

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

バージョン 8.2

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

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、641 ページの『特記事項』に記載されている情報をお読みください。

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

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

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

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

(URL は、変更になる場合があります)

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

原 典： SC27-1121-02
IBM DB2 Information Integrator
SQL Replication 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 1994, 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
本書の使用方法	xi
本書で使用される規則と用語	xiii
構文図の見方	xv
ロード・マップ	xvi

DB2 レプリケーション バージョン 8

での変更点

バージョン 8.2 での新機能	xvii
新しいレプリケーションの解決策	xvii
新機能	xvii
バージョン 8.1.4 での新機能	xviii
新機能	xviii
パフォーマンスの向上	xviii
バージョン 8 フィックスパック 2 での新機能	xviii
ユーザビリティの改善	xviii
パフォーマンスの向上	xix
新機能	xix
コントロール表の変更	xix
バージョン 8.1 での新機能	xix
ユーザビリティの改善	xix
パフォーマンスの向上	xxi
新規のユーザー・インターフェース	xxi
新機能	xxii
保守容易性の向上	xxv
レプリケーション・システム・コマンドの変更	xxvi
コントロール表の変更	xxvii
サポートされなくなった機能	xxix

第 1 部 レプリケーションのガイド

第 1 章 SQL レプリケーションの計画

移行計画	3
メモリーの計画	3
キャプチャー・プログラムによって使用されるメモリー	3
アプライ・プログラムによって使用されるメモリー	5
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	5
ストレージの計画	6
ログの影響の計画	6
ターゲット表およびコントロール表のストレージ要件の計画	8
一時ファイルのストレージ要件の計画	9
競合検出の計画	11
非 DB2 リレーショナル・ソースの計画	12
キャプチャー・トリガーの場合のトランザクション・スループット率の計画	12

非 DB2 リレーショナル・ソース・サーバーの場合のログの影響の計画	12
Oracle ソース・サーバーの場合のロックの計画	13
既存のトリガーとキャプチャー・トリガーとの共存の計画	13
コード・ページ・トランザクションの計画	13
互換コード・ページを持つデータベース間でのデータのレプリケーション	13
レプリケーションに合わせた各国語サポート (NLS) の構成	14
DB2 UDB for z/OS のレプリケーションの計画	15
パフォーマンス・チューニング	16

第 2 章 SQL レプリケーション用のサー

バーの構成

レプリケーション・サーバーに対するアクセスのコントロール	17
レプリケーションのための接続要件	17
レプリケーション用のユーザー ID の許可	19
管理の許可要件	19
キャプチャー・プログラムの許可要件	21
非 DB2 リレーショナル・データベースでのキャプチャー・トリガーの場合の許可要件	23
アプライ・プログラムの許可要件	23
レプリケーション・アラート・モニターの場合の許可要件	25
レプリケーション用のユーザー ID およびパスワードの保管 (Linux、UNIX、Windows)	26
レプリケーション・コントロール表のセットアップ	26
コントロール表の作成 (Linux、UNIX、Windows)	26
コントロール表の作成 (z/OS)	27
コントロール表の作成 (OS/400)	27
非 DB2 リレーショナル・ソース用のコントロール表の作成	28
キャプチャー・コントロール表のセットを複数作成する	28
複数のデータベース・パーティション上でのコントロール表の作成	29
レプリケーション・プログラムのセットアップ	29
レプリケーション・プログラムのセットアップ (Linux、UNIX、Windows)	30
キャプチャー・プログラムおよびアプライ・プログラムのセットアップ (OS/400)	34
レプリケーション・プログラムのセットアップ (z/OS)	36
複数データベース・パーティションのキャプチャー	36
ジャーナルのセットアップ (OS/400)	36
ソース表用ジャーナルの作成 (OS/400)	37
ジャーナルおよびジャーナル・レシーバーの管理 (OS/400)	38

第 3 章 表およびビューを SQL レプリケーション・ソースとして登録する 43

DB2 表をソースとして登録する	43
非 DB2 リレーショナル表をソースとして登録する	45
ソース表用の登録オプション	47
列のサブセットの登録 (垂直方向のサブセット化)	47
フル・リフレッシュ・コピーと変更キャプチャー・レプリケーション	48
変更後イメージ列と変更前イメージ列	50
変更前イメージ接頭部	52
エラー発生時におけるキャプチャー・プログラムの停止	53
キャプチャー・プログラムが更新を保存する方法	54
変更の再キャプチャーの防止 (Update-anywhere レプリケーション)	55
競合検出の設定 (Update-anywhere レプリケーション)	59
リモート・ジャーナリングを使用する表の登録 (OS/400)	61
主キーの代わりに相対レコード番号 (RRN) を使用する (OS/400)	62
レプリケーション・ソースとしてのビューの動作	62
単一の表に対するビュー	62
複数の表の結合に対するビュー	63
表のビューをソースとして登録する	65
CCD 表をソースとして保守する (IMS)	66

第 4 章 SQL レプリケーションのソースのサブスクリプト 69

ソースおよびターゲットの分類方法の計画	69
サブスクリプション・セット・メンバー数の計画	70
アプライ修飾子ごとのサブスクリプション・セット数の計画	71
サブスクリプション・セットの作成	72
サブスクリプション・セットのオプション処理	74
セットがアクティブかどうかの指定	74
アプライ・プログラムが取り出すデータに相当する分数の指定	75
アプライ・プログラムが参照保全性のあるターゲット表をロードする方法の決定	77
アプライ・プログラムがセットのメンバーへの変更を複製する方法の指定	78
サブスクリプション・セット用の SQL ステートメントまたはストアード・プロシージャの定義	79
サブスクリプション・セットのレプリケーションのスケジューリング	80
サブスクリプション・セット内におけるソース表およびビューのターゲット表およびビューへのマッピング	82
ターゲット・タイプの選択	85
読み取り専用ターゲット表の定義	87
複数層構成における中間層の定義	91
読み取り/書き込みターゲットの定義 (Update-anywhere)	94
既存の表をターゲット表として使用する	96

すべてのターゲット表タイプに共通のプロパティー	97
ターゲットに適用するソース列	97
ターゲットに適用するソース行	98
ソース列からターゲット列にマップする方法	99
ターゲット・キー	100
アプライ・プログラムがターゲット・キー変更オプションを使用してターゲット・キー列を更新する方法	101

第 5 章 SQL レプリケーションでの特殊なデータ・タイプのレプリケーション . 105

レプリケーションにおける一般的なデータの制約事項	105
ラージ・オブジェクトのレプリケーション	106
DATALINK 値のレプリケーション	107
ASNDLCOPY 出力ルーチンの設定と使用方法	109
DLFM_ASNCOPYD の設定と使用方法 (Linux、UNIX、Windows)	111
ASNDLCOPYD の設定と使用方法 (OS/400)	112

第 6 章 SQL レプリケーション環境におけるデータのサブセット化 115

登録時におけるデータのサブセット化	115
ビューを使用したソース・データのサブセット化	116
特定の行のキャプチャーを防止するために CD 表にトリガーを定義する (Linux、UNIX、Windows、z/OