Linux、UNIX、Windows: モニター・コントロール・サーバーを指定しない場合、このパラメーターはデフォルトで DB2DBDFT 環境変数の値になります。 z/OS: デフォルトは DSN です。
monitor_qual=mon_qual	レプリケーション・アラート・モニター・プログラムが使用するモニター修飾子を指定します。モニター修飾子は、モニター対象のサーバーおよび関連するモニター条件を識別します。 モニター修飾子は指定しなければなりません。モニター修飾子名には大文字小文字の区別があり、最大 18 文字です。

表 37. Linux、UNIX、Windows、および z/OS オペレーティング・システム用 asnmcmd 呼び出しパラメーター定義 (続き)

パラメーター	定義
chgparms	<p>レプリケーション・アラート・モニターの実行中に、以下の稼働パラメーターのうち 1 つ以上を変更することを指定します。</p> <ul style="list-style-type: none"> • monitor_interval • autoprune • alert_prune_limit • trace_limit • max_notifications_per_alert • max_notifications_minutes <p>1 つの chgparms サブコマンド中に複数のパラメーターを指定したり、必要に応じてこれらのパラメーター値を変更したりできます。変更内容は IBMSNAP_MONPARMS 表の値を一時的にオーバーライドしますが、この表に保管はされません。レプリケーション・アラート・モニターを停止して再始動する際には、IBMSNAP_MONPARMS 中の値が使用されます。 asnmon: レプリケーション・アラート・モニターの始動には、このサブコマンドを使用してオーバーライドできるパラメーターの説明が記載されています。</p> <p>重要: 変更するパラメーターは、chgparms サブコマンドの直後に指定する必要があります。</p>
reinit	<p>レプリケーション・アラート・モニター・プログラムがそのコントロール表を読み取り、連絡先、アラート条件、およびパラメーター用にそのメモリー内に持つデータをリフレッシュすることを指定します。すべての値が読み取られると、モニター・プログラムはサーバー上で条件のチェック・サイクルを開始します。このサイクルが完了したら、monitor_interval で指定した時間が経過した後、次のモニター・サイクルを開始します。</p>
status	<p>レプリケーション・アラート・モニター内の各スレッド (管理、シリアライゼーション、およびワーカー) の状態を示すメッセージを受け取ることを指定します。</p>
qryparms	<p>レプリケーション・アラート・モニターの現行の稼働パラメーター値を標準出力 (stdout) に書き込む場合に指定します。</p>
stop	<p>レプリケーション・アラート・モニターを通常の方法で停止することを指定します。</p>

asnmcmd の例

次の例は、**asnmcmd** コマンドの使用法を示します。

例 1

指定したモニター修飾子について、レプリケーション・アラート・モニターを停止します。


```
asnmcmd monitor_server=wsdb monitor_qual=monqual stop
```

例 2

レプリケーション・アラート・モニターのスレッドの状態を示すメッセージを受け取ります。

```
asnmcmd monitor_server=wsdb monitor_qual=monqual status
```

例 3

モニター・コントロール表からの最新の値を使用して、レプリケーション・アラート・モニターをリフレッシュします。

```
asnmcmd monitor_server=wsdb monitor_qual=monqual reinit
```

例 4

指定期間中にレプリケーション・アラート・モニターが送信する通知の最大数を、デフォルトの 3 から少なくします。

```
asnmcmd monitor_server=wsdb monitor_qual=monqual  
chgparms max_notifications_per_alert=2
```

例 5

レプリケーション・アラート・モニターの現行稼働パラメーター値を標準出力に送信します。

```
asnmcmd monitor_server=wsdb monitor_qual=monqual qrypams
```

関連資料:

- 379 ページの『asmon: レプリケーション・アラート・モニターの始動』

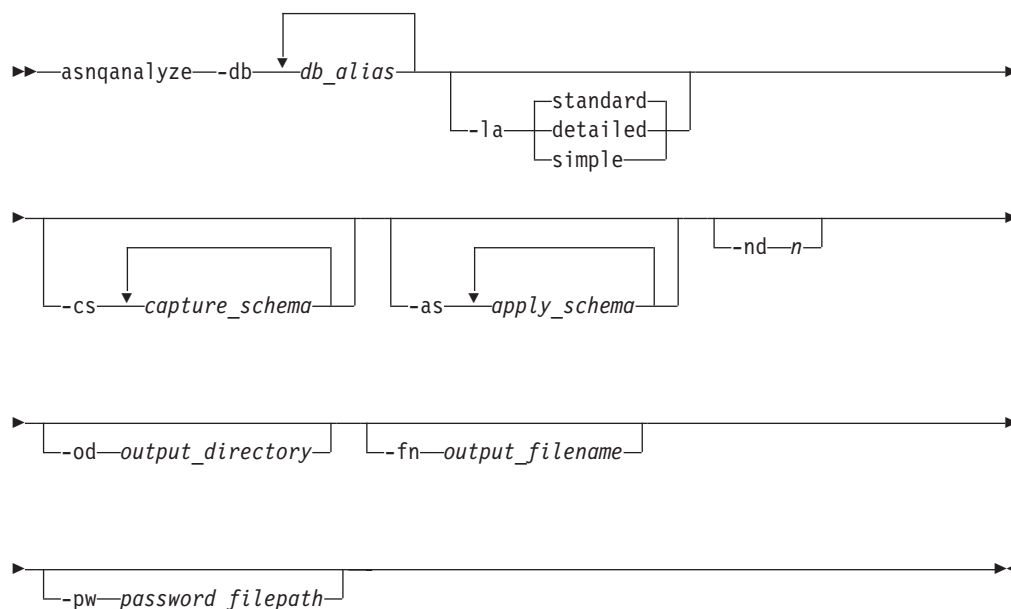
asnqanalyze: Q レプリケーション・アナライザーの操作

Q レプリケーションおよびイベント発行環境の状態についてのレポートを生成するには、**asnqanalyze** コマンドを使用します。このコマンドは、Q レプリケーション・アナライザーを呼び出します。これは、Q キャプチャーまたは Q アプライのコントロール表の状態についての定形の HTML レポートを生成します。

アナライザーは、Linux、UNIX、または Windows オペレーティング・システム上でのみ実行できます。しかしながら、z/OS においても DB2 UDB サブシステムに接続して、コントロール表を分析することは可能です。**asnqanalyze** コマンドは、オペレーティング・システム・プロンプトで実行します。

アナライザーの使用方法の詳細については、www.ibm.com/software/data/dpropr/library.html を参照してください。

構文



パラメーター

表 38 は、**asnqanalyze** コマンドの呼び出しパラメーターを定義します。

表 38. *Linux*、*UNIX*、および *Windows* オペレーティング・システム用 *asnqanalyze* 呼び出しパラメーター定義

パラメーター	定義
-db <i>db_alias</i>	Q キャプチャー・サーバーまたは Q アプライ・サーバーを指定します。 データベース別名を少なくとも 1 つ指定する必要があります。複数のデータベース別名がある場合は、ブランク・スペースを使用して値を区切ります。

表 38. Linux、UNIX、および Windows オペレーティング・システム用 *asnqanalyze* 呼び出しパラメーター定義 (続き)

パラメーター	定義
-la <i>level_of_analysis</i>	<p>報告される分析のレベルを指定します。</p> <p>standard (デフォルト) コントロール表の内容と、Q キャプチャー・プログラムおよび Q アプライ・プログラムからの状況情報を含むレポートを生成します。</p> <p>detailed 標準レポートの情報に加えて以下の情報を生成します。</p> <ul style="list-style-type: none"> • Q キャプチャーのチューニング・パラメーター • Q キャプチャー・プログラムがキャプチャーを行わない理由 • Q アプライのチューニング・パラメーター • Q アプライ・プログラムがアプライを行わない理由 • 不正なまたは非効率な表スペースのロック・サイズ値 • DB2 UDB for z/OS の SYSTABLEPART システム・カタログ表から選択した情報 • ソースおよびターゲット表に対する参照保全制約に関する情報 <p>simple 標準レポートで情報を生成しますが、IBMQREP_SRC_COLS 表および IBMQREP_TRG_COLS 表からの詳細情報は含まれません。</p>
-cs <i>capture_schema</i>	<p>分析する Q キャプチャー・スキーマの名前を指定します。複数の Q キャプチャー・スキーマを指定する場合、ブランク・スペースを使用して値を区切ります。 -cs キーワードを指定しない場合、指定されたデータベース別名に関するすべての Q キャプチャー・スキーマが分析されます。</p>
-as <i>apply_schema</i>	<p>分析する Q アプライ・スキーマの名前を指定します。複数の Q アプライ・スキーマを指定する場合、ブランク・スペースを使用して値を区切ります。 -as キーワードを指定しない場合、指定されたデータベース別名に関するすべての Q アプライ・スキーマが分析されます。</p>
-nd <i>n</i>	<p>以下のコントロール表から検索する情報のデータ範囲 (0 から 30 日) を指定します。</p> <ul style="list-style-type: none"> • IBMQREP_CAPTRACE • IBMQREP_CAPMON • IBMQREP_CAPQMON • IBMQREP_SIGNAL • IBMQREP_APPLYTRACE • IBMQREP_APPLYMON <p>デフォルトは 3 日です。</p>

表 38. Linux、UNIX、および Windows オペレーティング・システム用 *asnqanalyze* 呼び出しパラメーター定義 (続き)

パラメーター	定義
-od <i>output_directory</i>	アナライザー・レポートを保管するディレクトリーを指定します。デフォルトは、現行ディレクトリーです。
-fn <i>output_filename</i>	アナライザー・レポート出力を含むファイルの名前を指定します。 アナライザーを実行するオペレーティング・システムのファイル命名規則を使用します。ファイル名がすでに存在する場合、ファイルは上書きされます。デフォルトのファイル名は、 <i>asnqanalyze.htm</i> です。
-pw <i>password_filepath</i>	パスワード・ファイルの名前とパスを指定します。このパラメーターを指定しない場合、アナライザーは現行ディレクトリーで <i>asnpwd.aut</i> ファイルを探します。

asnqanalyze の例

次の例は、**asnqanalyze** コマンドの使用方法を示しています。

例 1

proddb1 という名前のデータベース上のコントロール表を分析する例です。

```
asnqanalyze -db proddb1
```

例 2

proddb1 と *proddb2* のデータベース上のコントロール表について、詳細レベルの分析を入手する例です。

```
asnqanalyze -db proddb1 proddb2 -la detailed
```

例 3

proddb1 と *proddb2* のデータベース上のコントロール表からの最後の 2 日分の情報を分析する例です。

```
asnqanalyze -db proddb1 proddb2 -nd 2
```

例 4

proddb1 と *proddb2* のデータベース上のコントロール表の単純レベルの分析を入手し、*c:\\$QLLIB* ディレクトリーに保管されているパスワード・ファイルを使用して、*c:\mydir* ディレクトリーの下に *anzout* という名前のファイルにアナライザー出力を書き込む例です。

```
asnqanalyze -db proddb1 proddb2 -la simple -od c:\mydir -fn anzout -pw c:\$QLLIB
```

例 5

asnmcmd

Q キャプチャー・サーバー qcapsvr1 上の特定の Q キャプチャー・スキーマを分析する例です。

```
asnqanalyze -db qcapsvr1 -cs ELB
```

例 6

Q アプライ・サーバー qappsvr1 上の特定の Q アプライ・スキーマを分析する例です。

```
asnqanalyze -db qappsvr1 -as CLS
```

例 7

特定の Q キャプチャーおよび Q アプライ・スキーマを分析する例です。

```
asnqanalyze -db qcapsvr1 qappsvr1 -cs qcapp1 -as qapp1
```

例 8

コマンド・ヘルプを表示するには、以下のように入力します。

```
asnqanalyze
```

関連概念:

- 359 ページの『Q レプリケーションおよびイベント発行用のシステム・コマンド -- 概要』

関連タスク:

- 43 ページの『リモート・サーバー用のユーザー ID およびパスワードの保管』

関連資料:

- 359 ページの『ロードマップ: Q レプリケーションおよびイベント発行のシステム・コマンド』

asnscri: レプリケーション・プログラムを開始する DB2 レプリケーション・サービスの作成

Windows の Service Control Manager (SCM) に DB2 レプリケーション・サービスを作成するには、**asnscri** コマンドを使用し、**asnqcap**、**asnqapp**、**asnmon**、**asnscap**、および **asnapply** コマンドを呼び出します。**asnscri** コマンドは Windows オペレーティング・システムで実行します。

構文

```
asnscri [-QC] [-QA] [-M] [-C] [-A] db2_instance account password [asnqcap_command] [asnqapp_command] [asnmon_command] [asnscap_command] [asnapply_command]
```

パラメーター

表 39 は、**asnscri** コマンドの呼び出しパラメーターを定義します。

表 39. Windows オペレーティング・システム用の *asnscri* 呼び出しパラメーター定義

パラメーター	定義
-QC	Q キャプチャー・プログラムの始動を指定します。
-QA	Q アプライ・プログラムの始動を指定します。
-M	レプリケーション・アラート・モニター・プログラムの始動を指定します。
-C	キャプチャー・プログラムの始動を指定します。
-A	アプライ・プログラムの始動を指定します。
<i>db2_instance</i>	ユニークな DB2 レプリケーション・サービスを識別するために使用される DB2 インスタンスを指定します。DB2 インスタンス名は最大 8 文字です。
<i>account</i>	Windows へのログオンに使用するアカウント名を指定します。アカウント名はピリオドと円記号 (¥) から始まる必要があります。
<i>password</i>	このアカウント名で使用するパスワードを指定します。パスワードに特殊文字が含まれる場合は、各特殊文字の前に円記号 (¥) を入力します。
<i>asnqcap_command</i>	<p>Q キャプチャー・プログラムを開始するための完全な asnqcap コマンドを指定します。文書化されている asnqcap コマンド構文と該当の asnqcap パラメーターを使用します。</p> <p>重要: DB2PATH 環境変数が定義されていない場合、asnqcap コマンドに capture_path パラメーターを含めて、作業ファイルのロケーションを指定する必要があります。DB2PATH 変数が定義されていて、capture_path を指定すると、capture_path パラメーターは DB2PATH 変数をオーバーライドします。</p> <p>asnscri コマンドは、ユーザーが入力した asnqcap パラメーターの構文の妥当性を検査しません。</p>
<i>asnqapp_command</i>	<p>Q アプライ・プログラムを開始するための完全な asnqapp コマンドを指定します。文書化されている asnqapp コマンド構文と該当の asnqapp パラメーターを使用します。</p> <p>重要: DB2PATH 環境変数が定義されていない場合、asnqapp コマンドに apply_path パラメーターを含めて、作業ファイルのロケーションを指定する必要があります。DB2PATH 変数が定義されていて、apply_path を指定すると、apply_path パラメーターは DB2PATH 変数をオーバーライドします。</p> <p>asnscri コマンドは、ユーザーが入力した asnqapp パラメーターの構文の妥当性を検査しません。</p>

表 39. Windows オペレーティング・システム用の *asnscri* 呼び出しパラメーター定義 (続き)

パラメーター	定義
<i>asnmon_command</i>	<p>レプリケーション・アラート・モニター・プログラムを開始するための完全な asnmon コマンドを指定します。文書化されている asnmon コマンド構文と該当の asnmon パラメーターを使用します。</p> <p>重要: DB2PATH 環境変数が定義されていない場合、asnmon コマンドに monitor_path パラメーターを含めて、ログ・ファイルのロケーションを指定する必要があります。DB2PATH 変数が定義されていて、monitor_path を指定すると、monitor_path パラメーターは DB2PATH 変数をオーバーライドします。</p> <p>asnscri コマンドは、ユーザーが入力した asnmon パラメーターの構文の妥当性を検査しません。</p>
<i>asnscap_command</i>	<p>キャプチャー・プログラムを開始するための完全な asnscap コマンドを指定します。文書化されている asnscap コマンド構文および該当の asnscap パラメーターを使用します。</p> <p>重要: DB2PATH 環境変数が定義されていない場合、asnscap コマンドに capture_path パラメーターを含めて、作業ファイルのロケーションを指定する必要があります。DB2PATH 変数が定義されていて、capture_path を指定すると、capture_path パラメーターは DB2PATH 変数をオーバーライドします。</p> <p>asnscri コマンドは、ユーザーが入力した asnscap パラメーターの構文の妥当性を検査しません。</p>
<i>asnapply_command</i>	<p>アプライ・プログラムを開始するための完全な asnapply コマンドを指定します。文書化されている asnapply コマンド構文と該当の asnapply パラメーターを使用します。</p> <p>重要: DB2PATH 環境変数が定義されていない場合、asnapply コマンドに apply_path パラメーターを含めて、作業ファイルのロケーションを指定する必要があります。DB2PATH 変数が定義されていて、apply_path を指定すると、apply_path パラメーターは DB2PATH 変数をオーバーライドします。</p> <p>asnscri コマンドは、ユーザーが入力した asnapply パラメーターの構文の妥当性を検査しません。</p>

asnscri の例

次の例は、**asnscri** コマンドの使用法を示しています。

例 1

inst1 という名前の DB2 インスタンスの下で Q キャプチャー・プログラムを呼び出す DB2 レプリケーション・サービスを作成するには、次のように指定します。


```
asnscri -QC inst1 .%joesmith password asnqcap capture_server=mydb1
capture_schema=QC1 capture_path=X:%logfiles
```

例 2

inst2 という名前の DB2 インスタンスの下で、ログオン・アカウント .%joesmith およびパスワード my\$pwd を使用して、Q アプライ・プログラムを呼び出す DB2 レプリケーション・サービスを作成するには、次のように指定します。

```
asnscri -QA inst2 .%joesmith my%$pwd asnqapp apply_server=mydb2 apply_schema =as2
apply_path=X:%sqllib
```

例 3

inst1 という名前の DB2 インスタンスの下でキャプチャー・プログラムを呼び出す DB2 レプリケーション・サービスを作成するには、次のように指定します。

```
asnscri -C inst1 .%joesmith password asncap capture_server=sampled
capture_schema=ASN capture_path=X:%logfiles
```

例 4

inst2 という名前の DB2 インスタンスの下で、ログオン・アカウント .%joesmith およびパスワード my\$pwd を使用して、アプライ・プログラムを呼び出す DB2 レプリケーション・サービスを作成するには、次のように指定します。

```
asnscri -A inst2 .%joesmith my%$pwd asnapply control_server=db2 apply_qual=aq2
apply_path=X:%sqllib
```

例 5

inst3 という名前の DB2 インスタンスの下でレプリケーション・アラート・モニター・プログラムを呼び出す DB2 レプリケーション・サービスを作成するには、次のように指定します。

```
asnscri -M inst3 .%joesmith password asnmon monitor_server=db3 monitor_qual=mq3
monitor_path=X:%logfiles
```

例 6

inst4 という名前の DB2 インスタンスの下でキャプチャー・プログラムを呼び出す DB2 レプリケーション・サービスを作成し、デフォルトの作業ファイル・ディレクトリを完全修飾された **capture_path** でオーバーライドするには、次のように指定します。

```
asnscri -C inst4 .%joesmith password X:%sqllib%bin%asncap capture_server=scdb
capture_schema=ASN capture_path=X:%logfiles
```

asnscri: DB2 レプリケーション・サービスのリスト

Windows の Service Control Manager (SCM) 中の DB2 レプリケーション・サービスをリストするには、**asnscri** コマンドを使用します。オプションで、このコマンドを使用して、個々のサービスに関する詳細情報をリストできます。**asnscri** コマンドは Windows オペレーティング・システムで実行します。

構文

```
▶▶ asnscri [DETAILS] ▶▶
```

パラメーター

表 40 は、**asnscri** コマンドの呼び出しパラメーターを定義します。

表 40. Windows オペレーティング・システム用の *asnscri* 呼び出しパラメーター定義

パラメーター	定義
details	システム上のすべての DB2 レプリケーション・サービスに関する詳細データをリストしたいことを指定します。

asnscri の例

次の例は、**asnscri** コマンドの使用方法を示しています。

例 1

システム上の DB2 レプリケーション・サービスの名前をリストするには、以下のようになります。

```
asnscri
```

以下にコマンド出力の例を示します。

```
DB2.DB2.SAMPLE.QAPP.ASN
DB2.DB4.SAMPLE.QCAP.ASN
```

例 2

システム上のすべてのサービスに関する詳細情報をリストするには、以下のようになります。

```
asnscri details
```

以下にコマンド出力の例を示します。

```
DB2.DB2.SAMPLE.QAPP.ASN
Display Name: DB2 DB2 SAMPLE QAPPLY ASN
Image Path:  ASNSERV DB2.DB2.SAMPLE.APP.AQ1 -ASNQAPPLY QAPPLY_SERVER=SAMPLE AP
             PLY_SCHEMA=ASN QAPPLY_PATH=C:\PROGRA~1\SQLLIB
Dependency:  DB2-0

DB2.DB4.SAMPLE.QCAP.ASN
```

```

Display Name: DB2 DB4 SAMPLE QAPPLY ASN
Image Path:  ASNSERV DB2.DB4.SAMPLE.APP.AQ1 -ASNQCAP QCAPTURE_SERVER=SAMPLE CA
             PTURE_SCHEMA=ASN QCAPTURE_PATH=C:\PROGRA~1\SQLLIB
Dependency:   DB4-0

```

asnsdrop: DB2 レプリケーション・サービスのドロップ

Windows オペレーティング・システムの Windows Service Control Manager (SCM) から、DB2 レプリケーション・サービスをドロップするには、**asnsdrop** コマンドを使用します。(DB2 レプリケーション・サービスの作成には **asnscri** コマンドを使用します。)

構文

```

>> asnsdrop service_name
         ALL

```

パラメーター

表 41 は、**asnsdrop** コマンドの呼び出しパラメーターを定義します。

表 41. Windows オペレーティング・システム用の **asnsdrop** 呼び出しパラメーター定義

パラメーター	定義
<i>service_name</i>	DB2 レプリケーション・サービスの完全修飾名を指定します。DB2 レプリケーション・サービス名を知るには、Windows SCM に入ります。Windows オペレーティング・システムでは、DB2 レプリケーション・サービスの「プロパティ」ウィンドウをオープンするとサービス名がわかります。 DB2 レプリケーション・サービス名にスペースが含まれる場合は、サービス名全体を二重引用符で囲ってください。
ALL	すべての DB2 レプリケーション・サービスをドロップしたいことを指定します。

asnsdrop の例

次の例は、**asnsdrop** コマンドの使用法を示しています。

例 1

DB2 レプリケーション・サービスをドロップする例です。

```
asnsdrop DB2.SAMPLEDB.SAMPLEDB.CAP.ASN
```

例 2

A S N というスキーマ名の DB2 レプリケーション・サービスをドロップする例です。

```
asnsdrop "DB2.SAMPLEDB.SAMPLEDB.CAP.A S N"
```

例 3

asnsdrop

すべての DB2 レプリケーション・サービスをドロップする例です。

```
asnsdrop ALL
```

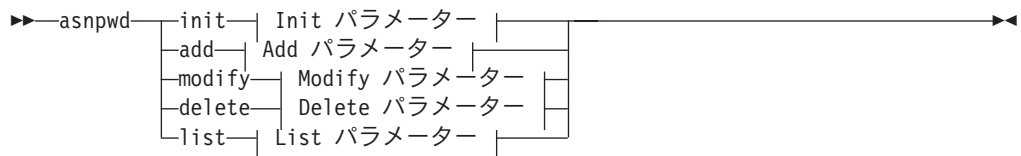
asnpwd: パスワード・ファイルの作成および保守

Linux、UNIX、および Windows のパスワード・ファイルの作成および変更は、**asnpwd** コマンドを使用して行います。このコマンドは、コマンド行またはシェル・スクリプト内で実行します。

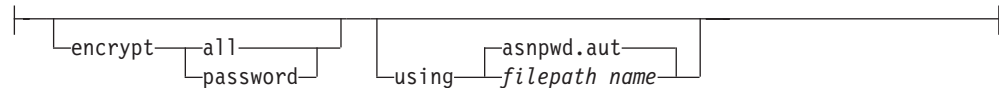
このコマンドのパラメーター・キーワードは、大文字小文字の区別をしません。

パラメーターを指定せずに **asnpwd** コマンドの後に ? または誤りのパラメーターを指定すると、コマンド・ヘルプが表示されます。

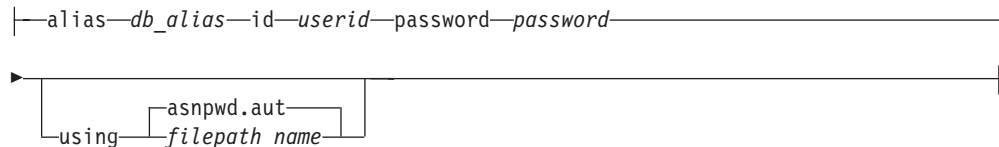
構文



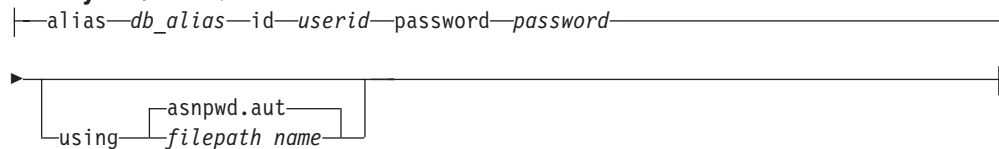
Init パラメーター:



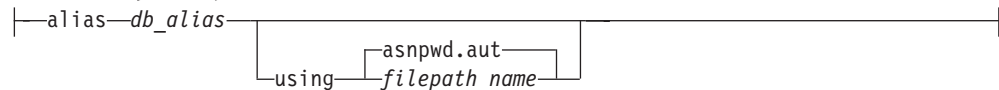
Add パラメーター:



Modify パラメーター:



Delete パラメーター:



List パラメーター:



パラメーター

表 42 は、**asnpwd** コマンドの呼び出しパラメーターを定義します。

表 42. Linux、UNIX、および Windows オペレーティング・システム用 **asnpwd** 呼び出しパラメーター定義

パラメーター	定義
init	空のパスワード・ファイルを作成することを指定します。すでに存在するパスワード・ファイルに init パラメーターを指定すると、このコマンドは失敗します。
add	パスワード・ファイルに項目を追加することを指定します。パスワード・ファイルにすでに存在する項目に add パラメーターを指定すると、このコマンドは失敗します。パスワード・ファイル内の既存の項目を変更するには、 modify パラメーターを使用します。
modify	パスワード・ファイル内の項目のパスワードまたはユーザー ID を変更することを指定します。
delete	パスワード・ファイルから項目を削除することを指定します。
list	パスワード・ファイル内の別名およびユーザー ID の項目をリストするよう指定します。このパラメーターは、パスワード・ファイルが encrypt パラメーターを使用して作成された場合のみ使用できます。パスワードを list コマンドによって表示することはできません。
encrypt	暗号化するファイル内の項目を指定します。 <p>all (デフォルト)</p> 指定したファイル内のすべての項目を暗号化し、ファイル内のデータベース別名、ユーザー名、およびパスワードをリストできないようにします。このオプションにより、パスワード・ファイル内の情報漏れを削減できます。
	<p>password</p> 指定したファイル内のパスワードの項目を暗号化します。このオプションでは、ユーザーは、パスワード・ファイル内に保管されているデータベース別名およびユーザー名をリストすることは許可されます。パスワードを表示することはできません。

表 42. Linux、UNIX、および Windows オペレーティング・システム用 asnpwd 呼び出しパラメーター定義 (続き)

パラメーター	定義
using <i>filepath_name</i>	<p>パスワード・ファイルのパスと名前を指定します。ご使用のオペレーティング・システムのファイル名規則に従ってください。Windows 上の有効なパスワード・ファイルの例は、C:\\$qllib¥mypwd.aut です。</p> <p>パスワード・ファイルのパスと名前を指定する場合、そのパスおよびパスワード・ファイルはすでに存在するものでなければなりません。 init パラメーターを使用し、パスワード・ファイルのパスと名前を指定する場合、そのパスはすでに存在するものでなければならず、コマンドはユーザーに代わってパスワード・ファイルを作成します。</p> <p>このパラメーターを指定しない場合、デフォルトのファイル名は asnpwd.aut、デフォルトのファイル・パスは現行ディレクトリーです。</p>
alias <i>db_alias</i>	<p>ユーザー ID がアクセス権を持つデータベースの別名を指定します。別名は、どのように入力しても必ず英大文字になります。</p>
id <i>userid</i>	<p>データベースへのアクセス権を持つユーザー ID を指定します。</p>
password <i>password</i>	<p>指定したユーザー ID のパスワードを指定します。このパスワードには大文字小文字の区別があり、パスワード・ファイル内では暗号化されます。</p>

戻りコード

asnpwd コマンドは、正常終了したときにゼロの戻りコードを戻します。コマンドが失敗する場合、ゼロ以外の戻りコードが戻されます。

asnpwd の例

次の例は、**asnpwd** コマンドの使用法を示しています。

例 1

現行ディレクトリーに、デフォルト名 asnpwd.aut を使用してパスワード・ファイルを作成します。

```
asnpwd INIT
```

例 2

c:\¥myfiles ディレクトリーに pass1.aut という名前のパスワード・ファイルを作成します。

```
asnpwd INIT USING c:\¥myfiles¥pass1.aut
```

例 3

encrypt all パラメーターを使用して mypwd.aut という名前のパスワード・ファイルを作成するには、次のようにします。

```
asnpwd INIT ENCRYPT ALL USING mypwd.aut
```

例 4

encrypt password パラメーターを使用して mypwd.aut という名前のパスワード・ファイルを作成するには、次のようにします。

```
asnpwd INIT ENCRYPT PASSWORD USING mypwd.aut
```

例 5

encrypt password パラメーターを使用してデフォルトのパスワード・ファイルを作成するには、次のようにします。

```
asnpwd INIT ENCRYPT PASSWORD
```

例 6

oneuser というユーザー ID とそのパスワードを、c:\myfiles ディレクトリー内の pass1.aut という名前のパスワード・ファイルに追加し、このユーザー ID に db1 データベースへのアクセス権を付与します。

```
asnpwd ADD ALIAS db1 ID oneuser PASSWORD mypwd using c:\myfiles\pass1.aut
```

例 7

c:\myfiles ディレクトリー内の pass1.aut という名前のパスワード・ファイル内の項目のユーザー ID またはパスワードを変更します。

```
asnpwd MODIFY Alias sample ID chglocalid PASSWORD chgmajorpwd
USING c:\myfiles\pass1.aut
```

例 8

c:\myfiles ディレクトリー内の pass1.aut という名前のパスワード・ファイルから、sample というデータベース別名を削除します。

```
asnpwd delete alias sample USING c:\myfiles\pass1.aut
```

例 9

コマンド・ヘルプを表示します。

```
asnpwd
```

例 10

デフォルトのパスワード・ファイル内の項目をリストします。

```
asnpwd LIST
```


例 11

pass1.aut という名前のパスワード・ファイル内の項目をリストします。

```
asnpwd LIST USING pass1.aut
```

このコマンドからの出力は、パスワード・ファイルの初期設定方法によって異なります。

- **encrypt all** パラメーターを使用して初期設定された場合、以下のメッセージが発行されます。

```
ASN1986E "Asnpwd" : "". The password file "pass1.aut" contains encrypted information that cannot be listed.
```

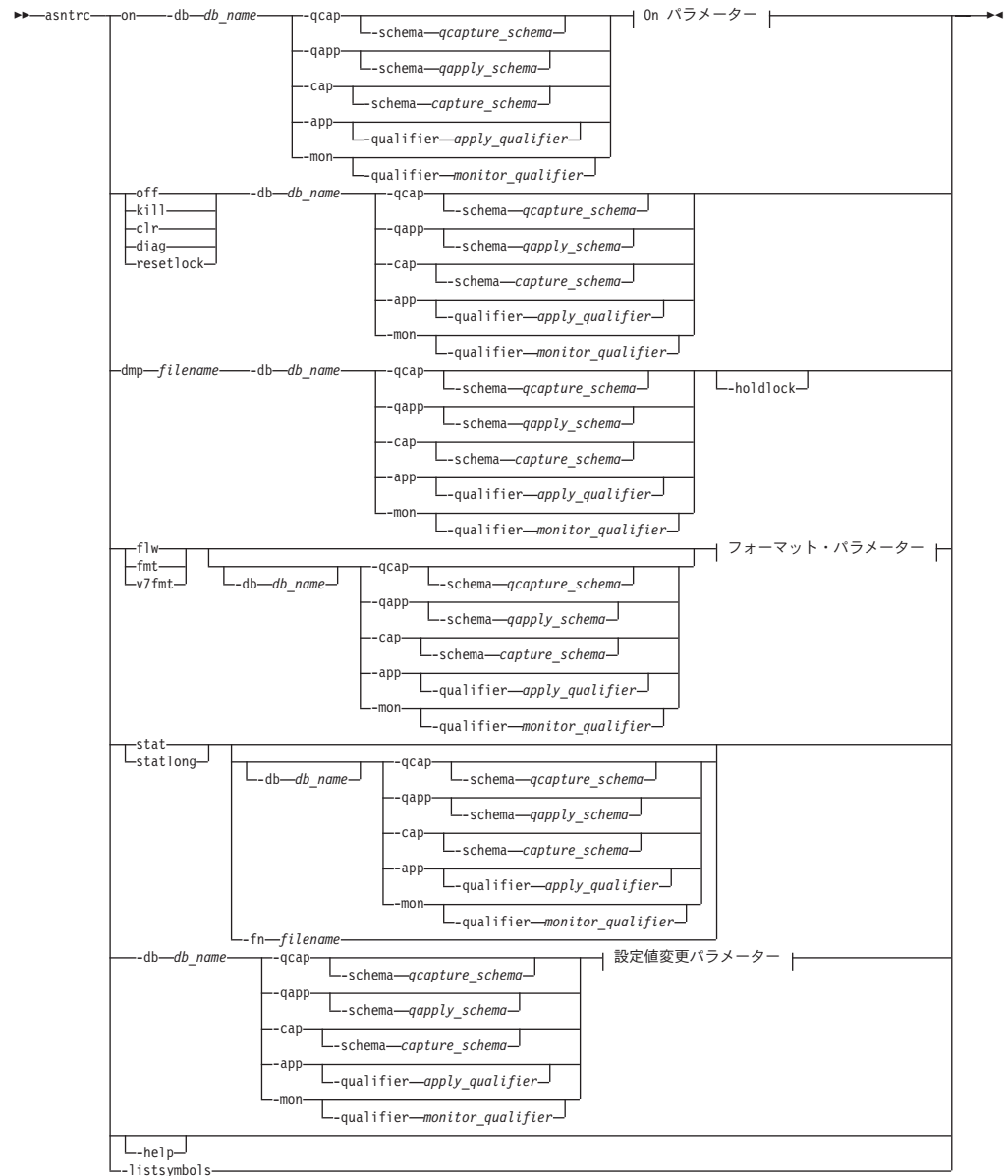
- **encrypt all** パラメーターを使用して初期設定されなかった場合、以下の詳細がリストされます。

```
asnpwd LIST USING pass1.aut  
Alias: SAMPLE ID: chglocalid  
Number of Entries: 1
```

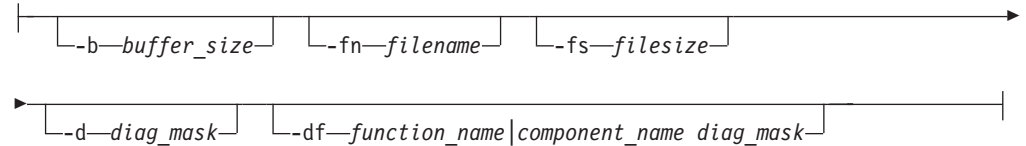
asntrc: レプリケーション・トレース機能の操作

Linux、UNIX、Windows、および z/OS 上の UNIX System Services (USS) でトレース機能を実行するには、**asntrc** コマンドを使用します。トレース機能は、Q キャプチャー・プログラム、Q アプライ・プログラム、キャプチャー・プログラム、アプライ・プログラムおよびレプリケーション・アラート・モニター・プログラムから、プログラム・フロー情報をログに記録します。このトレース情報を IBM ソフトウェア・サポートに提供して、トラブルシューティングに役立てることができます。このコマンドは、オペレーティング・システムのプロンプトまたはシェル・スクリプト内で実行します。

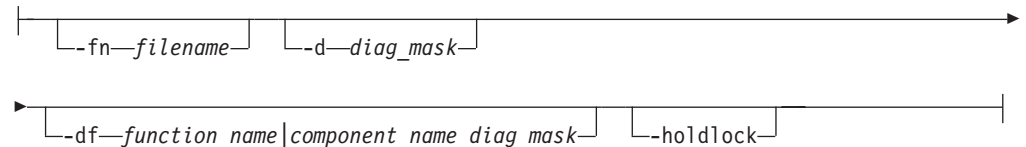
構文



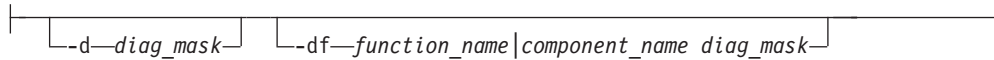
On パラメーター:



フォーマット・パラメーター:



設定値変更パラメーター:



パラメーター

表 43 は、**asntrc** コマンドの呼び出しパラメーターを定義します。

表 43. Linux、UNIX、Windows、および z/OS オペレーティング・システム用 *asntrc* 呼び出しパラメーター定義

パラメーター	定義
on	特定の Q キャプチャー・プログラム、Q アプライ・プログラム、キャプチャー・プログラム、アプライ・プログラム、またはレプリケーション・アラート・モニター・プログラムについて、トレース機能をオンにすることを指定します。トレース機能は、トレース処理中に使用する共有メモリー・セグメントを作成します。
-db <i>db_name</i>	トレースするデータベースの名前を指定します。 <ul style="list-style-type: none"> ・トレースする Q キャプチャー・プログラム用の Q キャプチャー・サーバーの名前を指定する。 ・トレースする Q アプライ・プログラム用の Q アプライ・サーバーの名前を指定する。 ・トレースするキャプチャー・プログラム用のキャプチャー・コントロール・サーバーの名前を指定する。 ・トレースするアプライ・プログラム用のアプライ・コントロール・サーバーの名前を指定する。 ・トレースするレプリケーション・アラート・モニター・プログラム用のモニター・コントロール・サーバーの名前を指定する。
-qcap	Q キャプチャー・プログラムをトレースすることを指定します。Q キャプチャー・プログラムは -schema パラメーターで識別されます。
-schema <i>qcapture_schema</i>	トレースする Q キャプチャー・プログラムの名前を指定します。Q キャプチャー・プログラムは指定した Q キャプチャー・スキーマにより識別されます。このパラメーターは -qcap パラメーターと一緒に使用します。
-qapp	Q アプライ・プログラムをトレースすることを指定します。Q アプライ・プログラムは -schema パラメーターで識別されます。
-schema <i>qapply_schema</i>	トレースする Q アプライ・プログラムの名前を指定します。Q アプライ・プログラムは指定した Q アプライ・スキーマにより識別されます。このパラメーターは -qapp パラメーターと一緒に使用します。
-cap	キャプチャー・プログラムをトレースすることを指定します。キャプチャー・プログラムは -schema パラメーターで識別されます。

表 43. Linux、UNIX、Windows、および z/OS オペレーティング・システム用 asntrc 呼び出しパラメーター定義 (続き)

パラメーター	定義
-schema <i>capture_schema</i>	トレースするキャプチャー・プログラムの名前を指定します。キャプチャー・プログラムは指定したキャプチャー・スキーマにより識別されます。このパラメーターは -cap パラメーターと一緒に使用します。
-app	アプライ・プログラムをトレースすることを指定します。アプライ・プログラムは -qualifier パラメーターで識別されます。
-qualifier <i>apply_qualifier</i>	トレースするアプライ・プログラムの名前を指定します。このアプライ・プログラムは、指定したアプライ修飾子により識別されます。このパラメーターは -app パラメーターと一緒に使用します。
-mon	レプリケーション・アラート・モニター・プログラムをトレースすることを指定します。レプリケーション・アラート・モニター・プログラムは -qualifier パラメーターで識別されます。
-qualifier <i>monitor_qualifier</i>	トレースするレプリケーション・アラート・モニター・プログラムの名前を指定します。このレプリケーション・アラート・モニター・プログラムは、指定したモニター修飾子により識別されます。このパラメーターは -mon パラメーターと一緒に使用します。
off	特定の Q キャプチャー・プログラム、Q アプライ・プログラム、キャプチャー・プログラム、アプライ・プログラム、またはレプリケーション・アラート・モニター・プログラムについて、トレース機能をオフにし、使用中の共有メモリー・セグメントを解放することを指定します。
kill	トレース機能を強制的に異常終了させることを指定します。 このパラメーターは、何らかの問題により、トレース機能を off パラメーターでオフにできない場合のみ使用してください。
clr	トレース・バッファーをクリアすることを指定します。このパラメーターは、トレース・バッファーの内容を消去しますが、バッファーはアクティブのままにします。
diag	トレース機能の実行中に、フィルター設定を表示することを指定します。
resetlock	トレース機能のバッファー・ラッチを解放することを指定します。このパラメーターは、エラー状態が起り、トレース・プログラムがバッファー・ラッチを保留したまま終了した場合に、バッファー・ラッチをエラー状態からリカバリーできるようにします。
dmp <i>filename</i>	トレース・バッファーの現在の内容をファイルに書き込むことを指定します。

表 43. Linux、UNIX、Windows、および z/OS オペレーティング・システム用 *asnlrc* 呼び出しパラメーター定義 (続き)

パラメーター	定義
-holdlock	トレース機能がバッファをコピーするためのメモリーが不足している場合でも、ロックを保留している間に、トレース機能がファイルのダンプまたはコマンドの出力を完了できることを指定します。
flw	トレース機能が作成し、共有メモリーまたはファイルに保管したサマリー情報を表示することを指定します。この情報には、プログラム・フローが含まれ、それぞれの処理およびスレッドごとに、関数と呼び出しのスタック構造がわかるように字下げして表示されます。
fnt	トレース機能が作成し、共有メモリーまたはファイルに保管した詳細情報を表示することを指定します。このパラメーターは、トレースしたデータ構造の内容全体を発生順に表示します。
v7fnt	トレース機能が作成し、共有メモリーまたはファイルに保管した情報を表示することを指定します。このトレース情報はバージョン 7 のフォーマットで表示されます。
stat	トレース機能の状況を表示することを指定します。この状況情報には、トレース・バージョン、アプリケーション・バージョン、項目数、バッファ・サイズ、使用中のバッファ量、状況コード、およびプログラム・タイム・スタンプが含まれます。
statlong	トレース機能の状況に z/OS バージョン・レベル情報を追加して表示することを指定します。この追加情報には、アプリケーション内の各モジュールのサービス・レベルが含まれ、長ストリングのテキストとして表示されます。
-fn filename	ミラーリングされたトレース情報を含むファイル名を指定します。ここには、トレース機能からのすべての出力が含まれます。
-help	有効なコマンド・パラメーターを記述と一緒に表示します。
-listsymbols	-df パラメーターで利用できる有効な関数およびコンポーネント ID を表示します。
-b buffer_size	トレース・バッファのサイズをバイト単位で指定します。数値の後に、キロバイトなら K を、メガバイトなら M を指定できます。これらの文字には大文字小文字の区別はありません。
-fs filesize	ミラーリングされたトレース情報ファイルのサイズ制限をバイト単位で指定します。

表 43. Linux、UNIX、Windows、および z/OS オペレーティング・システム用 asnlrc 呼び出しパラメーター定義 (続き)

パラメーター	定義
-d <i>diag_mask</i>	<p>トレース機能により記録されるトレース・レコードのタイプを指定します。トレース・レコードは、以下の診断マスク番号により分類されます。</p> <ol style="list-style-type: none"> 1 フロー・データ。関数の入力点と出口点が含まれます。 2 基本データ。トレース機能が検出したすべての主要なイベントが含まれます。 3 詳細データ。主要なイベントとその記述が含まれます。 4 パフォーマンス・データ。 <p>重要: 診断マスク番号の大きいものは、診断マスク番号の小さいものを包含していません。</p> <p>これらの番号を 1 つ以上入力し、必要なトレース・レコードだけを含む診断マスクを作成することができます。たとえば、-d 4 を指定すると、パフォーマンス・データだけが記録されます。フローとパフォーマンスのデータだけを記録するには -d 1,4 と指定し、すべてのトレース・レコードを記録するには -d 1,2,3,4 (デフォルト) と指定します。番号はコンマで区切ります。</p> <p>トレース機能がグローバル・トレース・レコードを記録しないようにするには、診断マスク番号 0 (ゼロ) を入力します。トレース機能に新しい診断マスク番号を指定する前に、診断レベルをリセットするには、-d 0 を入力します。</p>
-df <i>function_name component_name</i> <i>diag_mask</i>	<p>特定の関数またはコンポーネント ID をトレースすることを指定します。</p> <p>関数またはコンポーネント ID 名の後に診断マスク番号 (1、2、3、4) を入力します。1 つ以上の番号を入力できます。番号はコンマで区切ります。</p>

asnlrc の例

以下の例は、**asnlrc** コマンドの使用法を示しています。これらの例は、Linux、UNIX、Windows、または z/OS オペレーティング・システムで実行できます。

例 1

キャプチャー・プログラムの実行をトレースするには、次のように入力します。

1. 最大のバッファー・サイズとファイル・サイズを持つトレース・ファイル名を指定して、トレース機能を開始します。

```
asnlrc on -db mydb -cap -schema myschema -b 256k -fn myfile.trc -fs 500m
```

2. キャプチャー・プログラムを開始し、適切な長さの時間実行させます。

3. トレース機能がオンになっている間に、共有メモリーからデータを直接表示しません。

トレース機能から処理とスレッドのサマリー情報を表示するには、次のように指定します。

```
asnlrc flw -db mydb -cap -schema myschema
```

キャプチャー・ログ・リーダーからのみ、フロー、基本、詳細、およびパフォーマンスのデータ・レコードを表示するには、次のように指定します。

```
asnlrc fmt -db mydb -cap -schema myschema -d 0  
-df "Capture Log Read" 1,2,3,4
```

4. トレース機能を停止します。

```
asnlrc off -db mydb -cap -schema myschema
```

トレース・ファイルには、キャプチャー・プログラムの始動時点からトレース機能をオフにした時点までに生成された、すべてのキャプチャー・プログラム・トレース・データが含まれます。

5. トレース機能を停止した後、生成されたバイナリー・ファイルからのデータをフォーマットします。

```
asnlrc flw -fn myfile.trc
```

および

```
asnlrc fmt -fn myfile.trc -d 0 -df "Capture Log Read" 1,2,3,4
```

例 2

レプリケーション・アラート・モニター・プログラムのトレース機能を開始するには、次のように指定します。

```
asnlrc on -db mydb -mon -qualifier monq
```

例 3

アプライ・プログラムのパフォーマンス・データだけをトレースするには、次のように指定します。

```
asnlrc on -db mydb -app -qualifier aql -b 256k -fn myfile.trc -d 4
```

例 4

キャプチャー・プログラムのすべてのフローおよびパフォーマンス・データをトレースするには、次のように指定します。

```
asnlrc on dbserve1 -cap -schema myschema -b 256k  
-fn myfile.trc -d 1,4
```

例 5

すべてのグローバルなパフォーマンス・データおよび、キャプチャー・プログラムの、特定のキャプチャー・ログ・リーダーのフロー・データをトレースするには、次のように指定します。


```
asnlrc on -db mydb -cap -schema myschema -b 256k -fn myfile.trc -d 4
      -df "Capture Log Read" 1
```

例 6

キャプチャー・プログラムをトレースし、トレース機能の特定の時点のイメージを表示して保管するには、次のようにします。

1. 最新のレコードを保留するために十分なバッファ・サイズを指定して、トレース・コマンドを始動します。

```
asnlrc on -db mydb -cap -schema myschema -b 4m
```

2. キャプチャー・プログラムを開始し、適切な長さの時間実行させます。

3. 共有メモリーに保管された、特定の時点の詳細なトレース情報を表示します。

```
asnlrc fmt -db mydb -cap -schema myschema
```

4. 特定の時点のトレース情報をファイルに保管します。

```
asnlrc dmp myfile.trc -db mydb -cap -schema myschema
```

5. トレース機能を停止します。

```
asnlrc off -db mydb -cap -schema myschema
```

共有セグメントを使用した asnlrc の例

スタンドアロン・トレース機能 **asnlrc** は、共有セグメントを使用して、トレース対象の Q キャプチャー・プログラム、Q アプライ・プログラム、キャプチャー・プログラム、アラート・プログラム、またはレプリケーション・アラート・モニター・プログラムそれぞれと通信します。ファイルが指定されていない場合、トレース項目を保持するためにも共有セグメントが使用されます。それ以外の場合、正しい共有セグメントを突き合わせてトレースを制御するために、**asnlrc** コマンドおよびトレース対象の各プログラムの両方に、マッチング・オプションを指定しなければなりません。以下の例は、Q キャプチャー・プログラム、Q アプライ・プログラム、キャプチャー・プログラム、アラート・プログラム、またはアラート・モニター・プログラムとの関連でトレース機能を使用する場合に、指定する必要のあるオプションを示しています。

Q キャプチャー・プログラムについては、**asnlrc** コマンドに **-db** パラメーターで指定したデータベースが、**asnlqcap** コマンドに **capture_server** パラメーターで指定したデータベースと一致している必要があります。

```
asnlrc -db ASN6 -schema EMI -qcap
asnlqcap capture_server=ASN6 capture_schema=EMI
```

Q アプライ・プログラムについては、**asnlrc** コマンドに **-db** パラメーターで指定したデータベースが、**asnlqapp** コマンドに **apply_server** パラメーターで指定したデータベースと一致している必要があります。

```
asnlrc -db TSN3 -schema ELB -qapp
asnlqapp apply_server=TSN3 apply_schema=ELB
```

キャプチャー・プログラムについては、**asnlrc** コマンドに **-db** パラメーターで指定したデータベースが、**asnlcap** コマンドに **capture_server** パラメーターで指定したデータベースと一致している必要があります。

asntrc

```
asntrc -db DSN6 -schema JAY -cap  
asnrcap capture_server=DSN6 capture_schema=JAY
```

アプライ・プログラムについては、**asntrc** コマンドに **-db** パラメーターで指定したデータベースが、**asnapply** コマンドに **control_server** パラメーターで指定したデータベースと一致している必要があります。

```
asntrc -db SVL_LAB_DSN6 -qualifier MYQUAL -app  
asnapply control_server=SVL_LAB_DSN6 apply_qual=MYQUAL
```

レプリケーション・アラート・モニター・プログラムについては、**asntrc** コマンドに **-db** パラメーターで指定したデータベースが、**asnmon** コマンドに **monitor_server** パラメーターで指定したデータベースと一致している必要があります。

```
asntrc -db DSN6 -qualifier MONQUAL -mon  
asnmon monitor_server=DSN6 monitor_qual=MONQUAL
```

asntdiff: ソース表とターゲット表とのデータの比較

ソース表をターゲット表と比較して両者の間の違いのリストを生成するには、**asntdiff** コマンドを使用します。**asntdiff** コマンドは、Linux、UNIX、Windows、または z/OS 上のオペレーティング・システム上のシステム・プロンプトかシェル・スクリプトで実行してください。

asntdiff コマンドは、Linux、UNIX、Windows、z/OS、および iSeries オペレーティング・システム上で DB2 UDB 表を比較します。

構文

```
▶▶—asntdiff—db=server—schema=schema—where=WHERE_clause—————▶▶  
▶▶—diff=table_name—————▶▶
```

パラメーター

表 44 は、**asnltdiff** コマンドの呼び出しパラメーターを定義します。

表 44. Linux、UNIX、および Windows オペレーティング・システム用 **asnltdiff** 呼び出しパラメーター定義

パラメーター	定義
db=server	<p>比較対象のソース表とターゲット表に関する情報を保管するデータベースの DB2 UDB 別名を指定します。この値は、Q レプリケーションか SQL レプリケーションのどちらを使用するかに応じて変わります。</p> <p>Q レプリケーション この値は、IBMQREP_SUBS 表を含む Q キャプチャー・サーバーの名前になります。</p> <p>SQL レプリケーション この値は、IBMSNAP_SUBS_MEMBR 表を含むアプライ・コントロール・サーバーの名前になります。</p>
schema=schema	<p>Q レプリケーションの場合は Q キャプチャー・コントロール表のスキーマを指定し、SQL レプリケーションの場合はアプライ・コントロール表のスキーマを指定します。</p>
where=WHERE_clause	<p>比較対象のソース表とターゲット表に関する情報を保管するコントロール表の 1 行を固有に識別する、SQL WHERE 文節を指定します。WHERE 文節は、二重引用符で囲まなければならない。このパラメーターの値は、Q レプリケーションか SQL レプリケーションのどちらを使用するかに応じて変わります。</p> <p>Q レプリケーション WHERE 文節は IBMQREP_SUBS 表中の行を指定します。この表は SUBNAME 列を使用して、ソース表とターゲット表を含む Q サブスクリプションを識別します。</p> <p>SQL レプリケーション WHERE 文節は IBMSNAP_SUBS_MEMBR 表中の行を指定します。この表は SET_NAME 列、APPLY_QUAL 列、TARGET_SCHEMA 列と TARGET_TABLE 列を使用して、ソース表とターゲット表を含むサブスクリプション・セットのメンバーを識別します。</p>
diff=table_name	<p>ソース・データベース中に作成され、ソース表とターゲット表の間の違いを保管する表の名前を指定します。この表には、検出される違いごとに 1 行ずつあります。このパラメーターを組み込まない場合は、違いに関する表の名前は <code>schema.ASNLTDIFF</code> (<code>schema</code> は、比較対象のソース表とターゲット表に関する情報を含む Q キャプチャー・コントロール表かアプライ・コントロール表のスキーマ) になります。</p>

asntdiff の例

次の例は、**asntdiff** コマンドの使用方法を示しています。

例 1

Q レプリケーションにおいて、Q キャプチャー・スキーマが asn の Q キャプチャー・サーバー source_db 上で、Q サブスクリプション my_qsub 中に指定されているソース表とターゲット表の間の違いを検出するには、以下のようにします。

```
asntdiff db=source_db schema=asn where="where subname = 'my_qsub'"
```

例 2

SQL レプリケーションにおいて、アプライ・スキーマが asn のアプライ・コントロール・サーバー apply_db 上で、サブスクリプション・セット my_set 中に指定されているソース表とターゲット表 trg_table の間の違いを検出し、相違表の名前を diff_table にするには、以下のようにします。

```
asntdiff DB=apply_db schema=asn where="where set_name = 'my_set'
and target_table = 'trg_table'" diff=diff_table
```

関連概念:

- 359 ページの『Q レプリケーションおよびイベント発行用のシステム・コマンド -- 概要』

関連資料:

- 359 ページの『ロードマップ: Q レプリケーションおよびイベント発行のシステム・コマンド』
- 410 ページの『asntrep: ソース表とターゲット表の間の違いの修復』

asntrep: ソース表とターゲット表の間の違いの修復

ソース表とターゲット表の間の違いを修復して 2 つの表を同期化するには、**asntrep** コマンドを使用します。**asntrep** コマンドは、Linux、UNIX、または Windows 上のオペレーティング・システム上のシステム・プロンプトかシェル・スクリプトで実行してください。

構文

```
▶▶ asntrep db=server schema=schema where=WHERE_clause [diff=table_name] ▶▶
```

パラメーター

表 45 は、**asntrep** コマンドの呼び出しパラメーターを定義します。

表 45. Linux、UNIX、および Windows オペレーティング・システム用 **asntrep** 呼び出しパラメーター定義

パラメーター	定義
db=server	同期化したいソース表とターゲット表に関する情報を保管するデータベースの DB2 UDB 別名を指定します。この値は、Q レプリケーションか SQL レプリケーションのどちらを使用するかに応じて変わります。 Q レプリケーション この値は、IBMQREP_SUBS 表を含む Q キャプチャー・サーバーの名前になります。 SQL レプリケーション この値は、IBMSNAP_SUBS_MEMBR 表を含むアプライ・コントロール・サーバーの名前になります。
schema=schema	Q レプリケーションの場合は Q キャプチャー・コントロール表のスキーマを指定し、SQL レプリケーションの場合はアプライ・コントロール表のスキーマを指定します。
where=WHERE_clause	同期化対象のソース表とターゲット表に関する情報を保管するコントロール表の 1 行をユニークに識別する、SQL WHERE 文節を指定します。WHERE 文節は、二重引用符で囲まなければならないなりません。このパラメーターの値は、Q レプリケーションか SQL レプリケーションのどちらを使用するかに応じて変わります。 Q レプリケーション WHERE 文節は IBMQREP_SUBS 表中の行を指定します。この表は SUBNAME 列を使用して、ソース表とターゲット表を含む Q サブスクリプションを識別します。 SQL レプリケーション WHERE 文節は IBMSNAP_SUBS_MEMBR 表中の行を指定します。この表は SET_NAME 列、APPLY_QUAL 列、TARGET_SCHEMA 列と TARGET_TABLE 列を使用して、ソース表とターゲット表を含むサブスクリプション・セットのメンバーを識別します。
diff=table_name	asntdiff コマンドを使用してソース・データベース中に作成され、ソース表とターゲット表の間の違いを保管する表の名前を指定します。この表に保管される情報は、ソース表とターゲット表の同期化に使用されます。

asntrep の例

次の例は、**asntrep** コマンドの使用方を示しています。

例 1

Q レプリケーションにおいて、Q キャプチャー・スキーマが asn の Q キャプチャー・サーバー source_db 上で、Q サブスクリプション my_qsub 中に指定されているソース表とターゲット表 (両者の違いは表 q_diff_table に保管される) を同期化するには、以下のようにします。

```
asntrep db=source_db schema=asn where="where subname = 'my_qsub'" diff=q_diff_table
```

例 2

SQL レプリケーションにおいて、アプライ・スキーマが asn のアプライ・コントロール・サーバー apply_db 上で、サブスクリプション・セット my_set 中に指定されているソース表とターゲット表 trg_table (両者の違いは表 sql_diff_table に保管される) を同期化するには、以下のようにします。

```
asntrep DB=apply_db SCHEMA=asn WHERE="where set_name = 'my_set'
and target_table = 'trg_table'" diff=sql_diff_table
```

関連概念:

- 359 ページの『Q レプリケーションおよびイベント発行用のシステム・コマンド -- 概要』

関連資料:

- 359 ページの『ロードマップ: Q レプリケーションおよびイベント発行のシステム・コマンド』
- 408 ページの『asntdiff: ソース表とターゲット表とのデータの比較』

asnqmfmt: Q レプリケーションとイベント発行のメッセージのフォーマットおよび表示

Q レプリケーションおよびイベント発行で使用されるメッセージをフォーマットして表示するには、**asnqmfmt** コマンドを使用します。このコマンドは、Linux、UNIX、Windows、または z/OS 上の UNIX System Services (USS) のオペレーティング・システム・プロンプトまたはシェル・スクリプトで実行してください。

z/OS では、JCL を使用したメッセージ・フォーマット設定プログラムを操作できます。サンプル qrhlqual.SASNSAMP(ASNQMFMT) にサンプル JCL があります。532 ページの『Q レプリケーションおよびイベント発行のサンプル JCL プログラム (z/OS)』を参照してください。

構文

```

▶▶ asnqmfmt—queue_name—queue_manager_name—
└─o—┬──────────────────────────────────────────────────────────────────────────────────┘
      └─filepath_name—┘
▶┬──hex──┬──-l—number──┬──-delmsg──┬──-mqmd──┘
▶┬──oenc—output_encoding_name──┬──help──┘

```

パラメーター

表 46 では、**asnqmfmt** の呼び出しパラメーターについて説明します。

表 46. *asnqmfmt* 呼び出しパラメーターの定義

パラメーター	定義
<i>queue_name</i>	フォーマット、表示、およびオプションで削除したいメッセージを持つ WebSphere MQ キューの名前を指定します。
<i>queue_manager_name</i>	キューが定義されている WebSphere MQ キュー・マネージャーの名前を指定します。
-o <i>filepath_name</i>	定形出力を含むファイルの名前を指定します。 -o パラメーターを指定しない場合、定形メッセージは標準出力 (stdout) に書き込まれます。出力ファイルの名前が // で始まる場合、そのファイルは z/OS データ・セットに書き込まれます。デフォルトでは、ファイルは asnqmfmt コマンドを呼び出したディレクトリーで作成されます。ファイル名付きのパスを指定することにより、ディレクトリーを変更できます。
-hex	メッセージを 16 進数でフォーマットすることを指定します。このパラメーターを指定しない場合、メッセージはそのメッセージ・フォーマット・タイプとして、圧縮または XML のいずれかに応じて表示されます。
-l <i>number</i>	フォーマットしたいメッセージの数を指定します。
-delmsg	メッセージがフォーマットされた後にキューから削除されることを指定します。
-mqmd	フォーマットされるメッセージごとの WebSphere MQ メッセージ記述子を表示することを指定します。
-oenc <i>output_encoding_name</i>	メッセージのフォーマットに使用されるコード・ページを指定します。このパラメーターを指定しない場合、メッセージは、コマンドが呼び出されるオペレーティング・システムのデフォルト・コード・ページでフォーマットされます。
-help	有効なコマンド・パラメーターを記述と一緒に表示します。

asnqmfmt の例

次の例は、**asnqmfmt** コマンドの使用方を示しています。

例 1

キュー・マネージャー QMGR1 で定義されている、送信キュー Q1 上のすべてのメッセージを標準出力に表示するには、次のようにします。

```
asnqmfmt Q1 QMGR1
```

例 2

C:\qrepl ディレクトリーに保管されている Q1_messages という名前のファイルの送信キュー Q1 上のすべてのメッセージを表示するには、次のようにします (Windows のみ)。

```
asnqmfmt Q1 QMGR1 -o C:\qrepl\Q1_messages
```


例 3

キュー・マネージャー QMGR1 で定義されている管理キュー ADMNQ1 上のすべてのメッセージの 16 進数バージョンを標準出力に表示するには、次のようにします。

```
asnqfmt ADMNQ1 QMGR1 -hex
```

例 4

キュー・マネージャー QMGR1 で定義されている管理キュー ADMNQ1 上のすべてのメッセージのメッセージ本体およびメッセージ記述子を標準出力で表示し、その後キューからメッセージを削除するには、次のようにします。

```
asnqfmt ADMNQ1 QMGR1 -delmsg -mqmd
```

例 5

C:\%qrepl ディレクトリーに保管されている Q2_messages という名前のファイルの受信キュー Q2 上の最初の 100 のメッセージを表示するには、次のようにします (Windows のみ)。

```
asnqfmt Q2 QMGR2 -l 100 -o C:\%qrepl\%Q2_messages
```

関連概念:

- 359 ページの『Q レプリケーションおよびイベント発行用のシステム・コマンド -- 概要』

関連資料:

- 359 ページの『ロードマップ: Q レプリケーションおよびイベント発行のシステム・コマンド』

第 26 章 Q レプリケーションおよびイベント発行のコントロール表

Q レプリケーションおよびイベント発行のコントロール表 -- 概要

コントロール表は、リレーショナル・データベース表であり、Q レプリケーションおよびイベント発行プログラムの情報を保管する際に使用されます。これらのコントロール表は、Q キャプチャー・サーバー、Q アプライ・サーバー、およびモニター・コントロール・サーバーで保管されます。表を参照するには、以下のような 3 つの方法があります。

- クイック・リファレンスには、各サーバーの表、各表の列、そして各表の索引のリストが含まれます。『コントロール表の概観』を参照。
- 各サーバー上の表の概観については、以下を参照してください。
 - 421 ページの『Q キャプチャー・サーバーのコントロール表のリスト』
 - 422 ページの『Q アプライ・サーバーのコントロール表のリスト』
 - 424 ページの『モニター・コントロール・サーバーの表のリスト』
- 表とその列に関して、より詳しい説明が必要な場合は、以下を参照してください。
 - 425 ページの『Q キャプチャー・コントロール表の詳細な構造 -- 概説』
 - 448 ページの『Q アプライ・コントロール表の詳細な構造 -- 概説』
 - 471 ページの『モニター・コントロール表の詳細な構造 -- 概説』
 - 489 ページの『ピアツーピア・レプリケーションのバージョン管理用追加列の詳細な構造』

コントロール表の概観

416 ページの図 17 および 417 ページの図 18 は、Q キャプチャー・サーバーのコントロール表、各表の列、そして各表の索引およびキーを示しています。418 ページの図 19 および 419 ページの図 20 は、Q アプライ・サーバー上の表、各表の列、そして各表の索引およびキーを示しています。420 ページの図 21 および 421 ページの図 22 は、モニター・コントロール・サーバー上の表、各表の列、そして各表の索引を示しています。421 ページの図 23 は、ピアツーピア・レプリケーションのレプリケーション管理ツールを使用して、顧客表に追加するバージョン列を示しています。

Q キャプチャー・サーバーで使用されるコントロール表 (1/2)

IBMQREP_CAPPARMS

ユニーク索引: (QMGR)	
QMGR	VARCHAR(48) NOT NULL
REMOTE_SRC_SERVER	VARCHAR(18)
RESTARTQ	VARCHAR(48) NOT NULL
ADMINQ	VARCHAR(48) NOT NULL
STARTMODE	VARCHAR(6) NOT NULL WITH DEFAULT
MEMORY_LIMIT	INTEGER NOT NULL WITH DEFAULT
COMMIT_INTERVAL	INTEGER NOT NULL WITH DEFAULT
AUTOSTOP	CHAR(1) NOT NULL WITH DEFAULT
MONITOR_INTERVAL	INTEGER NOT NULL WITH DEFAULT
MONITOR_LIMIT	INTEGER NOT NULL WITH DEFAULT
TRACE_LIMIT	INTEGER NOT NULL WITH DEFAULT
SIGNAL_LIMIT	INTEGER NOT NULL WITH DEFAULT
PRUNE_INTERVAL	INTEGER NOT NULL WITH DEFAULT
SLEEP_INTERVAL	INTEGER NOT NULL WITH DEFAULT
LOGREUSE	CHAR(1) NOT NULL WITH DEFAULT
LOGSTDOUT	CHAR(1) NOT NULL WITH DEFAULT
TERM	CHAR(1) NOT NULL WITH DEFAULT
CAPTURE_PATH	VARCHAR(1040)
ARCH_LEVEL	CHAR(4) NOT NULL WITH DEFAULT

IBMQREP_SUBS

主キー: (SUBNAME)	
SUBNAME	VARCHAR(132) NOT NULL
SOURCE_OWNER	VARCHAR(30) 1 NOT NULL
SOURCE_NAME	VARCHAR(128) 2 NOT NULL
TARGET_SERVER	VARCHAR(18)
TARGET_ALIAS	VARCHAR(8)
TARGET_OWNER	VARCHAR(30) 1
TARGET_NAME	VARCHAR(128) 2
TARGET_TYPE	INTEGER
APPLY_SCHEMA	VARCHAR(128)
SENDQ	VARCHAR(48) NOT NULL
SEARCH_CONDITION	VARCHAR(2048)
SUB_ID	INTEGER
SUBTYPE	CHAR(1) NOT NULL WITH DEFAULT
ALL_CHANGED_ROWS	CHAR(1) NOT NULL WITH DEFAULT
BEFORE_VALUES	CHAR(1) NOT NULL WITH DEFAULT
CHANGED_COLS_ONLY	CHAR(1) NOT NULL WITH DEFAULT
HAS_LOADPHASE	CHAR(1) NOT NULL WITH DEFAULT
STATE	CHAR(1) NOT NULL WITH DEFAULT
STATE_TIME	TIMESTAMP NOT NULL WITH DEFAULT
STATE_INFO	CHAR(8)
STATE_TRANSITION	VARCHAR(256) FOR BIT DATA
SUBGROUP	VARCHAR(30)
SOURCE_NODE	SMALLINT NOT NULL WITH DEFAULT
TARGET_NODE	SMALLINT NOT NULL WITH DEFAULT
GROUP_MEMBERS	CHAR (254) FOR BIT DATA
OPTIONS_FLAG	CHAR(4) NOT NULL WITH DEFAULT
SUPPRESS_DELETES	CHAR(1) NOT NULL WITH DEFAULT
DESCRIPTION	VARCHAR(254)

IBMQREP_SRC_COLS

主キー: (SUBNAME, SRC_COLNAME)	
SUBNAME	VARCHAR(132) NOT NULL
SRC_COLNAME	VARCHAR(30) NOT NULL
IS_KEY	SMALLINT NOT NULL WITH DEFAULT

IBMQREP_SENDQUEUES

主キー: (SENDQ)	
ユニーク索引: (PUBQMAPNAME)	
PUBQMAPNAME	VARCHAR(128) NOT NULL
SENDQ	VARCHAR(48) NOT NULL
RECVO	VARCHAR(48)
MESSAGE_FORMAT	CHAR(1) NOT NULL WITH DEFAULT
MSG_CONTENT_TYPE	CHAR(1) NOT NULL WITH DEFAULT
STATE	CHAR(1) NOT NULL WITH DEFAULT
STATE_TIME	TIMESTAMP NOT NULL WITH DEFAULT
STATE_INFO	CHAR(8)
ERROR_ACTION	CHAR(1) NOT NULL WITH DEFAULT
HEARTBEAT_INTERVAL	INTEGER NOT NULL WITH DEFAULT
MAX_MESSAGE_SIZE	INTEGER NOT NULL WITH DEFAULT
APPLY_SERVER	VARCHAR(18)
APPLY_ALIAS	VARCHAR(8)
APPLY_SCHEMA	VARCHAR(128)
DESCRIPTION	VARCHAR(254)

IBMQREP_SRCH_COND

ASNOREQD	INTEGER
----------	---------

IBMQREP_SIGNAL

主キー: (SIGNAL_TIME)	
SIGNAL_TIME	TIMESTAMP NOT NULL WITH DEFAULT
SIGNAL_TYPE	VARCHAR(30) NOT NULL
SIGNAL_SUBTYPE	VARCHAR(30)
SIGNAL_INPUT_IN	VARCHAR(500)
SIGNAL_STATE	CHAR(1) NOT NULL WITH DEFAULT
SIGNAL_LSN	CHAR(10) FOR BIT DATA

IBMQREP_CAPTRACE

非固有索引: (TRACE_TIME)	
OPERATION	CHAR(8) NOT NULL
TRACE_TIME	TIMESTAMP NOT NULL
DESCRIPTION	VARCHAR(1024) NOT NULL

1 VARCHAR(128)、DB2 UDB for z/OS バージョン 8 新機能モード
 2 VARCHAR(18)、DB2 UDB for z/OS バージョン 8 互換モードまたはそれ以前

図 17. Q キャプチャー・サーバーで使用される表: これらの表は、Q キャプチャー・サーバーで Q キャプチャー・プログラム によって使用されます。各表の索引およびキーを構成する列は、表名の下に括弧内に示されています。

Q キャプチャー・サーバーで使用されるコントロール表 (2/2)

IBMQREP_CAPMON		IBMQREP_CAPENQ	
非固有索引: (MONITOR_TIME)		LOCKNAME	INTEGER
MONITOR_TIME	TIMESTAMP NOT NULL		
CURRENT_LOG_TIME	TIMESTAMP NOT NULL		
CAPTURE_IDLE	INTEGER NOT NULL		
CURRENT_MEMORY	INTEGER NOT NULL		
ROWS_PROCESSED	INTEGER NOT NULL		
TRANS_SKIPPED	INTEGER NOT NULL		
TRANS_PROCESSED	INTEGER NOT NULL		
TRANS_SPILLED	INTEGER NOT NULL		
MAX_TRANS_SIZE	INTEGER NOT NULL		
QUEUES_IN_ERROR	INTEGER NOT NULL		
RESTART_SEQ	CHAR(10) FOR BIT DATA NOT NULL		
CURRENT_SEQ	CHAR(10) FOR BIT DATA NOT NULL		
IBMQREP_CAPQMON		IBMQREP_ADMINMSG	
非固有索引: (MONITOR_TIME)		主キー: (MQMSGID)	
MONITOR_TIME	TIMESTAMP NOT NULL	MQMSGID	CHAR(24) FOR BIT DATA
SENDQ	VARCHAR(48) NOT NULL	MSG_TIME	TIMESTAMP NOT NULL WITH DEFAULT
ROWS_PUBLISHED	INTEGER NOT NULL		
TRANS_PUBLISHED	INTEGER NOT NULL		
CHG_ROWS_SKIPPED	INTEGER NOT NULL		
DELRROWS_SUPPRESSED	INTEGER NOT NULL		
ROWS_SKIPPED	INTEGER NOT NULL		

図 18. Q キャプチャー・サーバーで使用される表 (続き): これらの表は、Q キャプチャー・サーバーで Q キャプチャー・プログラム によって使用されます。各表の索引およびキーを構成する列は、表名の下に括弧内に示されています。

Q アプライ・サーバーで使用されるコントロール表 (1/2)

IBMQREP_APPLYPARMS

ユニーク索引: (QMGR)	
QMGR	VARCHAR(48) NOT NULL
MONITOR_LIMIT	INTEGER NOT NULL WITH DEFAULT
TRACE_LIMIT	INTEGER NOT NULL WITH DEFAULT
MONITOR_INTERVAL	INTEGER NOT NULL WITH DEFAULT
PRUNE_INTERVAL	INTEGER NOT NULL WITH DEFAULT
AUTOSTOP	CHAR(1) NOT NULL WITH DEFAULT
LOGREUSE	CHAR(1) NOT NULL WITH DEFAULT
LOGSTDOUT	CHAR(1) NOT NULL WITH DEFAULT
APPLY_PATH	VARCHAR(1040)
ARCH_LEVEL	CHAR(4) NOT NULL WITH DEFAULT
TERM	CHAR(1) NOT NULL WITH DEFAULT
PWDFILE	VARCHAR(48)
DEADLOCK_RETRIES	INTEGER NOT NULL WITH DEFAULT

IBMQREP_RECQUEUEES

主キー: (RECVQ)	
ユニーク索引: (REPQMAPNAME)	
REPQMAPNAME	VARCHAR(128) NOT NULL
RECVQ	VARCHAR(48) NOT NULL
SENDQ	VARCHAR(48)
ADMINQ	VARCHAR(48) NOT NULL
NUM_APPLY_AGENTS	INTEGER NOT NULL WITH DEFAULT
MEMORY_LIMIT	INTEGER NOT NULL WITH DEFAULT
CAPTURE_SERVER	VARCHAR(18) NOT NULL
CAPTURE_ALIAS	VARCHAR(8) NOT NULL
CAPTURE_SCHEMA	VARCHAR(30) NOT NULL WITH DEFAULT
STATE	CHAR(1) NOT NULL WITH DEFAULT
STATE_TIME	TIMESTAMP NOT NULL WITH DEFAULT
STATE_INFO	CHAR(8)
DESCRIPTION	VARCHAR(254)

IBMQREP_TRG_COLS

索引: (RECVQ, SUBNAME, TARGET_COLNAME)	
RECVQ	VARCHAR(48) NOT NULL
SUBNAME	VARCHAR(132) NOT NULL
SOURCE_COLNAME	VARCHAR(30) NOT NULL
TARGET_COLNAME	VARCHAR(30) NOT NULL
TARGET_COLNO	INTEGER
MSG_COL_CODEPAGE	INTEGER
MSG_COL_NUMBER	SMALLINT
MSG_COL_TYPE	SMALLINT
MSG_COL_LENGTH	INTEGER
IS_KEY	CHAR(1) NOT NULL

1 VARCHAR(128)、DB2 UDB for z/OS バージョン 8 新機能モード 2 VARCHAR(18)、DB2 UDB for z/OS バージョン 8 互換モードまたはそれ以前

IBMQREP_TARGETS

ユニーク索引: (TARGET_OWNER ASC, TARGET_NAME ASC, RECVQ ASC, SOURCE_OWNER ASC, SOURCE_NAME ASC)	
ユニーク索引: (SUBNAME, RECVQ)	
索引: (RECVQ, SUB_ID)	
SUBNAME	VARCHAR(132) NOT NULL
RECVQ	VARCHAR(48) NOT NULL
SUB_ID	INTEGER
SOURCE_SERVER	VARCHAR(18) NOT NULL
SOURCE_ALIAS	VARCHAR(8) NOT NULL
SOURCE_OWNER	VARCHAR(30) 1 NOT NULL
SOURCE_NAME	VARCHAR(128) 2 NOT NULL
SRC_NICKNAME_OWNER	VARCHAR(128)
SRC_NICKNAME	VARCHAR(128)
TARGET_OWNER	VARCHAR(30) 1 NOT NULL
TARGET_NAME	VARCHAR(128) 2 NOT NULL
TARGET_TYPE	INTEGER NOT NULL WITH DEFAULT
FEDERATED_TGT_SRVR	VARCHAR(18)
STATE	CHAR(1) NOT NULL WITH DEFAULT
STATE_TIME	TIMESTAMP NOT NULL WITH DEFAULT
STATE_INFO	CHAR(8)
SUBTYPE	CHAR(1) NOT NULL WITH DEFAULT
CONFLICT_RULE	CHAR(1) NOT NULL WITH DEFAULT
CONFLICT_ACTION	CHAR(1) NOT NULL WITH DEFAULT
ERROR_ACTION	CHAR(1) NOT NULL WITH DEFAULT
SPILLQ	VARCHAR(48)
OKSQLSTATES	VARCHAR(128)
SUBGROUP	VARCHAR(30)
SOURCE_NODE	SMALLINT NOT NULL WITH DEFAULT
TARGET_NODE	SMALLINT NOT NULL WITH DEFAULT
GROUP_INIT_ROLE	CHAR(1)
HAS_LOADPHASE	CHAR(1) NOT NULL WITH DEFAULT
LOAD_TYPE	SMALLINT NOT NULL WITH DEFAULT
DESCRIPTION	VARCHAR(254)

IBMQREP_EXCEPTIONS

EXCEPTION_TIME	TIMESTAMP NOT NULL WITH DEFAULT
RECVQ	VARCHAR(48) NOT NULL
SRC_COMMIT_LSN	CHAR(10) FOR BIT DATA NOT NULL
SRC_TRANS_TIME	TIMESTAMP NOT NULL
SUBNAME	VARCHAR(132) NOT NULL
REASON	CHAR(12) NOT NULL
SQLCODE	INTEGER
SQLSTATE	CHAR(5)
SQLERRMC	VARCHAR(70) FOR BIT DATA
OPERATION	VARCHAR(18) NOT NULL
TEXT	CLOB(32K) NOT NULL
IS_APPLIED	CHAR(1) NOT NULL
CONFLICT_RULE	CHAR(1)

IBMQREP_APPLYTRACE

非固有索引: (TRACE_TIME)	
OPERATION	CHAR(8) NOT NULL
TRACE_TIME	TIMESTAMP NOT NULL
DESCRIPTION	VARCHAR(1024) NOT NULL

図 19. Q アプライ・サーバーで使用される表: これらの表は、Q アプライ・サーバーで Q アプライ・プログラム によって使用されます。各表の索引およびキーを構成する列は、表名の下の括弧内に示されています。

Q アプライ・サーバーで使用されるコントロール表 (2/2)

IBMQREP_APPLYMON

非固有索引: (MONITOR_TIME)

MONITOR_TIME	TIMESTAMP NOT NULL
RECVQ	VARCHAR(48) NOT NULL
QSTART_TIME	TIMESTAMP NOT NULL
CURRENT_MEMORY	INTEGER NOT NULL
QDEPTH	INTEGER NOT NULL
END2END_LATENCY	INTEGER NOT NULL
QLATENCY	INTEGER NOT NULL
APPLY_LATENCY	INTEGER NOT NULL
TRANS_APPLIED	INTEGER NOT NULL
ROWS_APPLIED	INTEGER NOT NULL
TRANS_SERIALIZED	INTEGER NOT NULL
RI_DEPENDENCIES	INTEGER NOT NULL
RI_RETRIES	INTEGER NOT NULL
DEADLOCK_RETRIES	INTEGER NOT NULL
ROWS_NOT_APPLIED	INTEGER NOT NULL
MONSTER_TRANS	INTEGER NOT NULL
MEM_FULL_TIME	INTEGER NOT NULL
APPLY_SLEEP_TIME	INTEGER NOT NULL
SPILLED_ROWS	INTEGER NOT NULL
SPILLEDROWSAPPLIED	INTEGER NOT NULL
OLDEST_TRANS	TIMESTAMP NOT NULL
OKSQLSTATE_ERRORS	INTEGER NOT NULL
HEARTBEAT_LATENCY	INTEGER NOT NULL
KEY_DEPENDENCIES	INTEGER NOT NULL
UNIQ_DEPENDENCIES	INTEGER NOT NULL
UNIQ_RETRIES	INTEGER NOT NULL

IBMQREP_APPLYENQ

LOCKNAME	INTEGER
----------	---------

IBMQREP_DONEMSG

主キー: (RECVQ, MQMSGID)

RECVQ	VARCHAR(48) NOT NULL
MQMSGID	CHAR(24) FOR BIT DATA NOT NULL

IBMQREP_SPILLEDROW

主キー: (SPILLQ, MQMSGID)

SPILLQ	VARCHAR(48) NOT NULL
MQMSGID	CHAR(24) FOR BIT DATA NOT NULL

IBMQREP_DELTOMB

索引: (TARGET_NAME, TARGET_OWNER, VERSION_TIME, DESC, KEY_HASH)

TARGET_OWNER	VARCHAR(30) NOT NULL FOR Z/OS: VARCHAR(128) NOT NULL
TARGET_NAME	VARCHAR(128) NOT NULL
VERSION_TIME	TIMESTAMP NOT NULL
VERSION_NODE	SMALLINT NOT NULL
KEY_HASH	INTEGER NOT NULL
PACKED_KEY	VARCHAR(4096) FOR BIT DATA NOT NULL

IBMQREP_SAVERI

SUBNAME	VARCHAR(132) NOT NULL
RECVQ	VARCHAR(48) NOT NULL
CONSTNAME	VARCHAR(18) NOT NULL
TABSHEMA	VARCHAR(128) NOT NULL
TABNAME	VARCHAR(128) NOT NULL
REFTABSHEMA	VARCHAR(128) NOT NULL
REFTABNAME	VARCHAR(128) NOT NULL
ALTER_RI_DDL	VARCHAR(1680) NOT NULL
TYPE_OF_LOAD	CHAR(1) NOT NULL

IBMQREP_SPILLQS

主キー: (SPILLQ)

SPILLQ	VARCHAR(48) NOT NULL
SUBNAME	VARCHAR(132) NOT NULL
RECVQ	VARCHAR(48) NOT NULL

図 20. Q アプライ・サーバーで使用される表 (続き): これらの表は、Q アプライ・サーバーで Q アプライ・プログラム によって使用されます。各表の索引およびキーを構成する列は、表名の下に括弧内に示されています。

モニター・コントロール・サーバーで使用されるコントロール表 (1/2)

ASN.IBMSNAP_ALERTS

(MONITOR_QUAL, COMPONENT, SERVER_NAME, SCHEMA_OR_QUAL, SET_NAME, CONDITION_NAME, ALERT_CODE)

MONITOR_QUAL	CHAR(18) NOT NULL
ALERT_TIME	TIMESTAMP NOT NULL
COMPONENT	CHAR(1) NOT NULL
SERVER_NAME	CHAR(18) NOT NULL
SERVER_ALIAS	CHAR(8)
SCHEMA_OR_QUAL	VARCHAR(30)1 NOT NULL
SET_NAME	CHAR(18) NOT NULL WITH DEFAULT
CONDITION_NAME	CHAR(18) NOT NULL
OCCURRED_TIME	TIMESTAMP NOT NULL
ALERT_COUNTER	SMALLINT NOT NULL
ALERT_CODE	CHAR(10) NOT NULL
RETURN_CODE	INT NOT NULL
NOTIFICATION_SENT	CHAR(1) NOT NULL
ALERT_MESSAGE	VARCHAR(1024) NOT NULL

ASN.IBMSNAP_CONDITIONS

(MONITOR_QUAL, SERVER_NAME, COMPONENT, SCHEMA_OR_QUAL, SET_NAME, CONDITION_NAME)

MONITOR_QUAL	CHAR(18) NOT NULL
SERVER_NAME	CHAR(18) NOT NULL
COMPONENT	CHAR(1) NOT NULL
SCHEMA_OR_QUAL	VARCHAR(30)1 NOT NULL
SET_NAME	CHAR(18) NOT NULL WITH DEFAULT
SERVER_ALIAS	CHAR(8)
ENABLED	CHAR(1) NOT NULL
CONDITION_NAME	CHAR(18) NOT NULL
PARM_INT	INT
PARM_CHAR	VARCHAR(128)
CONTACT_TYPE	CHAR(1) NOT NULL
CONTACT	VARCHAR(127) NOT NULL

ASN.IBMSNAP_CONTACTGRP

(GROUP_NAME, CONTACT_NAME)

GROUP_NAME	VARCHAR(127) NOT NULL
CONTACT_NAME	VARCHAR(127) NOT NULL

ASN.IBMSNAP_CONTACTS

(CONTACT_NAME)

CONTACT_NAME	VARCHAR(127) NOT NULL
EMAIL_ADDRESS	VARCHAR(128) NOT NULL
ADDRESS_TYPE	CHAR(1) NOT NULL
DELEGATE	VARCHAR(127)
DELEGATE_START	DATE
DELEGATE_END	DATE
DESCRIPTION	VARCHAR(1024)

ASN.IBMSNAP_GROUPS

(GROUP_NAME)

GROUP_NAME	VARCHAR(127) NOT NULL
DESCRIPTION	VARCHAR(1024)

ASN.IBMSNAP_MONENQ

(索引なし)

MONITOR_QUAL	CHAR(18) NOT NULL
--------------	-------------------

ASN.IBMSNAP_MONPARMS

(MONITOR_QUAL)

MONITOR_QUAL	CHAR(18) NOT NULL
ALERT_PRUNE_LIMIT	INT WITH DEFAULT
AUTOPRUNE	CHAR(1) WITH DEFAULT
EMAIL_SERVER	VARCHAR(128)
LOGREUSE	CHAR(1) WITH DEFAULT
LOGSTDOUT	CHAR(1) WITH DEFAULT
NOTIF_PER_ALERT	INT WITH DEFAULT
NOTIF_MINUTES	INT WITH DEFAULT
MONITOR_ERRORS	VARCHAR(128)
MONITOR_INTERVAL	INT WITH DEFAULT
MONITOR_PATH	VARCHAR(1040)
RUNONCE	CHAR(1) WITH DEFAULT
TERM	CHAR(1) WITH DEFAULT
TRACE_LIMIT	INT WITH DEFAULT

1 DB2 for z/OS V8 互換モードかそれ以前の場合 VARCHAR (30)、DB2 for z/OS V8 新規関数モードの場合 VARCHAR (128)、DB2 for z/OS V8 互換モードかそれ以前の場合 VARCHAR (18)、DB2 for z/OS V8 新規関数モードの場合 VARCHAR (128)。

図 21. モニター・コントロール・サーバーで使用される表： これらの表は、モニター・コントロール・サーバーで、レプリケーション・アラート・モニター・プログラムによって使用されます。各表の主索引を構成する列は、表名の下に括弧内に示されています。

モニター・コントロール・サーバーで使用されるコントロール表 (2/2)

ASN.IBMSNAP_MONSERVERS		ASN.IBMSNAP_MONTRAIL	
(MONITOR_QUAL, SERVER_NAME)		(索引なし)	
MONITOR_QUAL	CHAR(18) NOT NULL	MONITOR_QUAL	CHAR(18) NOT NULL
SERVER_NAME	CHAR(18) NOT NULL	SERVER_NAME	CHAR(18) NOT NULL
SERVER_ALIAS	CHAR(8)	SERVER_ALIAS	CHAR(8)
LAST_MONITOR_TIME	TIMESTAMP NOT NULL	STATUS	SMALLINT NOT NULL
START_MONITOR_TIME	TIMESTAMP	LASTRUN	TIMESTAMP NOT NULL
END_MONITOR_TIME	TIMESTAMP	LASTSUCCESS	TIMESTAMP
LASTRUN	TIMESTAMP NOT NULL	ENDTIME	TIMESTAMP NOT NULL WITH DEFAULT
LASTSUCCESS	TIMESTAMP	LAST_MONITOR_TIME	TIMESTAMP NOT NULL
STATUS	SMALLINT NOT NULL	START_MONITOR_TIME	TIMESTAMP
		END_MONITOR_TIME	TIMESTAMP
		SQLCODE	INT
		SQLSTATE	CHAR(5)
		NUM_ALERTS	INT NOT NULL
		NUM_NOTIFICATIONS	INT NOT NULL
ASN.IBMSNAP_MONTRACE			
(MONITOR_QUAL, TRACE_TIME)			
MONITOR_QUAL	CHAR(18) NOT NULL		
TRACE_TIME	TIMESTAMP NOT NULL		
OPERATION	CHAR(8) NOT NULL		
DESCRIPTION	VARCHAR(1024) NOT NULL		

図 22. モニター・コントロール・サーバーで使用される表 (続き)：これらの表は、モニター・コントロール・サーバーで、レプリケーション・アラート・モニター・プログラムによって使用されます。各表の主索引を構成する列は、表名の下に括弧内に示されています。

カスタマー表：対等バージョン管理列

ibmqrepVERNODE	SMALLINT NOT NULL WITH DEFAULT
ibmqrepVERTIME	TIMESTAMP NOT NULL WITH DEFAULT

図 23. ピアツーピア・バージョン管理用追加列：これらの列は、ピアツーピア・レプリケーションのために、レプリケーション管理ツールによって追加されます。

関連概念：

- 415 ページの『Q レプリケーションおよびイベント発行のコントロール表 -- 概要』

Q キャプチャー・サーバーのコントロール表のリスト

Q キャプチャー・サーバーのコントロール表には、データ・ソース、Q サブスクリプションまたは XML 発行のオプション、Q キャプチャー・プログラムの稼働パラメーター、Q キャプチャー・パフォーマンス統計、および他のメタデータについての情報があります。この表は、レプリケーション・センターまたは SQL スクリプトで指定したオプションに基づいて作成されます。

表 47 では、Q キャプチャー・サーバーがあるコントロール表を説明しています。

表 47. Q キャプチャー・サーバーのコントロール表

表名	説明	参照ページ
IBMQREP_ADMINMSG	Q キャプチャー・プログラムが受信した管理メッセージが含まれる内部表。	425
IBMQREP_CAPENQ	1 つの Q キャプチャー・サーバーに対して既知のスキーマを持つ 1 つの Q キャプチャー・プログラムのみを実行することを確認。	426
IBMQREP_CAPMON	Q キャプチャー・プログラムのパフォーマンスに関する統計が含まれる。	427

表 47. Q キャプチャー・サーバーのコントロール表 (続き)

表名	説明	参照ページ
IBMQREP_CAPPARMS	Q キャプチャー・プログラムの操作をコントロールするためにユーザーが指定できるパラメーターを保持する。	428
IBMQREP_CAPQMON	Q キャプチャー・プログラムの送信キューごとのパフォーマンスに関する統計が含まれる。	432
IBMQREP_CAPTRACE	Q キャプチャー・プログラムからの通知、警告、およびエラー・メッセージが含まれる。	433
IBMQREP_SRCH_COND	Q キャプチャー・プログラムが使用する内部表で、Q サブスクリプションまたは XML 発行に指定した検索条件を評価する。	434
IBMQREP_SENDQUEUES	Q キャプチャー・プログラムがトランザクション、行の操作、ラージ・オブジェクト、および通知メッセージの送信に使用する WebSphere MQ キューに関する情報が含まれる。	434
IBMQREP_SIGNAL	Q キャプチャー・プログラムのプロンプトに使用されるシグナルがある。このシグナルは、ユーザーまたはサブスクライブ・アプリケーションにより挿入される。または Q キャプチャー・プログラムにより Q アプライ・プログラムまたはサブスクライブ・アプリケーションから制御メッセージを受信後に挿入される。	437
IBMQREP_SRC_COLS	Q サブスクリプションまたは XML 発行に複製または発行されるソース表内の列を識別する。	441
IBMQREP_SUBS	サブスクリプション・タイプ、ソース表、検索条件、データ送信オプション、ターゲット・ロード・オプション、および状態を含む、Q サブスクリプションおよび XML 発行の情報が含まれる。	442

関連概念:

- 415 ページの『Q レプリケーションおよびイベント発行のコントロール表 -- 概要』

関連タスク:

- 87 ページの『Q キャプチャー・プログラムと Q アプライ・プログラムのコントロール表の作成』

関連資料:

- 422 ページの『Q アプライ・サーバーのコントロール表のリスト』
- 424 ページの『モニター・コントロール・サーバーの表のリスト』

Q アプライ・サーバーのコントロール表のリスト

Q アプライ・サーバーのコントロール表サブスクリプションには、Q アプライの稼働パラメーター、Q サブスクリプション定義、パフォーマンス統計、および他のメタデータが含まれています。この表は、レプリケーション・センターまたは SQL スクリプトで指定したオプションに基づいて作成されます。

423 ページの表 48 では、Q アプライ・サーバーのコントロール表を説明していません。

表 48. Q アプライ・サーバーのコントロール表

表名	説明	参照ページ
IBMQREP_APPLYENQ	1 つの Q アプライ・サーバーに対して特定のスキーマを持つ唯一の Q アプライ・プログラムのみが実行されることを確認。	448
IBMQREP_APPLYMON	Q アプライ・プログラムの受信キューごとのパフォーマンスに関する統計がある。	449
IBMQREP_APPLYPARMS	Q アプライ・プログラムの操作をコントロールするためにユーザーが指定できるパラメーターがある。	452
IBMQREP_APPLYTRACE	Q アプライ・プログラムからの通知、警告、およびエラー・メッセージがある。	454
IBMQREP_DELTOMB	ピアツーピア・レプリケーションで競合削除を記録する Q アプライ・プログラムが使用する内部表。	455
IBMQREP_DONEMSG	どのメッセージが処理されたかを記録する Q アプライ・プログラムが使用する内部表。	456
IBMQREP_EXCEPTIONS	競合、エラー、またはロールバックのため、アプライできない行の変更がある。	457
IBMQREP_RECVQUEUES	トランザクション・メッセージを受信し、制御メッセージを送信するため Q アプライ・プログラムが使用し、Q アプライ・プログラムの稼働パラメーターがあるキューを識別する。	459
IBMQREP_SAVERI	ターゲットのロード中に、ドロップされる参照保全制約を保管するため、Q アプライ・プログラムが使用する内部表。	461
IBMQREP_SPILLEDROW	一時予備キューに送信される行を追跡するため、Q アプライ・プログラムが使用する内部表。	462
IBMQREP_SPILLQS	ターゲットへのアプライ前に、ソース表への変更を保持する一時予備キューを識別する。	463
IBMQREP_TRG_COLS	ソース列とターゲット列との間のマッピングについての情報がある。	464
IBMQREP_TARGETS	ターゲット表またはストアード・プロシージャの情報、および Q サブスクリプションのオプションがある。	465

関連概念:

- 415 ページの『Q レプリケーションおよびイベント発行のコントロール表 -- 概要』

関連タスク:

- 87 ページの『Q キャプチャー・プログラムと Q アプライ・プログラムのコントロール表の作成』

関連資料:

- 421 ページの『Q キャプチャー・サーバーのコントロール表のリスト』
- 424 ページの『モニター・コントロール・サーバーの表のリスト』

モニター・コントロール・サーバーの表のリスト

モニター・コントロール・サーバー上のコントロール表には、アラート条件が発生したときに、レプリケーション・アラート・モニターから、いつ、どのように、そしてだれに連絡するかに関する情報が含まれます。Linux、UNIX、Windows、および z/OS の場合は、レプリケーション・センターを使用して、ユーザーの指定に合わせてこれらのコントロール表を作成します。DataPropagator for iSeries にはモニター・コントロール表はありません。

表 49 では、モニター・コントロール・サーバー表を説明しています。

表 49. モニター・コントロール・サーバーのコントロール表

表名	説明	参照ページ
IBMSNAP_ALERTS	レプリケーション・アラート・モニターから発行されるすべてのアラートのレコードを保持する。	472
IBMSNAP_CONDITIONS	レプリケーション・アラート・モニターから担当者への連絡が必要なアラート条件と、特定の条件が発生したときの連絡先のグループまたは個人の名前が入っている。	474
IBMSNAP_CONTACTGRP	連絡先グループを構成する個人の連絡先が入っている。	481
IBMSNAP_CONTACTS	連絡先名に関連付けられたアラート条件が発生したときに、レプリケーション・アラート・モニターから各個人またはグループに通知する方法に関する情報が入っている。	488
IBMSNAP_GROUPS	各連絡先グループの名前と記述が入っている。	482
IBMSNAP_MONENQ	1 つのモニター修飾子に対して 1 つのレプリケーション・アラート・モニター・プログラムだけが確実に実行されるようにするために使用される。	483
IBMSNAP_MONPARMS	モニター・プログラムの操作をコントロールするためにユーザーが変更できるパラメーターが含まれている。	483
IBMSNAP_MONSERVERS	サーバーがレプリケーション・アラート・モニター・プログラム (モニター修飾子によって識別される) からモニターされた最後の時刻を保持する。	485
IBMSNAP_MONTRACE	モニター・プログラムからの重要なメッセージを保持する。	487
IBMSNAP_MONTRAIL	各モニター・サイクルに関する重要な情報を保持する。	483

関連概念:

- 415 ページの『Q レプリケーションおよびイベント発行のコントロール表 -- 概要』

関連タスク:

- 301 ページの『レプリケーション・アラート・モニターのコントロール表の作成』

関連資料:

- 422 ページの『Q アプライ・サーバーのコントロール表のリスト』
- 421 ページの『Q キャプチャー・サーバーのコントロール表のリスト』

Q キャプチャー・コントロール表の詳細な構造

Q キャプチャー・コントロール表の詳細な構造 -- 概説

以下のトピックでは、Q キャプチャー・サーバー上の各コントロール表について詳しく説明しています。コントロール表はアルファベット順にリストされます。各表にある列は、左から右に、出現順にリストされています。

- 『IBMQREP_ADMINMSG 表』
- 426 ページの 『IBMQREP_CAPENQ 表』
- 427 ページの 『IBMQREP_CAPMON 表』
- 428 ページの 『IBMQREP_CAPPARMS 表』
- 432 ページの 『IBMQREP_CAPQMON 表』
- 433 ページの 『IBMQREP_CAPTRACE 表』
- 434 ページの 『IBMQREP_SRCH_COND 表』
- 434 ページの 『IBMQREP_SENDQUEUES 表』
- 437 ページの 『IBMQREP_SIGNAL 表』
- 441 ページの 『IBMQREP_SRC_COLS 表』
- 442 ページの 『IBMQREP_SUBS 表』

関連概念:

- 415 ページの 『Q レプリケーションおよびイベント発行のコントロール表 -- 概要』
- 448 ページの 『Q アプライ・コントロール表の詳細な構造 -- 概説』
- 471 ページの 『モニター・コントロール表の詳細な構造 -- 概説』

IBMQREP_ADMINMSG 表

サーバー: Q キャプチャー・サーバー

デフォルト・スキーマ: ASN

主キー: MQMSGID

重要: SQL を使用してこの表を変更しないでください。この表の変更の方法が不適切であると、予期せぬ結果が生じたり、データが失われたりします。

IBMQREP_ADMINMSG 表は、Q キャプチャー・プログラムが使用する内部表で、時刻と受信する管理メッセージの ID を記録します。

表 50 では、IBMQREP_ADMINMSG 表の列の要旨を示します。

表 50. IBMQREP_ADMINMSG 表の列

列名	説明
MQMSGID	データ・タイプ: CHAR(24) FOR BIT DATA。 NULL 可能: いいえ メッセージの WebSphere MQ メッセージ ID。

表 50. IBMQREP_ADMINMSG 表の列 (続き)

列名	説明
MSG_TIME	<p>データ・タイプ: TIMESTAMP。 NULL 可能: デフォルトでは不可</p> <p>この表にメッセージが挿入されたときの Q キャプチャー・サーバーのタイム・スタンプ。デフォルト: 現在のタイム・スタンプ。</p>

関連概念:

- 425 ページの『Q キャプチャー・コントロール表の詳細な構造 -- 概説』

関連資料:

- 421 ページの『Q キャプチャー・サーバーのコントロール表のリスト』

IBMQREP_CAPENQ 表

サーバー: Q キャプチャー・サーバー

デフォルト・スキーマ: ASN

重要: SQL を使用してこの表を変更しないでください。この表の変更の方法が不適切であると、予期せぬ結果が生じたり、データが失われたりします。

IBMQREP_CAPENQ 表を使用して、以下のことを確認します。

- DB2 UDB for Linux、DB2 UDB for UNIX および DB2 UDB for Windows の場合は、1 つのデータベースに対して既知のスキーマを持つ 1 つの Q キャプチャー・プログラムのみを実行する。
- 非データ共有 DB2 UDB for z/OS の場合は、1 つのサブシステムに対して既知のスキーマを持つ 1 つの Q キャプチャー・プログラムのみを実行する。
- データ共有 DB2 UDB for z/OS の場合は、1 つのデータ共有グループに対して既知のスキーマを持つ 1 つの Q キャプチャー・プログラムのみを実行する。

Q キャプチャー・プログラムは実行時に、この表を排他的にロックする。Q キャプチャー・プログラムを 2 回開始すると、2 番目のインスタンスはこの表をロック待機している状態になります。その表は、空として作成されます。

表 51 では、IBMQREP_CAPENQ 表の列の要旨を示します。

表 51. IBMQREP_CAPENQ 表の列

列名	説明
LOCKNAME	<p>データ・タイプ: INTEGER。 NULL 可能: はい</p> <p>この列にはデータが含まれない。</p>

関連概念:

- 425 ページの『Q キャプチャー・コントロール表の詳細な構造 -- 概説』

関連資料:

- 421 ページの『Q キャプチャー・サーバーのコントロール表のリスト』

IBMQREP_CAPMON 表

サーバー: Q キャプチャー・サーバー

デフォルト・スキーマ: ASN

ユニークでない索引: MONITOR_TIME

重要: SQL を使用してこの表を変更しないでください。この表の変更の方法が不適切であると、予期せぬ結果が生じたり、データが失われたりします。

Q キャプチャー・プログラムは、IBMQREP_CAPMON 表に行を挿入し、一定期間内にパフォーマンス統計を記録します。IBMQREP_CAPPARMS 表の MONITOR_INTERVAL に指定した値により、Q キャプチャー・プログラムが IBMQREP_CAPMON 表に行を挿入する頻度を指示します。MONITOR_LIMIT 値は、整理の対象となるまでに、行がこの表の中に留まる分数を示します。

表 52 では、IBMQREP_CAPMON 表の列の要旨を示します。

表 52. IBMQREP_CAPMON 表の列

列名	説明
MONITOR_TIME	データ・タイプ: TIMESTAMP。 NULL 可能: いいえ この表に行が挿入されたときの Q キャプチャー・サーバーのタイム・スタンプ。
CURRENT_LOG_TIME	データ・タイプ: TIMESTAMP。 NULL 可能: いいえ Q キャプチャー・ログ読み取りプログラムで DB2 コミットが最後に確認された Q キャプチャー・サーバーのタイム・スタンプ。キャプチャー・プログラムがログの終わりに達した場合、この列には Q キャプチャー・サーバーの現在のタイム・スタンプがある。
CAPTURE_IDLE	データ・タイプ: INTEGER。 NULL 可能: いいえ ログの終わりに達した後、新規トランザクションの送信の待機中に、Q キャプチャー・プログラムがスリープする時間 (秒単位)。
CURRENT_MEMORY	データ・タイプ: INTEGER。 NULL 可能: いいえ Q キャプチャー・プログラムがログからトランザクションを再構成するのに使用するメモリーの量 (MB 単位)。
ROWS_PROCESSED	データ・タイプ: INTEGER。 NULL 可能: いいえ Q キャプチャー・プログラムがログから読み取る (挿入、更新、または削除操作別の) 行の数。
TRANS_SKIPPED	データ・タイプ: INTEGER。 NULL 可能: いいえ 変更が Q サブスクリプションまたは、XML 発行の一部ではない列に対して加えられるため、(変更済み行を含む) キューに書き込まれなかったトランザクションの数 (デフォルトでは、IBMQREP_SUBS 表の ALL_CHANGED_ROWS パラメーターは No に設定されている)。
TRANS_PROCESSED	データ・タイプ: INTEGER。 NULL 可能: いいえ Q キャプチャー・プログラムが処理した、トランザクションの数。

表 52. IBMQREP_CAPMON 表の列 (続き)

列名	説明
TRANS_SPILLED	データ・タイプ: INTEGER。NULL 可能: いいえ MEMORY_LIMIT しきい値を超えたので、Q キャプチャー・プログラムがファイルに対して書き出されたトランザクションの数。
MAX_TRAN_SIZE	データ・タイプ: INTEGER。NULL 可能: いいえ Q キャプチャー・プログラムが処理した最大のトランザクション (MB 単位)。
QUEUES_IN_ERROR	データ・タイプ: INTEGER。NULL 可能: いいえ メッセージを受信しないキューの数。
RESTART_SEQ	データ・タイプ: CHAR(10) FOR BIT DATA。NULL 可能: いいえ DB2 リカバリー・ログで Q キャプチャー・プログラムがウォーム・リスタート時に開始する時点の論理ログ・シーケンス番号。この値は、Q キャプチャー・プログラムが見つけた、コミットまたはアボート・レコードがまだ検出されていない、一番若いログ・シーケンス番号。
CURRENT_SEQ	データ・タイプ: CHAR(10) FOR BIT DATA。NULL 可能: いいえ Q キャプチャー・プログラムが読み取る、DB2 リカバリー・ログの最新の論理ログ・シーケンス番号。

関連概念:

- 283 ページの『Q レプリケーションおよびイベント発行プログラムの履歴およびパフォーマンス・データ』
- 425 ページの『Q キャプチャー・コントロール表の詳細な構造 -- 概説』

関連資料:

- 421 ページの『Q キャプチャー・サーバーのコントロール表のリスト』

IBMQREP_CAPPARMS 表

サーバー: Q キャプチャー・サーバー

デフォルト・スキーマ: ASN

ユニーク索引: QMGR

この表の情報は、SQL を使って更新できます。

IBMQREP_CAPPARMS 表は、Q キャプチャー・プログラムの操作をコントロールするためにユーザーが変更できる単一行を保持します。たとえば、Q キャプチャー・プログラムが開始時に使用する処理方式を指定できます。あるいは、送信キューにあるメッセージのコミットを Q アプライ・プログラムまたは使用するユーザー・アプリケーションに対して行う前に、Q キャプチャー・プログラムが待機する時間を設定できます。Q キャプチャー・プログラムは開始時にのみ、この表に加えられた変更を読み取ります。

重要: この表に行がない場合、または複数の行が入っている場合、Q キャプチャー・プログラムは実行しません。

表 53 では、IBMQREP_CAPPARMS 表の列の要旨を示します。

表 53. IBMQREP_CAPPARMS 表の列

列名	説明
QMGR	<p>データ・タイプ: VARCHAR(48)。NULL 可能: いいえ</p> <p>ソース・システム上の WebSphere MQ キュー・マネージャーの名前。</p>
REMOTE_SRC_SERVER	<p>データ・タイプ: VARCHAR(18)。NULL 可能: はい</p> <p>この列は将来の利用のために予約されている。</p>
RESTARTQ	<p>データ・タイプ: VARCHAR(48)。NULL 可能: いいえ</p> <p>Q キャプチャーのログ読み取りプログラムの再始動メッセージを保管するキューの名前。この名前は、それぞれの Q キャプチャー・プログラムごとに固有でなければならない。</p>
ADMINQ	<p>データ・タイプ: VARCHAR(48)。NULL 可能: いいえ</p> <p>Q アプライ・プログラムまたはユーザー・アプリケーションから制御メッセージを受信するキューの名前。この名前は、Q キャプチャー・プログラムごとに固有でなければならない。</p>
STARTMODE	<p>データ・タイプ: VARCHAR(6)。NULL 可能: デフォルトでは不可</p> <p>Q キャプチャー・プログラムが開始時にとるステップ。</p> <p>cold Q キャプチャー・プログラムは、再始動キューと管理キューをクリアし、N (新規) 状態または A (アクティブ) 状態のすべての Q サブスクリプションまたは XML 発行の処理を開始する。コールド・スタートの場合、Q キャプチャー・プログラムは DB2 リカバリー・ログの読み取りを、その末尾から開始する。</p> <p>warmsi Q キャプチャー・プログラムは、ログの読み取りを中止した位置から開始する。ただし、プログラムを初めて開始する場合を除く。この場合は、Q キャプチャー・プログラムはコールド・スタートに切り替わる。warmsi 始動モードでは、Q キャプチャー・プログラムは初めて開始する場合にのみコールド・スタートする。</p> <p>warmns Q キャプチャー・プログラムは、ログの読み取りを中止した位置から開始します。ウォーム・スタートできない場合、コールド・スタートに切り替わらない。warmns 始動モードでは、Q キャプチャー・プログラムが予期せずにコールド・スタートすることを避けられる。ウォーム・スタートした場合、Q キャプチャー・プログラムは終了したところから処理を再開する。Q キャプチャー・プログラムの開始後にエラーが発生すると、プログラムは終了し、すべての表は未変更のままになる。</p> <p>warmsa ウォーム・スタート情報が使用可能な場合は、Q キャプチャー・プログラムは、ログの読み取りを中止した位置から開始する。Q キャプチャー・プログラムは、ウォーム・スタートできない場合、コールド・スタートに切り替わる。</p> <p>ウォーム・スタートの場合、Q キャプチャー・プログラムは、I (非アクティブ) 状態ではない Q サブスクリプションまたは XML 発行のみ処理する。</p>

表 53. IBMQREP_CAPPARMS 表の列 (続き)

列名	説明
MEMORY_LIMIT	<p>データ・タイプ: INTEGER。 NULL 可能: デフォルトでは不可</p> <p>Q キャプチャー・プログラムがトランザクションの作成に使用するメモリーの量 (MB 単位)。この割り振りがすべて使用されると、メモリー内のトランザクションはファイルに書き出されます。</p>
COMMIT_INTERVAL	<p>データ・タイプ: INTEGER。 NULL 可能: デフォルトでは不可</p> <p>Q キャプチャー・プログラムが MQCMIT 呼び出しを発行する頻度 (ミリ秒単位)。この呼び出しは、WebSphere MQ キュー・マネージャーにシグナルを出し、Q アプライ・プログラムやユーザー・アプリケーションで利用可能なキューにあるデータ・メッセージおよび通知メッセージを作成する。</p>
AUTOSTOP	<p>データ・タイプ: CHAR(1)。 NULL 可能: デフォルトでは不可</p> <p>アクティブ DB2 ログの終わりに達したとき、Q キャプチャー・プログラムが停止するかどうかを示すフラグ。</p> <p>Y アクティブ DB2 ログの最後に達すると、Q キャプチャー・プログラムはすぐに停止する。</p> <p>N アクティブな DB2 ログの最後に達しても、Q キャプチャー・プログラムは実行を継続する。</p>
MONITOR_INTERVAL	<p>データ・タイプ: INTEGER。 NULL 可能: デフォルトでは不可</p> <p>Q キャプチャー・プログラムが IBMQREP_CAPMON 表と IBMQREP_CAPQMON 表に行を追加する頻度 (秒単位)。</p>
MONITOR_LIMIT	<p>データ・タイプ: INTEGER。 NULL 可能: デフォルトでは不可</p> <p>整理の対象となるまでに、行が IBMQREP_CAPMON 表および IBMQREP_CAPQMON 表の中に留まる分数。整理インターバルのたびに、この表中の行が現行タイム・スタンプに基づいてこの制限より古い場合、それらの行は整理される。</p>
TRACE_LIMIT	<p>データ・タイプ: INTEGER。 NULL 可能: デフォルトでは不可</p> <p>整理の対象となるまでに行が IBMQREP_CAPTRACE 表の中に留まる分数。整理インターバルのたびに、IBMQREP_CAPTRACE 表にある行が現行タイム・スタンプに基づいてこの制限より古い場合、それらの行は整理される。</p>
SIGNAL_LIMIT	<p>データ・タイプ: INTEGER。 NULL 可能: デフォルトでは不可</p> <p>整理の対象となるまでに行が IBMQREP_SIGNAL 表の中に留まる分数。整理インターバルのたびに、IBMQREP_SIGNAL 表中の行が現行タイム・スタンプに基づいてこの制限より古い場合、それらの行は整理される。</p>
PRUNE_INTERVAL	<p>データ・タイプ: INTEGER。 NULL 可能: デフォルトでは不可</p> <p>Q アプライ・プログラムが不必要になった IBMQREP_CAPMON 表、IBMQREP_CAPQMON 表、IBMQREP_SIGNAL 表および IBMQREP_CAPTRACE 表の行を自動的に整理する頻度 (秒単位)。</p>
SLEEP_INTERVAL	<p>データ・タイプ: INTEGER。 NULL 可能: デフォルトでは不可</p> <p>Q キャプチャー・プログラムが、アクティブ・ログおよびメモリー内に留まっているすべてのトランザクションの処理を行った後、活動停止中の秒数。</p>

表 53. IBMQREP_CAPPARMS 表の列 (続き)

列名	説明
LOGREUSE	<p>データ・タイプ: CHAR(1)。 NULL 可能: デフォルトでは不可</p> <p>Q キャプチャー・プログラムが Q キャプチャー・ログ・ファイルを再利用するか、ファイルに追加するかを示すフラグ。</p> <p>Y Q キャプチャー・プログラムは再始動時にログ・ファイルをクリアして、そのブランク・ファイルに書き込むことで、そのログ・ファイルを再利用します。</p> <p>N Q キャプチャー・プログラムは再始動時に、既存の Q キャプチャー・ログ・ファイルに新しい情報を追加する。</p>
LOGSTDOUT	<p>データ・タイプ: CHAR(1)。 NULL 可能: デフォルトでは不可</p> <p>Q キャプチャー・プログラムがログ・メッセージをログ・ファイル以外の出力に送信するかを示すフラグ。</p> <p>Y Q キャプチャー・プログラムはログ・メッセージをログ・ファイルとコンソール (stdout) の両方に送信する。</p> <p>N Q キャプチャー・プログラムは、ほとんどのログ・ファイル・メッセージをログ・ファイルにのみ送信する。</p> <p>初期化、停止、およびサブスクリプションのアクティブ化または非アクティブ化のメッセージは、コンソール (stdout) とログ・ファイルの両方に送信される。</p>
TERM	<p>データ・タイプ: CHAR(1)。 NULL 可能: デフォルトでは不可</p> <p>DB2 が静止するとき、Q キャプチャー・プログラムが停止するかどうかを示すフラグ。</p> <p>Y DB2 が静止するとき、Q キャプチャー・プログラムが停止する。</p> <p>N DB2 が静止するとき、Q キャプチャー・プログラムを継続して実行する。</p>
CAPTURE_PATH	<p>データ・タイプ: VARCHAR(1040)。 NULL 可能: はい</p> <p>Q キャプチャー・プログラムが作成するファイルが保管されるパス。</p>
ARCH_LEVEL	<p>データ・タイプ: CHAR(4)。 NULL 可能: デフォルトでは不可</p> <p>行にある定義の構造レベル。この列は、行を作成する基礎となった規則を識別する。このレベルは、IBM で定義されていて、バージョン 8.2 でこのレベルは 0802。</p>

関連概念:

- 425 ページの『Q キャプチャー・コントロール表の詳細な構造 -- 概説』

関連資料:

- 234 ページの『Q キャプチャー稼働パラメーターのデフォルト値』
- 235 ページの『Q キャプチャー・パラメーターの説明』
- 421 ページの『Q キャプチャー・サーバーのコントロール表のリスト』

IBMQREP_CAPQMON 表

サーバー: Q キャプチャー・サーバー

デフォルト・スキーマ: ASN

ユニークでない索引: MONITOR_TIME

重要: SQL を使用してこの表を変更しないでください。この表の変更の方法が不適切であると、予期せぬ結果が生じたり、データが失われたりします。

Q キャプチャー・プログラムは、IBMQREP_CAPQMON 表に行を挿入して、送信キューごとのパフォーマンスの統計を記録します。IBMQREP_CAPPARMS 表の MONITOR_INTERVAL に指定した値により、Q キャプチャー・プログラムがこの挿入を実行する頻度を指示します。MONITOR_LIMIT 値は、整理の対象となるまでに、行が IBMQREP_CAPQMON 表の中に留まる分数を示します。

表 54 では、IBMQREP_CAPQMON 表の列の要旨を示します。

表 54. IBMQREP_CAPQMON 表の列

列名	説明
MONITOR_TIME	データ・タイプ: TIMESTAMP 。 NULL 可能: いいえ この表に行が挿入されたときの Q キャプチャー・サーバーのタイム・スタンプ。
SENDQ	データ・タイプ: VARCHAR(97) 。 NULL 可能: いいえ モニター統計のこの行が通知する送信キューの名前。
ROWS_PUBLISHED	データ・タイプ: INTEGER 。 NULL 可能: いいえ Q キャプチャー・プログラムがこの送信キュー経由で送信する (それぞれ挿入、更新または削除操作を行う) 行の数。
TRANS_PUBLISHED	データ・タイプ: INTEGER 。 NULL 可能: いいえ Q キャプチャー・プログラムがこの送信キュー経由で送信するトランザクションの数。
CHG_ROWS_SKIPPED	データ・タイプ: INTEGER 。 NULL 可能: いいえ 変更が Q サブスクリプションまたは XML 発行の一部ではない列に対して加えられるため、この送信キューに書き込まれなかった変更済み行の数 (デフォルトでは、IBMQREP_SUBS 表の ALL_CHANGED_ROWS パラメーターは No に設定されている)。
DELROWS_SUPPRESSED	データ・タイプ: INTEGER 。 NULL 可能: いいえ Q サブスクリプションまたは XML 発行が削除を抑制するオプションを指定して作成されるために、この送信キューに書き込まれなかった行の削除操作の数。
ROWS_SKIPPED	データ・タイプ: INTEGER 。 NULL 可能: いいえ Q サブスクリプションまたは XML 発行で定義された検索条件に合致しなかったため、Q キャプチャー・プログラムがこの送信キューに送信しなかった行の数。

関連概念:

- 283 ページの『Q レプリケーションおよびイベント発行プログラムの履歴およびパフォーマンス・データ』
- 415 ページの『Q レプリケーションおよびイベント発行のコントロール表 -- 概要』

関連資料:

- 421 ページの『Q キャプチャー・サーバーのコントロール表のリスト』

IBMQREP_CAPTRACE 表

サーバー: Q キャプチャー・サーバー

デフォルト・スキーマ: ASN

ユニークでない索引: TRACE_TIME

重要: SQL を使用してこの表を変更しないでください。この表の変更の方法が不適切であると、予期せぬ結果が生じたり、データが失われたりします。

IBMQREP_CAPTRACE 表には、Q キャプチャー・プログラムからの通知、警告、およびエラー・メッセージがあります。

表 55 では、IBMQREP_CAPTRACE 表の列の要旨を示します。

表 55. IBMQREP_CAPTRACE 表の列

列名	説明
OPERATION	<p>データ・タイプ: CHAR(8)。NULL 可能: いいえ</p> <p>Q キャプチャー・プログラムからのメッセージのタイプ :</p> <p>INFO Q キャプチャー・プログラムの動作を説明する。</p> <p>WARNING Q キャプチャー・プログラムのエラー原因となる条件を説明する。</p> <p>ERROR Q キャプチャー・プログラムでのエラーを説明する。</p>
TRACE_TIME	<p>データ・タイプ: TIMESTAMP。NULL 可能: いいえ</p> <p>メッセージを送信キューに書き込んだ Q キャプチャー・サーバーの時刻。</p>
説明	<p>データ・タイプ: VARCHAR(1024)。NULL 可能: いいえ</p> <p>ASN メッセージ ID とメッセージ・テキスト。この列に入れられるテキストは英語のみである。</p>

関連概念:

- 283 ページの『Q レプリケーションおよびイベント発行プログラムの履歴およびパフォーマンス・データ』
- 425 ページの『Q キャプチャー・コントロール表の詳細な構造 -- 概説』

関連資料:

- 421 ページの『Q キャプチャー・サーバーのコントロール表のリスト』

IBMQREP_SRCH_COND 表

サーバー: Q キャプチャー・サーバー

デフォルト・スキーマ: ASN

重要: SQL を使用してこの表を変更しないでください。この表の変更の方法が不適切であると、予期せぬ結果が生じたり、データが失われたりします。

IBMQREP_SRCH_COND は、Q キャプチャー・プログラムが使用する内部表で、Q サブスクリプションまたは、XML 発行の検索条件文節を評価します。

表 56 では、IBMQREP_SRCH_COND 表の列の要旨を示します。

表 56. *IBMQREP_SRCH_COND* 表の列

列名	説明
ASNQREQD	データ・タイプ: CHAR(8)。 NULL 可能: はい この列にはデータが含まれない。

関連概念:

- 425 ページの『Q キャプチャー・コントロール表の詳細な構造 -- 概説』

関連資料:

- 421 ページの『Q キャプチャー・サーバーのコントロール表のリスト』

IBMQREP_SENDQUEUES 表

サーバー: Q キャプチャー・サーバー

デフォルト・スキーマ: ASN

主キー: SENDQ

ユニーク索引: PUBQMAPNAME

重要: SQL を使用してこの表を変更しないでください。この表の変更の方法が不適切であると、予期せぬ結果が生じたり、データが失われたりします。

IBMQREP_SENDQUEUES 表には、Q キャプチャー・プログラムがデータや通知メッセージを送信するときに使用する、WebSphere MQ キューに関する情報が含まれています。それぞれの Q キャプチャー・プログラムのインスタンスは、複数の送信キューの処理が可能です。送信キューはそれぞれ IBMQREP_SENDQUEUES 表で一意的に定義します。

435 ページの表 57 では、IBMQREP_SENDQUEUES 表の列の要旨を示します。

表 57. *IBMQREP_SENDQUEUES* 表の列

列名	説明
PUBQMAPNAME	<p>データ・タイプ: VARCHAR(128)。 NULL 可能: いいえ</p> <p>この送信キューのある発行キュー・マップの名前。 Q サブスクリプションの場合、この名前は <i>IBMQREP_RECVQUEUES</i> 表の <i>REPQMAPNAME</i> の値と一致していなければならない。</p>
SENDQ	<p>データ・タイプ: VARCHAR(48)。 NULL 可能: いいえ</p> <p>この送信キューのユニーク名。この名前は、リモート・キューのローカル定義または、ローカル・キューを表すことができる。キュー名にブランクは使用できない。</p>
RECVQ	<p>データ・タイプ: VARCHAR(48)。 NULL 可能: はい</p> <p>この Q サブスクリプションの受信キューの名前。これは、Q アプライ・サーバーがあるローカル・キューのことである。キュー名にブランクは使用できない。</p>
説明	<p>データ・タイプ: VARCHAR(254)。 NULL 可能: はい</p> <p>この送信キューのある発行キュー・マップの、ユーザー提供の説明。</p>
MESSAGE_FORMAT	<p>データ・タイプ: CHAR(1)。 NULL 可能: デフォルトでは不可</p> <p>送信キューに書き込むメッセージをエンコードするために使用されるフォーマット。</p> <p>C (デフォルト) Q キャプチャー・プログラムは、Q アプライ・プログラムで読み取られるように設計された圧縮形式のソース・データベースから、トランザクションをエンコードする。</p> <p>X Q キャプチャー・プログラムは、XML フォーマットのソース・データベースから、トランザクションをエンコードする。</p>
MSG_CONTENT_TYPE	<p>データ・タイプ: CHAR(1)。 NULL 可能: デフォルトでは不可</p> <p>キューに書き込まれるメッセージにデータベース・トランザクション全体が含まれるか、それとも行の操作のみが含まれるかを示すフラグ。</p> <p>T (デフォルト) メッセージには DB2 トランザクション内のすべての行操作 (更新、挿入、および削除) およびそのトランザクションに関する情報がある。</p> <p>R メッセージには、単一の更新、挿入、削除操作とともに、属している DB2 トランザクションに関する情報が含まれる。</p>
STATE	<p>データ・タイプ: CHAR(1)。 NULL 可能: デフォルトでは不可</p> <p>送信キューの状態を示す、Q キャプチャー・プログラムが挿入するフラグ。</p> <p>A (デフォルト) アクティブ。トランザクションはワーカー・スレッドによりこのキューに書き込まれている。最新のコミット済みトランザクションがキューに書き込まれている。</p> <p>I 非アクティブ。このキューで重大エラーが発生した。</p>

表 57. *IBMQREP_SENDQUEUES* 表の列 (続き)

列名	説明
STATE_TIME	<p>データ・タイプ: TIMESTAMP。 NULL 可能: デフォルトでは不可</p> <p>送信キューに最後に変更を行ったときの Q キャプチャー・サーバーのタイム・スタンプ。デフォルト: 現在のタイム・スタンプ</p>
STATE_INFO	<p>データ・タイプ: CHAR(8)。 NULL 可能: はい</p> <p>キューの状態を示す ASN メッセージの数。詳細については、<i>IBMQREP_CAPTRACE</i> 表または、Q キャプチャー診断ログを参照。</p>
ERROR_ACTION	<p>データ・タイプ: CHAR(1)。 NULL 可能: デフォルトでは不可</p> <p>送信キューがメッセージを受信できなくなると、Q キャプチャー・プログラムがとるべき手段を通知するフラグ。たとえば、キューがいっぱいになっていたり、キュー・マネージャーがこのキューの重大エラーであると報告しているなどの場合がある。</p> <p>I (デフォルト)</p> <p>Q キャプチャー・プログラムは、このキューのすべての Q サブスクリプションおよび XML 発行を無効にするが、別のキューへのメッセージの書き込みを継続する。</p> <p>S</p> <p>このキューでエラーを検出すると、Q キャプチャー・プログラムは停止する。</p>
HEARTBEAT_INTERVAL	<p>データ・タイプ: INTEGER。 NULL 可能: デフォルトでは不可</p> <p>Q キャプチャー・プログラムがこのキューにメッセージを送信して、発行する変更がないときに Q キャプチャー・プログラムが実行中であることを Q アプライ・プログラムまたはユーザー・アプリケーションに通知する頻度 (秒単位)。この値は、<i>COMMIT_INTERVAL</i> 値の倍数でなければならず、そうでない場合、倍数に最も近い値に丸められる (<i>COMMIT_INTERVAL</i> は <i>IBMQREP_CAPPARMS</i> 表で設定されている)。値 0 (デフォルト) は、Q キャプチャー・プログラムに、heartbeat メッセージを送信しないことを通知する。</p> <p>注: <i>IBMQREP_SENDQUEUES</i> 表で定義される <i>HEARTBEAT_INTERVAL</i> は、WebSphere MQ チャンネルに定義可能な <i>HBINT</i> (ハートビート・インターバル) パラメーターとは異なる。詳細については、「<i>WebSphere MQ MQSC</i> コマンド・リファレンス」を参照。</p>
MAX_MESSAGE_SIZE	<p>データ・タイプ: INTEGER。 NULL 可能: デフォルトでは不可</p> <p>この送信キューにメッセージを送信するときに使用するバッファの最大サイズ (KB)。このバッファのサイズは、メッセージが入っているすべてのキューに定義される WebSphere MQ 最大メッセージ長 (<i>MAXMSGL</i>) 属性より大きくなってはならない。そうなるとこの送信キューを使用するすべての Q サブスクリプションおよび XML 発行は無効にされる。デフォルト: 64 KB</p>
APPLY_SERVER	<p>データ・タイプ: VARCHAR(18)。 NULL 可能: はい</p> <p>Q アプライ・プログラムが実行しており、ターゲットを定義しているデータベースの名前。z/OS の場合は、ロケーション名となる。</p>
APPLY_ALIAS	<p>データ・タイプ: VARCHAR(8)。 NULL 可能: はい</p> <p><i>APPLY_SERVER</i> 列で命名された Q キャプチャー・サーバーに対応する DB2 データベース別名。</p>

表 57. *IBMQREP_SENDQUEUES* 表の列 (続き)

列名	説明
APPLY_SCHEMA	<p>データ・タイプ: VARCHAR(128)。NULL 可能: はい</p> <p>この送信キューからトランザクションをアプライしている Q アプライ・プログラムのスキーマ。</p>

関連概念:

- 62 ページの『WebSphere MQ オブジェクトの設定要件』
- 13 ページの『発行キュー・マップ』
- 6 ページの『レプリケーション・キュー・マップ』
- 425 ページの『Q キャプチャー・コントロール表の詳細な構造 -- 概説』

関連タスク:

- 180 ページの『発行キュー・マップの作成』
- 95 ページの『レプリケーション・キュー・マップの作成』

関連資料:

- 421 ページの『Q キャプチャー・サーバーのコントロール表のリスト』

IBMQREP_SIGNAL 表

サーバー: Q キャプチャー・サーバー

デフォルト・スキーマ: ASN

主キー: SIGNAL_TIME

この表の情報は、SQL を使って更新できます。

IBMQREP_SIGNAL 表を使用すると、ユーザー、ユーザー・アプリケーション、または Q アプライ・プログラムが、Q キャプチャー・プログラムと通信できます。ユーザーまたはユーザー・アプリケーションは、IBMQREP_SIGNAL 表に行を挿入して、Q キャプチャー・プログラムがソース表のログから変更のキャプチャーの開始や、Q サブスクリプションの非アクティブ化や、トランザクションの無視といった別の動作をとることを要求できます。Q アプライ・プログラムまたはユーザー・アプリケーションは、Q キャプチャー・プログラムに制御メッセージに送信して同一の要求を行い、対応するシグナルを IBMQREP_SIGNAL 表に挿入することができます。Q キャプチャー・プログラムは、IBMQREP_SIGNAL 表に挿入するログ・レコードを読み取ってから、シグナルを受信します。

この表の中で、SIGNAL_STATE 値が C (完全) のレコード、または整理の対象となるタイム・スタンプを持つレコードは、Q キャプチャー・プログラムによる整理が行われると削除されます。

438 ページの表 58 では、IBMQREP_SIGNAL 表の列の要旨を示します。

表 58. *IBMQREP_SIGNAL* 表の列

列名	説明
SIGNAL_TIME	<p>データ・タイプ: TIMESTAMP。 NULL 可能: デフォルトでは不可</p> <p>行を一意的に識別するために使用されるタイム・スタンプ。Q キャプチャー・プログラムはこの値を使用して、Q キャプチャー・シグナルの処理が終了した時刻を示す、シグナル表の中の正しい行を検出する。デフォルト: 現在のタイム・スタンプ</p>
SIGNAL_TYPE	<p>データ・タイプ: VARCHAR(30)。 NULL 可能: いいえ</p> <p>通知されたシグナルのタイプを示すフラグ。</p> <p>CMD 管理コマンド、asnlccmd、レプリケーション・センター、または別のアプリケーションにより挿入される行。使用可能なシグナルのサブタイプのリストについては、SIGNAL_SUBTYPE 列を参照。</p> <p>USER ユーザーから通知されたシグナル。Q キャプチャー・プログラムは、SIGNAL_LSN 列を、シグナルを挿入したときのログ・シーケンス番号で更新する。Q キャプチャー・プログラムはさらに、SIGNAL_STATE 列の値を保留 (P) から受信 (R) に更新する。</p>
SIGNAL_SUBTYPE	<p>データ・タイプ: VARCHAR(30)。 NULL 可能: はい</p> <p>Q キャプチャー・プログラムが行う CMD 型シグナルが要求する動作のタイプ。</p> <p>CAPSTART Q サブスクリプションまたは XML 発行の変更のキャプチャーを開始する。</p> <p>CAPSTOP Q サブスクリプションまたは XML 発行の変更のキャプチャーを停止する。</p> <p>QINERROR IBMQREP_SENDQUEUES 表で受信キューに定義されたエラー・アクションを実行する。</p> <p>LOADDONE Q アプライ・プログラムまたはユーザー・アプリケーションからのこのシグナルの受信を確認する。LOADDONE シグナルは、ターゲット表をロードした Q キャプチャー・プログラムを通知する。</p> <p>STOP 変更のキャプチャーを停止し、終了する。</p> <p>IGNORETRANS このシグナルが含まれている DB2 トランザクションを無視する。</p> <p>REINIT_SUB IBMQREP_SUBS 表、IBMQREP_SRC_COLS表、および IBMQREP_SENDQUEUES 表にある最新の値を使用している 1 つの Q サブスクリプションまたは XML 発行を、非アクティブにしてからアクティブにする。このシグナルは、ターゲットに新規のロードを行うことを要求しない。</p>

表 58. *IBMQREP_SIGNAL* 表の列 (続き)

列名	説明
SIGNAL_SUBTYPE	ADDCOL 1 つの列を、アクティブな単一方向の Q サブスクリプションまたは XML 発行に追加する。
(続き)	P2PNEW2MEMB ピアツーピア Q サブスクリプションの初期化に使用される内部シグナル。このシグナルは新規サーバーで挿入される。
	P2PMEMB2NEW ピアツーピア Q サブスクリプションの初期化に使用される内部シグナル。このシグナルはアクティブなサーバーで挿入される。
	P2PMEMB2INIT ピアツーピア Q サブスクリプションの初期化に使用される内部シグナル。このシグナルはアクティブなサーバーで挿入される。
	P2PSPOOLING ピアツーピア Q サブスクリプションの初期化に使用される内部シグナル。このシグナルは、新規サブスクリプションを開始するサーバーで挿入される。
	P2PLOADDONE ピアツーピア Q サブスクリプションの初期化に使用される内部シグナル。このシグナルは新規サーバーで挿入される。
	P2PSUBSTOP ピアツーピア Q サブスクリプションの非アクティブ化に使用される内部シグナル。このシグナルは、非アクティブ化されているサーバーで挿入される。
	P2PSUBSTOPPING ピアツーピア Q サブスクリプションの非アクティブ化に使用される内部シグナル。このシグナルは残りのアクティブなサーバーで挿入される。
	P2PREADYTOSTOP ピアツーピア Q サブスクリプションの非アクティブ化に使用される内部シグナル。このシグナルは、非アクティブ化されているサーバーで挿入される。
	P2PNORECAPTURE Q アプライ・プログラムが挿入して、Q キャプチャー・プログラムが変更を再キャプチャーしないようにするシグナル。双方向レプリケーションで使用する。

表 58. IBMQREP_SIGNAL 表の列 (続き)

列名	説明
SIGNAL_INPUT_IN	<p>データ・タイプ: VARCHAR(500)。 NULL 可能: はい</p> <p>SIGNAL_TYPE = USER の場合、この列は、ユーザー定義の入力を保持する。 SIGNAL_TYPE=CMD の場合、この値は以下のように SIGNAL_SUBTYPE 値によって変わる。</p> <p>CMD + CAPSTART Q サブスクリプションまたは XML 発行名。</p> <p>CMD + CAPSTOP Q サブスクリプションまたは XML 発行名。</p> <p>CMD + LOADDONE Q サブスクリプションまたは XML 発行名。</p> <p>CMD + STOP NULL (値は不要)。</p> <p>CMD + IGNORETRANS NULL (値は不要)。</p> <p>CMD + QINERROR ユーザー・アプリケーションの場合、エラー状態の送信キュー名。 Q アプリケーション・プログラムの場合、エラー状態の送信キューの名前、ASN メッセージ番号、およびスペースで区切ったトークン。</p> <p>CMD + REINIT_SUB Q サブスクリプションまたは XML 発行名。</p> <p>CMD + ADDCOL セミコロンで区切った、Q サブスクリプションまたは XML 発行名および列名。たとえば、QSUB1;COL10 のようになる。</p>
SIGNAL_STATE	<p>データ・タイプ: CHAR(1)。 NULL 可能: デフォルトでは不可</p> <p>シグナルの状況を示すフラグ。</p> <p>P (デフォルト) シグナルは保留中。キャプチャー・プログラムはまだシグナルを受け取っていない。</p> <p>R Q キャプチャー・プログラムはシグナルを受け取った。</p> <p>C Q キャプチャー・プログラムはシグナルの処理を完了した。</p> <p>F シグナルは失敗した。たとえば、Q キャプチャー・プログラムは、Q サブスクリプションまたは XML 発行に障害があるため、CAPSTART を処理できない。</p>
SIGNAL_LSN	<p>データ・タイプ: CHAR(10) FOR BIT DATA。 NULL 可能: いいえ</p> <p>SIGNAL 表に挿入するログ・レコードのログ・シーケンス番号。</p>

関連概念:

- 163 ページの『Q レプリケーションのターゲット表のロードのオプション -- 概要』
- 425 ページの『Q キャプチャー・コントロール表の詳細な構造 -- 概説』

関連タスク:

- 219 ページの『XML 発行の属性の変更』
- 221 ページの『既存の XML 発行への列の追加』
- 205 ページの『Q サブスクリプションの属性の変更』
- 207 ページの『既存の Q サブスクリプションへの列の追加』
- 253 ページの『Q サブスクリプションまたは XML 発行のアクティブ化』
- 254 ページの『Q サブスクリプションまたは XML 発行の非アクティブ化』
- 256 ページの『Q キャプチャー・プログラムの停止』

関連資料:

- 421 ページの『Q キャプチャー・サーバーのコントロール表のリスト』

IBMQREP_SRC_COLS 表

サーバー: Q キャプチャー・サーバー

デフォルト・スキーマ: ASN

主キー: SUBNAME、SRC_COLNAME

重要: SQL を使用してこの表を変更しないでください。この表の変更の方法が不適切であると、予期せぬ結果が生じたり、データが失われたりします。

IBMQREP_SRC_COLS 表では、キャプチャーされた変更をソース表で列にリストします。

表 59 では、IBMQREP_SRC_COLS 表の列の要旨を示します。

表 59. IBMQREP_SRC_COLS 表の列

列名	説明
SUBNAME	データ・タイプ: VARCHAR(30)。NULL 可能: いいえ このソース表の Q サブスクリプションまたは、XML 発行の名前。
SRC_COLNAME	データ・タイプ: VARCHAR(30)。NULL 可能: いいえ キャプチャーされる変更を示すソース表の列の名前。
IS_KEY	データ・タイプ: SMALLINT。NULL 可能: デフォルトでは不可 列がレプリケーションまたは発行に使用されるキーの一部であるかどうかを示すフラグ。ソースで固有な列セットはすべて使用できる。 0 (デフォルト) この列は、ユニーク・キーの一部ではない。トランザクション・メッセージでの配列は、ソース表での配列と同一である。 n この列は、ユニーク・キーの一部である。複数列のキーの場合、トランザクション・メッセージでの配列は、ユーザーの指定した数 <i>n</i> に基づいてエンコードされる。 ソース表からの列の少なくとも 1 つは、IBMQREP_SRC_COLS 表で 0 よりも大きい値でなければならず、そうでないなら、Q がサブスクリプションまたは、XML 発行が無効になる。

関連概念:

- 112 ページの『ターゲットの索引列またはキー列 (単一方向レプリケーション)』
- 425 ページの『Q キャプチャー・コントロール表の詳細な構造 -- 概説』

関連資料:

- 421 ページの『Q キャプチャー・サーバーのコントロール表のリスト』

IBMQREP_SUBS 表

サーバー: Q キャプチャー・サーバー

デフォルト・スキーマ: ASN

主キー: SUBNAME

重要: SQL を使用してこの表を変更しないでください。この表の変更の方法が不適切であると、予期せぬ結果が生じたり、データが失われたりします。

IBMQREP_SUBS 表には、Q サブスクリプションまたは XML 発行の情報が入っており、これは、サブスクリプションのタイプ、検索条件、データ送信オプション、ロード・オプション、およびサブスクリプションの状態などが含まれます。

表 60 では、IBMQREP_SUBS 表の列の要旨を示します。

表 60. IBMQREP_SUBS 表の列

列名	説明
SUBNAME	データ・タイプ: VARCHAR(132)。 NULL 可能: いいえ Q サブスクリプションまたは XML 発行名。 Q キャプチャー・プログラムの個々のインスタンスに対して、Q サブスクリプションまたは XML 発行名は固有でなければならない。
SOURCE_OWNER	データ・タイプ: VARCHAR(30)、DB2 UDB for z/OS バージョン 8 新機能モードの場合 VARCHAR(128)。 NULL 可能: いいえ この Q サブスクリプションまたは XML 発行のソース表があるスキーマ名または高位修飾子。
SOURCE_NAME	データ・タイプ: VARCHAR(128)、DB2 UDB for z/OS バージョン 7 およびバージョン 8 互換モードの場合 VARCHAR(18)。 NULL 可能: いいえ この Q サブスクリプションまたは XML 発行のソース表名。
TARGET_SERVER	データ・タイプ: VARCHAR(18)。 NULL 可能: はい Q アプライ・プログラムが実行しており、ターゲットを定義しているデータベースまたはサブシステムの名前。 z/OS の場合は、ロケーション名となる。
TARGET_ALIAS	データ・タイプ: VARCHAR(8)。 NULL 可能: はい TARGET_SERVER 列で命名された Q アプライ・サーバーに対応する DB2 データベース別名。

表 60. IBMQREP_SUBS 表の列 (続き)

列名	説明
TARGET_OWNER	<p>データ・タイプ: VARCHAR(30)、DB2 UDB for z/OS バージョン 8 新機能モードの場合 VARCHAR(128)。 NULL 可能: はい</p> <p>Q サブスクリプションのターゲット表またはストアード・プロシージャのスキーマ名または高位修飾子。</p>
TARGET_NAME	<p>データ・タイプ: VARCHAR(128)、DB2 UDB for z/OS バージョン 7 およびバージョン 8 互換モードの場合 VARCHAR(18)。 NULL 可能: はい</p> <p>Q サブスクリプションのターゲット表の名前。</p>
TARGET_TYPE	<p>データ・タイプ: INTEGER。 NULL 可能: はい</p> <p>レプリケーション・ターゲットのタイプを示すフラグ。</p> <p>1 ユーザー表</p> <p>2 将来用に予約済み。</p> <p>3 将来用に予約済み。</p> <p>4 将来用に予約済み。</p> <p>5 ストアード・プロシージャ</p>
APPLY_SCHEMA	<p>データ・タイプ: VARCHAR(128)。 NULL 可能: はい</p> <p>この Q サブスクリプションにトランザクションをアプライしている Q アプライ・プログラムのスキーマ。</p>
SENDQ	<p>データ・タイプ: VARCHAR(48)。 NULL 可能: いいえ</p> <p>Q キャプチャー・プログラムがこの Q サブスクリプションまたは XML 発行のトランザクション・データの送信に使用する WebSphere MQ キューの名前。それぞれのソース表に対して 1 つの送信キューがペアに割り当てられる。</p>
SEARCH_CONDITION	<p>データ・タイプ: VARCHAR(2048)。 NULL 可能: はい</p> <p>Q サブスクリプションまたは XML 発行で行をフィルタリングするために使用する検索条件。これは、直前にコロンを 1 つ付加したソース列名を持つ where 文節でなければならない。</p>
SUB_ID	<p>データ・タイプ: INTEGER。 NULL 可能: はい</p> <p>Q キャプチャー・プログラムが生成する整数で、サブスクリプション・スキーマ・メッセージ内で Q サブスクリプションと Q アプライ・プログラムを一意に識別するために使用する。</p>
SUBTYPE	<p>データ・タイプ: CHAR(1)。 NULL 可能: デフォルトでは不可</p> <p>Q サブスクリプションが関わっているレプリケーションのタイプ、またはこれが XML 発行であるかどうかを示すフラグ。</p> <p>U (デフォルト) 単一方向レプリケーション。これは XML 発行に使用する値である。</p> <p>B 双方向レプリケーション。</p> <p>P ピアツーピア・レプリケーション。</p>

表 60. IBMQREP_SUBS 表の列 (続き)

列名	説明
ALL_CHANGED_ROWS	<p>データ・タイプ: CHAR(1)。 NULL 可能: デフォルトでは不可</p> <p>以下は、Q サブスクリプションの一部である列のいずれにも変更がなかった場合でも、ソース表の行に変更があったときに Q キャプチャー・プログラムがメッセージを発行するかどうかを示すフラグ。</p> <p>N (デフォルト) Q キャプチャー・プログラムは、Q サブスクリプションの一部である列が変更された場合にのみ、メッセージを送信する。</p> <p>Y Q サブスクリプションの一部である列のいずれにも変更がなかった場合でも、ソース表の行に変更があると、Q キャプチャー・プログラムはその列を送信する。</p>
BEFORE_VALUES	<p>データ・タイプ: CHAR(1)。 NULL 可能: デフォルトでは不可</p> <p>更新操作ではこのフラグが、Q キャプチャー・プログラムが非キー列の変更後の値に加え、変更前の値を送信するかどうかを示す。削除ではこのフラグが、Q キャプチャー・プログラムがキー列の変更前の値に加え、非キー列の変更前の値を送信するかどうかを示す。</p> <p>N (デフォルト) Q キャプチャー・プログラムは、変更する非キー列の変更前の値を送信しない。キー列を変更すると、Q キャプチャー・プログラムは、キー列の変更前および変更後の値を両方送信する。キー列を含む DELETE ステートメントでは、変更前の値のみが送信される。</p> <p>Y Q サブスクリプションまたは XML 発行の一部であるソース表の非キー列に変更があると、Q キャプチャー・プログラムは変更前の値および変更後の値の両方を送信する。</p>
CHANGED_COLS_ONLY	<p>データ・タイプ: CHAR(1)。 NULL 可能: デフォルトでは不可</p> <p>列に変更があった場合にのみ、Q キャプチャー・プログラムが Q サブスクリプションまたは XML 発行の一部である列を発行するかどうかを示すフラグ。このフィールドは更新操作専用である。</p> <p>Y (デフォルト) Q キャプチャー・プログラムが更新済み行を送信すると、Q サブスクリプションまたは XML 発行の一部である変更された列のみを送信する。</p> <p>N 列が変更された場合はいつでも、Q キャプチャー・プログラムは Q サブスクリプションまたは XML 発行の一部である行にあるすべての列を送信する。</p>

表 60. IBMQREP_SUBS 表の列 (続き)

列名	説明
HAS_LOADPHASE	<p>データ・タイプ: CHAR(1)。 NULL 可能: デフォルトでは不可</p> <p>以下は Q サブスクリプションまたは XML 発行のターゲット表がソースからデータをロードするかどうかを示すフラグ。</p> <p>N (デフォルト) ターゲットはロードされない。</p> <p>I 自動ロード。 Q アプライ・プログラムは、IBMQREP_TARGETS 表で指定された LOAD_TYPE と、Q アプライ・サーバーおよび Q キャプチャー・サーバーのプラットフォームに応じて、CURSOR からの LOAD ユーティリティ、EXPORT/IMPORT ユーティリティ、または EXPORT/LOAD ユーティリティから呼び出します。</p> <p>E 手動ロード。 Q アプライ・プログラム以外のアプリケーションがターゲット表をロードする。この場合、ユーザーまたはレプリケーション・センターが LOADDONE シグナルを Q キャプチャー・プログラムで IBMQREP_SIGNAL 表に挿入するか、load done (ロード完了) メッセージの受信後に、Q キャプチャー・プログラムがこのシグナルを挿入する。</p>

表 60. IBMQREP_SUBS 表の列 (続き)

列名	説明
STATE	<p>データ・タイプ: CHAR(1)。 NULL 可能: デフォルトでは不可</p> <p>現在の Q サブスクリプションまたは XML 発行の状態を示す Q キャプチャー・プログラムが挿入するフラグ。初期状態は新規であり、STATE_INFO フィールドを最初に ASN7024I (新規の Q サブスクリプションまたは XML 発行) に設定する。</p> <p>N (デフォルト) Q サブスクリプションまたは XML 発行は新規である。Q キャプチャー・プログラムは、プログラムの開始時または再初期化時に自動的にこの Q サブスクリプションまたは XML 発行をアクティブにする。</p> <p>I Q サブスクリプションまたは、XML 発行は非アクティブである。Q キャプチャー・プログラムは、ログで CAPSTOP シグナルを確認している。そうでない場合、エラーが発生し、Q サブスクリプションまたは XML 発行が非アクティブになる。Q キャプチャー・プログラムは、この Q サブスクリプションまたは XML 発行のメッセージ送信を停止したが、それ以外は続行した。</p> <p>L Q サブスクリプションがロード中である。Q キャプチャー・プログラムは、CAPSTART シグナルを処理し、Q アプライ・プログラムまたは、ユーザー・アプリケーションにサブスクリプション・スキーマ・メッセージを送信した。Q キャプチャー・プログラムは、すべての列の変更前の値が含まれるトランザクション・メッセージを送信し、LOADDONE シグナルを待機している。</p> <p>A Q サブスクリプションまたは XML 発行はアクティブである。ロード段階で、Q キャプチャー・プログラムは、LOADDONE シグナルを処理し、Q アプライ・プログラムまたはユーザー・アプリケーションにロード完了受信メッセージを送信した。Q キャプチャー・プログラムは、Q サブスクリプションまたは XML 発行に定義したオプションに基づいてデータ・メッセージを送信している。</p> <p>T Q キャプチャー・プログラムが、このピアツーピア Q サブスクリプションのログで CAPSTART シグナルを読み取り、Q サブスクリプションがピアツーピア・グループ内で初期化されることを示す内部状態。</p> <p>G Q キャプチャー・プログラムが、このピアツーピア Q サブスクリプションのログで CAPSTOP シグナルを読み取り、Q サブスクリプションがピアツーピア・グループ内で非アクティブになることを示す内部状態。</p>
STATE_TIME	<p>データ・タイプ: TIMESTAMP。 NULL 可能: デフォルトでは不可</p> <p>Q サブスクリプションまたは XML 発行の状態の最終変更タイム・スタンプ。デフォルト: 現在のタイム・スタンプ</p>
STATE_INFO	<p>データ・タイプ: CHAR(8)。 NULL 可能: はい</p> <p>Q サブスクリプションの状態を示す ASN メッセージの数。詳細については、IBMQREP_CAPTRACE 表または、Q キャプチャー診断ログを参照する。</p>
STATE_TRANSITION	<p>データ・タイプ: VARCHAR(256) FOR BIT DATA。 NULL 可能: いいえ</p> <p>部分的な状態および関連情報の保存に使用する内部値。</p>

表 60. IBMQREP_SUBS 表の列 (続き)

列名	説明
SUBGROUP	<p>データ・タイプ: VARCHAR(30)。 NULL 可能: はい</p> <p>この Q サブスクリプションに含まれるピアツーピア・レプリケーション・グループの名前。この列は XML 発行には適用されない。</p>
SOURCE_NODE	<p>データ・タイプ: SMALLINT。 NULL 可能: デフォルトでは不可</p> <p>ピアツーピア Q サブスクリプションでのソース・サーバーの識別番号。この列は XML 発行には適用されない。デフォルト: 0</p>
TARGET_NODE	<p>データ・タイプ: SMALLINT。 NULL 可能: デフォルトでは不可</p> <p>ピアツーピア Q サブスクリプションでのターゲット・サーバーの識別番号。この列は XML 発行には適用されない。デフォルト: 0</p>
GROUP_MEMBERS	<p>データ・タイプ: CHAR(254) FOR BIT DATA。 NULL 可能: はい</p> <p>メンバーがピアツーピア・グループに参加または離脱した場合、この列は Q キャプチャー・プログラムによって更新される。</p>
OPTIONS_FLAG	<p>データ・タイプ: CHAR(4) FOR BIT DATA。 NULL 可能: デフォルトでは不可</p> <p>将来用に予約済み。</p>
SUPPRESS_DELETES	<p>データ・タイプ: CHAR(1)。 NULL 可能: デフォルトでは不可</p> <p>ソース表から削除された行を送信するかどうか Q キャプチャー・プログラムに通知するフラグ。</p> <p>N (デフォルト) 削除行を送信しない。</p> <p>Y 削除行を送信する。</p>
説明	<p>データ・タイプ: VARCHAR(254)。 NULL 可能: はい</p> <p>Q サブスクリプションまたは XML 発行のユーザー提供の説明。</p>

関連概念:

- 189 ページの『XML 発行の行をフィルター処理する検索条件』
- 188 ページの『Q キャプチャー・プログラムが XML 発行用のメッセージを発行する時期』
- 194 ページの『XML 発行用のメッセージに変更前の値を組み込むためのオプション』
- 193 ページの『XML 発行用のメッセージに未変更の列を組み込むためのオプション』
- 5 ページの『Q サブスクリプション』
- 108 ページの『行のフィルタリングのための検索条件 (単一方向レプリケーション)』
- 106 ページの『Q キャプチャー・プログラムがメッセージを送信する頻度 (単一方向レプリケーション)』
- 163 ページの『Q レプリケーションのターゲット表のロードのオプション -- 概要』

- 425 ページの『Q キャプチャー・コントロール表の詳細な構造 -- 概説』
- 12 ページの『XML 発行』

関連資料:

- 421 ページの『Q キャプチャー・サーバーのコントロール表のリスト』

Q アプライ・コントロール表の詳細な構造

Q アプライ・コントロール表の詳細な構造 -- 概説

以下のトピックでは、Q アプライ・サーバー上の各コントロール表について詳しく説明しています。コントロール表はアルファベット順にリストされます。各表にある列は、左から右に、出現順にリストされています。

- 『IBMQREP_APPENQ 表』
- 449 ページの『IBMQREP_APPLYMON 表』
- 452 ページの『IBMQREP_APPLYPARMS 表』
- 454 ページの『IBMQREP_APPLYTRACE 表』
- 455 ページの『IBMQREP_DELTOMB 表』
- 456 ページの『IBMQREP_DONEMSG 表』
- 457 ページの『IBMQREP_EXCEPTIONS 表』
- 459 ページの『IBMQREP_RECVQUEUES 表』
- 461 ページの『IBMQREP_SAVERI 表』
- 462 ページの『IBMQREP_SPILLEDROW 表』
- 463 ページの『IBMQREP_SPILLQS 表』
- 464 ページの『IBMQREP_TRG_COLS 表』
- 465 ページの『IBMQREP_TARGETS 表』

関連概念:

- 415 ページの『Q レプリケーションおよびイベント発行のコントロール表 -- 概要』
- 425 ページの『Q キャプチャー・コントロール表の詳細な構造 -- 概説』
- 471 ページの『モニター・コントロール表の詳細な構造 -- 概説』

IBMQREP_APPENQ 表

サーバー: Q アプライ・サーバー

デフォルト・スキーマ: ASN

重要: SQL を使用してこの表を変更しないでください。この表の変更の方法が不適切であると、予期せぬ結果が生じたり、データが失われたりします。

IBMQREP_APPENQ 表を使用して、以下のことを確認します。

- DB2 UDB for Linux、DB2 UDB for UNIX および DB2 UDB for Windows の場合は、1 つのデータベースに対して既知のスキーマを持つ 1 つの Q アプライ・プログラムのみが実行される。

- 非データ共有 DB2 UDB for z/OS の場合は、1 つのサブシステムに対して既知のスキーマを持つ 1 つの Q アプライ・プログラムのみが実行される。
- データ共有 DB2 UDB for z/OS の場合は、1 つのデータ共有グループに対して既知のスキーマを持つ 1 つの Q アプライ・プログラムのみが実行される。

Q アプライ・プログラムは実行中、この表を排他的にロックします。Q アプライ・プログラムを 2 回開始すると、2 番目のインスタンスはこの表をロック待機している状態になります。その表は、空として作成されます。

表 61 では、IBMQREP_APPENQ 表の列の要旨を示します。

表 61. IBMQREP_APPENQ 表の列

列名	説明
LOCKNAME	データ・タイプ: INTEGER。NULL 可能: はい この列にはデータが含まれない。

関連概念:

- 448 ページの『Q アプライ・コントロール表の詳細な構造 -- 概説』

関連資料:

- 422 ページの『Q アプライ・サーバーのコントロール表のリスト』

IBMQREP_APPLYMON 表

サーバー: Q アプライ・サーバー

デフォルト・スキーマ: ASN

ユニークでない索引: MONITOR_TIME

重要: SQL を使用してこの表を変更しないでください。この表の変更の方法が不適切であると、予期せぬ結果が生じたり、データが失われたりします。

Q アプライ・プログラムは周期的に IBMQREP_APPLYMON 表に受信キューごとに 1 行を挿入して、パフォーマンス統計を記録します。IBMQREP_APPLYPARMS 表の MONITOR_INTERVAL に値を指定し、Q アプライ・プログラムがこの行を挿入する回数を決定します。MONITOR_LIMIT 値は、整理の対象となるまでに、モニター表内に残っている行の数を決定します。

表 62 では、IBMQREP_APPLYMON 表の列の要旨を示します。

表 62. IBMQREP_APPLYMON 表の列

列名	説明
MONITOR_TIME	データ・タイプ: TIMESTAMP。NULL 可能: いいえ IBMQREP_APPLYMON 表に行が挿入されたときの Q アプライ・サーバーのタイム・スタンプ。
RECVQ	データ・タイプ: VARCHAR(48)。NULL 可能: いいえ Q アプライ・パフォーマンス統計のこの行に関する受信キューの名前。

表 62. IBMQREP_APPLYMON 表の列 (続き)

列名	説明
QSTART_TIME	<p>データ・タイプ: TIMESTAMP。 NULL 可能: いいえ</p> <p>受信キューが開始されたときの Q アプライ・サーバーのタイム・スタンプ。</p>
CURRENT_MEMORY	<p>データ・タイプ: INTEGER。 NULL 可能: いいえ</p> <p>キューからトランザクションを読み取るために使用される Q アプライ・ブラウザー・スレッドのメモリー量 (バイト数)。</p>
QDEPTH	<p>データ・タイプ: INTEGER。 NULL 可能: いいえ</p> <p>キュー項目数 (キューのメッセージ数)。</p>
END2END_LATENCY	<p>データ・タイプ: INTEGER。 NULL 可能: いいえ</p> <p>トランザクションがソース表にコミットされてから、ターゲットにコミットされるまでの平均経過時間 (ミリ秒)。</p>
QLATENCY	<p>データ・タイプ: INTEGER。 NULL 可能: いいえ</p> <p>Q キャプチャー・プログラムが、メッセージを送信キューに書き込んでから、Q アプライ・プログラムが受信キューからメッセージを受け取るまでの平均経過時間 (ミリ秒)。</p>
APPLY_LATENCY	<p>データ・タイプ: INTEGER。 NULL 可能: いいえ</p> <p>Q アプライ・プログラムが受信キューからトランザクションを読み取ってから、そのトランザクションがターゲットにコミットされるまでの平均経過時間 (ミリ秒)。</p>
TRANS_APPLIED	<p>データ・タイプ: INTEGER。 NULL 可能: いいえ</p> <p>Q アプライをターゲットにコミットする受信キューからのトランザクションの合計数。</p>
ROWS_APPLIED	<p>データ・タイプ: INTEGER。 NULL 可能: いいえ</p> <p>Q アプライ・プログラムをターゲットにアプライする受信キューからの挿入操作、更新操作、および削除操作の合計数。</p>
TRANS_SERIALIZED	<p>データ・タイプ: INTEGER。 NULL 可能: いいえ</p> <p>他のトランザクションと競合 (行の競合または参照保全競合のいずれかが理由) しているトランザクションの合計数。この場合、Q アプライ・プログラムは、並列処理を延期し、ソースでコミットされた順番でトランザクション内の行の変更をアプライする。</p>
RI_DEPENDENCIES	<p>データ・タイプ: INTEGER。 NULL 可能: いいえ</p> <p>検出された、トランザクションを直列化する参照保全競合の合計数。</p>
RI_RETRIES	<p>データ・タイプ: INTEGER。 NULL 可能: いいえ</p> <p>必要なトランザクションが並列で実行されたために、参照保全の競合が生じた場合に、Q アプライ・プログラムが行を再アプライして変更しなければならなかった回数。</p>
DEADLOCK_RETRIES	<p>データ・タイプ: INTEGER。 NULL 可能: いいえ</p> <p>ロック・タイムアウトとデッドロックのために Q アプライ・プログラムが行の変更を再アプライした回数。</p>

表 62. IBMQREP_APPLYMON 表の列 (続き)

列名	説明
ROWS_NOT_APPLIED	<p>データ・タイプ: INTEGER。 NULL 可能: いいえ</p> <p>アプライすることができず、IBMQREP_EXCEPTIONS 表に入れられた行の数。</p>
MONSTER_TRANS	<p>データ・タイプ: INTEGER。 NULL 可能: いいえ</p> <p>IBMQREP_RECVQUEUES 表にある受信キュー・セットに対して、MEMORY_LIMIT を超えるトランザクション数。</p>
MEM_FULL_TIME	<p>データ・タイプ: INTEGER。 NULL 可能: いいえ</p> <p>Q アプライ・プログラムのエージェントがトランザクションにアプライするために使用可能なメモリーをすべて使用しているため、受信キューからトランザクションがビルドできないときにかかる合計秒数。</p>
APPLY_SLEEP_TIME	<p>データ・タイプ: INTEGER。 NULL 可能: いいえ</p> <p>処理の待機中にその受信キューに対する Q アプライ・プログラムのエージェントがアイドル状態になっている合計秒数。</p>
SPILED_ROWS	<p>データ・タイプ: INTEGER。 NULL 可能: いいえ</p> <p>ターゲットをロードしている間に、Q アプライ・プログラムが一時予備キューに送信する行の数。</p>
SPILEDROWSAPPLIED	<p>データ・タイプ: INTEGER。 NULL 可能: いいえ</p> <p>ターゲットにアプライされた予備の行数。</p>
OLDEST_TRANS	<p>データ・タイプ: TIMESTAMP。 NULL 可能: いいえ</p> <p>モニターの間隔がコミットされている間に最も古いトランザクションが処理されたときの Q アプライ・サーバーのタイム・スタンプ。</p>
OKSQLSTATE_ERRORS	<p>データ・タイプ: INTEGER。 NULL 可能: いいえ</p> <p>IBMQREP_TARGETS 表の OKSQLSTATES フィールドで許容できると定義された SQL エラーの原因となった行を変更した数。 Q アプライ・プログラムはこのエラーを無視する。</p>
HEARTBEAT_LATENCY	<p>データ・タイプ: INTEGER。 NULL 可能: いいえ</p> <p>ハートビート・メッセージを Q キャプチャー・プログラムが送信してから、Q アプライ・プログラムが受信するまでの平均経過時間 (ミリ秒)。</p>
KEY_DEPENDENCIES	<p>データ・タイプ: INTEGER。 NULL 可能: いいえ</p> <p>検出された、トランザクションを直列化するレプリケーション・キー制約の合計数。</p>
UNIQ_DEPENDENCIES	<p>データ・タイプ: INTEGER。 NULL 可能: いいえ</p> <p>検出された、トランザクションを直列化するユニーク索引制約の合計数。</p>
UNIQ_RETRIES	<p>データ・タイプ: INTEGER。 NULL 可能: いいえ</p> <p>ユニーク索引制約のために並列にアプライされない行に Q アプライ・プログラムが再アプライを試行する回数。</p>

関連概念:

- 283 ページの『Q レプリケーションおよびイベント発行プログラムの履歴およびパフォーマンス・データ』
- 448 ページの『Q アプライ・コントロール表の詳細な構造 -- 概説』

関連資料:

- 422 ページの『Q アプライ・サーバーのコントロール表のリスト』

IBMQREP_APPLYPARMS 表

サーバー: Q アプライ・サーバー

デフォルト・スキーマ: ASN

ユニーク索引: QMGR

この表の情報は、SQL を使って更新できます。

IBMQREP_APPLYPARMS 表は、Q アプライ・プログラムの操作をコントロールするためにユーザーが変更できるパラメーターを保持します。たとえば、ユーザーは Q アプライ・プログラムが使用するスレッドの数を設定して、並列にトランザクションをアプライすることができます。また、整理する前に、Q アプライ・プログラムが IBMQREP_APPMON 表に保存するデータの長さを設定できます。Q アプライ・プログラムは開始時のみ、この表に加えられた変更を読み取ります。

IBMQREP_APPLYPARMS 表の行は、1 行だけです。この表に行がない場合、または複数の行が入っている場合、Q アプライ・プログラムは実行しません。

表 63 では、IBMQREP_APPLYPARMS 表の列の要旨を示します。

表 63. IBMQREP_APPLYPARMS 表の列

列名	説明
QMGR	データ・タイプ: VARCHAR(48)。NULL 可能: いいえ Q アプライ・サーバー上にある WebSphere MQ キュー・マネージャー名。
MONITOR_LIMIT	データ・タイプ: INTEGER。NULL 可能: デフォルトでは不可 整理の対象となるまでに、行が IBMQREP_APPLYMON 表の中に留まる分数。整理インターバルのたびに、IBMQREP_APPLYMON 表にある行が現行タイム・スタンプに基づいてこの制限より古い場合、それらの行は整理される。
TRACE_LIMIT	データ・タイプ: INTEGER。NULL 可能: デフォルトでは不可 整理の対象となるまでに、行が IBMQREP_APPLYTRACE 表の中に留まる分数。整理インターバルのたびに、IBMQREP_APPLYMON 表にある行が現行タイム・スタンプに基づいてこの制限より古い場合、それらの行は整理される。
MONITOR_INTERVAL	データ・タイプ: INTEGER。NULL 可能: デフォルトでは不可 Q アプライ・プログラムが IBMQREP_APPLYMON 表に行を追加する頻度 (秒単位)。
PRUNE_INTERVAL	データ・タイプ: INTEGER。NULL 可能: デフォルトでは不可 Q アプライ・プログラムが IBMQREP_APPLYMON 表と IBMQREP_APPLYTRACE 表の行を自動的に整理する頻度 (秒単位)。

表 63. IBMQREP_APPLYPARMS 表の列 (続き)

列名	説明
AUTOSTOP	<p>データ・タイプ: CHAR(1)。 NULL 可能: デフォルトでは不可</p> <p>すべての受信キューがいったん空になったとき、Q アプライ・プログラムを停止するかどうかを通知するフラグ。</p> <p>Y すべての受信キューがいったん空になったとき、Q アプライ・プログラムは停止する。</p> <p>N すべての受信キューがいったん空になった後も、Q アプライ・プログラムの実行は継続する。</p>
LOGREUSE	<p>データ・タイプ: CHAR(1)。 NULL 可能: デフォルトでは不可</p> <p>Q アプライ・プログラムが Q アプライ・ログ・ファイルを再利用するか、ファイルに追加するかを示すフラグ。</p> <p>Y Q アプライ・プログラムは再始動時にログ・ファイルをクリアして、そのブランク・ファイルに書き込むことで、そのログ・ファイルを再利用する。</p> <p>N Q アプライ・プログラムは再始動時に、既存の Q アプライ・ログ・ファイルに新しい情報を追加する。</p>
LOGSTDOUT	<p>データ・タイプ: CHAR(1)。 NULL 可能: デフォルトでは不可</p> <p>Q アプライ・プログラムがログ・ファイル以外のログ・メッセージを出力に送信するかを示すフラグ。</p> <p>Y Q アプライ・プログラムはログ・メッセージをログ・ファイルとコンソール (stdout) の両方に送信する。</p> <p>N Q アプライ・プログラムは、ほとんどのログ・メッセージをログ・ファイルにのみ送ります。</p> <p>初期化、停止、およびサブスクリプションのアクティブ化または非アクティブ化のメッセージは、パラメーターの設定にかかわらず、コンソール (stdout) とログ・ファイルの両方に送信されます。</p>
APPLY_PATH	<p>データ・タイプ: VARCHAR(1040)。 NULL 可能: はい</p> <p>Q アプライ・プログラムが作成するファイルのパスが保管される。デフォルトでは、これは Q アプライ・プログラムが開始されたディレクトリーです。</p>
ARCH_LEVEL	<p>データ・タイプ: CHAR(4)。 NULL 可能: デフォルトでは不可</p> <p>行にある定義の構造レベル。この列は、行を作成する基礎となった規則を識別する。このレベルは、IBM で定義されていて、バージョン 8.2 は 0802。</p>
TERM	<p>データ・タイプ: CHAR(1)。 NULL 可能: デフォルトでは不可</p> <p>DB2 が静止するとき、Q アプライ・プログラムが停止するかどうかを示すフラグ。</p> <p>Y DB2 が静止するとき、Q アプライ・プログラムが停止する。</p> <p>N DB2 が静止するとき、Q アプライ・プログラムを継続して実行する。</p>

表 63. IBMQREP_APPLYPARMS 表の列 (続き)

列名	説明
PWDFILE	<p>データ・タイプ: CHAR(48)。 NULL 可能: はい</p> <p>Q サブスクリプションがターゲットの内部ロードを要求したとき、Q キャプチャー・プログラムに接続するために Q アプライ・プログラムが使用する暗号化されたパスワード・ファイル。 asnpwd コマンドは、デフォルトでこのファイルを APPLY_PATH 列が保管されているディレクトリーに作成する。</p>
DEADLOCK_RETRIES	<p>データ・タイプ: INTEGER。 NULL 可能: デフォルトでは不可</p> <p>SQL デッドロック後に、Q アプライ・プログラムが変更をターゲット表に再アプライする回数、またはコントロール表に変更を加える回数。Q アプライ・プログラムは、次のデッドロックの再試行まで 1 秒待機する。</p>

関連概念:

- 448 ページの『Q アプライ・コントロール表の詳細な構造 -- 概説』

関連資料:

- 262 ページの『Q アプライ稼働パラメーターのデフォルト値』
- 263 ページの『Q アプライ・パラメーターの説明』
- 422 ページの『Q アプライ・サーバーのコントロール表のリスト』

IBMQREP_APPLYTRACE 表

サーバー: Q アプライ・サーバー

デフォルト・スキーマ: ASN

ユニークでない索引: TRACE_TIME

重要: SQL を使用してこの表を変更しないでください。この表の変更の方法が不適切であると、予期せぬ結果が生じたり、データが失われたりします。

IBMQREP_APPLYTRACE 表には、Q アプライ・プログラムからの通知、警告、およびエラー・メッセージが入ります。IBMQREP_APPLYPARMS 表にある TRACE_LIMIT パラメーターを使用して、この表のオート・プルーンをセットアップできます。

455 ページの表 64 では、IBMQREP_APPLYTRACE 表の列の要旨を示します。

表 64. IBMQREP_APPLYTRACE 表の列

列名	説明
OPERATION	<p>データ・タイプ: CHAR(8)。NULL 可能: いいえ</p> <p>Q アプライ・プログラムからのメッセージのタイプ :</p> <p>INFO Q アプライ・プログラムの動作を説明する。</p> <p>WARNING Q アプライ・プログラムのエラーの原因となる条件を説明する。</p> <p>ERROR Q アプライ・プログラムでのエラーを説明する。</p>
TRACE_TIME	<p>データ・タイプ: TIMESTAMP。NULL 可能: いいえ</p> <p>この表に行が挿入されたときの Q アプライ・サーバーの時刻。</p>
説明	<p>データ・タイプ: VARCHAR(1024)。NULL 可能: いいえ</p> <p>ASN メッセージ ID とメッセージ・テキスト。この列に入れられるテキストは英語のみである。</p>

関連概念:

- 283 ページの『Q レプリケーションおよびイベント発行プログラムの履歴およびパフォーマンス・データ』
- 448 ページの『Q アプライ・コントロール表の詳細な構造 -- 概説』

関連資料:

- 422 ページの『Q アプライ・サーバーのコントロール表のリスト』

IBMQREP_DELTOMB 表

サーバー: Q アプライ・サーバー

デフォルト・スキーマ: ASN

索引: TARGET_NAME、 TARGET_OWNER、 VERSION_TIME DESC、 KEY_HASH

重要: SQL を使用してこの表を変更しないでください。 この表の変更の方法が不適切であると、予期せぬ結果が生じたり、データが失われたりします。

IBMQREP_DELTOMB 表は、Q アプライ・プログラムにより使用される内部表で、ピアツーピア・レプリケーションの競合削除を記録します。 Q アプライ・プログラムはこの表を整理します。

456 ページの表 65 では、IBMQREP_DELTOMB 表の列の要旨を示します。

表 65. IBMQREP_DELTOMB 表の列

列名	説明
TARGET_OWNER	データ・タイプ: VARCHAR(30)、DB2 UDB for z/OS バージョン 8 新機能モードの場合 VARCHAR(128)。NULL 可能: いいえ 競合削除が記録されたターゲット表の所有者名。
TARGET_NAME	データ・タイプ: VARCHAR(128)。NULL 可能: いいえ 競合削除が記録された表の名前。
VERSION_TIME	データ・タイプ: TIMESTAMP。NULL 可能: いいえ 発信サーバーでの競合削除のタイム・スタンプ
VERSION_NODE	データ・タイプ: SMALLINT。NULL 可能: いいえ ピアツーピア・グループで競合削除の発生したサーバーを識別する。
KEY_HASH	データ・タイプ: INTEGER。NULL 可能: いいえ 競合削除のキーのハッシュ値。
PACKED_KEY	データ・タイプ: VARCHAR(4096)。NULL 可能: いいえ 競合削除のパック 10 進数キー値。

関連概念:

- 448 ページの『Q アプライ・コントロール表の詳細な構造 -- 概説』

関連資料:

- 422 ページの『Q アプライ・サーバーのコントロール表のリスト』

IBMQREP_DONEMSG 表

サーバー: Q アプライ・サーバー

デフォルト・スキーマ: ASN

主キー: RECVQ、MGMSGID

重要: SQL を使用してこの表を変更しないでください。この表の変更の方法が不適切であると、予期せぬ結果が生じたり、データが失われたりします。

IBMQREP_DONEMSG 表は Q アプライ・プログラムにより使用される内部表であり、受信したすべてのトランザクション・メッセージまたは管理メッセージを記録します。この表にある記録により、メッセージが (たとえば、システム障害が原因で) 削除される前に、2 回以上処理されることがないようにします。Q アプライ・プログラムは、この表にある項目を開始時または通常の実行時に除去します。

457 ページの表 66 では、IBMQREP_DONEMSG 表の列の要旨を示します。

表 66. IBMQREP_DONEMSG 表の列

列名	説明
RECVQ	データ・タイプ: VARCHAR(97)。NULL 可能: いいえ メッセージ到着時の受信キューの名前。
MQMSGID	データ・タイプ: CHAR(24) FOR BIT DATA。NULL 可能: いいえ メッセージの WebSphere MQ メッセージ ID。

関連概念:

- 448 ページの『Q アプライ・コントロール表の詳細な構造 -- 概説』

関連資料:

- 422 ページの『Q アプライ・サーバーのコントロール表のリスト』

IBMQREP_EXCEPTIONS 表

サーバー: Q アプライ・サーバー

デフォルト・スキーマ: ASN

IBMQREP_EXCEPTIONS 表には、SQL コードや、競合または SQL エラーによりアプライできない行の変更にに関する情報が入っています。SQLCODE、SQLSTATE、および SQLERRMC フィールドは、Q アプライ・プログラムによりロールバックされる行を NULL に設定します。

この表の大きさは、予想される競合やエラーの数に応じて異なります。ユーザーは SQL を使用して、表から不要な行を削除することができます。

表 67 では、IBMQREP_EXCEPTIONS 表の列の要旨を示します。

表 67. IBMQREP_EXCEPTIONS 表の列

列名	説明
EXCEPTION_TIME	データ・タイプ: TIMESTAMP。NULL 可能: デフォルトでは不可 エラーまたは競合が発生したときの Q アプライ・サーバーのタイム・スタンプ。 デフォルト: 現在のタイム・スタンプ
RECVQ	データ・タイプ: VARCHAR(48)。NULL 可能: いいえ トランザクション・メッセージが着信した受信キューの名前。
SRC_COMMIT_LSN	データ・タイプ: CHAR(10) FOR BIT DATA。NULL 可能: いいえ トランザクション用の Q キャプチャー・サーバーの論理ログ・シーケンス番号。
SRC_TRANS_TIME	データ・タイプ: TIMESTAMP。NULL 可能: いいえ トランザクション用の Q キャプチャー・サーバーのタイム・スタンプ。
SUBNAME	データ・タイプ: VARCHAR(128)。NULL 可能: いいえ トランザクションが属する Q サブスクリプションの名前。

表 67. IBMQREP_EXCEPTIONS 表の列 (続き)

列名	説明
REASON	<p>データ・タイプ: CHAR(12)。 NULL 可能: いいえ</p> <p>トランザクションを IBMQREP_EXCEPTIONS 表のログに記録する原因となるエラーまたは競合の説明。</p> <p>NOTFOUND 存在しない行の削除または、更新を試行する。</p> <p>DUPLICATE 既存の行の挿入を試行する。</p> <p>CHECKFAILED 競合検出の規則としてすべての値または変更された値をチェックするが、予期しない非キー値はチェックしない。</p> <p>SQLERROR IBMQREP_TARGETS 表の OKSQLSTATES 列の許容可能エラーのリストにはない SQL エラーが生じた。</p> <p>OKSQLSTATE IBMQREP_TARGETS 表の OKSQLSTATES 列の許容可能エラーのリストにある SQL エラーが生じた。</p> <p>P2PDUPKEY ピアツーピア・レプリケーションで、ターゲット行が既存の同一キーを持つものよりも新しかったため、キーの更新に失敗した。</p> <p>P2PNOTFOUND ピアツーピア・レプリケーションで、ターゲット行が存在しなかったため、削除または更新に失敗した。</p> <p>P2PVERLOSER ピアツーピア・レプリケーションで、ターゲット行が変更メッセージのある行よりも新しかったため、削除または更新に失敗した。</p>
SQLCODE	<p>データ・タイプ: INTEGER。 NULL 可能: はい</p> <p>トランザクションのため、DB2 により戻された SQL コード。</p>
SQLSTATE	<p>データ・タイプ: CHAR(5)。 NULL 可能: はい</p> <p>トランザクションのため、DB2 により戻された SQL の状態番号。</p>
SQLERRMC	<p>データ・タイプ: CHAR(70) FOR BIT DATA。 NULL 可能: いいえ</p> <p>トランザクションを実行する際に使用する SQLCA 構造からのエラー・メッセージ・トークン。</p>
OPERATION	<p>データ・タイプ: VARCHAR(18)。 NULL 可能: いいえ</p> <p>失敗した SQL 操作のタイプ。表示される値は、INSERT、INSERT(LOAD)、DELETE、DELETE(LOAD)、UPDATE、UPDATE(LOAD)、KEY UPDATE、KEY UPDATE(LOAD)。</p>

表 67. *IBMQREP_EXCEPTIONS* 表の列 (続き)

列名	説明
TEXT	<p>データ・タイプ: LONG VARCHAR(7800)。 NULL 可能: いいえ</p> <p>エラーを起こした行のデータの XML 記述。 XML 発行で使用されるのと同じスキーマを用いてメッセージをエンコードする。ルート・エレメントは、insertRow、deleteRow、または updateRow のいずれかである。可能な場合は、前後両方に値を含める。 XML 文書は、ターゲット・データベースのコード・ページでエンコードする。 <i>IBMQREP_EXCEPTIONS</i> 表を使用しているクライアント・アプリケーションの場合、文書のエンコード XML ヘッダー属性はターゲット・データベースのコード・ページを示すが、文書は、クライアント・アプリケーションのコード・ページにある。</p>
IS_APPLIED	<p>データ・タイプ: CHAR(1)。 NULL 可能: いいえ</p> <p><i>IBMQREP_EXCEPTIONS</i> 表にあっても、行がターゲット表にアプライされるかどうかを示すフラグ。</p> <p>Y Q サブスクリプションに <i>CONFLICT_ACTION</i> が F (force) を指定したので、行がアプライされた。</p> <p>N トランザクションはアプライされない。</p>
CONFLICT_RULE	<p>データ・タイプ: CHAR(1)。 NULL 可能: はい</p> <p><i>IBMQREP_EXCEPTIONS</i> 表に入れた行で生じた競合検出のタイプ。</p> <p>K キー値のみ値チェックされる。</p> <p>C キー値と同様に変更された非キー値もチェックされる。</p> <p>A すべての値がチェックされる。</p>

関連概念:

- 288 ページの『例外』
- 283 ページの『Q レプリケーションおよびイベント発行プログラムの履歴およびパフォーマンス・データ』
- 448 ページの『Q アプライ・コントロール表の詳細な構造 -- 概説』

関連資料:

- 422 ページの『Q アプライ・サーバーのコントロール表のリスト』

IBMQREP_RECVQUEUES 表

サーバー: Q アプライ・サーバー

デフォルト・スキーマ: ASN

主キー: RECVQ

ユニーク索引: REPQMAPNAME

重要: SQL を使用してこの表を変更しないでください。 この表の変更の方法が不適切であると、予期せぬ結果が生じたり、データが失われたりします。

IBMQREP_RECVQUEUES 表には、トランザクションをソースから受信するときに使用する、WebSphere MQ ローカル・キューに関する情報があります。それぞれの Q アプライ・プログラムが、複数の受信キューを処理することができます。それぞれの受信キューは、Q アプライ受信キュー表内の行によって一意に識別されます。

表 68 では、IBMQREP_RECVQUEUES 表の列の要旨を示します。

表 68. IBMQREP_RECVQUEUES 表の列

列名	説明
REPQMAPNAME	データ・タイプ: VARCHAR(128)。NULL 可能: いいえ 受信キューのあるレプリケーション・キュー・マップの名前。
RECVQ	データ・タイプ: VARCHAR(48)。NULL 可能: いいえ この Q サブスクリプションの受信キューの名前。
SENDQ	データ・タイプ: VARCHAR(48)。NULL 可能: はい この Q サブスクリプションを実行する Q キャプチャー・プログラムが使用する送信キューの名前。
ADMINQ	データ・タイプ: VARCHAR(48)。NULL 可能: いいえ Q アプライ・プログラムが、Q キャプチャー・プログラムへ制御およびエラー・メッセージを送信するときに使用する管理キューの名前。
NUM_APPLY_AGENTS	データ・タイプ: INTEGER。NULL 可能: デフォルトでは不可 この受信キューから並列にトランザクションをアプライするときに Q アプライ・プログラムが使用する、エージェント・スレッドの数。値 1 は、ソース表から受信した順でトランザクションを実行するように要求する。デフォルト: 16
MEMORY_LIMIT	データ・タイプ: INTEGER。NULL 可能: デフォルトでは不可 Q アプライ・プログラムがメッセージ用にこの受信キューから取得したバッファとして使用できるメモリの最大量 (MB 単位)。デフォルト: 32 MB
CAPTURE_SERVER	データ・タイプ: VARCHAR(18)。NULL 可能: いいえ この受信キューが稼働中のとき、Q キャプチャー・プログラムが使用するデータベースの名前。z/OS の場合は、ロケーション名となる。
CAPTURE_ALIAS	データ・タイプ: VARCHAR(8)。NULL 可能: いいえ CAPTURE_SERVER 列で命名された Q キャプチャー・サーバーに対応する DB2 データベース別名。
CAPTURE_SCHEMA	データ・タイプ: VARCHAR(30)、DB2 UDB for z/OS バージョン 8 新機能モードの場合 VARCHAR(128)。NULL 可能: デフォルトでは、いいえ この受信キューを使用している、Q キャプチャー・プログラムのスキーマ。デフォルト: ASN

表 68. IBMQREP_RECVQUEUES 表の列 (続き)

列名	説明
STATE	<p>データ・タイプ: CHAR(1)。 NULL 可能: デフォルトでは不可</p> <p>受信キューの現在の状態を示すフラグ。</p> <p>A (デフォルト) アクティブ: Q アプライ・プログラムが、このキューからトランザクションを処理し、アプライしている。</p> <p>I 非アクティブ: キューで重大エラーが発生した。</p>
STATE_TIME	<p>データ・タイプ: TIMESTAMP。 NULL 可能: デフォルトでは不可</p> <p>この受信キューで最後に状態を変更したときの Q アプライ・サーバーのタイム・スタンプ。デフォルト: 現在のタイム・スタンプ</p>
STATE_INFO	<p>データ・タイプ: CHAR(8)。 NULL 可能: はい</p> <p>キューの状態を示す ASN メッセージの数。詳細については、IBMQREP_APPLYTRACE 表または Q アプライ診断ログを参照。</p>
DESCRIPTION	<p>データ・タイプ: VARCHAR(254)。 NULL 可能: はい</p> <p>受信キューのあるレプリケーション・キュー・マップの、ユーザー提供の説明。</p>

関連概念:

- 62 ページの『WebSphere MQ オブジェクトの設定要件』
- 6 ページの『レプリケーション・キュー・マップ』
- 448 ページの『Q アプライ・コントロール表の詳細な構造 -- 概説』

関連タスク:

- 95 ページの『レプリケーション・キュー・マップの作成』

関連資料:

- 422 ページの『Q アプライ・サーバーのコントロール表のリスト』

IBMQREP_SAVERI 表

サーバー: Q アプライ・サーバー

デフォルト・スキーマ: ASN

重要: SQL を使用してこの表を変更しないでください。この表の変更の方法が不適切であると、予期せぬ結果が生じたり、データが失われたりします。

IBMQREP_SAVERI 表は、Q アプライ・プログラムによって使用される内部表であり、ターゲット表の参照保全制約に関する情報を保管します。Q アプライ・プログラムは、ターゲット表のロード中に、参照保全制約をドロップします。制約は、このコントロール表に保存され、表がロードされてからリストアされます。

462 ページの表 69 では、IBMQREP_SAVERI 表の列の要旨を示します。

表 69. IBMQREP_SAVERI 表の列

列名	説明
SUBNAME	データ・タイプ: VARCHAR(132)。NULL 可能: いいえ ターゲット表が属する Q サブスクリプションの名前。
RECVQ	データ・タイプ: VARCHAR(48)。NULL 可能: いいえ Q サブスクリプションを指定した受信キューの名前。
CONSTNAME	データ・タイプ: VARCHAR(18)。NULL 可能: いいえ 制約のユニーク名。
TABSCHEMA	データ・タイプ: VARCHAR(30)、DB2 UDB for z/OS バージョン 8 新機能モードの場合 VARCHAR(128)。NULL 可能: いいえ 制約が定義された子表のスキーマまたは高位修飾子。
TABNAME	データ・タイプ: VARCHAR(128)。NULL 可能: いいえ 制約が定義された子表の名前。
REFTABSCHEMA	データ・タイプ: VARCHAR(128)。NULL 可能: いいえ 制約が定義された親表のスキーマ。
REFTABNAME	データ・タイプ: VARCHAR(128) NULL 可能: いいえ 制約が定義された親表の名前。
ALTER_RI_DDL	データ・タイプ: VARCHAR(1680)。NULL 可能: いいえ 参照保全制約のリストアに使用される ALTER TABLE ステートメント。
TYPE_OF_LOAD	データ・タイプ: CHAR(1)。NULL 可能: いいえ ロード・フェーズのタイプを示すフラグ。 I 自動ロード。 E 手動ロード。

関連概念:

- 448 ページの『Q アプライ・コントロール表の詳細な構造 -- 概説』

関連資料:

- 422 ページの『Q アプライ・サーバーのコントロール表のリスト』

IBMQREP_SPILLEDROW 表

サーバー: Q アプライ・サーバー

デフォルト・スキーマ: ASN

主キー: SPILLQ、MQMSGID

重要: SQL を使用してこの表を変更しないでください。この表の変更の方法が不適切であると、予期せぬ結果が生じたり、データが失われたりします。

IBMQREP_SPILLEDROW 表は、Q アプライ・プログラムにより使用される内部表であり、ターゲットのロード中に一時予備キューに送信されるメッセージを記録します。Q アプライ・プログラムは、メッセージが予備キューから取られて、ターゲット表に適用された後に、この表からメッセージを示す行を削除します。

表 70 では、IBMQREP_SPILLEDROW 表の列の要旨を示します。

表 70. IBMQREP_SPILLEDROW 表の列

列名	説明
SPILLQ	データ・タイプ: VARCHAR(48)。NULL 可能: いいえ メッセージが一時的に保存される予備キューの名前。
MQMSGID	データ・タイプ: CHAR(24) FOR BIT DATA。NULL 可能: いいえ メッセージの WebSphere MQ メッセージ ID。

関連概念:

- 448 ページの『Q アプライ・コントロール表の詳細な構造 -- 概説』

関連資料:

- 422 ページの『Q アプライ・サーバーのコントロール表のリスト』

IBMQREP_SPILLQS 表

サーバー: Q アプライ・サーバー

デフォルト・スキーマ: ASN

主キー: SPILLQ

重要: SQL を使用してこの表を変更しないでください。この表の変更の方法が不適切であると、予期せぬ結果が生じたり、データが失われたりします。

IBMQREP_SPILLQS 表は、Q アプライ・プログラムにより使用される内部表であり、一時予備キューを記録し、ターゲット表のロード中にメッセージを保持させます。Q アプライ・プログラムは、予備キューが不要になると除去します。

表 71 では、IBMQREP_SPILLQS 表の列の要旨を示します。

表 71. IBMQREP_SPILLQS 表の列

列名	説明
SPILLQ	データ・タイプ: VARCHAR(48)。NULL 可能: いいえ Q サブスクリプションに使用する一時予備キューの名前。
SUBNAME	データ・タイプ: VARCHAR(132)。NULL 可能: いいえ Q サブスクリプションの名前。
RECVQ	データ・タイプ: VARCHAR(48)。NULL 可能: いいえ Q サブスクリプションに使用する受信キューの名前。

関連概念:

- 448 ページの『Q アプライ・コントロール表の詳細な構造 -- 概説』

関連資料:

- 422 ページの『Q アプライ・サーバーのコントロール表のリスト』

IBMQREP_TRG_COLS 表

サーバー: Q アプライ・サーバー

デフォルト・スキーマ: ASN

索引: RECVQ、SUBNAME、TARGET_COLNAME

重要: SQL を使用してこの表を変更しないでください。この表の変更の方法が不適切であると、予期せぬ結果が生じたり、データが失われたりします。

IBMQREP_TRG_COLS 表は、ソース表の列とターゲット表の列との間のマッピング、またはターゲットがストアド・プロシージャーの場合、ソース列とパラメーターとの間のマッピングを示します。Q アプライ・プログラムは、Q キャプチャー・プログラムからのスキーマ・メッセージで受信する情報に基づいてこの表に値を入力します。

表 72 では、IBMQREP_TRG_COLS 表の列の要旨を示します。

表 72. IBMQREP_TRG_COLS 表の列

列名	説明
RECVQ	データ・タイプ: VARCHAR(48)。NULL 可能: いいえ Q サブスクリプションに使用する受信キューの名前。
SUBNAME	データ・タイプ: VARCHAR(132)。NULL 可能: いいえ Q サブスクリプションの名前。
SRC_COLNAME	データ・タイプ: VARCHAR(30)。NULL 可能: いいえ ソース列の名前。
TARG_COLNAME	データ・タイプ: VARCHAR(30)。NULL 可能: いいえ ターゲット列の名前。ターゲットがストアド・プロシージャーである場合、この列には、Q アプライ・プログラムが、ソース列値を受け渡すパラメーターの名前が入ります。
TARG_COLNO	データ・タイプ: INTEGER。NULL 可能: はい ターゲット列に割り当てられた番号。ターゲットがストアド・プロシージャーである場合、この列には、Q アプライ・プログラムがソース列値を渡すパラメーターに割り当てられた番号が入ります。
MSG_COL_CODEPAGE	データ・タイプ: INTEGER。NULL 可能: はい 変更されたメッセージでソース列の値をエンコードするときに使用されるコード・ページの ID。

表 72. IBMQREP_TRG_COLS 表の列 (続き)

列名	説明
MSG_COL_NUMBER	データ・タイプ: SMALLINT。NULL 可能: はい 変更メッセージでのソース列の出現順序 (0 から始まる)。
MSG_COL_TYPE	データ・タイプ: SMALLINT。NULL 可能: はい ソース列の DB2 データ・タイプ。
MSG_COL_LENGTH	データ・タイプ: SMALLINT。NULL 可能: はい ソース列で定義された最大データ長。
IS_KEY	データ・タイプ: CHAR(1)。NULL 可能: いいえ ソース列がソース表のキーの一部であるかどうかを示すフラグ。このフラグの値がターゲット表のキー定義と一致しない場合、Q アプライ・プログラムはスキーマ・メッセージを拒否し、Q サブスクリプションを以下のように無効にする。 Y この列は、ソース表キーの一部である。 N この列は、ソース表キーの一部ではない。

関連概念:

- 112 ページの『ターゲットの索引列またはキー列 (単一方向レプリケーション)』
- 448 ページの『Q アプライ・コントロール表の詳細な構造 -- 概説』

関連資料:

- 422 ページの『Q アプライ・サーバーのコントロール表のリスト』

IBMQREP_TARGETS 表

サーバー: Q アプライ・サーバー

デフォルト・スキーマ: ASN

ユニーク索引: (TARGET_OWNER ASC、TARGET_NAME ASC、RECVQ ASC、SOURCE_OWNER ASC、SOURCE_NAME ASC)

ユニーク索引: SUBNAME、RECVQ

索引: RECVQ、SUB_ID

重要: SQL を使用してこの表を変更しないでください。この表の変更の方法が不適切であると、予期せぬ結果が生じたり、データが失われたりします。

IBMQREP_TARGETS 表には、ターゲット表またはストアード・プロシージャに関する情報と、それらが属する Q サブスクリプションについての情報があります。この表には、Q サブスクリプションのタイプと状態、デフォルトのエラー・アクション、および行の競合の処理規則に関する詳細が保管されています。

466 ページの表 73 では、IBMQREP_TARGETS 表の列の要旨を示します。

表 73. IBMQREP_TARGETS 表の列

列名	説明
SUBNAME	<p>データ・タイプ: VARCHAR(132)。 NULL 可能: いいえ</p> <p>Q サブスクリプションの名前。これは、ソース - ターゲットのペアごとに固有でなければならず、ブランクにすることはできない。</p>
RECVQ	<p>データ・タイプ: VARCHAR(48)。 NULL 可能: いいえ</p> <p>この Q サブスクリプション使用される受信キューの名前。</p>
SUB_ID	<p>データ・タイプ: INTEGER。 NULL 可能: はい</p> <p>Q キャプチャー・プログラムが生成する整数で、サブスクリプション・スキーマ・メッセージ内で Q サブスクリプションと Q アプライ・プログラムを一意に識別するために使用する。</p>
SOURCE_SERVER	<p>データ・タイプ: VARCHAR(18)。 NULL 可能: いいえ</p> <p>この Q サブスクリプションのソース表があるデータベースまたはサブシステムの名前。 z/OS の場合は、ロケーション名となる。</p>
SOURCE_ALIAS	<p>データ・タイプ: VARCHAR(18)。 NULL 可能: いいえ</p> <p>SOURCE_SERVER 列で命名された Q キャプチャー・サーバーに対応する DB2 データベース別名。</p>
SOURCE_OWNER	<p>データ・タイプ: VARCHAR(30)、DB2 UDB for z/OS バージョン 8 新機能モードの場合 VARCHAR(128)。 NULL 可能: いいえ</p> <p>この Q サブスクリプションのソース表があるスキーマ名または高位修飾子。</p>
SOURCE_NAME	<p>データ・タイプ: VARCHAR(128)、z/OS バージョン 7 およびバージョン 8 互換モードの場合 VARCHAR(18)。 NULL 可能: いいえ</p> <p>この Q サブスクリプションのソース表の名前。</p>
SRC_NICKNAME_OWNER	<p>データ・タイプ: VARCHAR(128)。 NULL 可能: はい</p> <p>Q アプライ・プログラムが非 z/OS プラットフォームで実行中に、CURSOR からの LOAD ユーティリティを使用する自動ロード用のソース表に割り当てられるニックネームのスキーマ。</p>
SRC_NICKNAME	<p>データ・タイプ: VARCHAR(128)。 NULL 可能: はい</p> <p>Q アプライ・プログラムが非 z/OS プラットフォームで実行中に CURSOR からの LOAD ユーティリティを使用する自動ロード用のソース表に割り当てられるニックネーム。</p>
TARGET_OWNER	<p>データ・タイプ: VARCHAR(30)、DB2 UDB for z/OS バージョン 8 新機能モードの場合 VARCHAR(128)。 NULL 可能: いいえ</p> <p>この Q サブスクリプションのターゲット表またはストアド・プロシージャのスキーマ名または高位修飾子。</p>
TARGET_NAME	<p>データ・タイプ: VARCHAR(128)、DB2 UDB for z/OS バージョン 7 およびバージョン 8 互換モードの場合 VARCHAR(18)。 NULL 可能: いいえ</p> <p>この Q サブスクリプションのターゲット表の名前。</p>

表 73. IBMQREP_TARGETS 表の列 (続き)

列名	説明
TARGET_TYPE	<p>データ・タイプ: INTEGER。NULL 可能: デフォルトでは不可</p> <p>レプリケーション・ターゲットのタイプを示すフラグ。</p> <p>1 (デフォルト) ユーザー表</p> <p>2 将来用に予約済み。</p> <p>3 将来用に予約済み。</p> <p>4 将来用に予約済み。</p> <p>5 ストアード・プロシージャ</p>
FEDERATED_TGT_SRVR	<p>データ・タイプ: VARCHAR(18)。NULL 可能: はい</p> <p>レプリケーションを行う時、DB2 以外のリレーショナル・ターゲットに対する Q サブスクリプション・ターゲットであるフェデレーテッド・リモート・サーバーの名前。</p>
STATE	<p>データ・タイプ: CHAR(1)。NULL 可能: デフォルトでは不可</p> <p>現在の Q サブスクリプションの状態を説明する Q アプライ・プログラムを挿入するフラグ。</p> <p>I (デフォルト) Q サブスクリプションが新規またはエラー状態にあるため、Q アプライはターゲットへの変更をアプライしない。Q アプライ・プログラムは Q サブスクリプションを受信し、新規のサブスクリプション・スキーマ・メッセージを待機するすべてのトランザクションを廃棄する。</p> <p>L Q キャプチャー・プログラムは、サブスクリプション・スキーマ・メッセージを送信して、Q サブスクリプションのアクティブ化を開始し、ソース表への変更を送信する。</p> <p>E ターゲット表は外部アプリケーションによってロードされる。Q アプライ・プログラムは、表のロード待機中にメッセージの変更を予備キューに書き込む。</p> <p>D ターゲット表がロードされると、Q アプライ・プログラムは、Q キャプチャー・プログラムへの load done メッセージを送信できる状態になる。このメッセージは、自動ロード用としてのみ出されます。</p> <p>F Q アプライ・プログラムが、予備キューからメッセージのアプライ中。</p> <p>T エラーが発生したため、Q アプライ・プログラムは終了した。Q サブスクリプションを非アクティブ化し、予備キューを空にして削除する。</p> <p>A Q アプライ・プログラムが、変更をターゲットにアプライ中。</p>
STATE_TIME	<p>データ・タイプ: TIMESTAMP。NULL 可能: デフォルトでは不可</p> <p>この Q サブスクリプションで最後に状態を変更したときの Q アプライ・サーバーのタイム・スタンプ。デフォルト: 現在のタイム・スタンプ</p>
STATE_INFO	<p>データ・タイプ: CHAR(8)。NULL 可能: はい</p> <p>Q サブスクリプションの状態を示す ASN メッセージの数。詳細については、IBMQREP_APPLYTRACE 表または Q アプライ診断ログを参照する。</p>

表 73. IBMQREP_TARGETS 表の列 (続き)

列名	説明
SUBTYPE	<p>データ・タイプ: CHAR(1)。 NULL 可能: デフォルトでは不可</p> <p>Q サブスクリプションが関わっているレプリケーションのタイプを示すフラグ。</p> <p>U (デフォルト) 単一方向レプリケーション。</p> <p>B 双方向レプリケーション。</p> <p>P ピアツーピア・レプリケーション。</p>
CONFLICT_RULE	<p>データ・タイプ: CHAR(1)。 NULL 可能: デフォルトでは不可</p> <p>Q アプライ・プログラムにターゲット表への競合する変更を検出する手段を通知するフラグ。挿入は、変更前の値がなく、キーが競合の検出に使用されなければならないので、常に K (キーのみチェック) 規則を使用してチェックを行う。双方向レプリケーションの場合、A (すべて変更) 競合検出規則が必要である。</p> <p>K (デフォルト) キーのみチェックする。Q アプライ・プログラムは、ターゲット表の現在の主キーの値を、ソース表から送信された以前のキー値と比較し、競合を検出する。これは、単一方向レプリケーションに許される唯一の競合の規則である。</p> <p>C 変更された列をチェックする。ターゲット列を更新する前に、Q アプライ・プログラムは、現在の値がソース列の変更前の値と一致しているかを確認する。削除された場合、Q アプライ・プログラムは、すべての列をチェックする。</p> <p>A すべての列をチェックする。行を更新または削除する前に、Q アプライ・プログラムは、すべての列にある現在の値が、ソース表の以前の値と一致しているかを確認する。</p> <p>V バージョンをチェックする。ピアツーピア・レプリケーションで、Q アプライ・プログラムは、行をアプライする前に、バージョン列をチェックする。</p>

表 73. IBMQREP_TARGETS 表の列 (続き)

列名	説明
CONFLICT_ACTION	<p>データ・タイプ: CHAR(1)。 NULL 可能: デフォルトでは不可</p> <p>行の変更が競合した場合に、Q アプライ・プログラムが実行することを指示するフラグ。</p> <p>I (デフォルト) Q アプライ・プログラムを、競合行にはアプライしないが、トランザクション内の他の行にはアプライする。</p> <p>F Q アプライ・プログラムは強制的に変更を試行する。これにより、Q キャプチャー・プログラムがすべての列を送信して、IBMQREP_SUBS 表で CHANGED_COLS_ONLY 値を N (いいえ) に設定しなければならない。これは、ターゲット表がロードするデフォルト値である。</p> <p>D Q アプライ・プログラムを、競合行にはアプライしないが、トランザクション内の他の行にはアプライする。これにより、Q サブスクリプションは使用不可になり、トランザクションのターゲットへのアプライが停止し、管理キューのある Q キャプチャー・プログラムにエラー報告を送信する。</p> <p>S Q アプライ・プログラムはトランザクションをロールバックし、コミットし、次いで停止する。</p> <p>Q Q アプライ・プログラムはキューからの読み取りを停止する。</p> <p>競合するすべての行は IBMQREP_EXCEPTIONS 表に挿入される。</p>
ERROR_ACTION	<p>データ・タイプ: CHAR(1)。 NULL 可能: デフォルトでは不可</p> <p>行の変更のアプライを妨げる (競合以外の) SQL が発生したときに、Q アプライ・プログラムが実行することを指示するフラグ。</p> <p>Q (デフォルト) Q アプライ・プログラムはキューからの読み取りを停止する。</p> <p>D Q アプライ・プログラムを、競合行にはアプライしないが、トランザクション内の他の行にはアプライする。これにより、Q サブスクリプションは使用不可になり、トランザクションのターゲットへのアプライが停止し、管理キューのある Q キャプチャー・プログラムにエラー報告を送信する。</p> <p>S Q アプライ・プログラムはトランザクションをロールバックし、コミットし、次いで停止する。</p> <p>競合するすべての行は IBMQREP_EXCEPTIONS 表に挿入される。</p>
SPILLQ	<p>データ・タイプ: VARCHAR(48)。 NULL 可能: はい</p> <p>ターゲットをロードするときに Q アプライ・プログラムが作成する一時予備キューの名前。</p>
OKSQLSTATES	<p>データ・タイプ: VARCHAR(128)。 NULL 可能: はい</p> <p>Q アプライ・プログラムがエラーとは見なさないスペース区切り SQLSTATE 値のリスト。 Q サブスクリプションを定義するときに、この値を指定する。</p>

表 73. *IBMQREP_TARGETS* 表の列 (続き)

列名	説明
SUBGROUP	<p>データ・タイプ: VARCHAR(30)。 NULL 可能: はい</p> <p>Q サブスクリプションに含まれるピアツーピア・レプリケーション・グループの名前。</p>
SOURCE_NODE	<p>データ・タイプ: SMALLINT。 NULL 可能: デフォルトでは不可</p> <p>ピアツーピア Q サブスクリプションでのソース・サーバーの識別番号。デフォルト: 0</p>
TARGET_NODE	<p>データ・タイプ: SMALLINT。 NULL 可能: デフォルトでは不可</p> <p>ピアツーピア Q サブスクリプションでのターゲット・サーバーの識別番号。デフォルト: 0</p>
GROUP_INIT_ROLE	<p>データ・タイプ: CHAR(1)。 NULL 可能: はい</p> <p>ピアツーピア Q サブスクリプションの初期化中にこのターゲット・サーバーが果たす役割。</p> <p>I サブスクリプションを初期化するため、CAPSTART シグナルを <i>IBMQREP_SIGNAL</i> 表に入れたピアツーピア・グループのイニシエーター。</p> <p>M サブスクリプションの初期化には使用しないピアツーピア・グループのサーバー。</p> <p>N ピアツーピア・グループを結合中の新規サーバー。</p>
HAS_LOADPHASE	<p>データ・タイプ: CHAR(1)。 NULL 可能: デフォルトでは不可</p> <p>ターゲット表がソースからデータをロードするかどうかを示すフラグ。</p> <p>N (デフォルト) ターゲットはロードされない。</p> <p>I 自動ロード。 Q サブスクリプションに指定された自動ロードのタイプおよび Q アプライ・サーバーのオペレーティング・システムに応じて、Q アプライ・プログラムは、CURSOR からの LOAD、EXPORT/IMPORT、または EXPORT/LOAD ユーティリティを呼び出す。表がロードされると、Q アプライ・プログラムは、Q キャプチャー・プログラムへの load done メッセージを送信できる状態になる。</p> <p>E 手動ロード。 Q アプライ・プログラム以外のアプリケーションがターゲット表をロードする。この場合、ユーザーまたはレプリケーション・センターが LOADDONE シグナルを Q キャプチャー・サーバーにある <i>IBMQREP_SIGNAL</i> 表に挿入するか、load done メッセージの受信後に、Q キャプチャー・プログラムがこのシグナルを挿入する。</p>

表 73. IBMQREP_TARGETS 表の列 (続き)

列名	説明
LOAD_TYPE	<p>データ・タイプ: SMALLINT。 NULL 可能: デフォルトでは不可</p> <p>ロード・タイプが内部であるとき、ターゲット表をロードするときに呼び出すユーティリティを示すフラグ。</p> <p>0 (デフォルト) Q アプライ・プログラムは、以下の 3 つのオプションからロード・ユーティリティを選択する。</p> <p>1 CURSOR からの LOAD ユーティリティを使用する。</p> <p>2 EXPORT/IMPORT ユーティリティを使用する。</p> <p>3 EXPORT/LOAD ユーティリティを使用する。</p>
説明	<p>データ・タイプ: VARCHAR(254)。 NULL 可能: はい</p> <p>Q サブスクリプションのユーザー提供の説明。</p>
SEARCH_CONDITION	<p>データ・タイプ: VARCHAR(2048)。 NULL 可能: はい</p> <p>Q サブスクリプションで行をフィルタリングするために使用する検索条件。これは、直前にコロンを 1 つ付加したソース列名を持つ where 文節でなければならない。</p>

関連概念:

- 5 ページの『Q サブスクリプション』
- 108 ページの『行のフィルタリングのための検索条件 (単一方向レプリケーション)』
- 114 ページの『ターゲット表での予期しない条件のオプション (単一方向レプリケーション)』
- 163 ページの『Q レプリケーションのターゲット表のロードのオプション -- 概要』
- 118 ページの『Q レプリケーションのエラー・オプション』
- 147 ページの『競合検出のオプション (双方向レプリケーション)』
- 448 ページの『Q アプライ・コントロール表の詳細な構造 -- 概説』

関連タスク:

- 209 ページの『Q サブスクリプションでの許容可能な SQL 状態の変更』

関連資料:

- 422 ページの『Q アプライ・サーバーのコントロール表のリスト』

モニター・コントロール表の詳細な構造

モニター・コントロール表の詳細な構造 -- 概説

以下のトピックでは、モニター・コントロール・サーバー上の各表について詳しく説明しています。コントロール表はアルファベット順にリストされます。各表にある列は、左から右に、出現順にリストされています。

- 『IBMSNAP_ALERTS 表』
- 474 ページの 『IBMSNAP_CONDITIONS 表』
- 481 ページの 『IBMSNAP_CONTACTGRP 表』
- 482 ページの 『IBMSNAP_CONTACTS 表』
- 483 ページの 『IBMSNAP_GROUPS 表』
- 483 ページの 『IBMSNAP_MONENQ 表』
- 483 ページの 『IBMSNAP_MONPARMS 表』
- 485 ページの 『IBMSNAP_MONSERVERS 表』
- 487 ページの 『IBMSNAP_MONTRACE 表』
- 488 ページの 『IBMSNAP_MONTRAIL 表』

関連概念:

- 415 ページの 『Q レプリケーションおよびイベント発行のコントロール表 -- 概要』
- 448 ページの 『Q アプライ・コントロール表の詳細な構造 -- 概説』
- 425 ページの 『Q キャプチャー・コントロール表の詳細な構造 -- 概説』

IBMSNAP_ALERTS 表

サーバー: モニター・コントロール・サーバー

索引: MONITOR_QUAL、COMPONENT、SERVER_NAME、SCHEMA_OR_QUAL、SET_NAME、CONDITION_NAME、ALERT_CODE

IBMSNAP_ALERTS 表は、レプリケーション・アラート・モニターから発行されるすべてのアラートのレコードを保持します。この表では、どのようなアラート条件が発生したか、どのサーバーで発生したか、そしていつアラートが検出されたかが記録されます。

表 74 では、IBMSNAP_ALERTS 表の列の要旨を示します。

表 74. IBMSNAP_ALERTS 表の列

列名	説明
MONITOR_QUAL	<p>データ・タイプ: CHAR(18)。 NULL 可能: いいえ。</p> <p>アラートを発行したレプリケーション・アラート・モニター・プログラムを示すモニター修飾子。</p>
COMPONENT	<p>データ・タイプ: CHAR(1)。 NULL 可能: いいえ。</p> <p>モニター対象のレプリケーション・コンポーネント。</p> <p>C キャプチャー・プログラム</p> <p>A アプライ・プログラム</p> <p>S Q キャプチャー・プログラム</p> <p>R Q アプライ・プログラム</p>

表 74. IBMSNAP_ALERTS 表の列 (続き)

列名	説明
SERVER_NAME	<p>データ・タイプ: CHAR(18)。 NULL 可能: いいえ。</p> <p>アラート条件が発生した、キャプチャー・コントロール・サーバー、アプライ・コントロール・サーバー、Q キャプチャー・サーバー、または Q アプライ・サーバーの名前。</p>
SERVER_ALIAS	<p>データ・タイプ: CHAR(8)。 NULL 可能: はい。</p> <p>アラート条件が発生した、キャプチャー・コントロール・サーバー、アプライ・コントロール・サーバー、Q キャプチャー・サーバー、または Q アプライ・サーバーの DB2 UDB 別名。</p>
SCHEMA_OR_QUAL	<p>データ・タイプ: VARCHAR(30)、DB2 UDB for z/OS バージョン 8 新機能モード・サブシステムの場合 VARCHAR(128)。 NULL 可能: いいえ。</p> <p>モニター対象のキャプチャー・スキーマ、アプライ・スキーマ、Q キャプチャー・スキーマ、または Q アプライ・スキーマ。</p>
SET_NAME	<p>データ・タイプ: CHAR(18)。 NULL 可能: デフォルトでは不可。 デフォルト: 現行サブスクリプション・セット。</p> <p>アプライ・プログラムでアラート条件が設定されている場合、この列は、モニター対象のサブスクリプション・セットの名前を指定します。セット名が指定されない場合は、アプライ修飾子レベルでモニターが行われます。つまり、特定のアプライ修飾子を持つすべてのセットがモニターされます。</p> <p>Q アプライの受信キュー項目数か予備キュー項目数のアラート条件が設定されている場合、この列は、モニター対象の受信キューか予備キューの名前を指定します。</p>
CONDITION_NAME	<p>データ・タイプ: CHAR(18)。 NULL 可能: いいえ。</p> <p>アラートが起動されたときにテストされた条件コード。</p>
OCCURRED_TIME	<p>データ・タイプ: TIMESTAMP。 NULL 可能: いいえ。</p> <p>キャプチャー・コントロール・サーバー、アプライ・コントロール・サーバー、Q キャプチャー・サーバー、または Q アプライ・サーバーでアラート条件が発生した時刻。</p>
ALERT_COUNTER	<p>データ・タイプ: SMALLINT。 NULL 可能: いいえ。</p> <p>連続するモニター・サイクルで、このアラートが以前に検出された回数。</p>
ALERT_CODE	<p>データ・タイプ: CHAR(10)。 NULL 可能: いいえ。</p> <p>アラートの発生時に発行されたメッセージ・コード。</p>
RETURN_CODE	<p>データ・タイプ: INT。 NULL 可能: いいえ。</p> <p>ユーザー条件から戻された整数値。</p>

IBMSNAP_ALERTS

表 74. IBMSNAP_ALERTS 表の列 (続き)

列名	説明
NOTIFICATION_SENT	<p>データ・タイプ: CHAR(1)。 NULL 可能: いいえ。</p> <p>通知メッセージが送信されたかどうかを示すフラグ。</p> <p>Y 通知メッセージが送信されました。</p> <p>E email_server パラメーターが指定されていなかったため、通知が送信されませんでした。</p> <p>N 通知の数が max_notifications_per_alert パラメーターで設定された限度にすでに達しているため、通知が送信されませんでした。</p>
ALERT_MESSAGE	<p>データ・タイプ: VARCHAR(1024)。 NULL 可能: いいえ。</p> <p>メッセージ・コードを含めて、送信されたメッセージのテキスト。</p>

IBMSNAP_CONDITIONS 表

サーバー: モニター・コントロール・サーバー

索引: MONITOR_QUAL、COMPONENT、SERVER_NAME、SCHEMA_OR_QUAL、SET_NAME、CONDITION_NAME

IBMSNAP_CONDITIONS 表には、レプリケーション・アラート・モニターから担当者への連絡が必要なアラート条件と、特定の条件が発生したときの連絡先のグループまたは個人の名前が含まれます。レプリケーション・アラート・モニターは、キャプチャー・コントロール・サーバー、アプライ・コントロール・サーバー、Q キャプチャー・サーバー、および Q アプライ・サーバー上の複数の条件の組み合わせをモニターできます。

表 75 では、IBMSNAP_CONDITIONS 表の列の要旨を示します。

表 75. IBMSNAP_CONDITIONS 表の列

列名	説明
SERVER_NAME	<p>データ・タイプ: CHAR(18)。 NULL 可能: いいえ。</p> <p>この条件がモニターされるキャプチャー・コントロール・サーバー、アプライ・コントロール・サーバー、Q キャプチャー・サーバー、または Q アプライ・サーバーの名前。</p>
COMPONENT	<p>データ・タイプ: CHAR(1)。 NULL 可能: いいえ。</p> <p>モニター対象のレプリケーション・コンポーネント。</p> <p>C キャプチャー・プログラム</p> <p>A アプライ・プログラム</p> <p>S Q キャプチャー・プログラム</p> <p>R Q アプライ・プログラム</p>

表 75. IBMSNAP_CONDITIONS 表の列 (続き)

列名	説明
SCHEMA_OR_QUAL	<p>データ・タイプ: VARCHAR(30)、DB2 UDB for z/OS バージョン 8 新機能モード・サブシステムの場合 VARCHAR(128)。 NULL 可能: いいえ。</p> <p>モニター対象のキャプチャー・スキーマ、アプライ・スキーマ、Q キャプチャー・スキーマ、または Q アプライ・スキーマ。</p>
SET_NAME	<p>データ・タイプ: CHAR(18)。 NULL 可能: いいえ。 デフォルト: 現行サブスクリプション・セット。</p> <p>アプライ・プログラムでアラート条件が設定されている場合、この列は、モニター対象のサブスクリプション・セットの名前を指定します。セット名が指定されない場合は、アプライ修飾子レベルでモニターが行われます。つまり、特定のアプライ修飾子を持つすべてのセットがモニターされます。</p> <p>Q アプライの受信キュー項目数か予備キュー項目数のアラート条件が設定されている場合、この列は、モニター対象の受信キューか予備キューの名前を指定します。</p>
MONITOR_QUAL	<p>データ・タイプ: CHAR(18)。 NULL 可能: いいえ。</p> <p>この条件について、キャプチャー・コントロール・サーバー、アプライ・コントロール・サーバー、Q キャプチャー・サーバー、または Q アプライ・サーバーをモニターしているレプリケーション・アラート・モニター・プログラムを識別するモニター修飾子。</p>
SERVER_ALIAS	<p>データ・タイプ: CHAR(8)。 NULL 可能: はい。</p> <p>この条件がモニターされるキャプチャー・コントロール・サーバー、アプライ・コントロール・サーバー、Q キャプチャー・サーバー、または Q アプライ・サーバーの DB2 UDB 別名。</p>
ENABLED	<p>データ・タイプ: CHAR(1)。 NULL 可能: いいえ。</p> <p>レプリケーション・アラート・モニターが次のモニター・サイクルでこの条件を処理するかどうかを示すフラグ。</p> <p>Y レプリケーション・アラート・モニターは次のモニター・サイクルでこの定義を処理します。</p> <p>N レプリケーション・アラート・モニターは次のモニター・サイクルでこの定義を無視します。</p>

IBMSNAP_CONDITIONS

表 75. IBMSNAP_CONDITIONS 表の列 (続き)

列名	説明
CONDITION_NAME	<p>データ・タイプ: CHAR(18)。 NULL 可能: いいえ。特定のキャプチャー・コントロール・サーバー、アプライ・コントロール・サーバー、Q キャプチャー・サーバー、または Q アプライ・サーバーで、レプリケーション・アラート・モニターがモニターしている条件の名前。キャプチャー・プログラムの条件は CAPTURE で始まります。アプライ・プログラムの条件は APPLY で始まります。Q キャプチャー・プログラムの条件は QCAPTURE で始まります。Q アプライ・プログラムの条件は QAPPLY で始まります。</p> <p>CAPTURE_STATUS キャプチャー・プログラムの状況。</p> <p>CAPTURE_ERRORS キャプチャー・プログラムがエラー・メッセージを通知するかどうか。</p> <p>CAPTURE_WARNINGS キャプチャー・プログラムが警告メッセージを通知するかどうか。</p> <p>CAPTURE_LASTCOMMIT 最後のモニター・サイクルでキャプチャー・プログラムが最後にデータをコミットした時刻。</p> <p>CAPTURE_CLATENCY キャプチャー・プログラムの現在の待ち時間。</p> <p>CAPTURE_HLATENCY キャプチャー・プログラムの待ち時間が特定の秒数を超えたかどうか。</p> <p>CAPTURE_MEMORY キャプチャー・プログラムが使用しているメモリーの量 (MB)。</p>

表 75. IBMSNAP_CONDITIONS 表の列 (続き)

列名	説明
CONDITION_NAME (続き)	<p>APPLY_STATUS アプライ・プログラムの状況。</p> <p>APPLY_SUBSFALING 失敗したサブスクリプション・セットがあるかどうか。</p> <p>APPLY_SUBSINACT 失敗した、または非アクティブのサブスクリプション・セットがあるかどうか。</p> <p>APPLY_ERRORS アプライ・プログラムがエラー・メッセージを通知するかどうか。</p> <p>APPLY_WARNINGS アプライ・プログラムが警告メッセージを通知するかどうか。</p> <p>APPLY_FULLREFRESH フル・リフレッシュが発生したかどうか。</p> <p>APPLY_REJTRANS (Update-anywhere) アプライ・プログラムがサブスクリプション・セットでトランザクションをリジェクトするかどうか。</p> <p>APPLY_SUBSDELAY アプライ・プログラムが、ユーザーが PARM_INT パラメーターで指定した時間よりも遅れるかどうか。</p> <p>APPLY_REWORKED アプライ・プログラムが、ターゲット表の行の再処理を行ったかどうか。</p> <p>APPLY_LATENCY アプライ・プログラムのエンドツーエンドの待ち時間がしきい値を超えるかどうか。</p>

IBMSNAP_CONDITIONS

表 75. IBMSNAP_CONDITIONS 表の列 (続き)

列名	説明
CONDITION_NAME (続き)	<p>QCAPTURE_STATUS Q キャプチャー・プログラムがダウンするかどうか。</p> <p>QCAPTURE_ERRORS Q キャプチャー・プログラムがエラー・メッセージを通知するかどうか。</p> <p>QCAPTURE_WARNINGS Q キャプチャー・プログラムが警告メッセージを通知するかどうか。</p> <p>QCAPTURE_LATENCY Q キャプチャーの待ち時間 (IBMQREP_CAPMON 表への最後の挿入と、 Q キャプチャー・プログラムが DB2 ログ中で読み取る最後のトランザクションのタイム・スタンプとの間の差) が、しきい値を超えるかどうか。</p> <p>QCAPTURE_MEMORY Q キャプチャー・プログラムが使用するメモリーがしきい値を超えるかどうか。</p> <p>QCAPTURE_TRANSIZE トランザクションが、IBMQREP_CAPMON 表に設定されている MAX_TRANS_SIZE (トランザクションの最大サイズ) を超えるかどうか。</p> <p>QCAPTURE_SUBSINACT Q サブスクリプションが I (非アクティブ) 状態に変更されたかどうか。</p>

表 75. IBMSNAP_CONDITIONS 表の列 (続き)

列名	説明
CONDITION_NAME (続き)	
QAPPLY_STATUS	Q アプライ・プログラムがダウンするかどうか。
QAPPLY_ERRORS	Q アプライ・プログラムがエラー・メッセージを通知するかどうか。
QAPPLY_WARNINGS	Q アプライ・プログラムが警告メッセージを通知するかどうか。
QAPPLY_LATENCY	キューの待ち時間 (メッセージが送信キューから受信キューに送られるのに要する時間) が、しきい値を超えるかどうか。
QAPPLY_EELATENCY	エンドツーエンドの待ち時間 (トランザクションがソースからターゲットに複製されるのに要する時間) が、しきい値を超えるかどうか。
QAPPLY_EXCEPTIONS	SQL エラーか競合のために、Q アプライが行を IBMQREP_EXCEPTIONS 表に挿入したかどうか。
QAPPLY_MEMORY	Q アプライ・プログラムが特定の受信キューからメッセージを読み取るのに使用したメモリーの量が、しきい値を超えるかどうか。
QAPPLY_SPILLQDEPTH	予備キュー上のメッセージの数がしきい値を超えるかどうか。
QAPPLY_QDEPTH	受信キュー上のメッセージの数がしきい値を超えるかどうか。

IBMSNAP_CONDITIONS

表 75. IBMSNAP_CONDITIONS 表の列 (続き)

列名	説明
PARM_INT	<p>データ・タイプ: INT。 NULL 可能: はい。</p> <p>条件の整数パラメーター。この列の値は、CONDITION_NAME 列の値によって異なります。</p> <p>CAPTURE_LASTCOMMIT しきい値 (秒単位)。</p> <p>CAPTURE_CLATENCY しきい値 (秒単位)。</p> <p>CAPTURE_HLATENCY しきい値 (秒単位)。</p> <p>CAPTURE_MEMORY しきい値 (MB 単位)。</p> <p>APPLY_SUBSDELAY しきい値 (秒単位)。</p> <p>APPLY_REWORKED 再処理される行のしきい値。</p> <p>APPLY_LATENCY しきい値 (秒単位)。</p> <p>QCAPTURE_LATENCY しきい値 (秒単位)</p> <p>QCAPTURE_MEMORY しきい値 (MB 単位)</p> <p>QCAPTURE_TRANSIZE しきい値 (MB 単位)</p> <p>QAPPLY_EELATENCY しきい値 (秒単位)</p> <p>QAPPLY_LATENCY しきい値 (秒単位)</p> <p>QAPPLY_MEMORY しきい値 (MB 単位)</p> <p>QAPPLY_SPILLQDEPTH メッセージ数のしきい値。</p> <p>QAPPLY_QDEPTH メッセージ数のしきい値。</p>

表 75. IBMSNAP_CONDITIONS 表の列 (続き)

列名	説明
PARM_CHAR	<p>データ・タイプ: VARCHAR(128)。 NULL 可能: はい。</p> <p>条件の文字パラメーター。この列には、条件が使用する追加のストリングが保持されます。</p> <p>CAPTURE_STATUS および APPLY_STATUS 条件は、この列の値を使用します。この列の値は、次の 3 つのパラメーターをコンマで区切って連結したストリングになります。</p> <ul style="list-style-type: none"> • キャプチャー・サーバーまたはアプライ・コントロール・サーバー • リモート DB2 インスタンス名 (サーバーがリモートの場合のみ) • リモート・ホスト名 <p>値が NULL またはゼロ長ストリングの場合、モニター・プログラムは次のデフォルトを使用します。</p> <ul style="list-style-type: none"> • キャプチャーまたはアプライ・コントロール・サーバーの CURRENT SERVER 値。 • UNIX サーバー上のリモート DB2 インスタンス名の値。この値は、サーバーの接続時に使用されたユーザー ID の名前です。Windows サーバーでは、この値は "DB" です。 • DB2 ノード・ディレクトリーのホスト名の値。
CONTACT_TYPE	<p>データ・タイプ: CHAR(1)。 NULL 可能: いいえ。</p> <p>この条件が発生したときに個人またはグループに連絡するかどうかを示すフラグ。</p> <p>C 個人連絡先</p> <p>G 連絡先のグループ</p>
CONTACT	<p>データ・タイプ: VARCHAR(127)。 NULL 可能: いいえ。</p> <p>この条件が発生したときに通知する個人連絡先、または連絡先のグループ。</p>

IBMSNAP_CONTACTGRP 表

サーバー: モニター・コントロール・サーバー

索引: GROUP_NAME、CONTACT_NAME

IBMSNAP_CONTACTGRP 表には、連絡先グループを構成する個人の連絡先が入っています。アラート条件が発生したときに、これらの個人のグループにレプリケーション・アラート・モニターから連絡がいくように指定できます。1 個人は、複数の連絡先グループに所属できます (列はユニークではありません)。

482 ページの表 76 では、IBMSNAP_CONTACTGRP 表の列の要旨を示します。

IBMSNAP_CONTACTGRP

表 76. IBMSNAP_CONTACTGRP 表の列

列名	説明
GROUP_NAME	データ・タイプ: VARCHAR(127)。 NULL 可能: いいえ。 連絡先グループの名前。
CONTACT_NAME	データ・タイプ: VARCHAR(127)。 NULL 可能: いいえ。 グループの一部である連絡先の名前。これらの個人は、モニター連絡先 (IBMSNAP_CONTACTS) 表で指定されます。

IBMSNAP_CONTACTS 表

サーバー: モニター・コントロール・サーバー

索引: CONTACT_NAME

IBMSNAP_CONTACTS 表は、個人 (またはそのグループ) に関連付けられたアラート条件が発生したときに、レプリケーション・アラート・モニターから個人に通知を行うために必要となる情報が入っています。1 行に 1 人指定します。

表 77 では、IBMSNAP_CONTACTS 表の列の要旨を示します。

表 77. IBMSNAP_CONTACTS 表の列

列名	説明
CONTACT_NAME	データ・タイプ: VARCHAR(127)。 NULL 可能: いいえ。 連絡先の名前。個人連絡先のみが許可されています。グループ名はサポートされていません。
EMAIL_ADDRESS	データ・タイプ: VARCHAR(128)。 NULL 可能: いいえ。 この連絡先の主な E メールまたはページのアドレス。
ADDRESS_TYPE	データ・タイプ: CHAR(1)。 NULL 可能: はい。 この連絡先の E メール・アドレスが、E メール・アカウントであるか、ページャー・アドレスであるかを示すフラグ。 E E メール・アドレスは E メール・アカウント用です。 P E メール・アドレスはページャー用です。
DELEGATE	データ・タイプ: VARCHAR(127)。 NULL 可能: はい。 代行期間中に通知を受ける連絡先の名前。個人連絡先名のみが許可されています。グループ名はサポートされていません。
DELEGATE_START	データ・タイプ: DATE。 NULL 可能: はい。 DELEGATE 列で指定された個人に通知を送信する場合の、代行期間の開始日。
DELEGATE_END	データ・タイプ: DATE。 NULL 可能: はい。 代行期間の終了日。
DESCRIPTION	データ・タイプ: VARCHAR(1024)。 NULL 可能: はい。 連絡先の説明。

IBMSNAP_GROUPS 表

サーバー: モニター・コントロール・サーバー

索引: GROUP_NAME

IBMSNAP_GROUPS 表には、各連絡先グループの名前と記述が入っています。1 つの行に 1 つのグループが指定されます。

表 78 では、IBMSNAP_GROUPS 表の列の要旨を示します。

表 78. IBMSNAP_GROUPS 表の列

列名	説明
GROUP_NAME	データ・タイプ: VARCHAR(127)。 NULL 可能: はい。 連絡先グループの名前。
DESCRIPTION	データ・タイプ: VARCHAR(1024)。 NULL 可能: はい。 連絡先グループの記述。

IBMSNAP_MONENQ 表

サーバー: モニター・コントロール・サーバー

索引: MONITOR_QUAL

IBMSNAP_MONENQ 表は、DB2 レプリケーションの将来のオプション用に予約済みです。

表 79 では、IBMSNAP_MONENQ 表の列の要旨を示します。

表 79. IBMSNAP_MONENQ 表の列

列名	説明
MONITOR_QUAL	データ・タイプ: CHAR(18)。 NULL 可能: いいえ。 DB2 レプリケーションの将来のオプション用に予約済み。

IBMSNAP_MONPARMS 表

サーバー: モニター・コントロール・サーバー

索引: MONITOR_QUAL

デフォルト・スキーマ: ASN

この表の情報は、SQL を使って更新できます。

IBMSNAP_MONPARMS 表は、レプリケーション・アラート・モニターの操作をコントロールするためにユーザーが変更できるパラメーターを保持します。これらのパラメーターを定義して、モニター・プログラムが整理を行う前にデータを CD 表および UOW 表の中に保持する時間の長さや、アラート条件が満たされるたびにモ

IBMSNAP_MONPARMS

モニター・プログラムが受け取る通知メッセージの数などを設定できます。ユーザーがこの表のパラメーターを変更しても、モニター・プログラムは始動時にしか変更を読み取りません。

表 80 では、IBMSNAP_MONPARMS 表の列の要旨を示します。

表 80. IBMSNAP_MONPARMS 表の列

列名	説明
MONITOR_QUAL	<p>データ・タイプ: CHAR(18)。 NULL 可能: いいえ。</p> <p>モニター修飾子は、これらのパラメーターの適用対象のレプリケーション・アラート・モニター・プログラムにパラメーターを一致させます。</p>
ALERT_PRUNE_LIMIT	<p>データ・タイプ: INT。 NULL 可能: デフォルトでは不可。 デフォルト: 10080 分 (7 日)。</p> <p>表から整理されるまでのデータの経過時間を示すフラグ。</p>
AUTOPRUNE	<p>データ・タイプ: CHAR(1)。 NULL 可能: デフォルトでは不可。 デフォルト: Y。</p> <p>モニター・プログラムが、不必要になった行を CD 表、UOW 表、シグナル表、トレース表、およびモニター表から自動的に除去するかどうかを示すフラグ。</p> <p>Y オート・プルーンはオン。</p> <p>N オート・プルーンはオフ。</p>
EMAIL_SERVER	<p>データ・タイプ: INT(128)。 NULL 可能: はい。</p> <p>SMTP プロトコルを使用する E メール・サーバーのアドレス。</p>
LOGREUSE	<p>データ・タイプ: CHAR(1)。 NULL 可能: デフォルトでは不可。 デフォルト: N。</p> <p>モニター・プログラムがモニター・ログ・ファイルに上書きするか、ファイルに追加するかを示すフラグ。</p> <p>Y モニター・プログラムは、最初にログ・ファイルを削除し、モニター・プログラムの再始動時にそれを再作成することにより、ログ・ファイルを再利用します。</p> <p>N モニター・プログラムは新しい情報をモニター・ログ・ファイルに追加します。</p>
LOGSTDOUT	<p>データ・タイプ: CHAR(1)。 NULL 可能: デフォルトでは不可。 デフォルト: N。</p> <p>モニター・プログラムがログ・ファイル・メッセージを送信するかどうかを示すフラグ。</p> <p>Y モニター・プログラムは、標準出力 (STDOUT) とログ・ファイルの両方にログ・ファイル・メッセージを送信します。</p> <p>N モニター・プログラムは、ほとんどのログ・ファイル・メッセージをログ・ファイルにのみ送ります。初期化メッセージは、標準出力 (STDOUT) とログ・ファイルの両方に送られます。</p>
NOTIF_PER_ALERT	<p>データ・タイプ: INT。 NULL 可能: デフォルトでは不可。 デフォルト: 3。</p> <p>アラート条件が満たされた場合に送信される通知メッセージの数。</p>
NOTIF_MINUTES	<p>データ・タイプ: INT。 NULL 可能: デフォルトでは不可。 デフォルト: 60。</p> <p>アラート条件が満たされた場合に通知メッセージを受け取る分数。</p>

表 80. IBMSNAP_MONPARMS 表の列 (続き)

列名	説明
MONITOR_ERRORS	<p>データ・タイプ: VARCHAR(128)。 NULL 可能: はい。</p> <p>レプリケーション・アラート・モニターの操作に関連するエラーが発生したときに通知メッセージを送信する E メール・アドレスを指定します。</p>
MONITOR_INTERVAL	<p>データ・タイプ: INT。 NULL 可能: デフォルトでは不可。 デフォルト: 300 (5 分)。</p> <p>レプリケーション・アラート・モニターが、選択されたアラート条件をモニターする頻度 (秒単位)。</p>
MONITOR_PATH	<p>データ・タイプ: VARCHAR(1040)。 NULL 可能: はい。</p> <p>モニター・プログラムからの出力が送信されるパス。</p>
RUNONCE	<p>データ・タイプ: CHAR(1)。 NULL 可能: デフォルトでは不可。 デフォルト: N。</p> <p>モニター・プログラムが、選択されたアラート条件をチェックするかどうかを示すフラグ。</p> <p>Y モニター・プログラムはアラート条件をチェックします。</p> <p>N モニター・プログラムはアラート条件をチェックしません。</p> <p>RUNONCE が y に設定されると、MONITOR_INTERVAL は無視されます。</p>
TERM	<p>データ・タイプ: CHAR(1)。 NULL 可能: デフォルトでは不可。 デフォルト: N。</p> <p>DB2 が静止したときにモニター・プログラムが終了するかどうかを示すフラグ。</p> <p>Y DB2 が終了するとモニター・プログラムは終了します。</p> <p>N モニター・プログラムはアクティブのまま、DB2 が再始動されるのを待ちます。</p>
TRACE_LIMIT	<p>データ・タイプ: INT。 NULL 可能: デフォルトでは不可。 デフォルト: 10080。</p> <p>整理の対象となるまでに、行が IBMSNAP_MONTRACE 表の中に留まる分数。整理プロセス時に、分数 (現在のタイム・スタンプから IBMSNAP_MONTRACE 表に行が挿入された時刻を引いたもの) が TRACE_LIMIT の値を超えると、モニター・トレース表の中の行を削除します。</p>

IBMSNAP_MONSERVERS 表

サーバー: モニター・コントロール・サーバー

索引: MONITOR_QUAL、SERVER_NAME

IBMSNAP_MONSERVERS 表は、レプリケーション・アラート・モニターが、キャプチャー・コントロール・サーバー、アプライ・コントロール・サーバー、Q キャプチャー・サーバー、または Q アプライ・サーバーを最後にモニターしたときの情報を保持します。

486 ページの表 81 では、IBMSNAP_MONSERVERS 表の列の要旨を示します。

IBMSNAP_MONSERVERS

表 81. IBMSNAP_MONSERVERS 表の列

列名	記述
MONITOR_QUAL	<p>データ・タイプ: CHAR(18)。 NULL 可能: いいえ。</p> <p>キャプチャー・コントロール・サーバー、アプライ・コントロール・サーバー、Q キャプチャー・サーバー、または Q アプライ・サーバーをモニターしているレプリケーション・アラート・モニターを識別するモニター修飾子。</p>
SERVER_NAME	<p>データ・タイプ: CHAR(18)。 NULL 可能: いいえ。</p> <p>レプリケーション・アラート・モニターによるモニター対象のキャプチャー・コントロール・サーバー、アプライ・コントロール・サーバー、Q キャプチャー・サーバー、または Q アプライ・サーバーの名前。</p>
SERVER_ALIAS	<p>データ・タイプ: CHAR(8)。 NULL 可能: はい。</p> <p>レプリケーション・アラート・モニターによるモニター対象のキャプチャー・コントロール・サーバー、アプライ・コントロール・サーバー、Q キャプチャー・サーバー、または Q アプライ・サーバーの DB2 UDB 別名。</p>
LAST_MONITOR_TIME	<p>データ・タイプ: TIMESTAMP。 NULL 可能: はい。</p> <p>レプリケーション・アラート・モニターが、キャプチャー・コントロール・サーバー、アプライ・コントロール・サーバー、Q キャプチャー・サーバー、または Q アプライ・サーバーに最後に接続したときの時刻 (このサーバーの)。この値は、コントロール表からメッセージをフェッチする下限値として使用され、最後に成功したモニター・サイクルの START_MONITOR_TIME の値と同じ値になります。</p>
START_MONITOR_TIME	<p>データ・タイプ: TIMESTAMP。 NULL 可能: はい。</p> <p>レプリケーション・アラート・モニターが、キャプチャー・コントロール・サーバー、アプライ・コントロール・サーバー、Q キャプチャー・サーバー、または Q アプライ・サーバーに接続したときの時刻 (キャプチャー・コントロール・サーバー、アプライ・コントロール・サーバー、Q キャプチャー・サーバー、または Q アプライ・サーバーの)。この値は、コントロール表からアラート・メッセージをフェッチする上限値として使用されます。</p>
END_MONITOR_TIME	<p>データ・タイプ: TIMESTAMP。 NULL 可能: はい。</p> <p>レプリケーション・アラート・モニターが、キャプチャー・コントロール・サーバー、アプライ・コントロール・サーバー、Q キャプチャー・サーバー、または Q アプライ・サーバーのモニターを終了したときの時刻 (このサーバーの)。</p>
LASTRUN	<p>データ・タイプ: TIMESTAMP。 NULL 可能: いいえ。</p> <p>レプリケーション・アラート・モニターが、キャプチャー・コントロール・サーバー、アプライ・コントロール・サーバー、Q キャプチャー・サーバー、または Q アプライ・サーバーの処理を最後に開始したときの時刻 (モニター・コントロール・サーバーの)。</p>
LASTSUCCESS	<p>データ・タイプ: TIMESTAMP。 NULL 可能: はい。</p> <p>レプリケーション・アラート・モニターが、キャプチャー・コントロール・サーバー、アプライ・コントロール・サーバー、Q キャプチャー・サーバー、または Q アプライ・サーバーの処理を最後に正常に完了した時刻の (モニター・コントロール・サーバーの) LASTRUN 列の値。このサーバーのモニターが繰り返し失敗するときには、この値が同じである可能性があります (この列の履歴は IBMSNAP_MONTRAIL 表にあります)。</p>

表 81. IBMSNAP_MONSERVERS 表の列 (続き)

列名	記述
STATUS	<p>データ・タイプ: SMALLINT。NULL 可能: いいえ。</p> <p>モニター・サイクルの状況を示すフラグ。</p> <p>-1 レプリケーション・アラート・モニターは、このサーバーを正常に処理できませんでした。</p> <p>0 レプリケーション・アラート・モニターは、このサーバーを正常に処理しました。</p> <p>1 レプリケーション・アラート・モニターは、このサーバーを現在処理中です。</p>

IBMSNAP_MONTRACE 表

サーバー: モニター・コントロール・サーバー

索引: MONITOR_QUAL、TRACE_TIME

IBMSNAP_MONTRACE 表には、レプリケーション・アラート・モニターの監査証跡情報が含まれます。モニター・プログラムによる処理はすべてこの表に記録されるため、モニター・プログラムの問題が発生した場合は、この表を参照すると便利です。

表 82 では、IBMSNAP_MONTRACE 表の列の要旨を示します。

表 82. IBMSNAP_MONTRACE 表の列

列名	記述
MONITOR_QUAL	<p>データ・タイプ: CHAR(18)。NULL 可能: いいえ。</p> <p>メッセージを発行したレプリケーション・アラート・モニターを示すモニター修飾子。</p>
TRACE_TIME	<p>データ・タイプ: TIMESTAMP。NULL 可能: いいえ。</p> <p>メッセージがこの表に挿入されたときのタイム・スタンプ。</p>
OPERATION	<p>データ・タイプ: CHAR(8)。NULL 可能: いいえ。</p> <p>メッセージを分類するために使用される値。</p> <p>ERROR エラー・メッセージ</p> <p>WARNING 警告メッセージ</p> <p>INFO 情報メッセージ</p>
DESCRIPTION	<p>データ・タイプ: VARCHAR(1024)。NULL 可能: いいえ。</p> <p>メッセージ・コードおよびテキスト。</p>

IBMSNAP_MONTRAIL 表

サーバー: モニター・コントロール・サーバー

索引: なし

IBMSNAP_MONTRAIL 表には、各モニター・サイクルに関する情報が入ります。レプリケーション・アラート・モニターは、モニター対象のキャプチャー・コントロール・サーバー、アプライ・コントロール・サーバー、Q キャプチャー・サーバー、および Q アプライ・サーバーごとに行を 1 つ挿入します。

表 83 では、IBMSNAP_MONTRAIL 表の列の要旨を示します。

表 83. IBMSNAP_MONTRAIL 表の列

列名	説明
MONITOR_QUAL	<p>データ・タイプ: CHAR(18)。NULL 可能: いいえ。</p> <p>キャプチャー・コントロール・サーバー、アプライ・コントロール・サーバー、Q キャプチャー・サーバー、または Q アプライ・サーバーをモニターしているレプリケーション・アラート・モニターを識別するモニター修飾子。</p>
SERVER_NAME	<p>データ・タイプ: CHAR(18)。NULL 可能: いいえ。</p> <p>レプリケーション・アラート・モニターによるモニター対象のキャプチャー・コントロール・サーバー、アプライ・コントロール・サーバー、Q キャプチャー・サーバー、または Q アプライ・サーバーの名前。</p>
SERVER_ALIAS	<p>データ・タイプ: CHAR(8)。NULL 可能: はい。</p> <p>レプリケーション・アラート・モニターによるモニター対象のキャプチャー・コントロール・サーバー、アプライ・コントロール・サーバー、Q キャプチャー・サーバー、または Q アプライ・サーバーの DB2 UDB 別名。</p>
STATUS	<p>データ・タイプ: SMALLINT。NULL 可能: いいえ。</p> <p>モニター・サイクルの状況を示すフラグ。</p> <p>-1 レプリケーション・アラート・モニターは、このサーバーを正常に処理できませんでした。</p> <p>0 レプリケーション・アラート・モニターは、このサーバーを正常に処理しました。</p> <p>1 レプリケーション・アラート・モニターは、このサーバーを現在処理中です。</p>
LASTRUN	<p>データ・タイプ: TIMESTAMP。NULL 可能: いいえ。</p> <p>レプリケーション・アラート・モニター・プログラムが、キャプチャー・コントロール・サーバー、アプライ・コントロール・サーバー、Q キャプチャー・サーバー、または Q アプライ・サーバーの処理を最後に開始したときの時刻 (モニター・コントロール・サーバーの)。</p>
LASTSUCCESS	<p>データ・タイプ: TIMESTAMP。NULL 可能: はい。</p> <p>レプリケーション・アラート・モニターが、キャプチャー・コントロール・サーバー、アプライ・コントロール・サーバー、Q キャプチャー・サーバー、または Q アプライ・サーバーの処理を最後に正常に完了したときの時刻 (モニター・コントロール・サーバーの)。</p>

表 83. IBMSNAP_MONTRAIL 表の列 (続き)

列名	説明
ENDTIME	<p>データ・タイプ: TIMESTAMP。 NULL 可能: デフォルトでは不可。デフォルト: 現行タイム・スタンプ。</p> <p>この表にこの行が挿入された時刻。</p>
LAST_MONITOR_TIME	<p>データ・タイプ: TIMESTAMP。 NULL 可能: はい。</p> <p>レプリケーション・アラート・モニターが、キャプチャー・コントロール・サーバー、アプライ・コントロール・サーバー、Q キャプチャー・サーバー、または Q アプライ・サーバーに最後に接続したときの時刻 (キャプチャー・コントロール・サーバー、アプライ・コントロール・サーバー、Q キャプチャー・サーバー、または Q アプライ・サーバーの)。この値は、コントロール表からメッセージをフェッチする下限値として使用され、直前の成功したモニター・サイクルの START_MONITOR_TIME の値と同じ値になります。</p>
START_MONITOR_TIME	<p>データ・タイプ: TIMESTAMP。 NULL 可能: はい。</p> <p>レプリケーション・アラート・モニターが、キャプチャー・コントロール・サーバー、アプライ・コントロール・サーバー、Q キャプチャー・サーバー、または Q アプライ・サーバーのモニターを最後に開始した時刻。</p>
END_MONITOR_TIME	<p>データ・タイプ: TIMESTAMP。 NULL 可能: はい。</p> <p>レプリケーション・アラート・モニターが、キャプチャー・コントロール・サーバー、アプライ・コントロール・サーバー、Q キャプチャー・サーバー、または Q アプライ・サーバーのモニターを最後に終了した時刻。</p>
SQLCODE	<p>データ・タイプ: INT。 NULL 可能: はい。</p> <p>このモニター・サイクル中に発生したエラーの SQLCODE。</p>
SQLSTATE	<p>データ・タイプ: CHAR(5)。 NULL 可能: はい。</p> <p>このモニター・サイクル中に発生したエラーの SQLSTATE。</p>
NUM_ALERTS	<p>データ・タイプ: INT。 NULL 可能: いいえ。</p> <p>このモニター・サイクル中に発生したアラート条件の数。</p>
NUM_NOTIFICATIONS	<p>データ・タイプ: INT。 NULL 可能: いいえ。</p> <p>このモニター・サイクル中に送信された通知の数。</p>

ピアツーピア・レプリケーションのバージョン管理用追加列の詳細な構造

ピアツーピア・レプリケーション・シナリオの一部であるユーザーのソースおよびターゲット表には、2 つのバージョンの列、つまり、タイム・スタンプ列と短整数列が必要です。この列は、トリガーにより保守されます。この 2 つの列を使用して、Q キャプチャーおよび Q アプライ・プログラムは、ピアツーピア・レプリケーションで必要な、バージョンに基づく競合の調査を実施できます。この列はまた、この 2 つのプログラムを使用して競合を解決し、Q サブスクリプション・グループ内にある表の収束を保守できます。この列の値は、行の最新のバージョンを反映しています。

レプリケーション・センターを使用して、ピアツーピア・レプリケーションの Q サブスクリプションを作成する時に、このレプリケーション列およびトリガーは、余分に作成されます。ピアツーピア・レプリケーションの Q サブスクリプションを作成する場合、この列の両方をサブスクライブしなければなりません。

表 84 では、ピアツーピア・レプリケーションに必要なユーザー表での追加の列を簡単に説明しています。

表 84. ピアツーピア・レプリケーションは、追加の列が必要です。

列名	説明
ibmqrepVERNODE	<p>データ・タイプ: SMALLINT。 NULL 可能: デフォルトでは不可</p> <p>ピアツーピア・グループ内でこの表があるデータベースまたはサブシステムを識別する数。デフォルト: 0</p>
ibmqrepVERTIME	<p>データ・タイプ: TIMESTAMP。 NULL 可能: デフォルトでは不可</p> <p>表で変更が生じた時刻を記録するタイム・スタンプ。デフォルト: 0001-01-01-00.00.00</p>

関連概念:

- 136 ページの『ピアツーピア・レプリケーション』
- 415 ページの『Q レプリケーションおよびイベント発行のコントロール表 -- 概要』

第 27 章 イベント発行の XML メッセージの構造

イベント発行の XML メッセージの構造 -- 概要

イベント発行で、Q キャプチャー・プログラムとユーザー・アプリケーションは、XML メッセージを交換します。

続くトピックでは、交換する XML メッセージとその構造について詳しく説明しています。

- 『XML メッセージ・タイプおよび要件 -- 概要』
- 493 ページの『Q キャプチャーからユーザー・アプリケーションへの XML メッセージの構造 -- 概要』
- 521 ページの『ユーザー・アプリケーションから Q キャプチャーへの XML メッセージの構造 -- 概説』

XML メッセージ・タイプおよび要件

XML メッセージ・タイプおよび要件 -- 概要

Q キャプチャー・プログラムは、XML メッセージを使用して、ユーザー・アプリケーションにトランザクションまたは行レベルの変更を送信します。Q キャプチャー・プログラムおよびユーザー・アプリケーションも XML メッセージを使用して通信します。以下のトピックでは、XML メッセージのタイプとその技術要件を説明します。

- 『XML メッセージ・タイプ』
- 492 ページの『XML メッセージの技術要件』
- 493 ページの『文字データで XML 区切り文字を処理する方法』

関連概念:

- 9 ページの『イベント発行』
- 491 ページの『イベント発行の XML メッセージの構造 -- 概要』

XML メッセージ・タイプ

Q キャプチャー・プログラムは、ユーザー・アプリケーションにデータ・メッセージと通知メッセージを送信し、ユーザー・アプリケーションは、Q キャプチャー・プログラムに制御メッセージを送信します。492 ページの表 85 では、3 つのタイプのメッセージについて説明します。

表 85. Q キャプチャー・プログラムおよびユーザー・アプリケーションが送信する XML メッセージ。

メッセージのタイプ	向き	説明
データ	Q キャプチャーからユーザー・アプリケーション	ソース表から以下のうち 1 つを含んでいる。 <ul style="list-style-type: none"> すべてまたは一部のトランザクション 単一行の操作 トランザクション中の行操作からのすべてまたは一部のラージ・オブジェクト (LOB) 値
通知	Q キャプチャーからユーザー・アプリケーション	Q キャプチャー・プログラムまたは XML 発行の状況に関する情報を提供する。
制御	ユーザー・アプリケーションから Q キャプチャー	Q キャプチャー・プログラムに XML 発行のアクティブ化または非アクティブ化、送信キューの無効化、またはターゲット表のロードの確認を要求する。

関連概念:

- 19 ページの『Q アプライ・プログラム』
- 15 ページの『Q キャプチャー・プログラム』
- 9 ページの『イベント発行』
- 491 ページの『XML メッセージ・タイプおよび要件 -- 概要』

関連資料:

- 521 ページの『ユーザー・アプリケーションから Q キャプチャーへの XML メッセージのリスト』
- 494 ページの『Q キャプチャーからユーザー・アプリケーションへの XML メッセージのリスト』

XML メッセージの技術要件

Q キャプチャー・プログラムは、以下の指針に従い、メッセージを XML 文書インスタンスの書式で生成します。

- メッセージは、XML 1.0 (第 2 版) の W3C 勧告 (2000 年 10 月 6 日) で指定されているように、UTF-8 (コード・ページ 1208) を使用してユニコードでエンコードされます。
- メッセージの構造は、XML スキーマ言語 (パート 1: 構造、パート 2: データ・タイプ) の W3C 勧告 (2001 年 5 月 2 日) に準拠しています。

ソース・データベースからの変更は、DB2[®] UDB Version 8.2 出荷時の IBM[®] International Components for Unicode (ICU4C) のバージョンを使用してメッセージに変換されます。

サブスクリプションするアプリケーションからの制御メッセージを解釈するため、Q キャプチャー・プログラムは、IBM XML パーサーの XML4C Version 5.3 を使用します。

関連参照

- Extensible Markup Language (XML) 1.0 (Second Edition)
- XML Schema Part 1: Structures
- XML Schema Part 2: Datatypes
- International Components for Unicode

関連概念:

- 491 ページの『XML メッセージ・タイプおよび要件 -- 概要』

文字データで XML 区切り文字を処理する方法

Q キャプチャー・プログラムからのデータ・メッセージでは、サブスクライブした列からの値は、列データ・タイプを説明する XML タグの間に表示されます。たとえば、ソース表の値 222 および Hello は、`<integer>222</integer>` および `<varchar>Hello</varchar>` とエンコードされます。

不等号括弧 (< または >) およびアンパーサンド (&) 文字は、XML 区切り文字に事前定義され、Q キャプチャー・プログラムは、次のように、列値内にあるとき、この文字を変換します。

- < を < に
- > を > に
- & を & に

また、アポストロフィ (') または、二重引用符 (") が、属性値内に表示されるとき、Q キャプチャー・プログラムは、これらの文字を以下のように変換します。

- ' を ' に
- " を " に

結果メッセージは、有効な XML 文書インスタンスです。

関連概念:

- 491 ページの『XML メッセージ・タイプおよび要件 -- 概要』

Q キャプチャーからユーザー・アプリケーションへの XML メッセージの構造

Q キャプチャーからユーザー・アプリケーションへの XML メッセージの構造 -- 概要

Q キャプチャー・プログラムは、ユーザー・アプリケーションにデータ・メッセージと通知メッセージの両方を送信します。このデータ・メッセージは、XML 発行の一部であるソース表に変更を伝達します。通知メッセージは、制御メッセージ経由でユーザー・アプリケーションの要求を確認するか、Q キャプチャー・プログラムの状況について報告するか、または XML 発行エラーを報告します。

続くトピックでは、Q キャプチャー・プログラムからユーザー・アプリケーションへのデータ・メッセージおよび通知メッセージの XML 構造を説明しています。

- 『Q キャプチャーからユーザー・アプリケーションへの XML メッセージのリスト』
- 495 ページの『msg: Q キャプチャーからユーザー・アプリケーションへの XML メッセージのルート・エレメント』
- 497 ページの『トランザクション・メッセージ』
- 506 ページの『行操作メッセージ』
- 508 ページの『ラージ・オブジェクト (LOB) メッセージ』
- 511 ページの『サブスクリプション非アクティブ化メッセージ』
- 512 ページの『ロード完了受信メッセージ』
- 513 ページの『エラー・レポート・メッセージ』
- 514 ページの『ハートビート・メッセージ』
- 515 ページの『サブスクリプション・スキーマ・メッセージ (subSchema)』
- 520 ページの『列の追加メッセージ』

関連概念:

- 491 ページの『イベント発行の XML メッセージの構造 -- 概要』
- 521 ページの『ユーザー・アプリケーションから Q キャプチャーへの XML メッセージの構造 -- 概説』

Q キャプチャーからユーザー・アプリケーションへの XML メッセージのリスト

Q キャプチャー・プログラムは、以下の 2 つのタイプの XML メッセージをユーザー・アプリケーションに送信します。

データ・メッセージ

ソース表への変更が含まれている。表 86 には、データ・メッセージのタイプの早見表がある。

通知メッセージ

Q キャプチャー・プログラムまたは XML 発行の状況の報告 495 ページの表 87 には、通知メッセージのタイプの早見表がある。

表 86. Q キャプチャー・プログラムからユーザー・アプリケーションへのデータ・メッセージ

メッセージ・タイプ	説明
トランザクション	ソース表への 1 回以上の挿入、削除、または更新操作が含まれる。これらの操作は同一のデータベース・トランザクションに属する。トランザクションのコミット情報も含まれる。
行操作	ソース表への 1 回の挿入、削除、または更新操作が含まれる。この行がその一部となっているデータベース・トランザクションのコミット情報も含まれる。
ラージ・オブジェクト (LOB)	ソース表の LOB 値からのデータの一部またはすべてが含まれる。LOB メッセージは、LOB 値が属しているトランザクション・メッセージおよび行操作メッセージとは別に送信される。

表 87. Q キャプチャー・プログラムからユーザー・アプリケーションへの通知メッセージ

メッセージ・タイプ	説明
サブスクリプションの非アクティブ化	ユーザー・アプリケーションに、Q キャプチャー・プログラムが XML 発行を非アクティブにすることを通知する。
ロード完了受信	Q キャプチャー・プログラムがターゲット表をロードしたというメッセージを受信したことを確認する。
エラー・レポート	ユーザー・アプリケーションに、Q キャプチャー・プログラムが XML 発行エラーを検出したことを通知する。
heartbeat	ユーザー・アプリケーションに、送信するデータ・メッセージがないが、Q キャプチャー・プログラムが実行を継続中であることを通知する。
サブスクリプション・スキーマ	ソース表およびソース表の列の情報を含める。また、データ送信オプション、送信キュー名、および Q キャプチャー・プログラムとソース・データベースの情報を含める。
列の追加	既存の XML 発行に追加した列の情報を含める。

関連概念:

- 15 ページの『Q キャプチャー・プログラム』
- 9 ページの『イベント発行』
- 491 ページの『イベント発行の XML メッセージの構造 -- 概要』

関連資料:

- 521 ページの『ユーザー・アプリケーションから Q キャプチャーへの XML メッセージのリスト』

msg: Q キャプチャーからユーザー・アプリケーションへの XML メッセージのルート・エレメント

msg エレメントは、Q キャプチャー・プログラムからユーザー・アプリケーションへのすべてのデータ・メッセージと通知メッセージのルート・エレメントです。

表 88 では、msg エレメントについて説明します。

表 88. msg エレメント (ユーザー・アプリケーションに対する Q キャプチャー・プログラム)

名前	プロパティ
msg	空でない、複合タイプ、複合内容

構造:

```
<xml version="1.0" encoding="UTF-8" ?>
<msg xmlns:xsi="XML_schema_instance"
      xsi:noNamespaceSchemaLocation="schema_document"
      version="version" dbName="database_name">
```

elements

</msg>

詳細:

XML_schema_instance

XML スキーマ・インスタンスの URL。イベント発行で、URL は、www.w3.org/2001/XMLSchema-instance です。XML データ・タイプ: ストリング。

schema_document

XML スキーマ文書のファイル名。XML ネームスペースは、メッセージが 1 つの XML スキーマしか参照しないので、イベント発行ではサポートされていません。Q キャプチャー・プログラムからユーザー・アプリケーションへのメッセージは、mqcap.xsd スキーマ文書を参照します。XML データ・タイプ: ストリング。

version

XML スキーマ・インスタンスのバージョン。DB2 UDB バージョン 8.2 では、このバージョンは 1.0.0 です。XML データ・タイプ: ストリング。

database_name

ソース・データベースまたはサブシステムの名前。XML データ・タイプ: ストリング。

elements

msg エLEMENTが入っているELEMENTの 1 つ。以下のようにメッセージごとにこれらのELEMENTのうち 1 つだけが表示されます。

- trans
- rowOp
- lob
- subDeactivated
- loadDoneRcvd
- heartbeat
- errorRpt
- subSchema

例:

以下の例は、メッセージ・ELEMENTを示しています。

```
<xml version="1.0" encoding="UTF-8" ?>
<msg xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="mqcap.xsd"
      version="1.0.0" dbName="DB1">
```

elements

</msg>

elements は、trans、rowOp、lob、subDeactivated、loadDoneRcvd、heartbeat、errorRpt、または subSchema のELEMENTの 1 つを表しています。

関連概念:

- 493 ページの『Q キャプチャーからユーザー・アプリケーションへの XML メッセージの構造 -- 概要』

関連資料:

- 494 ページの『Q キャプチャーからユーザー・アプリケーションへの XML メッセージのリスト』

トランザクション・メッセージ

transaction メッセージには、ソース表に対する 1 つ以上の行の挿入、更新、または削除操作が含まれています。transaction メッセージには、ソース・データベースでトランザクションがコミットされた時刻の情報、および時刻に基づくログ・シーケンス番号も含まれています。

transaction メッセージが送信キューに定義された最大メッセージ・サイズを超えると、Q キャプチャー・プログラムは、それを複数の transaction メッセージに分割できます。分割済みトランザクションの各メッセージは、トランザクション・エレメント (trans) のセグメント番号属性を使用して、番号付けされます。分割済みトランザクションのメッセージはすべて、コミット時間とコミット論理シーケンス番号が同一の値を共有します。

transaction メッセージ内で、trans エレメントには、行操作のタイプ、各列の属性、列値のデータ・タイプ、および値そのものを記述した他のエレメントの階層が入っています。続くセクションでは、trans エレメントが入ったエレメントを説明します。

- 『トランザクション・エレメント (trans)』
- 499 ページの『行操作エレメント (insertRow、updateRow、および deleteRow)』
- 500 ページの『列エレメント (col)』
- 502 ページの『single-column-value エレメント』
- 504 ページの『double-column-value エレメント』
- 505 ページの『変更前の値および変更後の値エレメント (beforeVal および afterVal)』

トランザクション・エレメント (trans)

トランザクション・エレメント (trans) は、msg エレメントに含まれており、3 つの行操作エレメント (insertRow、updateRow、または deleteRow) のうち 1 つを保持しています。

表 89 では、trans エレメントについて説明します。

表 89. trans のエレメントの説明

名前	プロパティ
trans	空でない、複合タイプ、複合内容

構造:


```
<trans isLast="is_last_indicator" segmentNum="segment_number"
      cmitLSN="commit_logical_sequence_number" cmitTime="commit_time">
    elements
</trans>
```

詳細:

is_last_indicator

transaction メッセージがデータベース・トランザクションの最終メッセージかどうかを示すブール値。最終メッセージである場合、値は 1 (真) です。最終メッセージではない場合、値は 0 (偽) です。 XML データ・タイプ: ブール

データベース・トランザクションに LOB 列の行操作があり、発行される LOB 値があるなら、LOB 値は、最終 transaction メッセージの後に分離した LOB メッセージに送信されます。この場合、データベース・トランザクションの最終メッセージは最終 transaction メッセージではなく、LOB メッセージです。

segment_number

分割済み transaction メッセージでメッセージのセグメント番号を示す正の整数。 XML データ・タイプ: positiveInteger。

commit_logical_sequence_number

トランザクションの COMMIT ステートメントのコミット論理シーケンス番号 (時刻に基づくログ・シーケンス番号)。 XML データ・タイプ: スtring。

commit_time

マイクロ秒単位で設定したグリニッジ標準時 (GMT) を使用したトランザクションの COMMIT ステートメントのタイム・スタンプ。 XML データ・タイプ: dateTime

elements

各 trans エレメントには、以下のエレメントが 1 つ以上入ります。

- insertRow
- updateRow
- deleteRow

例:

続く例では、行の挿入、行の更新、または、行の削除のエレメントのうち 1 つ以上が入ったトランザクションを示しています。

```
<xml version="1.0" encoding="UTF-8" ?>
<msg xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="mqcap.xsd"
      version="1.0.0" dbName="DB1">
  <trans isLast="1" segmentNum="1" cmitLSN="0000:0000::0000:06d6:87ab"
        cmitTime="2003-10-31T12:12:12.000122">
    insertRow, updateRow, or deleteRow
  </trans>
</msg>
```

insertRow、*updateRow*、または *deleteRow* は、『行操作エレメント (*insertRow*、*updateRow*、および *deleteRow*)』で説明されているエレメントを表しています。

行操作エレメント (*insertRow*、*updateRow*、および *deleteRow*)

トランザクション・エレメントで、行操作エレメント(*insertRow*、*updateRow*、および *deleteRow*) は、ソース表の行で実行される SQL 操作のタイプを記述します。これらのエレメントにはそれぞれ、サブスライブする列への変更を記述する 1 つ以上の列エレメント (*col*) が入っています。

表 90 では *insertRow*、*deleteRow*、および *updateRow* のエレメントを説明します。

表 90. *insertRow*、*deleteRow*、および *updateRow* のエレメントの説明

名前	プロパティ
<i>insertRow</i>	空でない、複合タイプ、複合内容
<i>deleteRow</i>	空でない、複合タイプ、複合内容
<i>updateRow</i>	空でない、複合タイプ、複合内容

構造:

```
<insertRow subName="XML_publication_name" srcOwner="source_owner"
           srcName="source_name" rowNum="row_number" hasLOBCols="LOB_indicator">
    elements
</insertRow>
```

```
<deleteRow subName="XML_publication_name" srcOwner="source_owner"
           srcName="source_name" rowNum="row_number" hasLOBCols="LOB_indicator">
    elements
</deleteRow>
```

```
<updateRow subName="XML_publication_name" srcOwner="source_owner"
           srcName="source_name" rowNum="row_number" hasLOBCols="LOB_indicator">
    elements
</updateRow>
```

詳細:

XML_publication_name

この行操作が属している XML 発行の名前。 XML データ・タイプ: ストリング。

source_owner

行操作が開始されたソース表のスキーマ。 XML データ・タイプ: ストリング。

source_name

ソース表の名前。 XML データ・タイプ: ストリング。

row_number

行操作にラージ・オブジェクト (LOB) 列があると、この属性が生成されて、データベース・トランザクションの行操作の位置番号を識別します。この属性にデフォルト値はありません。XML データ・タイプ: `positiveInteger`。

LOB_indicator

行操作に LOB 列があるかどうかを示すブール値。LOB 列がある場合、値は 1 (真) です。デフォルト値は 0 (偽) です。XML データ・タイプ: ブール

elements

`insertRow`、`updateRow`、または `deleteRow` エlement に含まれる 1 つ以上の列エlement (`col`)。

例:

以下の例では、`transaction` メッセージ内の `insertRow`、`updateRow`、および `deleteRow` エlement を示しています。

```
<xml version="1.0" encoding="UTF-8" ?>
<msg xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="mqcap.xsd" version="1.0.0"
      dbName="DB1">
  <trans isLast="1" segmentNum="1" cmitLSN="0000:0000::0000:06d6:87ab"
        cmitTime="2003-10-31T12:12:12.000122">
    <insertRow subName="S1" srcOwner="USER1" srcName="T1">
      column_element
    </insertRow>
    <deleteRow subName="S1" srcOwner="USER1" srcName="T1">
      column_element
    </deleteRow>
    <updateRow subName="S1" srcOwner="USER1" srcName="T1">
      column_element
    </updateRow>
  </trans>
</msg>
```

`column_element` は、『列エlement (`col`)』で説明されている列エlementを表しています。

列エlement (`col`)

列エlement (`col`) は、ソース表にサブスクライブした列の名前を記述し、その列が、発行に使用するキーの一部であるかどうかも通知します。挿入または削除操作の `col` エlement には単一の値しか入っていません。更新操作の場合、XML 発行に指定した送信データのオプションに応じて、`col` エlement には変更前の値と変更後の値を含めることができます。

表 91 では、col エlement について説明します。

表 91. col のElement の説明

名前	プロパティ
col	空でない、複合タイプ、複合内容

構造:

```
<col name="column_name" isKey="key_indicator">  
    single_or_double_column_value  
</col>
```

詳細:

column_name

ソース表のサブスクライブした列名。XML データ・タイプ: ストリング。

key_indicator

オプション: この列が発行に使用するキーの一部であるかどうかを示すブール値。デフォルトは 0 (偽) です。キー列である場合、値は 1 (真) です。XML データ・タイプ: ブール

single_or_double_column_value

列Element がソース表での挿入または削除操作の一部である場合、single-column-value Element の 1 つが入ります。更新操作の場合、この列Element に変更前の値と変更後の値の両方をもつ 2 列の値を入れることができます。

例:

以下の例では、単一系列の値をもつ行操作と、2 列の値をもつ更新操作を示しています。

```
<xml version="1.0" encoding="UTF-8" ?>  
<msg xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
    xsi:noNamespaceSchemaLocation="mqcap.xsd" version="1.0.0"  
    dbName="DB1">  
    <trans isLast="1" segmentNum="1" cmitLSN="0000:0000::0000:06d6:87ab"  
        cmitTime="2003-10-31T12:12:12.000122">  
        <insertRow subName="S1" srcOwner="USER1" srcName="T1">  
            <col name="COL1" isKey="1">  
                single_column_value  
            </col>  
            <col name="COL2">  
                single_column_value  
            </col>  
        </insertRow>  
        <updateRow subName="S1" srcOwner="USER1" srcName="T1">  
            <col name="COL1" isKey="1">  
                double_column_value
```

```

        </col>
        <col name="COL2">
            double_column_value
        </col>
    </updateRow>
</trans>
</msg>

```

single_column_value は、『single-column-value エlement』で説明されるElementを表し、*double_column_value* は、504 ページの『double-column-value Element』で説明されるElementを表しています。

single-column-value Element

single-column-value Elementには、ソース表からの実際の値が入ります。Q キャプチャー・プログラムは、挿入および削除操作に single-column-value Elementを使用します。これらのElementは、ソース列のデータ・タイプから命名され、他のElementはもちません。ソース表からの値が NULL の場合、このElementは空になり、xsi:nil 属性は、1 (真) に設定されます。

表 92 では、single-column-value Elementについて説明します。Elementはすべて複合タイプで、簡潔な内容です。LOB データを伝達する blob、clob、および dbclob Elementは、ラージ・オブジェクトからのデータが別の LOB メッセージに送信されるので、常に空です。blob、clob、および dbclob Elementは xsi:nil 属性をもちません。

表 92. single-column-value Elementの説明

名前	XML データ・タイプ	値のデータ・フォーマット
smallint	short	
integer	integer	
bigint	long	
float	float (32 bits)	[-]d.dddde[-+]dd
	double (64 bits)	[-]d.dddde[-+]dd
real	float	
double	double	
decimal	decimal	
date	date	YYYY-MM-DD
time	time	HH:MM:SS.SSS
timestamp	dateTime	YYYY-MM-DDTHH:MM:SS.SSS
char	string	
varchar	string	
long varchar	string	
bitchar	hexBinary	
bitvarchar	hexBinary	
bitlongvarchar	hexBinary	
graphic	string	

表 92. *single-column-value* エレメントの説明 (続き)

名前	XML データ・タイプ	値のデータ・フォーマット
<i>vargraphic</i>	string	
<i>longvargraphic</i>	string	
<i>rowid</i>	hexBinary	
<i>blob</i>	hexBinary	
<i>clob</i>	string	
<i>dbclob</i>	string	

構造:

```
<data_type xsi:nil="null_indicator">value</data_type>
```

詳細:

data_type

ソース表の列のデータ・タイプ。このデータ・タイプを使用してエレメントに命名します。

null_indicator

オプション: ソース列に NULL 値が入っているかどうかを示す整数。デフォルトは 0 (偽) です。ソース列に NULL 値が入っている場合、この属性の値は 1 (真) です。 *blob*、*clob*、および *dbclob* エレメントには、この属性はありません。 XML データ・タイプ: ブール

value

ソース列の実際の名前。ソース値が NULL または LOB 値であると、エレメントは空になります。

例:

以下の例では、整数データ・タイプのキー列にある *single-column-value* の 222 と、*varchar* データ・タイプの非キー列にある Hello を使用する挿入操作を示しています。例では、キー列にある整数データ・タイプの *single-column-value* の 222 の行の削除操作も示しています。

```
<xml version="1.0" encoding="UTF-8" ?>
<msg xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="mqcap.xsd" version="1.0.0"
  dbName="DB1">
  <trans isLast="1" segmentNum="1" cmitLSN="0000:0000::0000:06d6:87ab"
    cmitTime="2003-10-31T12:12:12.000122">
    <insertRow subName="S1" srcOwner="USER1" srcName="T1">
      <col name="COL1" isKey="1">
        <integer>222</integer>
      </col>
      <col name="COL2">
        <varchar>Hello</varchar>
      </col>
    </insertRow>
    <deleteRow subName="S1" srcOwner="USER1" srcName="T1">
      <col name="COL1" isKey="1">
        <integer>222</integer>
      </col>
    </deleteRow>
  </trans>
</msg>
```

```

        </col>
      </deleteRow>
    </trans>
  </msg>

```

double-column-value エレメント

double-column-value エレメントは、Q キャプチャー・プログラムが、ソース列から変更前の値および変更後の値の両方を送信する必要がある場合、更新操作で使用されます。XML メッセージで、Q キャプチャー・プログラムは、変更されたキー列の変更前の値を送信します。XML 発行に BEFORE_VALUES データ送信オプションが "Yes" を設定すると、変更された非キー列の変更前の値を送信します。変更前の値および変更後の値が同一である場合、変更後の値エレメント (afterValue) だけが使用されます。

blob、clob、および dbclob 以外のすべての double-column-value エレメントは、空ではなく、複合タイプで、複合の内容があります。エレメント blob、clob、および dbclob は常に空であり、NULL 可能ではありません。double-column-value エレメントには属性がありません。double-column-value エレメントの詳細については、502 ページの表 92 を参照してください。

構造:

```

<data_type>
  elements
</data_type>

```

詳細:

data_type
 ソース表の列のデータ・タイプ。このデータ・タイプを使用してエレメントに命名します。データ・タイプが blob、clob、または dbclob であると、エレメントは空であり、NULL 可能ではありません。

elements
 beforeValue または afterValue エレメントの一方もしくは両方、または、空の blob、clob、および dbclob。

例:

以下の例は、double-column-value エレメントを示しています。

- 変更されたキー列 (整数データ・タイプ)
- 変更された非キー列 (varchar データ・タイプ)。XML 発行の BEFORE_VALUES データ送信オプションは "No" に設定されています。

```

<xml version="1.0" encoding="UTF-8" ?>
<msg xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="mqcap.xsd" version="1.0.0"
  dbName="DB1">
  <trans isLast="1" segmentNum="1" cmitLSN="0000:0000::0000:06d6:87ab"
    cmitTime="2003-10-31T12:12:12.000122">
    <updateRow subName="S1" srcOwner="USER1" srcName="T1">
      <col name="COL1" isKey="1">

```



```

        <integer>
            beforeValue
            afterValue
        </integer>
    </col>
    <col name="COL2">
        <varchar>
            afterValue
        </varchar>
    </col>
</updateRow>
</trans>
</msg>

```

beforeValue および *afterValue* は、『変更前の値および変更後の値エレメント (beforeVal および afterVal)』で説明されるエレメントを表しています。

変更前の値および変更後の値エレメント (beforeVal および afterVal)

変更前の値および変更後の値エレメント (beforeVal および afterVal) には、ソース表からの実際の値が入っています。このエレメントは、XML 発行の BEFORE_VALUES データ送信オプションが "Yes" に設定されたとき、変更されたキー列または、変更された非キー列に対する更新操作に使用されます。XML 発行が、送信される変更前の値を呼び出し、ソース列のその値に変更が加えられなかった場合、afterVal エレメントだけが使用されます。ソース表からの値が NULL の場合、このエレメントは空になり、xsi:null 属性は、1 (真) に設定されます。

表 93 では、beforeVal および afterVal エレメントについて説明します。

表 93. *beforeVal* および *afterVal* のエレメントの説明

名前	プロパティ
beforeVal	NULL 可能、複合タイプ、単一の内容
afterVal	NULL 可能、複合タイプ、単一の内容、オプション

構造:

```

<beforeVal xsi:nil="null_indicator">value</beforeVal>
<afterVal xsi:nil="null_indicator">value</afterVal>

```

詳細:

null_indicator

オプション: ソース列の値が NULL であるかどうかを示す整数。デフォルトは 0 (偽) です。ソース列に NULL 値が入っている場合、この属性の値は 1 (真) です。XML データ・タイプ: ブール

value

ソース列の実際の名前。ソース値が NULL であると、エレメントは空になります。

例:

以下の例では、キー列の値 222 を変更せず (afterVal エレメントのみ使用)、同じ行にある varchar 列が、"Hello" から NULL に変更された更新操作を示します。この場合、XML 発行の BEFORE_VALUES オプションは、"Yes" に設定されます。

```
<xml version="1.0" encoding="UTF-8" ?>
<msg xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="mqcap.xsd" version="1.0.0"
      dbName="DB1">
  <trans isLast="1" segmentNum="1" cmitLSN="0000:0000::0000:06d6:87ab"
        cmitTime="2003-10-31T12:12:12.000122">
    <updateRow subName="S1" srcOwner="USER1" srcName="T1">
      <col name="COL1" isKey="1">
        <integer>
          <afterVal>222</afterVal>
        </integer>
      </col>
      <col name="COL2">
        <varchar>
          <beforeVal>Hello</beforeVal>
          <afterVal xsi:nil="1"/>
        </varchar>
      </col>
    </updateRow>
  </trans>
</msg>
```

関連概念:

- 493 ページの『Q キャプチャーからユーザー・アプリケーションへの XML メッセージの構造 -- 概要』

関連タスク:

- 180 ページの『発行キュー・マップの作成』

関連資料:

- 506 ページの『行操作メッセージ』
- 508 ページの『ラージ・オブジェクト (LOB) メッセージ』

行操作メッセージ

row operation メッセージには、ソース表からの 1 つの挿入、更新、または削除行操作が含まれています。row operation メッセージでは、メッセージ・エレメント (msg) に行操作エレメント (rowOp) が入っています。

row operation メッセージは、送信キューに定義した最大メッセージ・サイズを超えてはなりません。このサイズを超えた row operation メッセージは、複数のメッセージに分割できない場合があります。row operation メッセージで、トランザクションに属する挿入、更新、または削除は、同じコミット時間およびコミット論理シーケンス番号になります。LOB メッセージが row operation メッセージの後に続く場合、この row operation メッセージには、これがソース・データベースからの行操作の最終メッセージではないことを示す属性が含まれています。

表 94 では、rowOp エlementについて説明します。

表 94. rowOp のElementの説明

名前	プロパティ
rowOp	空でない、複合タイプ、複合内容

構造:

```
<rowOp cmitLSN="commit_logical_sequence_number"
      cmitTime="commit_time" isLast="is_last_indicator">
    elements
</rowOp>
```

詳細:

commit_logical_sequence_number

トランザクションの COMMIT ステートメントのコミット論理シーケンス番号 (時刻に基づくログ・シーケンス番号)。XML データ・タイプ: ストリング。

commit_time

マイクロ秒単位で設定されたグリニッジ標準時 (GMT) を使用した、トランザクションの COMMIT ステートメントのタイム・スタンプ。XML データ・タイプ: dateTime

is_last_indicator

オプション: row operation メッセージがソース・データベースからの行操作の最終メッセージかどうかを示すブール値。LOB メッセージが row operation メッセージの後に続く場合、この値は 0 (偽) に設定されます。この属性にデフォルト値はありません。XML データ・タイプ: ブール

elements

各 rowOp Elementには、以下のElementのうち 1 つが入ります。

- insertRow
- updateRow
- deleteRow

例:

以下の例では、insertRow、updateRow、または deleteRow element を含む行操作Elementを示しています。

```
<xml version="1.0" encoding="UTF-8" ?>
<msg xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="mqcap.xsd" version="1.0.0" dbName="DB1">
  <rowOp cmitLSN="0000:0000::0000:06d6:87ab"
        cmitTime="2003-10-31T12:12:12.000122">
    insertRow, deleteRow, or updateRow
  </rowOp>
</msg>
```

insertRow、*updateRow*、または *deleteRow* は、トランザクション・メッセージ で説明されているエレメントを表しています。

関連概念:

- 493 ページの『Q キャプチャーからユーザー・アプリケーションへの XML メッセージの構造 -- 概要』

関連タスク:

- 180 ページの『発行キュー・マップの作成』

関連資料:

- 497 ページの『トランザクション・メッセージ』
- 508 ページの『ラージ・オブジェクト (LOB) メッセージ』

ラージ・オブジェクト (LOB) メッセージ

ラージ・オブジェクト (LOB) メッセージは、ソース表の列からデータの一部または全部を伝送します。このソース表には、ラージ・オブジェクト値として、BLOB (バイナリー・ラージ・オブジェクト)、CLOB (文字ラージ・オブジェクト)、または DBCLOB (2 バイト文字ラージ・オブジェクト) が入っています。

それぞれの LOB メッセージには、ソース表にある最大で 1 つの LOB 値からのデータが含まれています。この値が、Q キャプチャー・プログラムで決定される LOB メッセージ・バッファー・サイズを超えると、Q キャプチャー・プログラムは LOB 値を、複数の LOB メッセージに分割することが可能です。このバッファー・サイズは、送信キューに定義した最大メッセージ・サイズにすることができます。同じ LOB 値の一部を含むすべてのメッセージは、同じ XML 発行名、ソース表所有者、ソース表名、行番号、および列名を持ちます。

LOB 値を含むメッセージは、LOB 値が属するトランザクションまたは行操作のあるメッセージの後に送信されます。isLast 属性は分割済み LOB 値の最終メッセージを示します。これはまた、トランザクションまたは行操作の最終メッセージです。

LOB メッセージ内で、ラージ・オブジェクト・エレメント (lob) は、メッセージ・エレメント (msg) に入れられ、単一の LOB 列値が入ります。

表 95 では、lob エレメントについて説明します。

表 95. lob のエレメントの説明

名前	プロパティ
lob	空でない、複合タイプ、複合内容

構造:

```
<lob isLast="is_last_indicator" subName="XML_publication_name"
  srcOwner="source_owner" srcName="source_name" rowNum="row_number"
  colName="column_name" totalDataLen="LOB_data_length"
  dataLen="segment_data_length">
```

LOB_column_value

</lob>

詳細:

is_last_indicator

これがトランザクションまたは行操作の最終メッセージかどうかを示すブール値。これが最終メッセージである場合、値は 1 (真) です。最終メッセージではない場合、値は 0 (偽) です。XML データ・タイプ: ブール

XML_publication_name

LOB 値が含まれる XML 発行の名前。XML データ・タイプ: ストリング。

source_owner

LOB が発生したソース表のスキーマ。XML データ・タイプ: ストリング。

source_name

ソース表の名前。XML データ・タイプ: ストリング。

row_number

データベース・トランザクション内で、LOB 値が含まれる行操作の位置番号。XML データ・タイプ: positiveInteger。

column_name

LOB 値が含まれるソース表の列の名前。XML データ・タイプ: ストリング。

LOB_data_length

ソース表に含まれている LOB 値の合計長 (バイト単位)。XML データ・タイプ: nonNegativeInteger。

segment_data_length

単一メッセージ・セグメントに含まれている LOB データの長さ (バイト単位)。XML データ・タイプ: nonNegativeInteger。

LOB_column_value

LOB 値のデータ・タイプを説明する 3 つの LOB 列値エレメントの 1 つ。この 3 つのエレメントとは、blob、clob、および dbclob です。

例:

以下の例では、LOB メッセージを示しています。

```
<xml version="1.0" encoding="UTF-8" ?>
<msg xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="mqcap.xsd"
  version="1.0.0" dbName="DB1">
  <lob isLast="0" subName="S1" srcOwner="USER1" srcName="T1" rowNum="3"
    colName="LOBCOL" totalDataLen="92675" dataLen="100">
    LOB_column_value
  </lob>
</msg>
```

LOB_column_value は、510 ページの『LOB 列値』で説明される 3 つのエレメントの 1 つです。

LOB 列値

3 つの LOB 列値エレメントには、それぞれソース表からの実 LOB データが含まれています。このエレメントは、そのデータ・タイプ blob、clob、または dbclob のいずれかに応じて命名されます。ソース表からの値が NULL の場合、このエレメントは空になり、xsi:nil 属性は 1 (真) に設定されます。

表 96 では、LOB 列値エレメントについて説明します。

表 96. LOB 列値のエレメントの説明

名前	プロパティ
blob	NULL 可能、複合タイプ、単一の内容
clob	NULL 可能、複合タイプ、単一の内容
dbclob	NULL 可能、複合タイプ、単一の内容

構造:

```
<data_type xsi:nil="null_indicator">  
  LOB_value  
</data_type>
```

詳細:

data_type

ソース表の列のデータ・タイプ。このデータ・タイプを使用してエレメントに命名します。

null_indicator

オプション: ソース列の値が NULL であるかどうかを示すブール値。デフォルトは 0 (偽) です。ソース列に NULL 値が含まれている場合、この属性は 1 (真) です。XML データ・タイプ: ブール

LOB_value

ソース表のラージ・オブジェクトからの実データ。

例:

以下の例では、CLOB (文字ラージ・オブジェクト) 値からのデータとして、合計で 92,675 バイトのうちの 100 バイトが入っている LOB メッセージを示しています。

```
<xml version="1.0" encoding="UTF-8" ?>  
<msg xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xsi:noNamespaceSchemaLocation="mqcap.xsd"  
  version="1.0.0" dbName="DB1">  
  <lob isLast="0" subName="S1" srcOwner="USER1" srcName="T1" rowNum="3"  
    colName="LOBCOL" totalDataLen="92675" dataLen="100">  
    <clob>LOB data</clob>  
  </lob>  
</msg>
```

関連概念:

- 72 ページの『ラージ・オブジェクト (LOB) 値のキュー項目数に関する考慮事項』
- 198 ページの『Q レプリケーションおよびイベント発行でのラージ・オブジェクト (LOB) データ・タイプに関する考慮事項』
- 32 ページの『Q レプリケーションおよびイベント発行の LOB データ・タイプのためのメモリー』
- 493 ページの『Q キャプチャーからユーザー・アプリケーションへの XML メッセージの構造 -- 概要』

関連資料:

- 497 ページの『トランザクション・メッセージ』
- 506 ページの『行操作メッセージ』

サブスクリプション非アクティブ化メッセージ

subscription deactivated メッセージにより、Q キャプチャー・プログラムがユーザー・アプリケーションから deactivate subscription メッセージを受信したことが確認できます。

subscription deactivated メッセージでは、メッセージ・エレメント (msg) にサブスクリプション非アクティブ化エレメント (subDeactivated) が入っています。

表 97 では、subDeactivated エレメントについて説明します。

表 97. subDeactivated のエレメントの説明

名前	プロパティ
subDeactivated	空、複合タイプ

構造:

```
<subDeactivated subName="XML_publication_name" srcOwner="source_owner"
  srcName="source_name" stateInfo="state_information"/>
```

詳細:

XML_publication_name

非アクティブ化した XML 発行の名前。XML データ・タイプ: ストリング。

source_owner

この XML 発行のソース表のスキーマ。XML データ・タイプ: ストリング。

source_name

ソース表の名前。XML データ・タイプ: ストリング。

state_information

XML 発行の状態に関する追加の情報。この属性に、ASN メッセージ番号は入っていません。XML データ・タイプ: ストリング。

例:

以下の例は、subscription deactivated メッセージを示しています。


```

<xml version="1.0" encoding="UTF-8" ?>
<msg xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="mqcap.xsd" version="1.0.0"
      dbName="DB1">
    <subDeactivated subName="S1" srcOwner="USER1" srcName="T1"
                    stateInfo="ASN7019I"/>
</msg>

```

関連概念:

- 493 ページの『Q キャプチャーからユーザー・アプリケーションへの XML メッセージの構造 -- 概要』

関連タスク:

- 254 ページの『Q サブスクリプションまたは XML 発行の非アクティブ化』

関連資料:

- 526 ページの『サブスクリプション・メッセージの非アクティブ化』

ロード完了受信メッセージ

load done received メッセージにより、Q キャプチャー・プログラムがユーザー・アプリケーションから load done received メッセージを受信したことが確認できます。load done メッセージは、ターゲット表がロードされていることを表します。

load done received メッセージでは、メッセージ・エレメント (msg) にロード完了受信エレメント (loadDoneRcvd) が入っています。

表 98 では、loadDoneRcvd エレメントについて説明します。

表 98. loadDoneRcvd のエレメントの説明

名前	プロパティ
loadDoneRcvd	空、複合タイプ

構造:

```

<loadDoneRcvd subName="XML_publication_name" srcOwner="source_owner"
              srcName="source_name" stateInfo="state_information"/>

```

詳細:

XML_publication_name

ターゲット表がロードされた XML 発行の名前。XML データ・タイプ: ストリング。

source_owner

この XML 発行のソース表のスキーマ。XML データ・タイプ: ストリング。

source_name

ソース表の名前。XML データ・タイプ: ストリング。

state_information

XML 発行の状態に関する追加の情報。この属性に、ASN メッセージ番号は入っていません。XML データ・タイプ: ストリング。

例:

以下の例は、load done received メッセージを示しています。

```
<xml version="1.0" encoding="UTF-8" ?>
<msg xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="mqcap.xsd" version="1.0.0"
      dbName="DB1">

      <loadDoneRcvd subName="S1" srcOwner="USER1" srcName="T1"
                    stateInfo="ASN7019I"/>

</msg>
```

関連概念:

- 163 ページの『Q レプリケーションのターゲット表のロードのオプション -- 概要』
- 493 ページの『Q キャプチャーからユーザー・アプリケーションへの XML メッセージの構造 -- 概要』

関連資料:

- 524 ページの『ロード完了メッセージ』

エラー・レポート・メッセージ

Q キャプチャー・プログラムは、制御メッセージ経由で出されたユーザー・アプリケーションの要求を実行できない場合、error report メッセージを送信します。たとえば、Q キャプチャー・プログラムが、XML 発行をアクティブまたは非アクティブにできない場合、または load done メッセージを確認できない場合、エラー・レポート・メッセージを送信します。Q キャプチャー・プログラムは、このエラーをログにも書き込みます。送信キューが使用不可のため、Q キャプチャー・プログラムがエラー・レポート・メッセージの送信ができない場合も、そのエラーをログを書き込みます。WebSphere MQ に関するエラーは、エラー・レポート・メッセージを生成しません。

error report メッセージでは、メッセージ・エレメント (msg) にエラー・レポート・エレメント (errorRpt) が入っています。

表 99 では、errorRpt エレメントについて説明します。

表 99. errorRpt のエレメントの説明

名前	プロパティ
errorRpt	空、複合タイプ

構造:

```
<errorRpt subName="XML_publication_name" srcOwner="source_owner"
  srcName="source_name" errorMsg="message_text"/>
```

詳細:

XML_publication_name

エラーを生成する XML 発行の名前。XML データ・タイプ: ストリング。

source_owner

この XML 発行のソース表のスキーマ。XML データ・タイプ: ストリング。

source_name

ソース表の名前。XML データ・タイプ: ストリング。

message_text

エラー・メッセージのテキスト。XML データ・タイプ: ストリング。

例:

以下の例では、Q キャプチャー・プログラムが XML 発行をアクティブにできなかった後に生成された error report メッセージを示しています。

```
<xml version="1.0" encoding="UTF-8" ?>
<msg xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="mqcap.xsd" version="1.0.0"
  dbName="DB1">

  <errorRpt subName="S1" srcOwner="USER1" srcName="T1"
    errorMsg="message_text"/>

</msg>
```

message_text は、エラー・メッセージのテキストです。

関連概念:

- 118 ページの『Q レプリケーションのエラー・オプション』
- 493 ページの『Q キャプチャーからユーザー・アプリケーションへの XML メッセージの構造 -- 概要』

ハートビート・メッセージ

heartbeat メッセージはユーザー・アプリケーションに Q キャプチャー・プログラムが依然として実行中であることを通知します。Q キャプチャー・プログラムは、キューに書き込むメッセージがない場合、送信キューのハートビート・インターバルが到着するたびに、このメッセージをアクティブな送信キューに書き込みます。Q キャプチャー・プログラムがこのインターバルの生じる前にログの最後に到達する場合、最終コミット時間に関する情報なしで、heartbeat メッセージを送信します。

heartbeat メッセージでは、メッセージ・エレメント (msg) にハートビート・エレメント (heartbeat) が入っています。

表 100 では、heartbeat エlementについて説明します。

表 100. heartbeat のElementの説明

名前	プロパティ
heartbeat	空、複合タイプ

構造:

```
<heartbeat sendQName="send_queue_name" lastCmitTime="last_commit_time"/>
```

詳細:

send_queue_name

Q キャプチャー・プログラムが heartbeat メッセージを書き込む送信キューの名前。XML データ・タイプ: ストリング。

last_commit_time

オプション: 最終コミット済みトランザクションのタイム・スタンプ (グリニッジ標準時 (GMT))。この属性はオプションで、デフォルト値はありません。XML データ・タイプ: dateTime

例:

以下の例は、heartbeat メッセージを示しています。

```
<xml version="1.0" encoding="UTF-8" ?>
<msg xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="mqcap.xsd" version="1.0.0"
  dbName="DB1">

  <heartbeat sendQName="Q1" lastcmitTime="2003-10-31T12:12:12.000122"/>

</msg>
```

関連概念:

- 62 ページの『WebSphere MQ オブジェクトの設定要件』
- 493 ページの『Q キャプチャーからユーザー・アプリケーションへの XML メッセージの構造 -- 概要』

関連資料:

- 434 ページの『IBMQREP_SENDQUEUES 表』

サブスクリプション・スキーマ・メッセージ (subSchema)

Q Cキャプチャー・プログラムは、subscription schema メッセージを送信して、XML 発行をアクティブにするかまたは再初期化するかを確認します。このメッセージは、ソース表および送信キューの名前、データ送信オプション、およびロード・フェーズの情報を含む XML 発行の詳細を伝えます。subscription schema メッセージは、activate subscription メッセージ、**reinit** コマンド、または、REINIT_SUB シグナルの応答として送信します。

subscription schema メッセージで、このメッセージ・エレメント(msg) には、1 つ以上の列エレメント (col.) があるサブスクリプション・スキーマ・エレメントがあります。以下のセクションでは、この 2 つのエレメントについて説明しています。

- 『サブスクリプション・スキーマ・エレメント (subSchema)』
- 518 ページの『サブスクリプション・スキーマ・メッセージの列エレメント (col)』

サブスクリプション・スキーマ・エレメント (subSchema)

この属性により、subSchema エレメントは XML 発行の詳細を提供します。subSchema エレメントには、1 つ以上の列エレメント (col) があります。

表 101 では、subSchema エレメントについて説明しています。

表 101. subSchema のエレメントの説明

名前	プロパティ
subSchema	空でない、複合タイプ、単一の内容

構造:

```
<subSchema subname="XML_publication_name"
  srcOwner="source_owner"
  srcName="source_name"
  sendQName="send_queue_name"
  allChangedRows="ALL_CHANGED_ROWS_option"
  beforeValues="BEFORE_VALUES_option"
  changedColsOnly="CHANGED_COLS_ONLY_option"
  hasLoadPhase="load_phase_option"
  dbServerType="operating_system"
  dbRelease="DB2_release_level"
  dbInstance="DB2_instance_name"
  capRelease="Q_capture_release_level">

  column_elements

</subSchema>
```

詳細:

XML_publication_name

アクティブにしたまたは再初期化した XML 発行の名前。XML データ・タイプ: ストリング。

source_owner

この XML 発行のソース表のスキーマ。XML データ・タイプ: ストリング。

source_name

ソース表の名前。XML データ・タイプ: ストリング。

send_queue_name

XML 発行に指定した送信キューの名前。XML データ・タイプ: ストリング。

ALL_CHANGED_ROWS_option

オプション: ALL_CHANGED_ROWS データ送信オプションが XML 発行に指

定されたかどうかを示すブール値。デフォルトは 0 (偽) です。このオプションが指定される場合、値は 1 (真) です。XML データ・タイプ: ブール

BEFORE_VALUES_option

オプション: BEFORE_VALUES データ送信オプションが XML 発行に指定されたかどうかを示すブール値。デフォルトは 0 (偽) です。このオプションが指定される場合、値は 1 (真) です。XML データ・タイプ: ブール

CHANGED_COLS_ONLY_option

オプション: CHANGED_COLS_ONLY データ送信オプションが XML 発行に指定されたかどうかを示すブール値。デフォルトは 0 (偽) です。このオプションが指定される場合、値は 1 (真) です。XML データ・タイプ: ブール

load_phase_option

オプション: XML 発行にロード・フェーズがあるかどうかを示す標識。デフォルトはロード・フェーズなしを意味する "none" です。ロード・フェーズが指定される場合、この値は "external" です。XML データ・タイプ: loadPhaseEnumType。

operating_system

オプション: ソース・データベースまたはサブシステムのオペレーティング・システム。デフォルトは QDB2/6000 (DB2 UDB for AIX) です。XML データ・タイプ: dbServerTypeEnumType。

DB2_release_level

ソース・データベースまたはサブシステムの DB2 リリース・レベル。XML データ・タイプ: スtring。

DB2_instance_name

ソース・データベースがある DB2 インスタンスの名前。XML データ・タイプ: スtring。

Q_capture_release_level

Q キャプチャー・プログラムのリリース・レベル。XML データ・タイプ: スtring。

column_elements

ソース表にある各列についての情報を送る 1 つ以上の列エレメント (col)。

表 102 では、subSchema エレメントの属性で使用される 2 つの XML データ・タイプについての追加の詳細を説明しています。

表 102. subSchema 属性の追加のデータ・タイプの説明

タイプ名	基本タイプ	値
loadPhaseEnumType	string	なし、外部
dbServerTypeEnumType	string	QDB2, QDB2/6000, QDB2/HPUX, QDB2/NT, QDB2/SUN, QDB2/LINUX, QDB2/Windows

注: QDB2 自体は、DB2 UDB for z/OS を暗黙指定する。

例:

以下の subscription schema メッセージは、XML 発行に送信され、BEFORE_VALUES データ送信オプションおよびロード・フェーズを指定します。

```
<xml version="1.0" encoding="UTF-8" ?>
<msg xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="mqcap.xsd" version="1.0.0"
      dbName="DB1">
  <subSchema subname="S1"
             srcOwner="USER1"
             srcName="T1"
             sendQName="Q1"
             beforeValues="yes"
             hasLoadPhase="external"
             dbServerType="QDB2/6000"
             dbRelease="8.2.0"
             dbInstance="DB2INST"
             capRelease="8.2.0">
    column_element
  </subSchema>
</msg>
```

column_element は、『サブスクリプション・スキーマ・メッセージの列エレメント (col)』で説明されている 1 つ以上の列エレメントを表しています。

サブスクリプション・スキーマ・メッセージの列エレメント (col)

subscription schema メッセージ内で、列エレメント (col) は、ソース表にある列ごとの情報を伝達します。

表 103 では、スキーマ・メッセージにある col エレメントについて説明します。

表 103. col のエレメントの説明

名前	プロパティ
col	空、複合タイプ

構造:

```
<col name="column_name"
      type="data_type"
      len="data_length"
      precision="data_precision"
      scale="decimal_scale"
      codepage="codepage_number"
      isKey="key_indicator"/>
```

詳細:

column_name

ソース表の列名。XML データ・タイプ: ストリング。

data_type

ソース列のデータ・タイプ。これは、dataTypeEnumType XML データ・タイプに定義したデータ・タイプの 1 つでなければなりません。リストについては、519 ページの表 104 を参照してください。XML データ・タイプ: dataTypeEnumType。

表 104. *dataTypeEnumType* の追加のデータ・タイプの説明

タイプ名	基本タイプ	値
<i>dataTypeEnumType</i>	string	smallint、integer、bigint、float、real、double、decimal、char、varchar、longvarchar、bitchar、bitvarchar、bitlongvarchar、graphic、vargraphic、longvargraphic、time、timestamp、date、rowid、blob、clob、dbclob

data_length

オプション: ソース列のデータの最大長。XML データ・タイプ: unsignedInt。

data_precision

オプション: 10 進数データ・タイプの場合の、数字の精度。XML データ・タイプ: unsignedShort。

decimal_scale

オプション: 10 進数データ・タイプの場合の、数字の位取り。XML データ・タイプ: unsignedShort。

codepage_number

オプション: 文字データ・タイプのコード・ページ。デフォルトは 0 です。XML データ・タイプ: unsignedShort。

key_indicator

オプション: これがキー列であるかどうかを示すブール値。デフォルトは 0 (偽) です。キー列である場合、値は 1 (真) です。XML データ・タイプ: ブール

例:

以下の例は、サブスクリプション・スキーマ・メッセージ内にある 2 つの列エレメントを示しています。

```
<xml version="1.0" encoding="UTF-8" ?>
<msg xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="mqcap.xsd" version="1.0.0"
  dbName="DB1">
  <subSchema subname="S1"
    srcOwner="USER1"
    srcName="T1"
    sendQName="Q1"
    beforeValues="yes"
    hasLoadPhase="external"
    dbServerType="QDB2/6000"
    dbRelease="8.2.0"
    dbInstance="DB2INST"
    capRelease="8.2.0">
    <col name="COL1" type="integer" len="4"/>
    <col name="COL2" type="varchar" len="50" codepage="1208"/>
  </subSchema>
</msg>
```

関連概念:

- 179 ページの『ソースからの発行 (イベント発行) の設定 -- 概要』
- 493 ページの『Q キャプチャーからユーザー・アプリケーションへの XML メッセージの構造 -- 概要』

列の追加メッセージ

add column メッセージはユーザー・アプリケーションに Q キャプチャー・プログラムが既存の XML 発行に列を追加したことを通知します。このメッセージは、ユーザーまたはユーザー・アプリケーションが IBMQREP_SIGNAL 表に ADDCOL シグナルを挿入すると、送信されます。

add column メッセージでは、メッセージ・エレメント (msg) に列の追加エレメント (addColumn) が入っています。列の追加エレメントには、追加されたソース表の列の情報を伝える列スキーマ (col) エレメントがあります。

表 105 では、addColumn エレメントについて説明します。

表 105. addColumn のエレメントの説明

名前	プロパティ
addColumn	空でない、複合タイプ、単一の内容

構造:

```
<addColumn subName="XML_publication_name" srcOwner="source_owner"
  srcName="source_name">
  column_element
</addColumn>
```

詳細:

XML_publication_name

列を追加した XML 発行の名前。XML データ・タイプ: ストリング。

srcOwner

この XML 発行のソース表のスキーマ。XML データ・タイプ: ストリング。

srcName

ソース表の名前。XML データ・タイプ: ストリング。

column_element

追加された列の詳細 (名前、データ・タイプ、データ長、および列がキー列であるかどうか) を含む列スキーマ (col) エレメント。

例:

以下の例は、add column メッセージを示しています。

```
<xml version="1.0" encoding="UTF-8" ?>
<msg xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="mqcap.xsd" version="1.0.0"
  dbName="DB1">
  <addColumn subName="S1" srcOwner="USER1" srcName="T1">
    column_element
  </addColumn>
</msg>
```

`column_element` は、「515 ページの『サブスクリプション・スキーマ・メッセージ (subSchema)』」で説明されている列エレメントを表しています。

関連概念:

- 493 ページの『Q キャプチャーからユーザー・アプリケーションへの XML メッセージの構造 -- 概要』

関連タスク:

- 221 ページの『既存の XML 発行への列の追加』

関連資料:

- 437 ページの『IBMQREP_SIGNAL 表』

ユーザー・アプリケーションから Q キャプチャーへの XML メッセージの構造

ユーザー・アプリケーションから Q キャプチャーへの XML メッセージの構造 -- 概説

ユーザー・アプリケーションは、XML メッセージをその管理キューに送信して Q キャプチャー・プログラムと通信します。これらのメッセージは制御メッセージとして知られています。ユーザー・アプリケーションはこのメッセージを使用して、ターゲット表がロードされたことを報告したり、Q キャプチャー・プログラムが XML 発行をアクティブまたは非アクティブにしたり、あるいは送信キューを無効にするように要求します。

続くトピックでは、ユーザー・アプリケーションから Q キャプチャー・プログラムへの制御メッセージの XML 構造を説明しています。

- 『ユーザー・アプリケーションから Q キャプチャーへの XML メッセージのリスト』
- 522 ページの『msg: ユーザー・アプリケーションから Q キャプチャーへの XML メッセージのルート・エレメント』
- 524 ページの『送信キュー・メッセージの無効化』
- 524 ページの『ロード完了メッセージ』
- 525 ページの『サブスクリプション・メッセージのアクティブ化』
- 526 ページの『サブスクリプション・メッセージの非アクティブ化』

関連概念:

- 491 ページの『イベント発行の XML メッセージの構造 -- 概要』

ユーザー・アプリケーションから Q キャプチャーへの XML メッセージのリスト

522 ページの表 106 では、ユーザー・アプリケーションから Q キャプチャー・プログラムへの、4 つのタイプの制御メッセージを説明しています。

表 106. ユーザー・アプリケーションから Q キャプチャー・プログラムへの制御メッセージ

メッセージ・タイプ	説明
Invalidate send queue	Q キャプチャー・プログラムに、指定したキュー・エラー・アクションを実行して送信キューを無効化するように要求する。
Load done	XML 発行のターゲット表がロードされたことを Q キャプチャー・プログラムに通知する。
Activate subscription	Q キャプチャー・プログラムに XML 発行のアクティブ化を要求する。
Deactivate subscription	Q キャプチャー・プログラムに XML 発行の非アクティブ化を要求する。

関連概念:

- 15 ページの『Q キャプチャー・プログラム』
- 9 ページの『イベント発行』
- 491 ページの『イベント発行の XML メッセージの構造 -- 概要』

関連資料:

- 494 ページの『Q キャプチャーからユーザー・アプリケーションへの XML メッセージのリスト』

msg: ユーザー・アプリケーションから Q キャプチャーへの XML メッセージのルート・エレメント

msg エレメントは、ユーザー・アプリケーションから Q キャプチャー・プログラムへのすべての制御メッセージのルート・エレメントです。

表 107 では、msg エレメントについて説明します。

表 107. msg (Q キャプチャー・プログラムに対するユーザー・アプリケーション) のエレメントの説明

名前	プロパティ
msg	空でない、複合タイプ、複合内容

構造:

```
<xml version="1.0" encoding="UTF-8" ?>
<msg xmlns:xsi="XML_schema_instance"
      xsi:noNamespaceSchemaLocation="schema_document"
      version="version">

  elements

</msg>
```

詳細:

XML_schema_instance

XML スキーマ・インスタンスの URL。イベント発行で、URL は、www.w3.org/2001/XMLSchema-instance です。XML データ・タイプ: ストリング。

schema_document

XML スキーマ文書のファイル名。XML ネームスペースは、メッセージが 1 つの XML スキーマしか参照しないので、イベント発行ではサポートされていません。ユーザー・アプリケーションから、Q キャプチャー・プログラムへのメッセージは、[mqsub.xsd](#) スキーマ文書を参照します。XML データ・タイプ: ストリング。

version

XML スキーマ・インスタンスのバージョン。DB2 UDB バージョン 8.2 では、このバージョンは 1.0.0 です。XML データ・タイプ: ストリング。

elements

msg エLEMENTが入っているELEMENTの 1 つ。以下のようにメッセージごとにこれらのELEMENTのうち 1 つだけが表示されます。

- invalidateSendQ
- loadDone
- activateSub
- deactivateSub

例:

続く例では、ユーザー・アプリケーションから Q キャプチャー・プログラムへのメッセージを示しています。

```
<xml version="1.0" encoding="UTF-8" ?>
<msg xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="mqsub.xsd" version="1.0.0">
  elements
</msg>
```

elements は、invalidateSendQ、loadDone、activateSub、deactivateSub のELEMENTの 1 つを表しています。

関連概念:

- 521 ページの『ユーザー・アプリケーションから Q キャプチャーへの XML メッセージの構造 -- 概説』

関連資料:

- 521 ページの『ユーザー・アプリケーションから Q キャプチャーへの XML メッセージのリスト』

送信キュー・メッセージの無効化

サブスクライブするアプリケーションは、送信キューでエラーを検出し、XML 発行に指定したエラー・アクションを Q キャプチャー・プログラムに実行させたい場合、`invalidate send queue` メッセージを送信します。

表 108 では、`invalidateSendQ` エlementについて説明します。

表 108. `invalidateSendQ` のElementの説明

名前	プロパティ
<code>invalidateSendQ</code>	空、複合タイプ

構造:

```
<invalidateSendQ sendQName="send_queue_name"/>
```

詳細:

`send_queue_name`

Q キャプチャー・プログラムに無効を要求する送信キューの名前。XML データ・タイプ: ストリング。

例:

以下の例は、`invalidate send queue` メッセージを示しています。

```
<xml version="1.0" encoding="UTF-8" ?>
<msg xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="mqsub.xsd" version="1.0.0">
  <invalidateSendQ sendQName="S1"/>
</msg>
```

関連概念:

- 521 ページの『ユーザー・アプリケーションから Q キャプチャーへの XML メッセージの構造 -- 概説』

関連資料:

- 521 ページの『ユーザー・アプリケーションから Q キャプチャーへの XML メッセージのリスト』

ロード完了メッセージ

`load done` メッセージは、ターゲット表がロードされたことを Q キャプチャー・プログラムに通知します。Q キャプチャー・プログラムは、`load done received` メッセージを送信して、`load done` メッセージに応答します。

表 109 では、`loadDone` Elementについて説明します。

表 109. `loadDone` のElementの説明

名前	プロパティ
<code>loadDone</code>	空、複合タイプ

構造:

```
<loadDone subName="XML_publication_name"/>
```

詳細:

XML_publication_name

ロード・フェーズを完了した XML 発行の名前。XML データ・タイプ: ストリング。

例:

以下の例は、load done メッセージを示しています。

```
<xml version="1.0" encoding="UTF-8" ?>
<msg xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="mqsub.xsd" version="1.0.0">

  <loadDone subName="S1"/>

</msg>
```

関連概念:

- 163 ページの『Q レプリケーションのターゲット表のロードのオプション -- 概要』
- 521 ページの『ユーザー・アプリケーションから Q キャプチャーへの XML メッセージの構造 -- 概説』

関連資料:

- 521 ページの『ユーザー・アプリケーションから Q キャプチャーへの XML メッセージのリスト』
- 512 ページの『ロード完了受信メッセージ』

サブスクリプション・メッセージのアクティブ化

activate subscription メッセージは、Q キャプチャー・プログラムに、XML 発行への変更のキャプチャー開始を指示します。

表 110 では、activateSub エlementについて説明します。

表 110. activateSub のElementの説明

名前	プロパティ
activateSub	空、複合タイプ

構造:

```
<activateSub subName="XML_publication_name"/>
```

詳細:

XML_publication_name

Q キャプチャー・プログラムにアクティブ化を要求する XML 発行の名前。
XML データ・タイプ: ストリング。

例:

以下の例は、activate subscription メッセージを示しています。

```
<xml version="1.0" encoding="UTF-8" ?>
<msg xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="mqsub.xsd" version="1.0.0">
    <activateSub subName="S1"/>
</msg>
```

関連概念:

- 521 ページの『ユーザー・アプリケーションから Q キャプチャーへの XML メッセージの構造 -- 概説』

関連タスク:

- 253 ページの『Q サブスクリプションまたは XML 発行のアクティブ化』

関連資料:

- 521 ページの『ユーザー・アプリケーションから Q キャプチャーへの XML メッセージのリスト』
- 515 ページの『サブスクリプション・スキーマ・メッセージ (subSchema)』

サブスクリプション・メッセージの非アクティブ化

deactivate subscription メッセージは、Q キャプチャー・プログラムに、XML 発行への変更のキャプチャー停止を指示します。

表 111 では、deactivateSub エレメントについて説明します。

表 111. deactivateSub のエレメントの説明

名前	プロパティー
deactivateSub	空、複合タイプ

構造:

```
<deactivateSub subName="XML_publication_name"/>
```

詳細:

XML_publication_name

Q キャプチャー・プログラムに非アクティブ化を要求する XML 発行の名前。
XML データ・タイプ: ストリング。

例:

以下の例は、deactivate subscription メッセージを示しています。

```
<xml version="1.0" encoding="UTF-8" ?>
<msg xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="mqsub.xsd" version="1.0.0">

      <deactivateSub subName="S1"/>

</msg>
```

関連概念:

- 521 ページの『ユーザー・アプリケーションから Q キャプチャーへの XML メッセージの構造 -- 概説』

関連タスク:

- 254 ページの『Q サブスクリプションまたは XML 発行の非アクティブ化』

関連資料:

- 521 ページの『ユーザー・アプリケーションから Q キャプチャーへの XML メッセージのリスト』
- 511 ページの『サブスクリプション非アクティブ化メッセージ』

付録 A. Q レプリケーションおよびイベント発行で使用するサンプル・プログラム

Q レプリケーションおよびイベント発行で使用するサンプル・プログラム -- 概要

サンプル・プログラムには、プログラムを作成して Q レプリケーション環境をセットアップおよび操作する際に使用できる、サンプル・コードが含まれています。続くいくつかのセクションでは、Linux オペレーティング・システム、UNIX オペレーティング・システム、Windows オペレーティング・システム、および z/OS オペレーティング・システム用のサンプル・プログラムについて説明しています。

- 『Q レプリケーションおよびイベント発行で使用するサンプル・プログラム (Linux、UNIX、Windows) -- 概要』
- 531 ページの『Q レプリケーションおよびイベント発行で使用するサンプル・プログラム (z/OS) -- 概要』

サンプル・プログラム (Linux、UNIX、Windows)

Q レプリケーションおよびイベント発行で使用するサンプル・プログラム (Linux、UNIX、Windows) -- 概要

続くいくつかのセクションでは、Linux、UNIX、および Windows オペレーティング・システムでの、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操作に役立つサンプル・プログラムについて説明しています。

- 『Q レプリケーションおよびイベント発行をセットアップするためのサンプル (Linux、UNIX、Windows)』
- 530 ページの『Q レプリケーション・プログラムを操作するためのサンプル (Linux、UNIX、Windows)』

Linux、UNIX、および Windows オペレーティング・システム用のサンプル・プログラムは、`sqllib/samples/repl` ディレクトリーにあります。

関連資料:

- 529 ページの『Q レプリケーションおよびイベント発行で使用するサンプル・プログラム -- 概要』

Q レプリケーションおよびイベント発行をセットアップするためのサンプル (Linux、UNIX、Windows)

530 ページの表 112 では、以下の例について示し、Q レプリケーションおよびイベント発行環境をセットアップするのに役立つ、サンプル・プログラムについて説明しています。

- Q レプリケーション・オブジェクトをセットアップするためのプログラム。
- レプリケーション用のコントロール表を作成するためのプログラム。
- ビジネス環境でイベント発行を使用するプログラム。

表 112. Linux、UNIX、および Windows オペレーティング・システム用のサンプル・プログラム

プログラム	説明
asnsqdefq	このサンプル・スクリプトは、Q レプリケーションおよびイベント発行が WebSphere MQ と対話する方法について示す例です。これには、ピアツーピア・レプリケーションでの 2 つのサーバーのキューおよびキュー定義を作成するコマンドが含まれています。このサンプル・スクリプトを使用すれば、Q レプリケーション環境を短時間で作成し、セットアップすることができます。
asnsqctlw.sql	このサンプル・スクリプトは、レプリケーション用のコントロール表を作成します。
asnsqwxml.zip	このファイルは、イベント発行用の Web ベース・アプリケーションの例を示します。このサンプルでは、ビジネス・シナリオで XML 発行を使用する方法について例示します。
asnsqspC.SQC	このサンプル・プログラムは、C で書かれたストアード・プロシージャの例です。
asnsqspcreate.sql	このサンプル・スクリプトは、SQL で書かれたストアード・プロシージャを作成します。
asnsqspSQL.sql	このサンプル・スクリプトは、SQL で書かれたストアード・プロシージャの例です。

AsnsqspC.mak は、asnsqspC.SQC サンプル・プログラムをコンパイルするために使用できる makefile です。これは、他のサンプル・プログラムと同じディレクトリーにあります。

関連資料:

- 529 ページの『Q レプリケーションおよびイベント発行で使用するサンプル・プログラム -- 概要』

Q レプリケーション・プログラムを操作するためのサンプル (Linux、UNIX、Windows)

表 113 では、Q キャプチャー・プログラム、Q アプライ・プログラム、およびレプリケーション・アラート・モニターを開始する際に役立つサンプル・プログラムについて説明しています。

表 113. Q レプリケーション・プログラムを操作するためのサンプル

プログラム	説明
qcapture_api.c	このサンプル・プログラムには、Q キャプチャー・プログラムを開始するためのコードが含まれています。
qapply_api.c	このサンプル・プログラムには、Q アプライ・プログラムを開始するためのコードが含まれています。

表 113. Q レプリケーション・プログラムを操作するためのサンプル (続き)

プログラム	説明
monitor_api.c	このサンプル・プログラムには、レプリケーション・アラート・モニターを開始するためのコードが含まれています。

表 114 では、530 ページの表 113 でリストされているプログラムをビルド、コンパイル、およびリンクするのに必要な makefile について説明しています。

表 114. サンプル・プログラムをビルドするための makefile

makefile	説明
qcapture_api_nt.mak	この makefile は、Windows オペレーティング・システムでの qcapture_api.c サンプル・プログラムのコードをビルドします。
qcapture_api_unix.mak	この makefile は、Linux および UNIX オペレーティング・システムでの qcapture_api.c サンプル・プログラムのコードをビルドします。
qapply_api_nt.mak	この makefile は、Windows オペレーティング・システムでの qapply_api.c サンプル・プログラムのコードをビルドします。
qapply_api_unix.mak	この makefile は、Linux および UNIX オペレーティング・システムでの qapply_api.c サンプル・プログラムのコードをビルドします。

関連資料:

- 529 ページの『Q レプリケーションおよびイベント発行で使用するサンプル・プログラム -- 概要』

サンプル・プログラム (z/OS)

Q レプリケーションおよびイベント発行で使用するサンプル・プログラム (z/OS) -- 概要

以下のセクションでは、z/OS オペレーティング・システムでの、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操作に役立つサンプル・プログラムについて説明しています。

- 532 ページの『Q レプリケーションおよびイベント発行をセットアップするためのサンプル (z/OS)』
- 532 ページの『Q レプリケーションおよびイベント発行のサンプル JCL プログラム (z/OS)』

SASNSAMP 区分データ・セット (PDS) には、z/OS オペレーティング・システム用のサンプル・プログラムが含まれています。

Q レプリケーションおよびイベント発行をセットアップするためのサンプル (z/OS)

表 115 では、以下の例について示し、Q レプリケーションおよびイベント発行環境をセットアップするのに役立つ、サンプル・プログラムについて説明しています。

- Q レプリケーション・オブジェクトをセットアップするためのプログラム。
- レプリケーション用のコントロール表を作成するためのプログラム。
- ビジネス環境でイベント発行を使用するプログラム。

表 115. 表 1. z/OS オペレーティング・システム用のサンプル・プログラム

プログラム	説明
asnqdefq	このサンプル・スクリプトは、Q レプリケーションおよびイベント発行が WebSphere MQ と対話する仕方について示す例です。これには、ピアツーピア・レプリケーションでの 2 つのサーバーのキューおよびキュー定義を作成するコマンドが含まれています。このスクリプトを使用すれば、Q レプリケーション環境をすぐに作成し、セットアップできます。
asnqctlz.sql	このサンプル・スクリプトには、レプリケーション用のコントロール表を作成するためのコードが含まれています。
asnqbnds	このサンプル・プログラムには、ローカル・サーバーで Q キャプチャー・プログラムおよび Q アプライ・プログラムをバインドするためのコードが含まれています。
asnqbndl	このサンプル・プログラムには、リモート Windows サーバーで Q アプライ・プログラムをバインドするためのコードが含まれています。
asnqspC.SQC	このサンプル・プログラムには、C で書かれたストアード・プロシージャ用のコード例が含まれています。

AsnqspC.mak は、asnqspC.SQC サンプル・プログラムをコンパイルするために使用できる makefile です。これは、他のサンプル・プログラムと同じディレクトリーにあります。

Q レプリケーションおよびイベント発行のサンプル JCL プログラム (z/OS)

ジョブ制御言語 (JCL) を使用して、z/OS オペレーティング・システムで Q レプリケーションの多くの機能を実行できます。表 116 では、JCL で作成されたサンプル・プログラムについて説明しています。

表 116. ジョブ制御言語 (JCL) の使用に関するサンプル・プログラム。

プログラム	説明
asnstpa	このサンプル・プログラムには、 asnacmd コマンドを使用して Q アプライ・プログラムを停止する、JCL が含まれています。
asnstpc	このサンプル・プログラムには、 asnccmd コマンドを使用して Q キャプチャー・プログラムを停止する、JCL が含まれています。
asnstra	このサンプル・プログラムには、Q アプライ・プログラムを開始するための JCL が含まれています。

表 116. ジョブ制御言語 (JCL) の使用に関するサンプル・プログラム。(続き)

プログラム	説明
asnstrc	このサンプル・プログラムには、Q キャプチャー・プログラムを開始するための JCL が含まれています。
asnstrm	このサンプル・プログラムには、レプリケーション・アラート・モニターを開始するための JCL が含まれています。
asnqtrc	このサンプル・プログラムには、Q キャプチャー・プログラムまたは Q アプライ・プログラムで asntrc プログラムを実行するための JCL が含まれています。
asntrdmp	このサンプル・プログラムには、asntrc プログラムによって生成されたバッファをファイルに書き込むための JCL が含まれています。
asntrflw	このサンプル・プログラムには、asntrc flw トレース・レポートを生成するための JCL が含まれています。
asntrfmt.c	このサンプル・プログラムには、asntrc fmt トレース・レポートを生成するための JCL が含まれています。
asnqtoff.c	このサンプル・プログラムには、asntrc プログラムを停止するための JCL が含まれています。
asnqton.c	このサンプル・プログラムには、asntrc プログラムを開始するための JCL が含まれています。
asntdiff.c	このサンプル・プログラムには、 asntdiff コマンドを実行するための JCL が含まれています。
asnqjspc	このサンプル・プログラムには、C ストアード・プロシージャのサンプルの、プリコンパイル、コンパイル、リンク・エディット、およびバインドを行うための JCL が含まれています。

関連タスク:

- 349 ページの『レプリケーション・プログラムのスケジューリング (z/OS)』

関連資料:

- 531 ページの『Q レプリケーションおよびイベント発行で使用するサンプル・プログラム (z/OS) -- 概要』
- 532 ページの『Q レプリケーションおよびイベント発行をセットアップするためのサンプル (z/OS)』

付録 B. ASNCLP: レプリケーションおよびイベント発行を管理するコマンド行インターフェース

ASNCLP プログラムは、Linux、UNIX®、または Windows® オペレーティング・システムで実行するコマンド行インターフェースです。ASNCLP プログラムは、特に 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使用するコマンドを提供しています。ASNCLP プログラムを使用して、レプリケーション・センターで実行できるタスクの多くを実行できます。ASNCLP プログラムを使用すれば、管理コマンドを対話式に発行できます。これらのコマンドの出力は、常に SQL スクリプトです。これは、即時に実行するか、または後で実行できます。また、ASNCLP プログラムへの入力ファイルで複数のコマンドを提供することもできます。さらに、コマンド行インターフェースを使用して、レプリケーションまたはイベント発行のさまざまな管理タスクを自動化することもできます。

ASNCLP のセットアップおよび使用に関する詳細については、Web 上の製品資料 (<http://www.ibm.com/software/data/dpropr/library.html>) を参照してください。

用語集

用語集

[ア行]

アーカイブ・ログ (archive log). (1) クローズされ、通常の処理には必要なくなったログ・ファイルの集まり。ロールフォワード・リカバリーで使うために保存される。(2) アクティブ・ログからコピーされたログ・レコードが入っている DB2 Universal Database for z/OS ログの部分。アーカイブ・ログには適さなくなったレコードが保留されている。

アプライ (apply). アプライ (apply). レプリケーションでは、レプリケーション・ターゲット表をリフレッシュしたり更新したりすること。

アプライ修飾子 (Apply qualifier). SQL レプリケーションでは、アプライ・プログラムの各インスタンスごとにユニークなレプリケーション・サブスクリプション・セットを識別する大文字小文字の区別のある文字ストリング。

アプライ待ち時間 (Apply latency). SQL レプリケーションでは、レプリケーションが 1 つのサイクルを完了するために必要なおおよその時間。「キャプチャー待ち時間 (Capture latency)」も参照。

アプライ・コントロール・サーバー (Apply control server). SQL レプリケーションでは、アプライ・コントロール表が含まれるデータベースで、サブスクリプション・セットおよびそのメンバーに関する情報を保管する。「アプライ・サーバー (Apply server)」と対比。

アプライ・サーバー (Apply server). SQL レプリケーションでは、アプライ・プログラムが実行中のシステム。「アプライ・コントロール・サーバー (Apply control server)」と対比。

アプライ・サイクル (Apply cycle). SQL レプリケーションでは、データがソース表からターゲット表へ複製される間の時間間隔。

アプライ・プログラム (Apply program). SQL レプリケーションでは、レプリケーション・ターゲット表を最新表示したり更新したりするのに使われるプログラム。「キャプチャー・プログラム (Capture program)」および「キャプチャー・トリガー (Capture trigger)」と対比。

アラート (alert). レプリケーションでは、レプリケーション中のイベントと条件を記述した通知。レプリケーション・アラート・モニターは、E メールかページャーを使用してアラートを送信する。

アラート条件 (alert condition). レプリケーションでは、レプリケーション・アラート・モニターがアラートを送信することになるレプリケーション環境の条件。アラート条件には 3 種類あり、それは状況によって起動されるアラート条件、イベントによって起動されるアラート条件、およびしきい値によって起動されるアラート条件である。レプリケーション・アラート・モニターがレプリケーション環境をチェックする際のアラート条件を選択できる。

異機種のレプリケーション (heterogeneous replication). DB2 と非 DB2 リレーショナル・データベース間のレプリケーション。「フェデレーテッド・データベース・システム (federated database system)」も参照。

イベント発行 (event publishing). DB2 リカバリー・ログからトランザクション・データをキャプチャーし、そのデータを XML メッセージとして発行する、データ発行ソリューション。XML メッセージは WebSphere MQ キューに発行され、1 つ以上のユーザー・アプリケーションはそのメッセージを検索して利用できる。

イベント・タイミング (event timing). SQL レプリケーションでは、レプリケーション・サブスクリプション・サイクルを開始するタイミングを制御する最も精密なメソッド。「インターバル・タイミング (interval timing)」と対比。

インターバル・タイミング (interval timing). SQL レプリケーションでは、サブスクリプション・サイクルを開始するタイミングを制御する最も簡単な方法。インターバル・タイミングを使用する際には、サブスクリプション・サイクルを開始する日付と時刻を指定し、サブスクリプション・サイクルを実行する頻度を示す時間間隔を設定する。「イベント・タイミング (event timing)」と対比。

ウォーム・スタート (warm start). レプリケーションでは、キャプチャー・プログラムまたは Q キャプチャー・プログラムを、オフになったポイントからトランザクションを読み取れるように開始するプロセス。「コールド・スタート (cold start)」と対比。

エージェント・スレッド (agent thread). Q レプリケーションでは、Q アプライ・プログラムのスレッドの 1 つで、ブラウザー・スレッドからトランザクションを受信し、このデータを同じサーバー上のターゲット表にアプライする。ブラウザー・スレッドごとに 1 つ以上のエージェント・スレッドを使用できる。

エンドツーエンド待ち時間 (end-to-end latency). レプリケーションでは、レプリケーションがソース・データベースから変更をキャプチャーし、それをターゲット・データベースにアプライするために必要なおおよその時間。「アプライ待ち時間 (Apply latency)」、「キャプチャー待ち時間 (Capture latency)」、「Q アプライ待ち時間 (Q Apply latency)」、および「Q キャプチャー待ち時間 (Q Capture latency)」も参照。

オブジェクト (object). (1) SQL で作成または操作できるもの。たとえば、表、ビュー、索引、パッケージなど。(2) オブジェクト指向設計またはオブジェクト指向プログラミングでは、データとそのデータに関連付けられた操作からなる抽象的なもの。(3) NetWare では、ネットワーク上で定義されたエンティティで、ファイル・サーバーへのアクセスを持つ。(4) インフォメーション・カタログ・センターでは、単位または特殊な情報のグループを示す項目。インフォメーション・カタログ・センターの各オブジェクトは情報を識別して記述するが、実際の情報は含まれない。たとえば、オブジェクトはレポートの名前を提供し、その作成日をリストし、さらにその目的を記述する。

[力行]

外部 CCD 表 (external CCD table). SQL レプリケーションでは、登録されたレプリケーション・ソースであるために、直接サブスクライブできる CCD 表。その独自の行がレジスター表の中に存在し、レジスター表では SOURCE_OWNER および SOURCE_TABLE 列として識別されている。「整合変更データ表 (consistent-change data table)」も参照。「内部 CCD 表 (internal CCD table)」と対比。

カスケード・リジェクト (cascade rejection). SQL レプリケーションでは、競合が検出され、しかもそれ自身がリジェクトされたトランザクションと関連しているためにレプリケーション・トランザクションをリジェクトするプロセス。

監査証跡 (audit trail). イベントのシーケンスをリンクする論理パスの書式データ。監査証跡は、レコードの内容に影響するトランザクションをトレースする。

管理キュー (administration queue). Q レプリケーションおよびイベント発行では、Q キャプチャー・プロ

ラムと Q アプライ・プログラムまたはユーザー・アプリケーションとの間の通信に使用される WebSphere MQ キュー。個々の Q キャプチャー・プログラムの管理キューは、ローカルな永続キューでなければならない。

キー (key). (1) レプリケーションでは、表、索引、または参照制約の記述内で識別される列または順番に並べた列の集合。同じ列が複数のキーの一部となることができる。(2) Q レプリケーションでは、Q サブスクリプションで指定されているソース表とターゲット表の両方にある 1 つ以上のマッチング列。

基礎集約表 (base aggregate table). SQL レプリケーションでは、レプリケーション・ソース表から集約されるデータが含まれるレプリケーション・ターゲット表のタイプ。「変更集約表 (change aggregate table)」と対比。

ギャップ (gap). SQL レプリケーションでは、キャプチャー・プログラムが一定範囲のログまたはジャーナル・レコードを読み取ることができず、変更データを失う可能性のある状態。

キャプチャー (capture). レプリケーションでは、イベント発行へのレプリケーションのためにソース・データベースから変更を収集すること。

キャプチャー待ち時間 (Capture latency). SQL レプリケーションでは、キャプチャー・プログラムが、CD 表にどのくらい前にデータをコミットしたかを示すおおよその測定。「アプライ待ち時間 (Apply latency)」も参照。

キャプチャー・コントロール・サーバー (Capture control server). (1) SQL レプリケーションでは、キャプチャー・コントロール表が含まれるデータベースで、登録済みレプリケーション・ソース表に関する情報を保管する。(2) キャプチャー・プログラムが実行中のシステム。

キャプチャー・スキーマ (Capture schema). SQL レプリケーションでは、キャプチャー・プログラムの特別なインスタンスで使用されるコントロール表を識別する名前。

キャプチャー・トリガー (Capture trigger). SQL レプリケーションでは、DB2 以外のリレショナル・ソース表に対して実行される削除、挿入、および更新操作を収集するためのメカニズム。「キャプチャー・プログラム (Capture program)」および「アプライ・プログラム (Apply program)」と対比。

キャプチャー・プログラム (Capture program). SQL レプリケーションでは、DB2 ソース表に加えられた変更をキャプチャーするために、データベース・ログまた

はジャーナルの記録を読み取るプログラム。「アプライ・プログラム (Apply program)」および「キャプチャー・トリガー (Capture trigger)」と対比。

キュー (queue). WebSphere MQ オブジェクト。メッセージ・キューイング・アプリケーションは、キューでのメッセージの書き込みとメッセージの読み取りができる。Q キャプチャー・プログラムと Q アプライ・プログラムは、キュー中にメッセージを挿入したりキューからメッセージを受け取ったりできる。キューは、キュー・マネージャーによって所有および管理される。

キュー待ち時間 (queue latency). Q レプリケーションおよびイベント発行では、Q キャプチャー・プログラムが送信キューに挿入してから、Q アプライ・プログラムが受信キューからトランザクションを受け取るまでの間の時間。

キュー・マップ (queue map). Q レプリケーションおよびイベント発行では、キューをリンクし、Q キャプチャー・プログラムと Q アプライ・プログラムがキューを使用するメッセージを処理する方法を定義するオブジェクト。発行キュー・マップとレプリケーション・キュー・マップの 2 種類のキュー・マップがある。「発行キュー・マップ (publishing queue map)」および「レプリケーション・キュー・マップ (replication queue map)」も参照。

行キャプチャー規則 (row-capture rules). SQL レプリケーションでは、キャプチャー・プログラムが行を CD 表にいつ書き込むか、書き込むかどうか、またはキャプチャー・トリガーが行を CCD 表にいつ書き込むか、書き込むかどうかを定義した登録済み列への変更を基にした規則。

競合検出 (conflict detection). 双方向レプリケーションおよび Update-anywhere レプリケーションでは、次のプロセスのいずれかを競合検出が参照する。

- キー制約や参照制約などの制約エラーの検出プロセス。
- ソース表とターゲット表の両方の同じ行が、同じレプリケーション・サイクルの中でユーザーまたはアプリケーションによって更新されたかどうかを検出するプロセス。

許可トークン (authorization token). (1) トランザクションに関連したトークン。(2) DB2 Universal Database for z/OS では相関 ID を指す。(3) DB2 Universal Database for iSeries では、トランザクションの原因となるジョブの名前を指す。

区切り ID (delimited identifier). 二重引用符 (") で囲まれたストリング。このストリングは、1 文字とその後

につづくゼロ個以上の文字 (そのおのおのが文字、数字、または下線文字) で構成されていなければならない。「通常識別子 (ordinary identifier)」も参照。

クライアント (client). データベース・サーバーと通信したりアクセスしたりするプログラム、またはプログラムが稼働するサーバー。

グローバル・レコード (global record). SQL レプリケーションでは、キャプチャー・プログラムの特別なインスタンスに対してグローバル・レプリケーション特性を定義する登録表の行。

結合 (join). 列値を突き合わせて複数の表からデータを取り出すための SQL 関係操作。

コールド・スタート (cold start). (1) SQL レプリケーションでは、キャプチャー・プログラムの前の操作からの再始動情報を使用しないで、そのプログラムを開始するプロセス。コールド・スタートを実行すると、フル・リフレッシュが行われ、アクティブ・サブスクリプションがすべて非アクティブ化されてからアクティブ化される。コールド・スタート・プロセスは、コールド・スタートの前に処理されなかったトランザクションをすべて削除する。「ウォーム・スタート (warm start)」と対比。(2) Q レプリケーションでは、Q キャプチャー・プログラムの前の操作からの再始動情報を使用しないで、そのプログラムを開始するプロセス。コールド・スタートを実行すると、フル・リフレッシュが行われる。コールド・スタートの前に処理されなかったすべてのトランザクションは、コールド・スタート後に処理される。ユーザーは、コールド・スタートの前に、キューからトランザクションを消去する作業を行う。「ウォーム・スタート (warm start)」と対比。(3) 初期プログラム・ロード・プロシージャを使用して、システムまたはプログラムを開始するプロセス。(4) DB2 Universal Database for z/OS がどのログ・レコードも処理しないで再始動するときのプロセス。

高可用性災害時リカバリー (high-availability disaster recovery). 必要な時点でいつでも従属アプリケーションが複製されたデータを使用でき、大災害時の障害によるデータ欠落から保護するレプリケーション構成。

コンデンス CCD 表 (condensed CCD table). SQL レプリケーションでは、行の最新の値のみを含み、各キー値ごとに 1 つの行のみを持つ CCD 表。「非コンデンス CCD 表 (noncondensed CCD table)」と対比。「整合変更データ表 (consistent-change data (CCD) table)」も参照。

コンデンス (condensed). SQL レプリケーションでは、データに加えられた変更の履歴ではなく、現行データが表に入っていることを示す表属性。コンデンス表に

は、表内の各主キー値につき 1 つ以上の行が含まれることはない。そのため、コンデンス表は、最新表示のための現行情報を提供するのに使うことができる。

コントロール表 (control table). 「レプリケーション・コントロール表 (replication control table)」を参照。

コントロール・サーバー (control server). SQL レプリケーションでは、キャプチャー・プログラム、アプライ・プログラム、またはレプリケーション・アラート・モニターに対するレプリケーション・コントロール表を含むデータベース・サーバー。「アプライ・コントロール・サーバー (Apply control server)」、「キャプチャー・コントロール・サーバー (Capture control server)」、「Q アプライ・サーバー (Q Apply server)」、「Q キャプチャー・サーバー (Q Capture server)」、および「モニター・コントロール・サーバー (Monitor control server)」も参照。

コントロール・センター (Control Center). DB2 のグラフィカル・インターフェースで、これを使用して DB2 データベースを管理したり、オブジェクトの作成やパフォーマンスのモニターを含むさまざまなタスクを実行できる。コントロール・センターは、データベース・オブジェクト (データベースや表など) とその相互関係を示す。

コントロール・メッセージ (control message). Q レプリケーションでは、Q アプライ・プログラムまたはユーザー・アプリケーションからのメッセージで、Q サブスクリプションや XML 発行のアクティブ化や非アクティブ化、送信キューの無効化、またはターゲット表のロードの確認について Q キャプチャー・プログラムに尋ねる。

コンプリート CCD 表 (complete CCD table). SQL レプリケーションでは、初期状態で、レプリケーション・ソース表またはビューからのすべての行と、ソース表またはビューからの任意の述部が含まれる CCD 表。「非コンプリート CCD 表 (noncomplete CCD table)」と対比。「整合変更データ表 (consistent-change data (CCD) table)」も参照。

[サ行]

サーバー (server). 「論理サーバー (logical server)」を参照。「アプライ・コントロール・サーバー (Apply control server)」、「アプライ・サーバー (Apply server)」、「キャプチャー・コントロール・サーバー (Capture control server)」、「コントロール・サーバー (Control server)」、「モニター・コントロール・サーバー (Monitor control server)」、「Q アプライ・サーバー

(Q Apply server)」、「Q キャプチャー・サーバー (Q Capture server)」、「ソース・サーバー (source server)」、および「ターゲット・サーバー (target server)」も参照。

再キャプチャー (recapture). Update-anywhere レプリケーションで、レプリカ表で変更をキャプチャーし、これらの変更をマスター表か他のレプリカ表に転送することを指す。

再作業 (rework). (1) レプリケーション・ターゲット表への挿入を、ターゲット表にすでに行が存在するために挿入が失敗した場合、更新へ変換すること。(2) レプリケーション・ターゲット表への更新を、ターゲット表に行が存在しないために更新が失敗した場合、挿入へ変換すること。

作業単位 (unit of work). (1) アプリケーション・プロセスにおけるリカバリー可能な一連の操作。アプリケーション・プロセスは、常に単一の作業単位であるが、コミットまたはロールバック操作のため、アプリケーション・プロセス全体には複数の作業単位が関与する。

DB2 Universal Database for z/OS マルチサイト更新操作では、単一の作業単位に複数のリカバリー単位 (unit of recovery) を組み込むことができる。「トランザクション (transaction)」の同義語。(2) インフォメーション・カタログ・センターでは、アプリケーション・プロセスにおけるリカバリー可能な一連の操作。アプリケーション・プロセスは、常に単一の作業単位であるが、コミットまたはロールバック操作のため、アプリケーション・プロセス全体には複数の作業単位が関与する。

作業単位 (UOW) 表 (unit-of-work (UOW) table).

SQL レプリケーションでは、データベース・ログまたはジャーナルから読み取ったコミット・レコードを含むキャプチャー・コントロール・サーバーに保管されたレプリケーション・コントロール表。レコードはトランザクションまたは UOW が正常にコミットされたことを示し、リカバリー単位 ID を組み込んでいる。このリカバリー単位 ID は、作業単位表と CD 表を結合させて、トランザクション間で整合性のある変更データを生成するために使用される。

作業ファイル (work file). SQL レプリケーションでは、サブスクリプション・セットの処理中にアプライ・プログラムが使用する一時ファイル。

サブスクリプション (subscription). (1) SQL レプリケーションでは、サブスクリプション・セットとサブスクリプション・セット・メンバーを作成するオブジェクト。SQL レプリケーションの場合は「登録 (registration)」、および Q レプリケーションの場合は

「*Q* サブスクリプション (*Q subscription*)」と対比。(2) 「サブスクリプション・セット (*subscription set*)」も参照。

サブスクリプション・サイクル (subscription cycle). SQL レプリケーションが所定のサブスクリプション・セットの変更済みデータを取り出し、変更内容をターゲット表に複製し、その状況と現在の進行を反映するように、該当するレプリケーション・コントロール表を更新するプロセス。

サブスクリプション・セット (subscription set). SQL レプリケーションで、サブスクリプション・サイクル中に変更されたデータのレプリケーションをコントロールするレプリケーション定義。サブスクリプション・セットにはサブスクリプション・セット・メンバーが含まれる場合がある。

サブスクリプション・セット・メンバー (subscription-set member). SQL レプリケーションで、登録済みレプリケーション・ソースをレプリケーション・ターゲットでマップするレプリケーション定義。各メンバーは、ターゲット表の構造と、ソース表から複製される行と列を定義する。

サブセット (subset). 表全体からターゲット表へ複製するのではなく、ソース表の一部からデータを複製すること。行単位または列単位のサブセットが可能である。

差分リフレッシュ・レプリケーション (differential-refresh replication). 「変更キャプチャー・レプリケーション (*change-capture replication*)」を参照。

参照制約 (referential constraints). 外部キーの非ヌル値が有効なのは親キーの値として表示されるときだけである、という参照保全規則。

参照保全 (referential integrity). 外部キーの値がすべて有効であるデータベースの状態。参照保全の保守には、参照制約が定義されている表のデータを変更するすべての操作に、参照制約 (*referential constraints*) の制約が必要である。

シグナル (signal). キャプチャー・プログラムと *Q* キャプチャー・プログラムとの通信を行える、レプリケーションの通信メカニズム。シグナルは、シグナル・コントロール表に挿入され、キャプチャー・プログラムか *Q* キャプチャー・プログラムがシグナル挿入に関するログ項目を読み取る際に受け取る SQL ステートメントである。

ジャーナリング (journaling). iSeries システムでは、物理ファイル・メンバーやアクセス・パスなどのオブジ

ェクトへの変更をジャーナルへ記録する処理、またはシステム関数やユーザー関数を使用してジャーナル項目をデポジットする処理。

ジャーナル ID (JID) (journal identifier (JID)). iSeries システムでは、あるオブジェクトに対してジャーナリングを開始すると、そのオブジェクトに割り当てられるユニークな ID。この JID 値を使用して、オブジェクトとジャーナル項目が関連付けられる。

ジャーナル (journal). iSeries システムでは、記録 (ジャーナル) されるオブジェクトを識別するシステム・オブジェクト、現行のジャーナル・レシーバー、およびジャーナル用システム上のすべてのジャーナル・レシーバーを指す。「ジャーナル・レシーバー (*journal receiver*)」も参照。

ジャーナル項目 (journal entry). iSeries システムでは、ジャーナルされた変更や、ジャーナルされたその他の活動についての情報を含む、ジャーナル・レシーバー内の 1 つのレコード。「ジャーナル・コード (*journal code*)」および「ジャーナル項目タイプ (*journal entry type*)」も参照。

ジャーナル項目タイプ (journal entry type). iSeries システムでは、システム生成ジャーナル項目の操作タイプ、またはユーザー生成ジャーナル項目のタイプを識別する、ジャーナル項目内の 2 文字フィールド。たとえば、PT は書き込み操作の項目タイプである。「ジャーナル・コード (*journal code*)」も参照。

ジャーナル・コード (journal code). iSeries システムでは、ジャーナル項目のカテゴリーを識別する、ジャーナル項目内の 1 文字コード。たとえば、F は、ファイルへの操作を識別し、R は、レコードへの操作を識別する、など。「ジャーナル項目 (*journal entry*)」も参照。

ジャーナル・レシーバー (journal receiver). iSeries システムでは、データベース・ファイルへの変更、他のジャーナル・オブジェクト、あるいはセキュリティ関連イベントへの変更などのイベントが発生した時に追加されるジャーナル項目が含まれるシステム・オブジェクト。「ジャーナル (*journal*)」も参照。

集約表 (aggregate table). SQL レプリケーションでは、ソース表からのデータの集約が含まれる読み取り専用のレプリケーション・ターゲット表。このデータは MIN、MAX、SUM、AVG といった SQL 列関数を基にしている。

主キー (primary key). 表の定義の一部であるユニーク・キー。主キーは、参照制約定義の親キーのデフォルト。表の行を固有に識別する列または複数の列の組み合わせ。

受信キュー (receive queue). Q レプリケーションでは、Q キャプチャー・プログラムによってキャプチャーされたトランザクションを受信するのに、Q アプライ・プログラムで使用される WebSphere MQ メッセージ・キュー。

述部 (predicate). 比較演算を明示または暗黙指定する検索条件のエレメント。

シリアルイゼーション (serialization). (1) 項目の連続配列。(2) リソースの保全性を保護するリソースへのアクセスを制御する処理。(3) Q レプリケーションでは、トランザクションをソースでコミットされた順序でアプライするプロセス。

ステージング表 (staging table). SQL レプリケーションでは、データがターゲット・データベースへ複製される前に、データを保管するために使用される CCD 表。ステージング・データに使用される CCD 表は、1 つ以上のターゲット表へデータを更新するための中間ソースとして機能する。「整合変更データ表 (*consistent-change data table*)」も参照。

整合変更データ (CCD) 表 (consistent-change data (CCD) table). SQL レプリケーションでは、履歴の保管、データの監査、またはデータのステージングに使用されるレプリケーション・ターゲット表のタイプ。CCD 表はレプリケーション・ソースでもある。「コンプリート CCD 表 (*complete CCD table*)」、「コンデンス CCD 表 (*condensed CCD table*)」、「外部 CCD 表 (*external CCD table*)」、「内部 CCD 表 (*internal CCD table*)」、「非コンプリート CCD 表 (*noncomplete CCD table*)」、および「非コンデンス CCD 表 (*noncondensed CCD table*)」も参照。

整理 (pruning). レプリケーションでは、キャプチャー・プログラム、Q キャプチャー・プログラム、アプライ・プログラム、または Q アプライ・プログラムで使用されるレプリケーション・コントロール表またはログ・ファイルから、古いデータを除去するタスク。

ソース表 (source table). レプリケーションでは、ターゲット表に複製されるデータを含む表。「ターゲット表 (*target table*)」と対比。

ソース・サーバー (source server). レプリケーションでは、ソース表を含むデータベースまたはサブシステム。

送信キュー (send queue). Q レプリケーションでは、Q キャプチャー・プログラムによってキャプチャーされたトランザクションを発行するのに、このプログラムで使用される WebSphere MQ メッセージ・キュー。

送信キューは、Q レプリケーションまたはイベント発行で使用できるが、これらの両方で同時に使用できない。

双方向レプリケーション (bidirectional replication). Q レプリケーションでは、レプリケーション構成の一種で、表のコピーの 1 つに加えられた変更内容が、その表の 2 つ目のコピーに複製される。2 つ目のコピーに加えられた変更内容は、1 つ目のコピーに複製し戻される。競合が発生した場合は、表のどちらのコピーを優先するか選択しなければならない。

[タ行]

ターゲット表 (target table). (1) SQL レプリケーションでは、登録済みレプリケーション・ソースから複製された変更に対する宛先である表。ターゲット表には、ユーザー・コピー表、ポイント・イン・タイム表、基礎集約表、変更集約表、CCD 表またはレプリカ表がある。(2) Q レプリケーションでは、Q サブスクリプションの一部であるソースから複製された変更に対する宛先である表。

ターゲット・サーバー (target server). (1) SQL レプリケーションでは、レプリケーション・ターゲット表、ビュー、またはストアド・プロシージャを含むデータベースまたはサブシステム。(2) Q レプリケーションでは、レプリケーション・ターゲット表またはストアド・プロシージャを含むデータベースまたはサブシステム。

タイム・スタンプ (timestamp). 年、月、日、時、分、秒、およびマイクロ秒で表される日時で構成される、7 つの部分から成る値を含むデータ・タイプ。

単一方向レプリケーション (unidirectional replication). Q レプリケーションでは、ソース表に加えられた変更内容が、WebSphere MQ キューを使用してターゲット表に複製されるか、データ操作のためにストアド・プロシージャに渡されるレプリケーション構成。ターゲット表に加えられた変更内容は、ソース表に複製され戻されない。

通常識別子 (ordinary identifier). (1) SQL では、名前を作成するために使用される文字。英字 (a~z および A~Z) の後に、ゼロ個以上の英字、記号、数字、または下線文字が続く。(2) DB2 Universal Database for z/OS では、英大文字の後にゼロ個以上の文字が続く。それぞれの文字は英大文字、数値、数字、または下線文字である。

通知メッセージ (informational message). Q レプリケーションおよびイベント発行では、Q キャプチャー・プログラムが送信して Q アプライ・プログラムまたは

ユーザー・アプリケーションに通知する、Q キャプチャー・プログラム、Q サブスクリプション、または XML 発行の状況に関するメッセージ。

データ分散レプリケーション (data distribution replication). レプリケーションでは、シングル・ソース表が含まれるレプリケーション構成。この構成から、変更が 1 つ以上の読み取り専用ターゲット表に複製される。ターゲット表へのレプリケーションが発生する前に、ソース表からのデータの完全セットが表に含まれている必要がある。

データベース管理システム (DBMS) (database management system (DBMS)). 「データベース・マネージャー (database manager)」を参照。

データベース・マネージャー (database manager). 効率のよいアクセス、整合性、リカバリー、データの現行性制御、プライバシー、およびセキュリティのための中央制御、データ独立性、および複合物理構造の各種サービスを提供することによってデータを管理するプログラム。

データベース・リカバリー・ログ (database recovery log). レプリケーションでは、ログ・レコード中にデータベースに対するすべての変更内容を記録する一連の 1 次および 2 次ログ・ファイル。

データ・ブロッキング (data blocking). SQL レプリケーションでは、アプライ・サイクル中にデータを変更する時間に相当する特定の分数 (時間) を複製する処理。

データ・メッセージ (data message). Q レプリケーションでは、ソース表からの以下のいずれかの内容を含むメッセージ。

- すべてまたは一部のトランザクション
- 単一行の操作
- トランザクション中の行操作からのすべてまたは一部のラージ・オブジェクト (LOB) 値

同期点 (synchpoint). SQL レプリケーションでは、最新のアプライ・サイクル中に適用された最新の変更の DB2 ログまたはジャーナル・レコードのシーケンス番号に対するレプリケーション・コントロール表の値。この値は、CD 表の整理を調整するためにも使用される。

同期レプリケーション (synchronous replication). リアルタイム・レプリケーションとしても認識されている。更新をソース・トランザクションの有効範囲内で連続して送達するレプリケーションのタイプ。

登録 (registration). (1) SQL レプリケーションでは、DB2 の表、ビュー、またはニックネームをレプリケー

ション・ソースとして登録する処理。「サブスクリプション (subscription)」と対比。(2) 「レプリケーション・ソース (replication source)」を参照。

特殊タイプ (distinct type). 内部的には既存のタイプ (そのソース・タイプ) として表示されるが、セマンティクスとしての用途においては別個で非互換のタイプと見なされるユーザー定義のデータ・タイプ。「ユーザー定義タイプ (UDT) (user-defined type) (UDT)」も参照。

独立補助記憶域プール (IASP) (independent auxiliary storage pool (IASP)). アドレス可能ディスク装置を構成する、ディスク装置またはディスク装置サブシステムから定義される 1 つ以上の記憶装置。独立補助プールには、オブジェクト、オブジェクトを含むディレクトリ、および許可所有権属性などの他のオブジェクト属性が入る。独立補助記憶域プールは、システムを再始動させずに使用可能 (オン)/使用不可 (オフ) にすることができる。独立補助プールは、クラスタリング環境内の複数のシステム間で切り替え可能となるか、または単一システムに私的に接続される。

トランザクション (transaction). サーバーとプログラム、2 つのサーバー、または 2 つのプログラム間での交換で、特定のアクションまたは結果が伴うもの。たとえば、カスタマーの預金の入力、カスタマーの貸借の更新など。「作業単位 (unit of work)」の同義語。

トランザクション整合レプリケーション (transaction-consistent replication). SQL レプリケーションでは、すべてのトランザクションの更新が、ターゲット表に複製されるレプリケーション処理のタイプ。「トランザクション・ベース・レプリケーション (transaction-based replication)」と対比。

トランザクション・ベース・レプリケーション (transaction-based replication). SQL レプリケーションでは、すべてのトランザクションがソース表でコミットされている時に、ターゲット表に複製されるレプリケーション処理のタイプ。「トランザクション整合レプリケーション (transaction-consistent replication)」と対比。

トランザクション・モード処理 (transaction-mode processing). SQL レプリケーションでは、レプリケーション・サブスクリプション・セット処理のタイプで、アプライ・プログラムがソース CD 表からデータを検索し、次にデータをソースで使用したシーケンスと同じコミット・シーケンスでターゲット表に適用する。アプライ・プログラムは、順次ではなく、すべてのサブスクリプション・セット・メンバーをまとめてトランザクションを処理する。「表モード処理 (table-mode processing)」と対比。

トリガー (trigger). (1) 特定の SQL が実行されているときに、データベース・マネージャーが間接的に呼び出すデータベースのオブジェクト。「キャプチャー・トリガー (Capture trigger)」も参照。(2) DB2 データベースに保管され、DB2 表にあるイベントが発生したときに実行される一連の SQL ステートメント。

トレース (trace). (1) レプリケーションでは、キャプチャー・プログラム、Q キャプチャー・プログラム、アプライ・プログラム、Q アプライ・プログラム、またはレプリケーション・アラート・モニターのモニター、監査、およびパフォーマンス・データを収集する能力を提供する機能を指す。(2) モニター、監査、パフォーマンス、アカウントिंग、統計、および保守容易性 (グローバル) 関連のデータをモニターし、収集する能力を提供する DB2 Universal Database for z/OS の機能。

[ナ行]

内部 CCD 表 (internal CCD table). SQL レプリケーションでは、登録されたレプリケーション・ソースでないために、直接サブスクリプションできない CCD 表。レジスター表にはその独自の行が存在しない。関連した登録済みレプリケーション・ソースの行に対する CCD_OWNER および CCD_TABLE 列によって識別される。「外部 CCD 表 (external CCD table)」と対比。「整合変更データ表 (consistent-change data (CCD) table)」も参照。

ニックネーム (nickname). (1) データ・ソース・オブジェクト (表やビューなど) を参照するためにフェデレーテッド・サーバーが使用する ID。(2) DB2 以外のデータベースの物理データベース・オブジェクト (表やストアド・プロシージャなど) を表すために、DB2 V8 for Informix ソース・データベースまたは DB2 II データベースで定義された名前。

[ハ行]

バイナリー・ラージ・オブジェクト (BLOB) (binary large object (BLOB)). サイズが 0 バイトから 2 ギガバイト未満の範囲のバイト・シーケンスを含むデータ・タイプ。このストリングには、関連したコード・ページおよび文字セットがない。BLOB にはイメージ、音声、およびビデオ・データが含まれる。「文字ラージ・オブジェクト (character large object)」および「2 バイト文字ラージ・オブジェクト (double-byte character large object)」も参照。

パッケージ (package). SQL ステートメントを実行するために使用されるプログラム準備中に作成される制御構造。

発行キュー・マップ (publishing queue map). イベント発行では、メッセージ送信用の送信キューと、送信キューを使用するすべてのトランザクションが Q キャプチャー・プログラムで処理される方法に関する設定を含むオブジェクト。「レプリケーション・キュー・マップ (replication queue map)」および「キュー・マップ (queue map)」も参照。

ピアツーピア・レプリケーション (peer-to-peer replication). Q レプリケーションでは、対等な表の間のレプリケーション構成の一種で、いずれかの表に加えられた変更内容が他の表に複製され、収束が保守される。ピアツーピア・レプリケーションでは、サーバーは 2 つでも 3 つ以上でもよい。「Update-anywhere レプリケーション (Update-anywhere replication)」と対比。「multi-tier レプリケーション (multi-tier replication)」も参照。

非コミット読み取り (UR) (uncommitted read (UR)). アプリケーションが他のトランザクションの非コミット変更にアクセスするのを可能にする分離レベル。他のアプリケーションが表を消去または変更しようとしないう限り、アプリケーションは自分が読み取っている行以外のアプリケーションをロックしない。

非コンデンス CCD 表 (noncondensed CCD table). SQL レプリケーションでは、キー値ごとに、複数の行が含まれた CCD 表。これらの複写行は、表の行にある値に対する変更履歴を表す。「コンデンス CCD 表 (condensed CCD table)」と対比。「整合変更データ表 (consistent-change data (CCD) table)」も参照。

非コンプリート CCD 表 (noncomplete CCD table). SQL レプリケーションでは、初期状態は空で、レプリケーション・ソースに変更が行われるにつれて行が追加される CCD 表。「コンプリート CCD 表 (complete CCD table)」と対比。「整合変更データ表 (consistent-change data (CCD) table)」も参照。

非常時接続 (occasionally connected). SQL レプリケーションでは、常にネットワークに接続している訳ではないターゲット・サーバーが含まれるレプリケーション構成。この構成により、ユーザーはローカル・データベースをソースのデータと短期間同期化するために、基本データ・ソースに接続することができる。

非同期レプリケーション (asynchronous replication). レプリケーションでは、ソース表を更新した元のトランザクションの有効範囲外にあるターゲット表へ、ソース表からデータをコピーする処理。「同期レプリケーション (synchronous replication)」と対比。

ビュー (view). (1) 照会によって生成されたデータからなる論理表。基本表の基礎設定に基づいており、ビュー

のデータは基本表で実行されている SELECT タイプの照会によって判別される。(2) オブジェクトに関する情報とオブジェクトに含まれる情報を参照する方法。それぞれのビューは、オブジェクトに関する異なった情報を明らかにする場合がある。

表モード処理 (table-mode processing). SQL レプリケーションでは、レプリケーション・サブスクリプション・セット処理のタイプで、アプライ・プログラムがソース CD 表からすべてのデータを検索し、次にデータをターゲット表ごとに適用 (1 回に 1 メンバー) し、最終的にこの作業をコミットする。「トランザクション・モード処理 (transaction-mode processing)」と対比。

フェデレーテッド・データベース・システム (federated database system). 分散データベース管理システム (DBMS) の特殊タイプ。他のサーバーにあるデータの照会と操作が可能になる。データは、Oracle、Sybase、Microsoft SQL Server、Informix、Teradata などのデータベース・マネージャーにあるか、あるいはスプレッドシート、Web サイト、データマートなどのリストまたは格納先にある。SQL ステートメントは、1 つの文内で、複数のデータベース・マネージャーや個々のデータベース (複数) を参照することができる。たとえば、DB2 Universal Database 表、Oracle の表、および Sybase のビューを結合することができる。

複数方向レプリケーション (multidirectional replication). Q レプリケーション環境では、ピアツーピア・レプリケーションまたは双方向レプリケーションを含むレプリケーション構成。

プッシュ構成 (push configuration). SQL レプリケーションでは、アプライ・プログラムが、ターゲット・サーバー以外のソース・サーバーまたはレプリケーション・サーバーで実行するレプリケーション構成。アプライ・プログラムはソース・サーバーからの更新をターゲットにアプライするためにプッシュする。「プル構成 (pull configuration)」と対比。

ブラウザー・スレッド (browser thread). Q レプリケーションでは、Q アプライ・プログラム・スレッドの 1 つで、受信キューからメッセージを受け取り、そのメッセージを 1 つ以上のエージェント・スレッドに渡してターゲットにアプライされるようにする。

プル構成 (pull configuration). SQL レプリケーションでは、アプライ・プログラムがターゲット・サーバーで実行するレプリケーション構成。アプライ・プログラムはソース・サーバーからの更新をターゲットにアプライするためにプルする。「プッシュ構成 (push configuration)」と対比。

フル・リフレッシュ (full refresh). (1) SQL レプリケーションでは、すべてのデータのプロセスが登録に一致し、レプリケーション・ソース表に対するサブスクリプション・セット述部がターゲット表にコピーされる。ターゲット表のロードとも呼ばれる。フル・リフレッシュは、ターゲット表の既存のデータをすべて置換する。

「変更キャプチャー・レプリケーション (change-capture replication)」と対比。(2) Q レプリケーションでは、Q サブスクリプションの検索条件に一致するレプリケーション・ソース表のデータが、すべてターゲット表にコピーされるプロセス。フル・リフレッシュは、ターゲット表の既存のデータをすべて置換する。

ブロッキング (blocking). SQL レプリケーションでは、アプリケーションをバインドするときに指定するオプション。このオプションを指定すると、通信サブシステムが複数行の情報をキャッシュできるようになるため、FETCH ステートメントで各要求を 1 行ずつネットワークに伝送する必要がなくなる。「ブロック・フェッチ (block fetch)」も参照。

ブロック・フェッチ (block fetch). 大量の行を一緒に検索 (またはフェッチ) する DB2 の機能。ブロック・フェッチを使用すると、ネットワークにまたがって送信されるメッセージの数をかなり減らすことができる。データを更新しないカーソルのみに適用する。

プロモート (promote). SQL レプリケーションでは、ソースを再度登録したりサブスクリプション・セットを再度作成したりせずに、あるデータベースから別のデータベースへ、サブスクリプション・セットまたは登録済みソースのレプリケーション定義をコピーすること。

変更キャプチャー・レプリケーション (change-capture replication). SQL レプリケーションでは、レプリケーション・ソース表に対して行われた変更をキャプチャーし、これをレプリケーション・ターゲット表にアプライする処理。「フル・リフレッシュ (full refresh)」と対比。

変更後イメージ (after-image). SQL レプリケーションでは、変更データ (CD) 表またはデータベース・ログカージャーナルに記録されたソース表の列の更新後の内容。「変更前イメージ (before-image)」と対比。

変更後の値 (after-value). Q レプリケーションでは、ソース表の列の更新後の内容。

変更集約表 (change aggregate table). SQL レプリケーションでは、CD 表の内容を基に集約されるデータが含まれるレプリケーション・ターゲット表のタイプ。「基礎集約表 (base aggregate table)」と対比。

変更前イメージ (before-image). SQL レプリケーションでは、トランザクションによる更新の前のレプリケーション・ソース表の列の内容。変更データ (CD) 表、またはデータベース・ログかジャーナルに記録された内容。「変更後イメージ (after-image)」と対比。

変更データ (CD) 表 (change data (CD) table). SQL レプリケーションでは、レプリケーション・ソース表の変更済みデータの入った、キャプチャー・コントロール・サーバーのレプリケーション表。

変更前の値 (before-value). Q レプリケーションでは、トランザクションによる更新の前のレプリケーション・ソース表の列の内容。

ポイント・イン・タイム表 (point-in-time table). SQL レプリケーションでは、すべてまたは一部のソース表と内容が一致するレプリケーション・ターゲット表のタイプ。ソース・システムで特定の行が挿入または更新された大体の時刻を識別する追加列を含む。

保持制限整理 (retention-limit pruning). SQL レプリケーションでは、ユーザーが指定する制限より先のキャプチャー・プログラムによる CD または UOW 表の整理。

補助記憶域プール (ASP) (auxiliary storage pool (ASP)). 補助記憶域を構成する、記憶装置 (storage device) または記憶装置サブシステムから定義される 1 つ以上の記憶装置 (storage unit)。ASP は、記憶装置 (storage device) の障害の影響を制限し、リカバリー時間を削減するためにデータを編成する方法を提供する。

ホット・スポット更新 (hot-spot update). 短い期間で、同じ行に対して行われる一連の繰り返し更新。

[マ行]

マスター表 (master table). SQL レプリケーション、特に Update-anywhere レプリケーションでは、レプリカ表のデータに対するオリジナルのソース表を指す。レプリケーション競合検出が使用可能な場合、マスター表への変更は保存されるが、レプリカ表への変更はリジェクトされる。「Update-anywhere レプリケーション (Update-anywhere replication)」、「レプリカ表 (replica table)」、および「競合検出 (conflict detection)」も参照。

待ち時間 (latency). ソースに加えた更新をターゲットに複製するのに必要な時間。

メンバー (member). 「サブスクリプション・セット・メンバー (subscription-set member)」を参照。

文字ラージ・オブジェクト (CLOB) (character large object (CLOB)). サイズが 0 バイトから 2 ギガバイト未満の範囲の文字シーケンス (1 バイトまたはマルチバイト、あるいはその両方) を含むデータ・タイプ。一般には、CLOB の値は、文字ストリングが VARCHAR タイプの制限を超過した場合にはいつでも使用される。文字ラージ・オブジェクト・ストリングともいう。「バイナリー・ラージ・オブジェクト (binary large object)」および「2 バイト文字ラージ・オブジェクト (DBCLOB) (double-byte character large object (DBCLOB))」も参照。

モニター修飾子 (monitor qualifier). レプリケーションでは、レプリケーション・アラート・モニターの特定のインスタンスを識別する大文字小文字の区別のある文字ストリング。

モニター・コントロール・サーバー (Monitor control server). レプリケーションでは、モニター・コントロール表が含まれるデータベースで、レプリケーション・アラート・モニターがモニターするアラート条件に関する情報を保管する。

[ヤ行]

ユーザー定義タイプ (UDT) (user-defined type) (UDT). データベース・マネージャーにもともとあったものではなく、ユーザーにより作成されたデータ・タイプ。DB2 Universal Database では、ユーザー定義タイプの代わりに「特殊タイプ (distinct type)」という用語を使用する。

ユーザー・コピー表 (user copy table). SQL レプリケーションでは、登録済みソース表の全部または一部と一致する内容を持つレプリケーション・ターゲット表。ユーザー・データ列のみを含む。

ユニーク索引 (unique index). 表に同一のキー値がないことを保証する索引。

ユニーク・キー (unique key). 値が同じものがないように制約されているキー。

ユニコード (Unicode). ISO 10646 標準のサブセットである国際文字エンコード・スキーム。各文字はユニークな 2 バイト・コードで定義される。

予備エージェント・スレッド (spill agent thread). Q レプリケーションでは、予備キュー中で待っているトランザクションをアプライし、予備キューが空になったり削除されたりするとブラウザー・スレッドに通知するスレッド。

予備キュー (spill queue). Q レプリケーションでは、Q アプライ・プログラムが作成し、ターゲット表のロード中にソース表で生じるトランザクションを保持する動的キュー。その後 Q アプライ・プログラムは、これらのトランザクションをアプライしてから、予備キューを削除する。

予備ファイル (spill file). SQL レプリケーションでは、アプライ・プログラムが作成する一時ファイルで、ターゲット表を更新するためのデータを保持するために使用される。

[ラ行]

ラージ・オブジェクト (LOB) (large object (LOB)). サイズが 0 バイトから 2 ギガバイト未満の範囲のバイト・シーケンスを含むデータ・タイプ。バイナリー・ラージ・オブジェクト (binary large object) (バイナリー)、文字ラージ・オブジェクト (character large object) (1 バイト文字または混合文字)、および 2 バイト文字ラージ・オブジェクト (double-byte character large object) (2 バイト文字) の 3 つのタイプのラージ・オブジェクトがある。「バイナリー・ラージ・オブジェクト (BLOB) (binary large object (BLOB))」、「文字ラージ・オブジェクト (CLOB) (character large object (CLOB))」、および「2 バイト文字ラージ・オブジェクト (DBCLOB) (double-byte character large object (DBCLOB))」を参照。

リアルタイム・レプリケーション (real-time replication). 「同期レプリケーション (synchronous replication)」を参照。

リジェクトされたトランザクション (rejected transaction). マスター表と競合するレプリカ表からの 1 つ以上の更新データを含むトランザクション。

リモート・データベース (remote database). 現在使用しているサーバー以外のワークステーションに物理的に置かれているデータベース。「ローカル・データベース (local database)」と対比。

リレーショナル・データベース管理システム (RDBMS) (relational database management system (RDBMS)). リレーショナル・データベースへのアクセスを編成および提供するハードウェアとソフトウェアの集まり。

レジスター (register). SQL レプリケーションでは、DB2 の表、ビュー、またはニックネームをレプリケーション・ソースとして定義すること。

レプリカ表 (replica table). SQL レプリケーション、特に Update-anywhere (任意の場所で更新) レプリケーシ

ョンでは、ローカルに更新でき、サブスクリプション・セット定義を通して、マスター表から更新情報も受け取ることができるタイプのターゲット表。レプリケーション競合検出が使用可能な場合、レプリカ表への変更はリジェクトされるが、マスター表への変更は保存される。

「Update-anywhere レプリケーション (Update-anywhere replication)」、「マスター表 (master table)」、および「競合検出 (conflict detection)」も参照。

レプリケーション (replication). 複数の場所にある定義済みデータ集合を保守するための処理。これには、特定の変更内容がある場所 (ソース) から別の場所 (ターゲット) にコピーしたり、2 つの場所にあるデータを同期化することが関係する。

レプリケーション管理者 (replication administrator). (1) Q レプリケーションでは、Q サブスクリプションや XML 発行の作成に責任を負うユーザー。このユーザーは Q キャプチャー・プログラムおよび Q アプライ・プログラムも実行できる。(2) SQL レプリケーションでは、レプリケーション・ソースの登録およびサブスクリプション・セットの作成に責任を負うユーザー。このユーザーはキャプチャー・プログラムおよびアプライ・プログラムも実行できる。

レプリケーション・アナライザー (Replication Analyzer). レプリケーションでは、セットアップの問題、構成エラー、およびパフォーマンスの問題に応じたレプリケーション環境を分析できるプログラム。

レプリケーション・アラート・モニター (Replication Alert Monitor). レプリケーションでは、キャプチャー・プログラム、アプライ・プログラム、Q キャプチャー・プログラム、Q アプライ・プログラムの操作をチェックし、指定されたアラート条件を検出すると 1 人以上のユーザーにアラートを送信するプログラム。

レプリケーション・キュー・マップ (replication queue map). Q レプリケーションでは、送信キューと受信キューをリンクするオブジェクト。レプリケーション・キュー・マップには、送信キューを使用するすべてのトランザクションが Q キャプチャー・プログラムで処理される方法と、受信キューを使用するすべてのトランザクションが Q アプライ・プログラムで処理される方法に関する設定に関する設定が含まれる。「発行キュー・マップ (publishing queue map)」および「キュー・マップ (queue map)」も参照。

レプリケーション・コントロール表 (replication control table). レプリケーションでは、レプリケーション定義または制御情報が保管されている表。

レプリケーション・センター (Replication Center).
レプリケーションでは、レプリケーション環境の定義、操作、保守、およびモニターに使用するグラフィカル・ユーザー・インターフェース。一式の DB2 Administration Client ツールの一部である。

レプリケーション・ソース (replication source). (1) SQL レプリケーションでは、レプリケーションのソースとして登録されている表、ビュー、またはニックネーム。この表に行った変更はサブスクリプション・セット・メンバーで定義されたターゲット表にキャプチャーされ、コピーされる。「サブスクリプション・セット (subscription set)」および「サブスクリプション・セット・メンバー (subscription-set member)」も参照。(2) Q レプリケーションでは、レプリケーションのソースになる表。このタイプの表に行った変更は Q サブスクリプションまたは XML 発行で定義されたターゲット表にキャプチャーされ、コピーされる。「Q サブスクリプション (Q subscription)」および「XML 発行 (XML publication)」も参照。

レプリケーション・ターゲット (replication target).
(1) SQL レプリケーションでは、登録されているレプリケーション・ソースから複製された変更内容の宛先になっている表、ビュー、またはニックネーム。アプライ・プログラムはこれらの変更内容をアプライする。「ターゲット表 (target table)」も参照。(2) Q レプリケーションでは、ソースから複製された変更内容の宛先になっている表またはストアド・プロシージャ。Q アプライ・プログラムはこれらの変更内容をアプライする。「ターゲット表 (target table)」も参照。

ローカル・データベース (local database). 使用中のサーバーに物理的に存在するデータベース。「リモート・データベース (remote database)」と対比。

ロード・フェーズ (load phase). Q レプリケーションでは、ターゲット表がソース表のデータでロードされ、2 つの表が同期化される段階。Q アプライ・プログラムでは自動ロードを使用してロード・プロセスが処理されるので、ロード・ユーティリティを自分で指定することもできるが、Q アプライ・プログラムに使用可能な最善のユーティリティを選択させてもよい。手動ロードの場合には、ターゲット表をロードしてから、その表がロードされた時点でレプリケーション・プログラムに通知する。

ログ (log). (1) システムで行われた変更を記録するファイル。(2) DB2 Universal Database for z/OS 実行中に発生するイベントおよびその順序を記述する、レコードの集合。記録された情報は、DB2 Universal Database for z/OS 実行中に障害が起きた場合のリカバリーに使用さ

れる。(3) 「データベース・リカバリー・ログ (database recovery log)」を参照。

ロック (locking). データの整合性を確保するためにデータベース・マネージャーが使用する機構。ロックにより、複数の並行ユーザーが非整合データにアクセスできないようにする。

ロック (lock). (1) イベントまたはデータへのアクセスのシリアライズのための手段。(2) あるアプリケーション・プロセスが加えた非コミットの変更が、別のアプリケーション・プロセスで認識されないようにしたり、アプリケーション・プロセスが、別のプロセスからアクセスされているデータを変更しないようにしたりするための手段。

論理サーバー (logical server). レプリケーションでは、Linux、UNIX、および Windows 上では DB2 データベース。z/OS では DB2 を実行するサブシステム。

[数字]

2 バイト文字ラージ・オブジェクト (DBCLOB) (double-byte character large object (DBCLOB)).
サイズが 0 バイトから 2 ギガバイトまでの範囲の 2 バイト・シーケンスを含むデータ・タイプ。2 バイト・テキストのラージ・オブジェクトの保管に使われるデータ・タイプ。「2 バイト文字ラージ・オブジェクト・ストリング (double-byte character large object string)」ともいう。このようなストリングには常に関連したコード・ページがある。「バイナリー・ラージ・オブジェクト (BLOB) (binary large object (BLOB))」および「文字ラージ・オブジェクト (CLOB) (character large object (CLOB))」も参照。

A

ASP. 「補助記憶域プール (ASP) (auxiliary storage pool (ASP))」を参照。

B

BLOB. 「バイナリー・ラージ・オブジェクト (BLOB) (binary large object (BLOB))」を参照。

C

CCD 表 (CCD table). 「整合変更データ表 (consistent-change data (CCD) table)」を参照。

CD 表 (CD table). 「変更データ表 (change data table)」を参照。

CLOB. 「文字ラージ・オブジェクト (CLOB) (*character large object (CLOB)*)」を参照。

D

DB2 以外のリレーショナル・データベース・サーバー (non-DB2 relational database server). IBM 以外のベンダーからの Informix データベース・サーバーまたはリレーショナル・データベース・サーバー。

DB2 レプリケーション (DB2 replication). 「SQL レプリケーション (*SQL replication*)」を参照。

DBCLOB. 「2 バイト文字ラージ・オブジェクト (*DBCLOB (double-byte character large object (DBCLOB))*)」を参照。

DBMS. データベース管理システム (*Database management system*)。

I

IASP. 「独立補助記憶域プール (*IASP (Independent Auxiliary Storage Pool (IASP))*)」を参照。

L

LOB. 「ラージ・オブジェクト (*LOB (large object (LOB))*)」を参照。

M

multi-tier レプリケーション (multi-tier replication). SQL レプリケーションでは、変更があるデータベースのレプリケーション・ソースから、別のデータベースのレプリケーション・ターゲットに複製され、別のデータベースのレプリケーション・ターゲットへ再度複製されるレプリケーション構成。

N

NULL 値 (null value). 値が指定されていないパラメーターの位置。

NULL 可能 (nullable). 列、関数パラメーター、または結果に値がなくてもよい状態。

O

ODBC. 「*Open Database Connectivity (ODBC)*」を参照。

ODBC ドライバー (ODBC driver). ODBC 関数呼び出しを実行し、データ・ソースと対話するドライバー。

Open Database Connectivity (ODBC). 呼び出し可能 SQL (SQL プリプロセッサを必要としない) を使用して、データベース管理システムにアクセスできるようにするアプリケーション・プログラム・インターフェース (API)。ODBC アーキテクチャーを使用すると、データベース・ドライバー (*database driver*) と呼ばれるモジュールを追加することができる。これらのモジュールはアプリケーションを、選択したデータベース管理システムにランタイムにリンクする。アプリケーション・プログラムがサポートされるすべてのデータベース管理システムのモジュールに直接的にリンクされる必要はない。

Q

Q アプライ待ち時間 (Q Apply latency). Q レプリケーションでは、Q アプライ・プログラムが受信キューからトランザクションを受け取った後に、トランザクションがターゲット表にアプライされる時間。

Q アプライ・サーバー (Q Apply server). Q レプリケーションでは、Q アプライ・プログラムのコントロール表があり、Q アプライ・プログラムを実行しているデータベースまたはサブシステム。ターゲット表やその他のレプリケーション定義に関する情報を保管するコントロール表の 1 つ以上の集合が含まれる。

Q アプライ・スキーマ (Q Apply schema). Q レプリケーションでは、Q アプライ・プログラムとそのコントロール表の ID。

Q アプライ・プログラム (Q Apply program). Q レプリケーションでは、受信キューからトランザクションを読み取り、これらの変更内容を 1 つ以上のターゲット表にアプライするかストアド・プロシージャーに渡すプログラム。

Q キャプチャー待ち時間 (Q Capture latency). Q レプリケーションでは、現行の Q キャプチャー・プログラムが DB2 リカバリー・ログの読み取りに要するおおよその時間。Q キャプチャー待ち時間は、Q キャプチャー・プログラムがパフォーマンス・データを保管した時刻と、このプログラムがデータの保管時にログ中で読み取った、最後にコミットされたトランザクションのタイム・スタンプとの間の時間を測定する。例えば、Q キャプチャー・プログラムが午前 10 時にパフォーマンス・データを保管し、最後にコミットされたトランザクションのタイム・スタンプが午前 9 時 59 分だった場合は、Q キャプチャー待ち時間は 1 分になる。

Q キャプチャー・サーバー (Q Capture server). Q レプリケーションおよびイベント発行では、Q キャプチャー・プログラムのコントロール表があり、Q キャプチャー・プログラムを実行しているデータベースまたはサブシステム。Q サブスクリプション、XML 発行、およびその他のレプリケーションや発行の定義に関する情報を保管するコントロール表の 1 つ以上の集合が含まれる。

Q キャプチャー・スキーマ (Q Capture schema). Q レプリケーションでは、Q キャプチャー・プログラムとそのコントロール表の ID。

Q キャプチャー・トランザクション待ち時間 (Q Capture transaction latency). Q レプリケーションでは、Q キャプチャー・プログラムが DB2 リカバリー・ログ中でトランザクションのコミット・ステートメントを読み取った時点から、Q キャプチャー・プログラムがそのトランザクションを含むメッセージを送信キュー上に挿入するまでの時間。

Q キャプチャー・プログラム (Q Capture program). Q レプリケーションおよびイベント発行では、DB2 リカバリー・ログを読み取り、DB2 ソース表に対する変更内容をキャプチャーし、1 つ以上の送信キューを使用してその変更内容を伝送するプログラム。

Q サブスクリプション (Q subscription). Q レプリケーションでは、ソース表とターゲット表またはストアード・プロシージャとの間のマッピングを識別し、どの変更内容を複製するか指定するオブジェクト。Q キャプチャー・プログラムは、ソース表から変更内容を複製し、それらの変更内容を圧縮形式で送信キューに挿入する。次に、Q アプライ・プログラムが受信キューから圧縮メッセージを受け取り、変更内容をターゲット表にアプライするか、データ操作のためにストアード・プロシージャに渡す。Q サブスクリプションは、XML 発行で発行されるデータを複製しない点で、XML 発行とは異なるオブジェクトである。

Q サブスクリプション・グループ (Q subscription group). Q レプリケーションでは、同一の論理表のレプリケーションに関する Q サブスクリプションのグループ。

Q レプリケーション (Q replication). WebSphere MQ メッセージ・キューを使用して、大ボリュームで待ち時間の短いレプリケーションを行うレプリケーションの解決策。競合検出、競合解決、および収束を伴う対等な解決策を提供する。

R

RDBMS. 「リレーショナル・データベース管理システム (relational database management system)」を参照。

S

SQL レプリケーション (SQL replication). ステージング表を使用するレプリケーション表のタイプ。

U

UDT. 「ユーザー定義タイプ (user-defined type)」を参照。

Update-anywhere レプリケーション

(Update-anywhere replication). SQL レプリケーションでは、すべての表が、登録済みソースおよび読み取り-書き込みターゲットの両方であるレプリケーション構成。1 つの表は他の表のすべてをフル・リフレッシュするための 1 次ソース表である。この構成では、ソースおよびターゲット表の間には、暗黙的なレプリケーション階層はない。「ピアツーピア・レプリケーション (peer-to-peer replication)」と対比。「multi-tier レプリケーション (multi-tier replication)」、「マスター表 (master table)」、および「レプリカ表 (replica table)」も参照。

X

XML 発行 (XML publication). イベント発行では、どの変更内容をソース表からユーザー・アプリケーションに発行するのか識別するオブジェクト。Q キャプチャー・プログラムは、ソース表から変更内容を発行し、それらの変更内容を XML 形式で送信キューに挿入する。これらの XML メッセージを検索して使用するには、Q アプライ・プログラム以外のアプリケーションを用意する。Q サブスクリプションは XML 発行で発行されるデータを複製しないので、Q サブスクリプションと XML 発行は異なるオブジェクトである。

ユーザー補助

動作が制限されていたり視力が限られているなどの身体障害を持つユーザーは、ユーザー補助を使用して、ソフトウェア・プロダクトを首尾よく使用することができます。DB2 Information Integrator バージョン 8 の主要なユーザー補助機能は以下のとおりです。

- マウスの代わりにキーボードを使用してすべての機能を操作することができます。
- フォントのサイズと色をカスタマイズすることができます。
- アラート・キューは表示または音声のどちらでも受け取ることができます。
- DB2 は Java™ Accessibility API を使用するユーザー補助アプリケーションをサポートしています。
- DB2 資料は利用しやすいフォーマットで提供されています。

キーボード入力およびナビゲーション

コントロール・センター、データウェアハウス・センター、およびレプリケーション・センターなどの DB2 データベース・ツールは、キーボードだけを使用して操作することができます。たいいていの操作は、マウスの代わりにキーまたはキーの組み合わせを使用して実行することができます。

UNIX ベースのシステムでは、キーボード・フォーカスの位置が強調表示されます。この強調表示は、ウィンドウのアクティブな領域と、キー・ストロークが有効になる場所を示します。

利用可能なモニター

DB2 データベース・ツールは、ユーザー・インターフェースを拡張し、弱視のユーザーのためのユーザー補助を向上させています。これらのユーザー補助の強化には、カスタマイズ可能なフォントのプロパティのサポートも含まれています。

フォントの設定

DB2 データベース・ツールの場合、「ツール設定」ノートブックを使用して、メニューとウィンドウ内のテキストの色、サイズ、およびフォントを選択することができます。

色の非依存

この製品のすべての機能は、色を識別する必要なく使用することができます。

代替アラート・キュー

「ツール設定」ノートブックを使用して、オーディオまたはビジュアル・キューのどちらでアラートを受け取るかを指定することができます。

支援テクノロジーとの互換性

DB2 Information Integrator のグラフィカル・インターフェースは、Java Accessibility API をサポートしており、身体障害を持つユーザーがスクリーン・リーダーや他の支援テクノロジーを使用することができます。

利用しやすい資料

DB2 ファミリー製品の資料は HTML フォーマットで入手可能です。資料は、ご使用のブラウザの画面設定に応じて表示することができます。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ができます。

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
AIX
DB2
DB2 Extenders
DB2 Universal Database
iSeries
Java
MVS
OS/390
RACF
Solaris
Sun
UNIX
WebSphere
Windows
z/OS
zSeries

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]

アーカイブ・ログ, オンにする 78
アーカイブ・ログをオンにする 78
アクティブ化
 Q サブスクリプションまたは XML 発行 253
値に基づく競合 288
圧縮メッセージ, レプリケーション・キュー・マップ経由の 95
圧縮メッセージ, Q レプリケーション・レプリケーション・キュー・マップ経由の 6
アプライ・プログラム
 アラート条件 294
アラート条件
 アプライ・プログラムの 294
 概要 294
 キャプチャー・プログラムの 294
 選択 305
 通知基準 313
 ASNMAIL exit ルーチン 299
 E メール通知 298
 list 294
 Q アプライ・プログラムの 294
 Q キャプチャー・プログラムの 294
イベント発行
 イベント発行のキューの指定 180
 エラー, リカバリー 325
 オブジェクト 9
 オブジェクト名
 大文字小文字 356
 命名規則 353
 環境変数, 設定 77
 キュー・マネージャーのクラスター 73
 許可要件
 概要 39
 管理 42
 レプリケーション・アラート・モニター 42
 Q キャプチャー 39
計画
 ストレージ要件 32
 メモリー, 概要 25

イベント発行 (続き)
計画 (続き)
 メモリーおよびストレージ 25
コード・ページ 38
コマンド行プロセッサ 535
コントロール表
 概要 415
 再編成 324
 作成 87
 図示 415
 整理 323
 ドロップ 227
 保守 322
コンプレッション・ディクショナリー 321
サーバーの構成, 概要 75
作成
 単一の XML 発行 183
 発行キュー・マップ 180
 XML 発行 185
サンプル・プログラム
 概要 529
 Linux, UNIX, Windows 529
 z/OS 532
システム・サービス 335
紹介 9
ストレージ要件
 WebSphere MQ 69
制約事項, データ・タイプ 197
接続要件 75
セットアップ 179
説明 9
ソース列のサブセット 188
データベース, 構成
 Linux, UNIX の 76
 Windows の 76
 z/OS の 85
データ変換
 概要 37
 説明 38
パスワードの設定 39
パッケージ, バインド
 概要 81
 レプリケーション・アラート・モニター 81
 Q キャプチャー 81
パッケージおよびプランの再バインド 326
発行キュー・マップと XML 発行のグループ化 179

イベント発行 (続き)
プログラム
 紹介 15
 Q キャプチャー・プログラム 15
変更 219
変更前の値に含まれる 194
保守
 概要 315
 コールド・スタート, 回避 232
 ソース・オブジェクト 316
 ソース・システム 315
 ソース・ログ, 保存 316
 ログ・データ, 保存 318, 319
メッセージ内容 9
メッセージに含まれる列 193
メモリー, Q キャプチャー・プログラム 26
レポートの表示 279
Q キャプチャーのメッセージ送信時 188
Q キャプチャー・プログラム 15
RUNSTATS ユーティリティ 324
WebSphere MQ
 オブジェクト 47, 53
 セットアップ, 概要 47
XML 発行 12
 行のフィルタリング 189
 検索条件 189
 WHERE 文節 189
XML 発行に列を追加 221
XML 発行のキー列 192
XML 発行の作成 182
XML 発行のソース列 188
XML メッセージ
 エラー・レポート・メッセージ 513
 技術要件 492
 行操作メッセージ 506
 区切り文字, 文字データの 493
 構造, 概要 491
 構造, Q キャプチャーから 493
 構造, Q キャプチャーに 521
 サブスクリプション非アクティブ化メッセージ 511
 サブスクリプション・スキーマ・メッセージ (subSchema) 515
 サブスクリプション・メッセージのアクティブ化 525
 サブスクリプション・メッセージの非アクティブ化 526
 制御メッセージ 491

イベント発行 (続き)

XML メッセージ (続き)

送信キュー・メッセージの無効化
524

タイプの概要 491

タイプのリスト 491

通知メッセージ 491

データ・メッセージ 491

トランザクション・メッセージ
497

ハートビート・メッセージ 514

ラージ・オブジェクト (LOB) メッ
セージ 508

列の追加メッセージ 520

ロード完了受信メッセージ 512

ロード完了メッセージ 524

msg: ルート・エレメント 495,
522

Q キャプチャーからのメッセージ
のリスト 494

Q キャプチャーに 521

イベント発行コマンド

asnqanalyze 386

asnqcap 361

asnqccmd 368

asnqfmt 412

asnslst 394

asntdiff 408

asntrep 410

list 359

roadmap 359

ウォーム・スタート

アクティブなサブスクリプション

XML メッセージへの影響 253

ログ・データでの影響 316

CAPSTART シグナルへの影響 253

CAPSTOP シグナル 254

startmode パラメーター 234, 235

startmode パラメーター、3 つのタイプ
361

エラー

アラート条件でモニター 291

キュー・エラー

発行キュー・マップ 180

レプリケーション・キュー・マップ
95

ストアード・プロシージャの論理の
処理 126

操作の 313

単一方向レプリケーションの処理 114

ピアツーピア・レプリケーション 136

リカバリー 325

レプリケーション

アラート条件、

APPLY_ERRORS 294

エラー (続き)

レプリケーション (続き)

アラート条件、

CAPTURE_ERRORS 294

アラート条件、

QAPPLY_ERRORS 294

アラート条件、

QCAPTURE_ERRORS 294

レプリケーション・キュー・マップ
211

レプリケーション・サービス 345

monitor_errors パラメーター 308

Q レプリケーションの処理 118

SQL 273, 294

SQL 状態 209

XML メッセージ 513

エラー・メッセージ

レプリケーション・アラート・モニタ
ーから 283

Q アプライ・プログラムから 283

Q キャプチャー・プログラムから
283

エラー・レポート XML メッセージ 513

エンドツーエンド待ち時間 283, 285

[力行]

稼働パラメーター、ストアード・プロシー
ジャー 126

環境変数

設定

レプリケーション 77

管理キュー

設定要件 62

レプリケーション・キュー・マップで
6, 95

Q アプライの 48

Q キャプチャーの 48

キー列

イベント発行の 192

Q サブスクリプションの 112

Q レプリケーションの 112

XML 発行の 192

キャプチャー・プログラム

アラート条件 294

キュー

管理

Q アプライの 48

Q キャプチャーの 48

再始動 48

受信

説明 48

メッセージ処理の開始 273

メッセージ処理の停止 275

送信 48

発行キュー・マップで 13, 180

キュー (続き)

予備 48

レプリケーション・キュー・マップで
6, 95

キュー項目数、ラージ・オブジェクト
(LOB) 値の 72

キューの再始動

説明 48

Q キャプチャーの使用法 15

キュー待ち時間 283, 285

キュー・マップ

発行

削除 226

作成 13

説明 13

属性の変更 223

変更 223

レプリケーション

削除 215

作成 95

説明 6

変更 211

キュー・マネージャー

クラスター 73

設定要件 62

説明 48

旧データ値

XML 発行に含まれる 194

行

フィルタリング

イベント発行 189

Q レプリケーション 108

行 (水平方向) のサブセット化 108

イベント発行 189

競合

値に基づくおよびバージョンに基づく
288

IBMQREP_EXCEPTIONS 表の 288

競合検出

双方向レプリケーション 133

ピアツーピア・レプリケーション 136

SQL 状態

変更 209

競合検出、双方向レプリケーション 147

行操作 XML メッセージ 506

許可

レプリケーションの、概要 39

レプリケーションの、管理 42

レプリケーションの、

Linux, UNIX, Windows 40

レプリケーションの、z/OS 40

レプリケーション・アラート・モニタ
ーの 42

Q アプライ・プログラムの 40

Q キャプチャー・プログラムの 39

許可 (続き)

WebSphere MQ

レプリケーション・アラート・モニター 70

Q アプライ・プログラム 70

Q キャプチャー・プログラム 70

空間データ・タイプ

制約事項 197

グループ化、レプリケーション・オブジェクト

発行キュー・マップと XML 発行

179

レプリケーション・キュー・マップおよび Q サブスクリプション 93

計画

ストレージ要件

イベント発行、概要 25, 32

Q レプリケーション、概要 25, 32

WebSphere MQ 69

メモリー

イベント発行、概要 25

Q レプリケーション、概要 25

検査、プログラムの状況 279

コード・ページ

イベント発行の

概要 37

説明 38

イベント発行の XML メッセージ 38

Q アプライ・プログラム 37

Q キャプチャー・プログラム

イベント発行の 38

Q レプリケーションの 37

Q レプリケーションの

概要 37

説明 37

コールド・スタート

回避 232

再始動メッセージ 15

Q サブスクリプションへの影響 253,

254

startmode パラメーター 235

XML 発行への影響 253, 254

更新

発行キュー・マップ 223

レプリケーション・キュー・マップ

211

Q サブスクリプション 205

XML 発行 219

コマンド

レプリケーション・コマンドを参照

359

コマンド行プロセッサ (CLP)

ASNCLP プログラムも参照 535

固有のデータ・タイプ 197

コントロール表

イベント発行

概要 415

図示 415

再編成 324

作成

レプリケーション・アラート・モニター

301

Q キャプチャーおよびアプライ

87

図示 415

整理 323

ドロップ 216, 227

分析 386

保守 322

モニター・コントロール・サーバー

リスト 424, 471

IBMSNAP_ALERTS 472

IBMSNAP_CONDITIONS 474

IBMSNAP_CONTACTGRP 481

IBMSNAP_CONTACTS 482

IBMSNAP_GROUPS 483

IBMSNAP_MONENQ 483

IBMSNAP_MONPARMS 483

IBMSNAP_MONSERVERS 485

IBMSNAP_MONTRAIL 488

Q アプライ・サーバー

リスト 422, 448

IBMQREP_APPENQ 448

IBMQREP_APPLYMON 449

IBMQREP_APPLYPARMS 452

IBMQREP_APPLYTRACE 454

IBMQREP_DELTOMB 455

IBMQREP_DONEMSG 456

IBMQREP_EXCEPTIONS 457

IBMQREP_RECVQUEUES 459

IBMQREP_SAVERI 461

IBMQREP_SPILLEDROW 462

IBMQREP_SPILLQS 463

IBMQREP_TARGETS 465

IBMQREP_TRG_COLS 464

Q キャプチャー・サーバー

リスト 421, 425

IBMQREP_ADMINMSG 425

IBMQREP_CAPENQ 426

IBMQREP_CAPMON 427

IBMQREP_CAPPARMS 428

IBMQREP_CAPQMON 432

IBMQREP_CAPTRACE 433

IBMQREP_SENDQUEUES 434

IBMQREP_SIGNAL 437

IBMQREP_SRCH_COND 434

IBMQREP_SRC_COLS 441

IBMQREP_SUBS 442

Q レプリケーション

概要 415

コントロール表 (続き)

Q レプリケーション (続き)

図示 415

RUNSTATS ユーティリティ 324

コンプレッション・ディクショナリー

321

[サ行]

再始動メッセージ、Q キャプチャー・プ

ログラム 15

再初期化

キュー 199, 211

発行キュー・マップ 223

レプリケーション・アラート・モニター

199, 307

Q アプライ・プログラム 199

Q キャプチャー・プログラム 205,

219

Q サブスクリプション 205

XML 発行 219

削除

発行キュー・マップ 226

レプリケーション・キュー・マップ

215

Q サブスクリプション 213

XML 発行 225

作成

アラート条件 199

コントロール表

レプリケーション・アラート・モニター

301

レプリケーション・センターから

199

Q キャプチャーおよび Q アプライ

87

発行キュー・マップ 180

モニター 304

レプリケーションの WebSphere MQ

オブジェクト 47

レプリケーション・キュー・マップ

95

レプリケーション・サービス 344

Q サブスクリプション

双方向レプリケーション 144

単一方向レプリケーション、概要

98

単一方向レプリケーション、単一

100

単一方向レプリケーション、複数

103

ピアツーピア・レプリケーション、

概要 153

ピアツーピア・レプリケーション、

2 つのサーバー 153

作成 (続き)

- Q サブスクリプション (続き)
 - ピアツーピア・レプリケーション、
3 つ以上のサーバー 155

XML 発行

- 概要 182
- 単一の 183
- 複数の 185
- レプリケーション・センターから
199

サブスクリプション XML メッセージの
アクティブ化 525

サブスクリプション XML メッセージの
非アクティブ化 526

サブスクリプション非アクティブ化 XML
メッセージ 511

サブスクリプション・スキーマ XML メ
ッセージ 515

サブセット化

- ソース行
 - Q サブスクリプション 108
 - XML 発行 189
- ソース列
 - Q サブスクリプション 105
 - XML 発行 188

参照制約

ピアツーピア・レプリケーション 136

参照保全

ピアツーピア・レプリケーション 136

サンプル・プログラム

- イベント発行
 - 概要 529
 - Linux、UNIX、Windows 529
 - z/OS 531、532

Q レプリケーション

- 概要 529
- JCL 532
- Linux、UNIX、Windows 529、530
- z/OS 531、532

システム開始タスク

レプリケーション・プログラムの実行
339

システム・コマンド

- レプリケーション・コマンドを参照
359
- asnlqcmd 376
- asnlqanalyze 386
- asnlqapp 372
- asnlqcap 361
- asnlqccmd 368
- asnlqmfmt 412
- asnlqslst 394
- asnlqdiff 408
- asnlqtrp 410

始動

イベント発行 253

始動 (続き)

- 双方向レプリケーション 158
- 単一方向レプリケーション 253
- ピアツーピア・レプリケーション
2 つのサーバー 158
- 3 つ以上のサーバー 160
- レプリケーション・アラート・モニタ
ー
 - さまざまな方式 306

Q アプライ・プログラム 259、372

Q キャプチャー・プログラム 229

自動リスタート・マネージャー (ARM)

- 説明 340
- レプリケーション・プログラムの再始
動 341

自動ロード

- 概要 164
- 使用するユーティリティ 165
- 定義 163
- ニックネームを指定 167
- z/OS の場合の考慮事項 166

受信キュー

- キューの指定、Q レプリケーションの
95
- 設定要件 62
- 説明 48
- 命名規則 353
- メッセージ処理の開始 273
- メッセージ処理の停止 275
- レプリケーション・キュー・マップ 6
- レプリケーション・キュー・マップで
95
- Q アプライ・プログラムの処理 93
- Q レプリケーションのメモリー 29

手動ロード

- 説明 169
- 定義 163
- 使用可能化
 - 変更キャプチャー 78

診断ファイル

- ストレージ要件 34
- 垂直方向 (列) のサブセット化
 - イベント発行 188
 - Q サブスクリプション 105
 - Q レプリケーション 105
 - XML 発行 188

水平方向 (行) のサブセット化

- Q サブスクリプション 108
- Q レプリケーション 108

スキーマ

- 命名規則
 - 双方向レプリケーションの 133
 - ピアツーピア・レプリケーションの
136
- Q アプライ・プログラムの
紹介 15

スキーマ (続き)

Q アプライ・プログラムの (続き)

- 説明 21
- 双方向レプリケーションの規則
133
- ドロップ 216
- ピアツーピア・レプリケーションの
規則 136

Q キャプチャー・プログラムの

- 紹介 15
- 説明 21
- 双方向レプリケーションの規則
133
- ドロップ 216、227
- ピアツーピア・レプリケーションの
規則 136

ストアド・プロシージャ

- パラメーター
 - ソース列にマッピング 129
 - 操作タイプの識別 126
 - トランザクションの識別 129
 - 抑止列の識別 128

Q サブスクリプション

- エラー・オプション 118
- 概要 121
- 説明 122
- パラメーター、キー列 112
- 予期しない条件、ターゲットで
114

Q レプリケーション

- エラー・オプション 118
- 概要 121
- 説明 122
- パラメーター、キー列 112
- 予期しない条件、ターゲットで
114

Q レプリケーションの書き込み 124

ストレージ

- 計画
 - イベント発行 25
 - Q レプリケーション 25
- DB2 UDB ログ 33
- WebSphere MQ 69
- ストレージ要件
 - 計画
 - イベント発行、概要 32
 - Q レプリケーション、概要 32
 - 診断ファイル用 34
 - トレース・ファイル 35
 - Q キャプチャー・プログラム 34

スループット

- Q アプライ・プログラム 283
- Q キャプチャー・プログラム 283

スレッド

状況の検査 279

スレッド (続き)

- Q レプリケーション
 - 状況の検査 281
 - admin 281
 - agent 19, 281
 - browser 19, 281
 - holdl 281
 - housekeeping 281
 - prune 281
 - serialization 281
 - worker 281

制御メッセージ 491

制約事項

- 空間データ・タイプ 197
- 固有のデータ・タイプ 197
- データ・タイプ 197
- ユーザー定義のデータ・タイプ 197
- 要約データ・タイプ 197
- DATALINK 値 197
- DB2 エクステンダー・ラージ・オブジェクト 198
- EDITPROC 文節 197
- FIELDPROC 文節 197
- LONG VARCHAR データ・タイプ 197
- VALIDPROC 文節 197

整理

- Q アプライ・プログラム
 - asqacmd コマンド 376
 - IBMQREP_APPLTRACE コントロール表 323
 - IBMQREP_APPLYMON コントロール表 323
 - monitor_limit パラメーター 323
 - trace_limit パラメーター 323
- Q キャプチャー
 - IBMQREP_CAPQMON コントロール表 323
- Q キャプチャー・プログラム
 - IBMQREP_CAPMON コントロール表 323
 - IBMQREP_CAPTRACE コントロール表 323
 - IBMQREP_SIGNAL コントロール表 323
 - monitor_limit パラメーター 323
 - signal_limit パラメーター 323
 - trace_limit パラメーター 323

整理インターバル

- レプリケーション・アラート・モニター 312

設定

- パスワード 39
- ユーザー ID 39

セットアップ

- 単一方向レプリケーション 91

セットアップ (続き)

- 複数方向レプリケーション 133
- レプリケーション・アラート・モニター 300
- Q レプリケーションの WebSphere MQ 47
- ソース - ターゲットのペア
 - 双方向レプリケーションの
 - 作成 144
 - 説明 133
 - 単一方向レプリケーションの 91
 - 作成 103
 - 1 つ作成 100
 - ピアツープア・レプリケーションの
 - 説明 136
 - 2 つのサーバーの作成 153
 - 列のマッピング、Q レプリケーション 111
 - 3 つ以上のサーバーでのピアツープア・レプリケーション 155
- ソース表
 - ストアード・プロシージャーにマッピング 129
 - 双方向レプリケーションの 133, 144
 - ターゲットへのマッピング、Q レプリケーション 111
 - 単一方向レプリケーションの
 - 指定 103
 - 説明 91
 - 1 つ指定 100
 - ピアツープア・レプリケーションの 136
 - ピアツープア・レプリケーションの、3 つ以上のサーバー 155
 - 保守 316
 - 命名規則 353
 - 列の追加 207, 221
 - 2 つのサーバーでピアツープア・レプリケーションの 153
- Q サブスクリプションの 5
 - 行のフィルタリング 108
 - 検索条件 108
 - ソース列とターゲット列のマッピング 111
 - Q キャプチャー・メッセージ 106
- XML 発行で 12
 - 行のフィルタリング 189
 - 検索条件 189
- Q キャプチャーのメッセージ送信時 188
- WHERE 文節 189
- XML 発行の
 - 単一の表 183
 - 複数の表 185

ソース列

- サブセットの発行
 - イベント発行 188
 - XML 発行 188
- サブセットのレプリケーション 105
- ストアード・プロシージャーにマッピング 129
- ターゲット列へのマッピング
 - Q レプリケーション 111
- 抑止列を識別するパラメーター 128
- XML 発行で発行 193
- ソース・サーバー
 - コード・ページ
 - イベント発行の 38
 - Q レプリケーションの 37
 - 指定
 - イベント発行の 185
 - 双方向レプリケーションの 144
 - 単一方向レプリケーションの 100
 - 単一方向レプリケーションの、複数 103
 - ピアツープア・レプリケーションの、2 つのサーバー 153
 - ピアツープア・レプリケーションの、3 つ以上のサーバー 155
 - XML 発行の 183
 - Q キャプチャーのメッセージ送信時
 - Q サブスクリプションの 106
 - XML 発行の 188
- ソース・システム
 - 定義 315
 - 保守 315
- ソース・データベース
 - 構成 78
- ソース・ログ、保存 316
- 相違検出表 329
- 操作
 - レプリケーション・アラート・モニター 303
 - Q アプライ・プログラム 259, 376
 - Q キャプチャー・プログラム 229, 368
 - 操作、データの
 - ストアード・プロシージャー、Q サブスクリプションの
 - 概要 121
 - 書き込み 124
 - 説明 122
 - Q レプリケーション
 - ストアード・プロシージャー、概要 121
 - ストアード・プロシージャー、書き込み 124
 - ストアード・プロシージャー、説明 122

送信キュー

- イベント発行 180
- キューの指定、Q レプリケーションの 95
- 設定要件 62
- 説明 48
- 発行キュー・マップ 13, 180
- 命名規則 353
- レプリケーション・キュー・マップ 95
- レプリケーション・キュー・マップで 6
- Q キャプチャー・プログラムの処理
イベント発行で 179
- Q レプリケーションの 93

送信キュー XML メッセージの無効化 524

双方向レプリケーション

- エラー・オプション 118
 - キューの指定 95
 - 競合検出 133, 147
 - 始動 158
 - 説明 3, 133
 - 停止 159
 - レプリケーション・オブジェクト 133
 - レプリケーション・キュー・マップの
作成 95
 - ロード・オプション 173
 - Q サブスクリプションの作成 144
 - WebSphere MQ オブジェクト 55
- ## ソフトウェア要件
- レプリケーション・センター 86

[タ行]

ターゲット表

- キー列
 - Q サブスクリプション 112
 - XML 発行 192
- 許可要件 40
- 索引列、Q サブスクリプション 112
- 修復 332
- 接続要件 75
- ソースへのマッピング
 - Q レプリケーション 111
- 双方向レプリケーションの 133, 144
- ターゲット・オブジェクト・プロファイル 99
- ターゲット・キー
 - Q サブスクリプション 112
 - XML 発行 192
- 単方向レプリケーションの
指定、複数 103
- 説明 91
- 予期しない条件 114
- 1 つ指定 100

ターゲット表 (続き)

- トランザクションの適用 19
 - ニックネーム 167
 - ピアツーピア・レプリケーションの 136
 - ピアツーピア・レプリケーションの、3
つ以上のサーバー 155
 - 変更のストレージ・スペース 33
 - 保守 325
 - 命名規則 353
 - 列の追加 207
 - ロード・オプション
 - 概要 163
 - 自動 165
 - 自動ロード 163, 167
 - 手動 169
 - 手動ロード 163
 - 推奨事項 164
 - 双方向レプリケーション 173
 - 単方向レプリケーション 173
 - ピアツーピア・レプリケーション 173
 - ロードなし 163, 172
 - 2 つのサーバーでピアツーピア・レ
プリケーションの 153
 - IBMQREP_TARGETS コントロール表 422, 465
 - Q サブスクリプションの
説明 5
 - 予期しない条件 114
- ## ターゲット・オブジェクト・プロファイル 99
- ターゲット・キー
 - Q サブスクリプション 112
 - Q レプリケーション 112
 - XML 発行 192
 - ターゲット・サーバー
指定
 - 双方向レプリケーションの 144
 - 単方向レプリケーションの 100, 103
 - ピアツーピア・レプリケーション
の、3 つ以上のサーバー 155
 - 2 つのサーバーでピアツーピア・レ
プリケーションの 153
 - ターゲット・オブジェクト・プロファ
イル 99
 - ターゲット・データベース
 - Q アプライの構成 80
 - 単方向レプリケーション
 - エラー・オプション 118
 - キューの指定 95
 - ストアード・プロシージャー
概要 121
 - 書き込み 124
 - 説明 122

単方向レプリケーション (続き)

- セットアップ
 - 説明 91
 - レプリケーション・キュー・マップ
および Q サブスクリプションの
グループ化 93
 - 説明 3, 91
 - レプリケーション・キュー・マップの
作成 95
 - ロード・オプション 173
 - Q サブスクリプション
行のフィルタリング 108
 - 検索条件 108
 - 作成、概要 98
 - 作成、複数 103
 - ソース列 105
 - ソース列とターゲット列のマッピ
ング 111
 - ターゲットのキー列 112
 - ターゲットの索引列 112
 - 予期しない条件、ターゲットで 114
 - 1 つ作成 100
 - Q キャプチャー・メッセージ 106
 - WHERE 文節 108
 - WebSphere MQ オブジェクトに必要な
同一システム 50
 - リモート 52
- ## チャンネル・オブジェクト、設定要件 62
- ## 通知メッセージ 491
- ## データのトランスフォーム、Q レプリケ ーション
- ストアード・プロシージャー、概要 121
 - ストアード・プロシージャー、説明 122
 - ストアード・プロシージャーの書き込
み 124
- ## データのフィルタリング、検索条件を使用 した 15
- ## データベース
- Linux、UNIX および Windows の構成 76
 - Q アプライの構成
ターゲット 80
 - Q キャプチャーの構成
ソース 78
 - z/OS 用の構成 85
- ## データ変換
- イベント発行、説明 38
 - 概要 37
 - Q レプリケーション、説明 37
- ## データ例外、概要 288
- ## データ・タイプ
- 考慮事項
 - 発行時の 197

データ・タイプ (続き)
考慮事項 (続き)
レプリケーション時 197
制約事項 197
BLOB、レプリケーション 198
CLOB、レプリケーション 198
DBCLOB、レプリケーション 198
LOB、レプリケーション 198
データ・トランスフォーメーション、Q
レプリケーション
ストアード・プロシージャ、概要
121
ストアード・プロシージャ、書き込
み 124
ストアード・プロシージャ、説明
122
データ・メッセージ 491
停止
イベント発行 254
双方向レプリケーション 159
単一方向レプリケーション 254
ピアツーピア・レプリケーション
2つのサーバー 159
3つ以上のサーバー 161
レプリケーション・アラート・モニタ
ー 199, 314
Q アプライ・プログラム 199, 276
Q キャプチャー・プログラム 199,
256
Q サブスクリプション 254
Q サブスクリプションまたは XML 発
行 253
XML 発行 254
デフォルト値
Q アプライ・パラメーター 262
Q キャプチャー・パラメーター 234
統計
Q アプライ・プログラム 283
Q キャプチャー・プログラム 283
トラブルシューティング
プログラムの状況の検査 281
トラブルシューティング・コマンド
asnqanalyze 386
トランザクション XML メッセージ 497
トランザクションのパラメーター、Q レ
プリケーションのストアード・プロシ
ージャ 129
トランザクション待ち時間、Q キャプチ
ャー 285
トランスフォーム、データの
ストアード・プロシージャ、Q サブ
スクリプションの
概要 121
トレース・ファイル、ストレージ要件 35
ドロップ
コントロール表 216, 227

ドロップ (続き)
スキーマ
Q キャプチャー 227
Q キャプチャーおよび Q アプライ
216

[ナ行]

ニックネーム
レプリケーション
自動ロードの指定 167

[ハ行]

バージョンに基づく競合 288
ハートビート XML メッセージ 514
ハートビート・インターバル
発行キュー・マップで 13, 180
レプリケーション・キュー・マップで
6, 95
バインド、パッケージ
レプリケーション・アラート・モニタ
ー (Linux, UNIX, Windows) 84
Q アプライ
(Linux, UNIX, Windows) 83
Q キャプチャー
(Linux, UNIX, Windows) 82
パスワード
セットアップ、概要 39
保管、リモート・サーバー用の 43
パッケージ
バインド
概要 81
レプリケーション・アラート・モニ
ター
(Linux, UNIX, Windows) 81, 84
Q アプライ・プログラム
(Linux, UNIX, Windows) 81, 83
Q キャプチャー・プログラム
(Linux, UNIX, Windows) 81, 82

発行
イベント発行
メッセージに含まれる列 193
Q キャプチャーのメッセージ送信
時 188
イベント発行のセットアップ 179
データのサブセット
XML 発行の行のフィルタリング
189
発行キュー・マップと XML 発行のグ
ループ化 179
変換済みデータ 38
Q キャプチャー・プログラムと 15
Q キャプチャーのメッセージ送信
時 188

発行 (続き)
Q キャプチャー・プログラムと (続き)
XML 発行に含まれる列 193
XML 発行
説明 12
変更前の値 194
メッセージに含まれる列 193
Q キャプチャーのメッセージ送信
時 188
XML 発行のソース列のサブセット
188
XML 発行の変更前の値 194
発行キュー・マップ
イベント発行で 9
削除 226
作成 180
説明 13
変更 223
XML 発行でのグループ化 179
バッチ・モード、レプリケーション・プロ
グラム 338
パフォーマンス、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
283
パラメーター
ストアード・プロシージャ、レプリ
ケーション
キー列へのマッピング、Q サブス
クリプション 112
ソース列にマッピング 129
操作タイプの識別 126
トランザクションの識別 129
抑止列の識別 128
Q レプリケーション、概要 121
Q レプリケーション、説明 122
Q レプリケーションの書き込み
124
レプリケーション・アラート・モニタ
ー
説明 308
デフォルト値 307
alert_prune_limit 308
autoprune 308
email_server 308
max_notifications_per_alert 308
max_notification_minutes 308
monitor_errors 308
monitor_limit 308
monitor_path 308
runonce 308
trace_limit 308
Q アプライ・プログラム
概要 262
実行中の変更 272, 376
デフォルト値 262
変更 19, 271

パラメーター (続き)

Q アプライ・プログラム (続き)

- 保存の変更 271
- apply_path 263
- apply_schema 263
- apply_server 263
- autostop 263
- deadlock_retries 263
- logreuse 263
- logstdout 263
- monitor_interval 263
- monitor_limit 263
- pwdfile 263
- qmgr 263
- term 263
- trace_limit 263

Q キャプチャー・プログラム

- 概要 233
- 実行中の変更 251, 368
- 説明 235
- デフォルト値 234
- 変更 248
- 変更の方式 249
- 保存の変更 250
- add_partition 235
- adminq 235
- autostop 235
- capture_path 235
- capture_schema 235
- capture_server 235
- commit_interval 235
- logreuse 235
- logstdout 235
- memory_limit 235
- monitor_interval 235
- monitor_limit 235
- prune_interval 235
- qmgr 235
- restartq 235
- signal_limit 235
- sleep_interval 235
- startmode 235
- term 235
- trace_limit 235

WebSphere MQ 62

パラメーター、呼び出し

- レプリケーション・アラート・モニター
- UNIX 版 380
- Windows 版 380
- z/OS 版 380

Q アプライ・プログラム 259, 372

Q キャプチャー・プログラム 229, 361

パラメーターの変更

Q アプライ・プログラム 271, 272

パラメーターの変更 (続き)

Q キャプチャー・プログラム 250, 251

非アクティブ化

Q サブスクリプション 254
XML 発行 254

ピアツーピア・レプリケーション

- エラー・オプション 118
- キューの指定 95
- 競合検出 136
- 参照保全 136

始動

- 2 つのサーバー 158
- 3 つ以上のサーバー 160

説明 3, 136

停止

- 2 つのサーバー 159
- 3 つ以上のサーバー 161

列のバージョン管理 489

レプリケーション・オブジェクト

- 2 つのサーバー 136
- 3 つ以上のサーバー 136

レプリケーション・キュー・マップの作成 95

2 つのサーバー

- オブジェクト 136
- 始動 158
- 停止 159
- ロード・オプション 173

2 リモート・サーバー

WebSphere MQ オブジェクト 55

3 つ以上のサーバー

- オブジェクト 136
- 始動 160
- 停止 161

ロード・オプション 173

3 つ以上のリモート・サーバー

WebSphere MQ オブジェクト 58

Q サブスクリプション

- 概要 153
- 2 つのサーバーの作成 153
- 3 つ以上のサーバーの作成 155

表

- IBMQREP_ADMINMSG 425
- IBMQREP_APPENQ 448
- IBMQREP_APPLYMON 449
- IBMQREP_APPLYPARMS 452
- IBMQREP_APPLYTRACE 454
- IBMQREP_CAPENQ 426
- IBMQREP_CAPMON 427
- IBMQREP_CAPPARMS 428
- IBMQREP_CAPQMON 432
- IBMQREP_CAPTRACE 433
- IBMQREP_DELTOMB 455
- IBMQREP_DONEMSG 456
- IBMQREP_EXCEPTIONS 457

表 (続き)

- IBMQREP_RECQUEUEUES 459
- IBMQREP_SAVERI 461
- IBMQREP_SENDQUEUEUES 434
- IBMQREP_SIGNAL 437
- IBMQREP_SPILLEDROW 462
- IBMQREP_SPILLQS 463
- IBMQREP_SRCH_COND 434
- IBMQREP_SRC_COLS 441
- IBMQREP_SUBS 442
- IBMQREP_TARGETS 465
- IBMQREP_TRG_COLS 464
- IBMSNAP_ALERTS 472
- IBMSNAP_CONDITIONS 474
- IBMSNAP_CONTACTGRP 481
- IBMSNAP_CONTACTS 482
- IBMSNAP_GROUPS 483
- IBMSNAP_MONENQ 483
- IBMSNAP_MONPARMS 483
- IBMSNAP_MONSERVICES 485
- IBMSNAP_MONTRAIL 488
- 表修復ユーティリティー 332, 410
- 表相違検出ユーティリティー 329, 408
- フィルタリング
- 行
- 検索条件、イベント発行 189
- 検索条件、Q レプリケーション 108
- 複数方向レプリケーション
- セットアップ 133
- 双方向およびピアツーピア・レプリケーション間の選択 141
- 双方向レプリケーション 133
- ピアツーピア・レプリケーション 136
- プロファイル
- ターゲット・オブジェクト・プロファイル、Q レプリケーションの 99
- 並列処理
- Q アプライ・プログラムの使用 19
- Q レプリケーションの使用 19
- 並列ロード、z/OS 上 166
- 変換
- イベント発行のデータ
- 概要 37
- 説明 38
- Q レプリケーションのデータ
- 概要 37
- 説明 37
- 変更
- イベント発行環境 219
- 発行キュー・マップ 223
- レプリケーション・キュー・マップ 211
- Q アプライ・パラメーター 271
- Q キャプチャー・パラメーター 248
- Q サブスクリプション属性 205

変更 (続き)
 Q レプリケーション環境 205
 SQL 状態 209
 XML 発行属性 219
 変更キャプチャー
 始動 253
 停止 254
 変更後の値
 キー列の 122
 検索条件の一致 108
 パラメーター、キー列の 129
 XML 発行に含まれる 194
 変更のキャプチャー
 始動 253
 停止 254
 変更前の値
 ストアード・プロシージャー 122
 双方向レプリケーション、競合検出
 147
 データのフィルタリング 15
 非キー列 129
 IBMQREP_SUBS 表 442
 LOB データ・タイプ 198
 LOB データ・タイプの制約事項 147
 LOB 列 144
 ROWID 列 198
 XML 発行に含まれる 194
 保守
 イベント発行環境
 概要 315
 コントロール表 322
 ソース・オブジェクト 316
 ターゲット表 325
 ログ・データ、保存 318, 319
 Q レプリケーション環境
 概要 315
 コールド・スタート、回避 232
 コントロール表 322
 ソース・オブジェクト 316
 ターゲット表 325
 ログ・データ、保存 318, 319

[マ行]

待ち時間
 概要 285
 Q アプライ・プログラム 283
 Q キャプチャー・プログラム 283
 マッピング
 ソース列からターゲット列への
 Q レプリケーション 111
 命名規則
 ターゲット・オブジェクト・プロファイ
 ル、Q レプリケーションの 99
 Q レプリケーションおよびイベント発
 行の 353

メッセージ
 フォーマットおよび表示 412
 レプリケーション・アラート・モニタ
 ーから 283
 Q アプライ・プログラムから 283
 Q キャプチャー・プログラムから
 履歴および効率データ 283
 Q サブスクリプションの 106
 XML 発行の Q キャプチャーのメ
 ッッセージ送信時 188
 WebSphere MQ
 サイズ制限 68
 メッセージ処理
 Q アプライの始動 273
 Q アプライの停止 275
 メッセージ・バッファ
 ラージ・オブジェクト (LOB) 値 72
 メモリー
 アラート条件
 APPLY_MEMORY 294
 CAPTURE_MEMORY 294
 QAPPLY_MEMORY 294
 QCAPTURE_MEMORY 294
 計画
 イベント発行、概要 25
 Q レプリケーション、概要 25
 レプリケーション・アラート・モニタ
 ー 32
 LOB データ・タイプ 32
 Q アプライ・プログラム 29
 Q キャプチャー・プログラム 26
 Q キャプチャー・プログラムによる超
 過 34
 モニター
 レプリケーション 291, 303
 モニター修飾子
 命名規則 353
 レプリケーション 291
 モニター・コントロール・サーバー
 コントロール表
 リスト 471
 コントロール表のリスト 424
 IBMSNAP_ALERTS コントロール表
 472
 IBMSNAP_CONDITIONS コントロー
 ル表 474
 IBMSNAP_CONTACTGRP コントロー
 ル表 481
 IBMSNAP_CONTACTS コントロー
 ル表 482, 483
 IBMSNAP_MONENQ コントロール表
 483
 IBMSNAP_MONPARMS コントロー
 ル表 483
 IBMSNAP_MONSERVERS コントロー
 ル表 485

モニター・コントロール・サーバー (続
 き)
 IBMSNAP_MONTRAIL コントロー
 ル表 488
 問題行 288

[ヤ行]

ユーザー ID
 保管、リモート・サーバー用の 43
 ユーザー定義タイプ (UDT)
 Q レプリケーション 197
 ユーザー・アプリケーション
 からの XML メッセージのリスト
 521
 ユーティリティ
 表修復 410
 表相違検出 408
 メッセージ・フォーマッター 412
 CURSOR からの LOAD 165, 167
 EXPORT および IMPORT 165
 EXPORT および LOAD 165
 Q レプリケーション・アナライザー
 386
 要約データ・タイプ 197
 予期しない条件、ターゲットで
 ストアード・プロシージャーで処理
 126
 単一方向レプリケーション 114
 Q サブスクリプション 114
 抑止パラメーター
 ストアード・プロシージャー、Q レ
 プリケーションの 128
 予備キュー
 ストレージ要件 70
 設定要件 62
 説明 48
 呼び出しパラメーター
 レプリケーション・アラート・モニタ
 ー
 UNIX 版 380
 Windows 版 380
 z/OS 版 380
 Q アプライ・プログラム 259
 Q キャプチャー・プログラム 229
 予備ファイル
 Q アプライ・プログラムのストレージ
 70

[ラ行]

ラージ・オブジェクト (LOB) XML メ
 ッッセージ 508

- ラージ・オブジェクト (LOB) データ・タイプ
 - 双方向レプリケーションの制約事項 147
- ランタイム・パラメーター
 - Q アプライの変更 272
 - Q キャプチャーの変更 251
- リモート・サーバー
 - パスワードおよびユーザー ID の保管 43
- 例外
 - レプリケーションの、概要 288
- 列
 - キー列
 - イベント発行 192
 - Q サブスクリプション 112
 - Q レプリケーション 112
 - XML 発行 192
 - 既存の XML 発行に追加 221
 - サブセット化
 - イベント発行 188
 - Q サブスクリプション 105
 - Q レプリケーション 105
 - XML 発行 188
 - 発行
 - イベント発行 188
 - XML 発行 188
 - マッピング
 - Q サブスクリプションの 111
 - Q レプリケーションの 111
 - レプリケーション
 - Q サブスクリプション 105
 - Q レプリケーション 105
 - Q サブスクリプションに追加 207
 - XML 発行で発行 193
- 列 (垂直方向) のサブセット化
 - イベント発行 188
 - Q サブスクリプション 105
 - Q レプリケーション 105
 - XML 発行 188
- 列の追加メッセージ 520
- 列のバージョン管理 489
- レプリケーション
 - ストアード・プロシージャに、Q サブスクリプションの 129
 - ソース列とターゲット列のマッピング
 - Q レプリケーション 111
 - 双方向レプリケーション
 - 競合検出 147
 - 説明 133
 - 単一方向レプリケーション
 - ストアード・プロシージャの使用、概要 121
 - セットアップ 91
 - 説明 91
 - 予期しない条件 114

- レプリケーション (続き)
 - データのサブセット
 - 行のフィルタリング、Q サブスクリプションの 108
 - ソース列、Q レプリケーションの 105
 - ピアツーピア・レプリケーション 136
 - 複数方向レプリケーション、セットアップ 133
 - 変換済みデータ 37
 - BLOB データ・タイプ 198
 - CLOB データ・タイプ 198
 - DBCLOB データ・タイプ 198
 - Q アプライ・プログラム 19
 - Q キャプチャー・プログラム 15
 - Q サブスクリプション
 - ストアード・プロシージャ、概要 121
 - ストアード・プロシージャ、説明 122
 - 説明 5
 - 双方向レプリケーションの作成 144
 - 単一方向レプリケーションに 1 つ作成 100
 - 単一方向レプリケーションの作成、概要 98
 - 単一方向レプリケーションの作成、複数 103
 - ピアツーピアの作成、2 つのサーバーの 153
 - ピアツーピア・レプリケーションの作成 153
 - ピアツーピア・レプリケーションの作成、3 つ以上のサーバー 155
 - メッセージ、Q キャプチャーから 106
 - ROWID 列 198
- レプリケーション・アラート・モニター
 - アラート 291
 - アラート条件
 - イベント 291
 - 概要 294
 - しきい値 291
 - 状況 291
 - 選択 305
 - E メール通知 298
 - list 294
 - 許可要件 42
 - コントロール表
 - IBMSNAP_ALERTS 472
 - IBMSNAP_CONDITIONS 474
 - IBMSNAP_CONTACTGRP 481
 - IBMSNAP_CONTACTS 482
 - IBMSNAP_GROUPS 483
 - IBMSNAP_MONENQ 483

- レプリケーション・アラート・モニター (続き)
 - コントロール表 (続き)
 - IBMSNAP_MONPARMS 483
 - IBMSNAP_MONSERVERS 485
 - IBMSNAP_MONTRAIL 488
 - コントロール表、作成 301
 - コントロール表の整理 323
 - 再初期化 307
 - 実行
 - システム開始タスクとして 339
 - バッチ (z/OS) で 338
 - 始動 306
 - 状況の検査 279
 - スケジューリング 347
 - Linux、UNIX の 347
 - Windows の 348
 - z/OS の 349
 - セットアップ 300
 - 説明 291
 - 操作 303
 - 停止 314
 - パッケージ、バインド 81
 - パラメーター
 - アラート条件の通知基準 313
 - 設定 311
 - 説明 308
 - 操作エラーの通知基準 313
 - データの整理インターバル 312
 - デフォルト値 307
 - レプリケーション・アラート・モニターの実行頻度 312
 - alert_prune_limit 308
 - autoprunce 308
 - email_server 308
 - max_notifications_per_alert 308
 - max_notification_minutes 308
 - monitor_errors 308
 - monitor_interval 308
 - monitor_limit 308
 - monitor_path 308
 - runonce 308
 - trace_limit 308
 - メッセージの表示 283
 - メモリー 32
 - モニター
 - 再初期化 307
 - 作成 304
 - レプリケーションのモニター、概要 291
 - 連絡先 291
 - 連絡先グループ 291
 - 連絡先情報の定義 302
 - JCL を使用して始動 338
- レプリケーション・キュー・マップ
 - 削除 215

レプリケーション・キュー・マップ (続き)
 作成 95
 説明 6
 双方向レプリケーション 133
 属性の変更 211
 単一方向レプリケーション 91
 ピアツーピア・レプリケーション 136
 Q サブスクリプションでグループ化 93
 レプリケーション・コマンド
 実行および保存、レプリケーション・センターから 201
 asnqacmd 376
 asnqanalyze 386
 asnqapp 372
 asnqcap 361
 asnqccmd 368
 asnqfmt 412
 asnslst 394
 asntdiff 408
 asntrep 410
 z/OS の
 MODIFY 339
 レプリケーション・サービス
 管理 342
 作成 344
 始動 345
 説明 342
 停止 345
 ドロップ 346
 名前 342
 表示名 342
 リストの表示 346
 listing 394
 レプリケーション・センター
 コマンドの実行 201
 操作コマンド 199
 ソフトウェア前提条件 86
 ターゲット・オブジェクト・プロファイル 99
 SQL スクリプト 199
 レプリケーション・ソース表
 Q サブスクリプションの 5
 XML 発行で 12
 レプリケーション・ターゲット表
 Q サブスクリプションの 5
 レプリケーション・プログラム
 イベント発行で 15
 スキーマ 21
 スケジューリング 347
 パフォーマンスの検査 279
 Q アプライ・プログラム
 スキーマ 21
 説明 19

レプリケーション・プログラム (続き)
 Q キャプチャー・プログラム
 スキーマ 21
 説明 15
 Q レプリケーションの 15
 連絡先
 説明 291
 定義 302
 連絡先グループ 291
 ロード完了 XML メッセージ 524
 ロード完了受信 XML メッセージ 512
 ロードなし
 説明 172
 定義 163
 ロード・オプション、レプリケーション
 概要 163
 自動 165
 概要 164
 z/OS の場合の考慮事項 166
 手動 169
 双方向レプリケーション 173
 単一方向レプリケーション 173
 なし 172
 ピアツーピア・レプリケーション
 2 つのサーバー 173
 3 つ以上のサーバー 173
 ログ読み取り
 Q キャプチャー進行の検査 283
 ログ・データ
 保存
 Linux、UNIX の 318
 Windows の 318
 z/OS 319
 ログ・データの保存
 Linux、UNIX の 318
 Windows の 318
 z/OS 319
 論理表
 双方向レプリケーション 133
 ピアツーピア・レプリケーション 136

A

ADDCOL シグナル
 Q サブスクリプションの 207
 XML 発行の 221
 add_partition パラメーター、Q キャプチャー (Linux、UNIX、Windows) 235, 361
 admin スレッド 281
 adminq パラメーター、Q キャプチャー 235
 agent スレッド 281
 説明 19
 待ち時間 285
 メッセージ・サイズ 68

agent スレッド (続き)
 レプリケーション・キュー・マップ 211
 レプリケーション・キュー・マップで 95
 alert_prune_limit パラメーター、レプリケーション・アラート・モニター 308
 APPLHEAPSZ データベース構成パラメーター
 Q アプライの設定 80
 Q キャプチャーの設定 78
 apply_path パラメーター、Q アプライ 263, 372
 apply_schema パラメーター、Q アプライ 263, 372, 376
 apply_server パラメーター、Q アプライ 263, 372, 376
 ARM (自動リスタート・マネージャー)
 説明 340
 レプリケーション・プログラムの再始動 341
 ASNCLP プログラム 535
 ASNMAIL exit ルーチン 299
 asnqacmd コマンド 376
 asnqanalyze コマンド 386
 asnqapp コマンド 372
 asnqcap コマンド 361
 asnqccmd コマンド 368
 asnqfmt コマンド 412
 asnslst コマンド 394
 asntdiff コマンド 408
 asntrep コマンド 410
 autoprune パラメーター、レプリケーション・アラート・モニター 308
 autostop パラメーター
 Q アプライ 263, 372
 Q キャプチャー 235, 361

B

BLOB (バイナリー・ラージ・オブジェクト) データ・タイプ
 メッセージのバッファー・サイズ 32
 メモリー 32
 ラージ・オブジェクト (LOB) XML メッセージ 508
 列値 508
 レプリケーション 198
 browser スレッド 281
 始動 273
 説明 19
 停止 275

C

capture_path パラメーター、Q キャプチャー 235, 361
capture_schema パラメーター、Q キャプチャー 235, 361
capture_server パラメーター、Q キャプチャー 235, 361
chgparms パラメーター
Q アプライ 376
Q キャプチャー 368
CHLTYPE パラメーター、WebSphere MQ 62
CLOB (文字ラージ・オブジェクト) データ・タイプ
メッセージのバッファー・サイズ 32
メモリー 32
ラージ・オブジェクト (LOB) XML メッセージ 508
列値 508
レプリケーション 198
CLP (コマンド行プロセッサ)
ASNCLP プログラムも参照 535
col エレメント、サブスクリプション・スキーマ・メッセージの 515
commit_interval パラメーター、Q キャプチャー 235, 361
CURSOR からの LOAD ユーティリティ 165

D

DATALINK データ・タイプ 197
DB2 UDB ログ
ストレージ要件 33
DB2 エクステンダー
制約事項 198
DB2CODEPAGE
環境変数
レプリケーション・プログラムの設定 77
DB2INSTANCE
環境変数
レプリケーション・プログラムの設定 77
DBCLOB (2 バイト文字ラージ・オブジェクト) データ・タイプ
メッセージのバッファー・サイズ 32
メモリー 32
ラージ・オブジェクト (LOB) XML メッセージ 508
列値 508
レプリケーション 198
DBHEAP データベース構成パラメーター
Q アプライの設定 80
Q キャプチャーの設定 78

deadlock_retries パラメーター、Q アプライ 263, 372
DEFPSIST パラメーター、WebSphere MQ 62
DEFSOPT パラメーター、WebSphere MQ 62
DEFTYPE パラメーター、WebSphere MQ 62
DISCINT パラメーター、WebSphere MQ 62
DSNUTILS ストアード・プロシージャ
一、z/OS の 166

E

E メール通知、レプリケーション 298
EDITPROC 文節 197
email_server パラメーター、レプリケーション・アラート・モニター 308
EXPORT ユーティリティおよび
IMPORT ユーティリティ 165
EXPORT ユーティリティおよび LOAD
ユーティリティ 165
EXTSHM
環境変数
レプリケーション・プログラムの設定 77

F

FIELDPROC 文節 197

H

HBINT パラメーター
発行キュー・マップで 180
レプリケーション・キュー・マップで 95
WebSphere MQ 62
hold1 スレッド 281
housekeeping スレッド 281

I

IBMQREP_ADMINMSG コントロール表 425
IBMQREP_APPENQ コントロール表 448
IBMQREP_APPLYMON コントロール表 449
IBMQREP_APPLYPARMS コントロール表
更新 271
説明 452
IBMQREP_APPLYTRACE コントロール表 454
IBMQREP_CAPENQ コントロール表 426

IBMQREP_CAPMON コントロール表 427
IBMQREP_CAPPARMS コントロール表
更新 250
説明 428
IBMQREP_CAPQMON コントロール・サーバー 432
IBMQREP_CAPTRACE コントロール表 433
IBMQREP_DELTOMB コントロール表 455
IBMQREP_DONEMSG コントロール表 456
IBMQREP_EXCEPTIONS コントロール表 457
IBMQREP_RECVQUEUEES コントロール表 459
IBMQREP_SAVERI コントロール表 461
IBMQREP_SENDQUEUEES コントロール表 434
IBMQREP_SIGNAL コントロール表 437
IBMQREP_SPILLEDROW コントロール表 462
IBMQREP_SPILLQS コントロール表 463
IBMQREP_SRCH_COND コントロール表 434
IBMQREP_SRC_COLS コントロール表 441
IBMQREP_SUBS コントロール表 442
IBMQREP_TARGETS コントロール表 465
IBMQREP_TRG_COLS コントロール表 464
IBMSNAP_ALERTS コントロール表 472
IBMSNAP_CONDITIONS コントロール表 474
IBMSNAP_CONTACTGRP コントロール表 481
IBMSNAP_CONTACTS コントロール表 482
IBMSNAP_GROUPS コントロール表 483
IBMSNAP_MONENQ コントロール表 483
IBMSNAP_MONPARMS コントロール表 483
IBMSNAP_MONSERVERS コントロール表 485
IBMSNAP_MONTRAIL コントロール表 488

J

Java ランタイム環境 (JRE) 86
JCL
レプリケーション・アラート・モニターの開始 338

JCL (続き)

レプリケーション・コマンドの変更
339

レプリケーション・プログラムの変更
339

Q アプライ・プログラムの始動 337

Q キャプチャー・プログラムの始動
336

Q レプリケーションのサンプル・プロ
グラム 532

JCL バッチ・モード 338

JRE (Java ランタイム環境) 86

L

LOB (ラージ・オブジェクト) データ・タ
イプ

キュー項目数考慮事項 72

双方向レプリケーションの制約事項
147

メッセージのバッファ・サイズ 32

メモリー 32

ラージ・オブジェクト (LOB) XML メ
ッセージ 508

列値 508

レプリケーション 198

LOCKLIST データベース構成パラメータ
ー

Q アプライの設定 80

Q キャプチャーの設定 78

LOCKTIMEOUT データベース構成パラメ
ーター

Q アプライの設定 80

Q キャプチャーの設定 78

LOGBUFSZ データベース構成パラメータ
ー

Q アプライの設定 80

Q キャプチャーの設定 78

LOGFILSIZ データベース構成パラメータ
ー

Q アプライの設定 80

Q キャプチャーの設定 78

LOGPRIMARY データベース構成パラメ
ーター

Q アプライの設定 80

Q キャプチャーの設定 78

LOGRETAİN データベース構成パラメー
ター 78

logreuse パラメーター

Q アプライ 263, 372

Q キャプチャー 235

LOGSECOND データベース構成パラメー
ター

Q アプライの設定 80

Q キャプチャーの設定 78

logstdout パラメーター

Q アプライ 263, 372

Q キャプチャー 235, 361

LONG VARCHAR データ・タイプ 197

LONG VARGRAPHIC データ・タイプ
197

M

MAXAPPLS データベース構成パラメータ
ー

Q アプライの設定 80

Q キャプチャーの設定 78

MAXDEPTH パラメーター、WebSphere

MQ 62

MAXLOCKS データベース構成パラメー
ター

Q アプライの設定 80

Q キャプチャーの設定 78

MAXMSGL パラメーター、WebSphere

MQ 62

max_notifications_per_alert パラメーター、
レプリケーション・アラート・モニター
308

max_notification_minutes パラメーター、レ
プリケーション・アラート・モニター
308

memory_limit パラメーター、Q キャプチ
ャー 235, 361

MODIFY コマンド 339

monitor_errors パラメーター、レプリケー
ション・アラート・モニター 308

monitor_interval パラメーター

Q アプライ 263, 372

Q キャプチャー 235, 361

monitor_limit パラメーター

レプリケーション・アラート・モニタ
ー 308

Q アプライ 263, 372

Q キャプチャー 235, 361

monitor_path パラメーター、レプリケーシ
ョン・アラート・モニター 308

msg ルート・エレメント、XML メッセー
ジの 495, 522

MSGDLVSQ パラメーター、WebSphere

MQ 62

N

NUMTCB パラメーター、z/OS ロードの
場合 166

O

OKSQLSTATES

ピアツーピア・レプリケーション 136

Q レプリケーションで変更 209

P

prune スレッド 281

prune パラメーター

Q アプライ 376

Q キャプチャー 368

prune_interval パラメーター

Q アプライ 372

Q キャプチャー 235, 361

pwdfile パラメーター

Q アプライ 372

Q キャプチャー 361

pwdfile パラメーター、Q アプライ 263

Q

Q アプライ待ち時間 285

Q アプライ・コントロール表

作成 87

ドロップ 216

リスト 422, 448

Q アプライ・サーバー

コントロール表のリスト 422, 448

IBMQREP_APPENQ コントロール表
448

IBMQREP_APPLYMON コントロール
表 449

IBMQREP_APPLYPARMS コントロー
ル表 452

IBMQREP_APPLYTRACE コントロー
ル表 454

IBMQREP_DELTOMB コントロール表
455

IBMQREP_DONEMSG コントロール表
456

IBMQREP_EXCEPTIONS コントロー
ル表 457

IBMQREP_RECVQUEUEES コントロー
ル表 459

IBMQREP_SAVERI コントロール表
461

IBMQREP_SPILLEDROW コントロー
ル表 462

IBMQREP_SPILLQS コントロール表
463

IBMQREP_TARGETS コントロール表
465

IBMQREP_TRG_COLS コントロール表
464

- Q アプライ・スキーマ
 - 双方向レプリケーションの規則 133
 - ピアツーピア・レプリケーションの規則 136
 - 命名規則 353
- Q アプライ・スループット 283
- Q アプライ・プログラム
 - アラート条件 294
 - エラー・オプション 118
 - 概要 19
 - 競合検出、双方向レプリケーション 147
 - 許可要件 40
 - コード・ページ 37
 - コントロール表、作成 87
 - コントロール表の整理 323
 - 実行
 - システム開始タスクとして 339
 - バッチ (z/OS) で 338
 - 始動
 - 説明 259
 - asnlqapp コマンド 372
 - JCL を使用して 337
 - 受信キューの処理 93
 - 紹介 15
 - 状況の検査 279, 376
 - スキーマ
 - 説明 21
 - 双方向レプリケーションの規則 133
 - ピアツーピア・レプリケーションの規則 136
 - スケジューリング 347
 - Linux、UNIX の 347
 - Windows の 348
 - z/OS の 349
 - ストアード・プロシージャ
 - 概要 121
 - 説明 122
 - ストレージ要件 70
 - 説明 19
 - 操作 259
 - 双方向レプリケーション 133
 - ターゲット・データベース、構成 80
 - 単一方向レプリケーション 91
 - データ変換 37
 - 停止 276, 376
 - トランザクションのアプライ、並列で 19
 - トレース・ファイル 35
 - パッケージ、バインド 81
 - パラメーター
 - 実行中の変更 272
 - 説明 262, 263
 - デフォルト値 262
 - 変更 271
- Q アプライ・プログラム (続き)
 - パラメーター (続き)
 - 保存の変更 271
 - apply_path 263
 - apply_schema 263
 - apply_server 263
 - autostop 263
 - deadlock_retries 263
 - logreuse 263
 - logstdout 263
 - monitor_interval 263
 - monitor_limit 263
 - pwdfile 263
 - qmgr 263
 - term 263
 - trace_limit 263
- ピアツーピア・レプリケーション 136
 - メッセージ処理の開始 273, 376
 - メッセージ処理の停止 275, 376
 - メモリー 29
 - 予期しない条件の処理 114
 - 呼び出しパラメーター 372
 - ランタイム・パラメーター値の検査 376
- agent スレッド
 - 説明 19
 - 定義済み 281
 - レプリケーション・キュー・マップで 95
- asnlqcmd コマンド 376
- browser スレッド
 - 説明 19
 - 定義済み 281
- prune コマンド 376
- Q キャプチャー待ち時間 285
- Q キャプチャー・コントロール表
 - 作成 87
 - ドロップ 216, 227
 - リスト 421, 425
- Q キャプチャー・サーバー
 - イベント発行のコード・ページ 38
 - コントロール表のリスト 421, 425
 - IBMQREP_ADMINMSG コントロール表 425
 - IBMQREP_CAPENQ コントロール表 426
 - IBMQREP_CAPMON コントロール表 427
 - IBMQREP_CAPPARMS コントロール表 428
 - IBMQREP_CAPQMON コントロール表 432
 - IBMQREP_CAPTRACE コントロール表 433
 - IBMQREP_SENDQUEUEUES コントロール表 434
- Q キャプチャー・サーバー (続き)
 - IBMQREP_SIGNAL コントロール表 437
 - IBMQREP_SRCH_COND コントロール表 434
 - IBMQREP_SRC_COLS コントロール表 441
 - IBMQREP_SUBS コントロール表 442
- Q キャプチャー・スキーマ
 - 双方向レプリケーションの規則 133
 - ピアツーピア・レプリケーションの規則 136
 - 命名規則 353
- Q キャプチャー・スループット
 - メモリー使用量の検査 26
 - モニター 283
- Q キャプチャー・トランザクション待ち時間 283, 285
- Q キャプチャー・プログラム
 - アラート条件 294
 - 概要 15
 - 許可要件 39
 - コード・ページ 37
 - イベント発行の 38
 - コールド・スタート
 - 回避 232
 - 再始動メッセージ 15
 - Q サブスクリプションへの影響 253
 - startmode パラメーター 235
 - XML 発行への影響 253
 - コントロール表、作成 87
 - コントロール表の整理 323
 - 再始動メッセージ
 - 説明 15
 - Q キャプチャー・プログラムの使用法 15
 - 再初期化 368
 - 実行
 - システム開始タスクとして 339
 - バッチ (z/OS) で 338
 - 実行中にコマンドを送信 368
 - 始動
 - 説明 229
 - asnlqcap コマンド 361
 - JCL 336
 - 紹介 15
 - 状況の検査 279, 368
 - スキーマ
 - 説明 21
 - 双方向レプリケーションの規則 133
 - ピアツーピア・レプリケーションの規則 136
 - スケジューリング 347
 - Linux、UNIX の 347

- Q キャプチャー・プログラム (続き)
 - スケジューリング (続き)
 - Q キャプチャー・メッセージ 106
 - Windows の 348
 - z/OS の 349
 - 説明 15
 - ソース・データベース、
 - Linux、UNIX、Windows 用の構成 78
 - 操作 229
 - 送信キューの再初期化 368
 - 送信キューの処理
 - イベント発行で 179
 - Q レプリケーションの 93
 - 双方向レプリケーション 133
 - 単一方向レプリケーション 91
 - 通信 15
 - データ変換
 - イベント発行の 38
 - Q レプリケーションの 37
 - 停止 256, 368
 - デフォルトのパラメーター値 234
 - トレース・ファイル、ストレージ要件 35
 - パッケージ、バインド 81
 - パラメーター 233
 - 実行中の変更 251
 - 説明 235
 - 変更 248
 - 変更の方式 249
 - 保存の変更 250
 - add_partition 235
 - adminq 235
 - autostop 235
 - capture_path 235
 - capture_schema 235
 - capture_server 235
 - commit_interval 235
 - logreuse 235
 - logstout 235
 - memory_limit 235
 - monitor_interval 235
 - prune_interval 235
 - qmgr 235
 - restartq 235
 - signal_limit 235
 - sleep_interval 235
 - startmode 235
 - term 235
 - trace_limit 235
 - パラメーターの変更 15
 - ピアツーピア・レプリケーション 136
 - 変更のキャプチャー 15
 - メモリー 26, 34
 - ランタイム・パラメーター値の検査 368
 - Q キャプチャー・プログラム (続き)
 - 履歴データの表示 283
 - prune コマンド 368
 - Q キャプチャーのメッセージ送信時
 - Q サブスクリプションの 106
 - XML 発行の 188
 - XML 発行で送信する列 193
 - XML 発行の変更前の値 194
 - XML メッセージ 494, 521
 - Q キャプチャー・メッセージ
 - Q キャプチャーのメッセージ送信時
 - Q サブスクリプションの 106
 - XML 発行の 188
 - XML 発行の変更前の値 194
 - XML メッセージに含まれる列 193
 - Q サブスクリプション
 - アクティブ化 253
 - エラー・オプション 118
 - オプション
 - ソース列とターゲット列のマッピング 111
 - Q キャプチャー・メッセージ 106
 - 既存への列の追加 207
 - 競合検出、双方向レプリケーション 147
 - 行のフィルタリング 108
 - 検索条件 108
 - 削除 213
 - 作成
 - 双方向レプリケーション 144
 - 単一方向レプリケーション、概要 98
 - 単一方向レプリケーション、単一 100
 - 単一方向レプリケーション、複数 103
 - ピアツーピア・レプリケーション、概要 153
 - ピアツーピア・レプリケーション、2 つのサーバー 153
 - ピアツーピア・レプリケーション、3 つ以上のサーバー 155
 - 自動ロードのニックネームを指定 167
 - ストアド・プロシージャ
 - 概要 121
 - 説明 122
 - パラメーター、ソース列にマッピング 129
 - パラメーター、操作タイプ 126
 - パラメーター、トランザクションの識別 129
 - パラメーター、抑止列 128
 - 説明 5
 - ソース列のサブセット 105
 - 双方向レプリケーション 133
- Q サブスクリプション (続き)
 - 双方向レプリケーションの競合検出 147
 - 属性の変更 205
 - ターゲットのキー列 112
 - ターゲットの索引列 112
 - ターゲット・オブジェクト・プロファイル 99
 - 単一方向レプリケーション 91
 - 非アクティブ化 254
 - ピアツーピア・レプリケーション 136
 - 命名規則 353
 - 予期しない条件、ターゲットで 114
 - レプリケーション・キュー・マップとグループ化 93
 - CAPSTART シグナル 253
 - CAPSTOP シグナル 254
 - WHERE 文節 108
- Q レプリケーション
 - エラー、リカバリー 325
 - エラー・オプション 118
 - オブジェクト名
 - 大文字小文字 356
 - 命名規則 353
 - 環境変数、設定 77
 - 既存の Q サブスクリプションへの列の追加 207
 - キューの指定 95
 - キュー・マネージャーのクラスター 73
 - 競合検出
 - 双方向レプリケーション 133
 - ピアツーピア・レプリケーション 136
 - 競合検出、双方向レプリケーション 147
 - 許可要件
 - 概要 39
 - 管理のための 42
 - レプリケーション・アラート・モニターの 42
 - Q アプライの 40
 - Q キャプチャーの 39
 - 計画
 - ストレージ要件 32
 - メモリーおよびストレージ 25
 - コード・ページ 37
 - コマンド行プロセッサ 535
 - コントロール表
 - 概要 415
 - 再編成 324
 - 作成 87
 - 図示 415
 - 整理 323
 - ドロップ 216
 - 保守 322

- Q レプリケーション (続き)
 - コントロール表 (続き)
 - リスト、レプリケーション・アラート・モニター 424
 - リスト、Q アプライ 422
 - リスト、Q キャプチャー 421
 - コンプレッション・ディクショナリー 321
 - サーバーの構成、概要 75
 - サンプル・プログラム
 - 概要 529
 - JCL 532
 - Linux、UNIX、Windows 529, 530
 - z/OS 531, 532
 - システム・サービス 335
 - 紹介 3
 - 推奨事項、ターゲット表のロードの 164
 - ストアド・プロシージャ
 - 概要 121
 - 書き込み 124
 - パラメーター、ソース列にマッピング 129
 - パラメーター、操作タイプ 126
 - パラメーター、トランザクションの識別 129
 - パラメーター、抑止列 128
 - ストレージ要件
 - トレース・ファイル 35
 - 予備キュー 70
 - DB2 UDB ログ 33
 - WebSphere MQ 69
 - 制約事項、データ・タイプ 197
 - 接続要件 75
 - 説明 3
 - 双方向レプリケーション
 - 競合検出 133, 147
 - 始動 158
 - 説明 3, 133
 - 停止 159
 - Q サブスクリプションの作成 144
 - ターゲット・オブジェクト・プロフィール 99
 - 単一方向レプリケーション
 - 行のフィルタリング 108
 - 検索条件 108
 - セットアップ 91
 - 説明 3, 91
 - ソース列、Q サブスクリプションの 105
 - ソース列とターゲット列のマッピング 111
 - ソース列のサブセット 105
 - ターゲットのキー列 112
 - ターゲットの索引列 112
- Q レプリケーション (続き)
 - 単一方向レプリケーション (続き)
 - 単一 Q サブスクリプションの作成 100
 - 予期しない条件、ターゲットで 114
 - Q キャプチャー・メッセージ 106
 - Q サブスクリプションの作成、概要 98
 - Q サブスクリプションの作成、複数 103
 - データベース、構成
 - Linux、UNIX の 76
 - Windows の 76
 - z/OS の 85
 - データ変換
 - 概要 37
 - 説明 37
 - パスワードの設定 39
 - パッケージ
 - レプリケーション・アラート・モニター (Linux、UNIX、Windows) 81
 - パッケージ、バインド
 - 概要 81
 - Q アプライ (Linux、UNIX、Windows) 81
 - Q キャプチャー (Linux、UNIX、Windows) 81
 - パッケージおよびプランの再バインド 326
 - ピアツーピア・レプリケーション
 - 競合検出 136
 - 説明 3, 136
 - 列のバージョン管理 489
 - 2 つのサーバーの始動 158
 - 2 つのサーバーの停止 159
 - 3 つ以上のサーバーの始動 160, 161
 - Q サブスクリプションの作成、概要 153
 - Q サブスクリプションの作成、2 つのサーバー 153
 - Q サブスクリプションの作成、3 つ以上のサーバー 155
 - 複数方向レプリケーション
 - セットアップ 133
 - 双方向およびピアツーピア・レプリケーション間の選択 141
 - 双方向レプリケーション 133
 - ピアツーピア・レプリケーション 136
 - プログラム
 - 紹介 15
 - スキーマ 21
 - Q アプライ・プログラム 19
- Q レプリケーション (続き)
 - プログラム (続き)
 - Q キャプチャー 15
 - 変更 205
 - 保守
 - 概要 315
 - コールド・スタート、回避 232
 - ソース・オブジェクト 316
 - ソース・システム 315
 - ソース・ログ、保存 316
 - ターゲット表 325
 - ログ・データ、保存 318, 319
 - メモリー
 - Q アプライ・プログラム 29
 - Q キャプチャー・プログラム 26
 - レプリケーション・キュー・マップ 6
 - レプリケーション・キュー・マップおよび Q サブスクリプションのグループ化 93
 - レポートの表示 279
 - ASNCLP プログラム 535
 - Q アプライ・プログラム 19
 - スキーマ 21
 - 双方向レプリケーション 133
 - 単一方向レプリケーション 91
 - ピアツーピア・レプリケーション 136
 - Q キャプチャー・プログラム 15
 - スキーマ 21
 - 双方向レプリケーション 133
 - 単一方向レプリケーション 91
 - ピアツーピア・レプリケーション 136
 - Q サブスクリプション
 - エラー・オプション 118
 - 行のフィルタリング 108
 - 検索条件 108
 - 説明 5
 - 双方向 144
 - 単一方向 103
 - ピアツーピア、2 つのサーバー 153
 - ピアツーピア、3 つ以上のサーバー 155
 - 命名規則 353
 - WHERE 文節 108
 - RUNSTATS ユーティリティ 324
 - WebSphere MQ
 - オブジェクト 47
 - セットアップ、概要 47
 - Q レプリケーション・アナライザー
 - 説明 386
 - パラメーター 386
 - Q レプリケーション・コマンド
 - asnaqcmd 376
 - asnaqanalyze 386

Q レプリケーション・コマンド (続き)

- asnqapp 372
- asnqcap 361
- asnqccmd 368
- asnqfmt 412
- asnslist 394
- asntdiff 408
- asntrep 410
- list 359
- road map 359
- qmgr パラメーター
 - Q アプライ 263
 - Q キャプチャー 235
- qrypargs パラメーター
 - Q アプライ 376
 - Q キャプチャー 368

R

- reinit パラメーター
 - Q キャプチャー 368
- reinitq パラメーター
 - Q アプライ 376
 - Q キャプチャー 368
- restartq パラメーター、Q キャプチャー 235
- ROWID 列、レプリケーション 198
- runonce パラメーター、レプリケーション・アラート・モニター 308
- RUNSTATS ユーティリティ 324

S

- SCM (Service Control Manager)
 - 説明 342
- serialization スレッド 281
- Service Control Manager (SCM)
 - 説明 342
- SHARE パラメーター、WebSphere MQ 62
- signal_limit パラメーター、Q キャプチャー 235, 361
- sleep_interval パラメーター、Q キャプチャー 235, 361
- SQL エラー
 - 受け入れ可能 288
 - 予期しない 288
- SQL 状態
 - Q レプリケーションで受け入れ可能ピアツープアの設定 136
 - 変更 209
- SQL スクリプト、レプリケーション・センターから実行および保存 200

SQLSTATE

- Q レプリケーションで受け入れ可能ピアツープアの設定 136
- 変更 209
- startmode パラメーター、Q キャプチャー 235, 361
- startq パラメーター、Q アプライ 376
- status パラメーター
 - Q アプライ 376
 - Q キャプチャー 368
- STMTHEAP データベース構成パラメーター
 - Q アプライの設定 80
 - Q キャプチャーの設定 78
- stop パラメーター
 - Q アプライ 376
 - Q キャプチャー 368
- stopq パラメーター、Q アプライ 376

T

- term パラメーター
 - Q アプライ 263, 372
 - Q キャプチャー 235, 361
- trace_limit パラメーター
 - レプリケーション・アラート・モニター 308
 - asnmon コマンドで使用 382
 - Q アプライ 263, 372
 - Q キャプチャー 235, 361
- TSO
 - レプリケーション・プログラムの実行、バッチで 338
 - Q アプライ・プログラムの始動 337
 - Q キャプチャー・プログラムの始動 336

V

- VALIDPROC 文節 197

W

- WebSphere MQ
 - イベント発行のセットアップ 47
 - オブジェクト
 - イベント発行 53
 - 管理キュー 48
 - 許可要件 70
 - 再始動キュー 48
 - 受信キュー 48
 - 接続要件 70
 - 設定要件 62
 - 送信キュー 48
 - 双方向レプリケーション 55

WebSphere MQ (続き)

- オブジェクト (続き)
 - 単一方向レプリケーション、同一システム 50
 - 単一方向レプリケーション、リモート 52
 - のリスト 48
 - 発行キュー・マップで 13
 - ピアツープア・レプリケーション (2 リモート・サーバー) 55
 - ピアツープア・レプリケーション (3 つ以上のリモート・サーバー) 58
 - 予備キュー 48
 - レプリケーション・キュー・マップで 6
- 許可要件
 - レプリケーション・アラート・モニター 70
 - Q アプライ・プログラム 70
 - Q キャプチャー・プログラム 70
- ストレージ要件 69
- 発行キュー・マップ
 - 作成 180
 - 説明 13
- パラメーター
 - CHLTYPE 62
 - DEFPSIST 62
 - DEFSOPT 62
 - DEFTYPE 62
 - DISCINT 62
 - HBINT 62
 - MAXDEPTH 62
 - MAXMSG 62
 - MSGDLVSQ 62
 - SHARE 62
- メッセージのサイズ制限
 - Q アプライ 68
 - Q キャプチャー 68
- メッセージ分割 68
- レプリケーション・キュー・マップ
 - 作成 95
 - 説明 6
- Q レプリケーションのセットアップ 47
- WHERE 文節
 - データのフィルタリング
 - Q サブスクリプションの行 108
 - XML 発行の行 189
- Windows Service Control Manager (SCM)
 - 説明 342
 - レプリケーション・サービス 342
 - レプリケーション・サービスのリスト 394
 - asnslist コマンド 394
- worker スレッド 281

X

XML 発行

- アクティブ化 253
- イベント発行で 9
- キー列 192
- 既存への列の追加 221
- 行のフィルタリング 189
- 検索条件 189
- 削除 225
- 作成
 - 概要 182
 - 単一の 183
 - 複数の 185
- 説明 12
- ソース列のサブセット 188
- 属性の変更 219
- 発行キュー・マップの処理 179
- 非アクティブ化 254
- 変更前の値に含まれる 194
- 命名規則 353
- メッセージに含まれる列 193
- Q キャプチャー・メッセージ 188
- WHERE 文節 189

XML メッセージ

- イベント発行
 - 技術要件 492
 - 区切り文字、文字データの 493
 - Q キャプチャーに 521
- エラー・レポート・メッセージ 513
- 行操作メッセージ 506
- 構造
 - 概要 491
 - Q キャプチャーから 493
 - Q キャプチャーに 521
- サブスクリプション非アクティブ化メッセージ 511
- サブスクリプション・スキーマ・メッセージ 515
- サブスクリプション・メッセージのアクティブ化 525
- サブスクリプション・メッセージの非アクティブ化 526
- 制御メッセージ 491
- 送信キュー・メッセージの無効化 524
- タイプの概要 491
- タイプのリスト 491
- 通知メッセージ 491
- データ・メッセージ 491
- トランザクション・メッセージ 497
- ハートビート・メッセージ 514
- 発行キュー・マップ経由の 13, 180
- ラージ・オブジェクト (LOB) メッセージ 508
- 列の追加メッセージ 520
- ロード完了受信メッセージ 512

XML メッセージ (続き)

- ロード完了メッセージ 524
- msg: ルート・エレメント 495, 522
- Q キャプチャーからのメッセージのリスト 494
- XML 発行に含まれる列 193
- XML 発行の変更前の値 194

Z

z/OS

- ロード考慮事項 166

z/OS コンソール

- レプリケーション・プログラムの実行、バッチで 338
- Q アプライ・プログラムの始動 337
- Q キャプチャー・プログラムの始動 336

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 <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、最新のテクニカル・ライブラリーに関する情報、資料の注文、クライアントのダウンロード、ニュースグループ、フィックスパック、ニュース、および Web リソースのリンクが含まれています。

お住まいの国や地域の IBM 事務所の所在地を調べる場合は、Web 上で IBM Directory of Worldwide Contacts (www.ibm.com/planetwide) を参照してください。

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 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。
- 電子メール (E メール) で comments@us.ibm.com に送信する。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) を必ず含めてください。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

SC88-9893-00



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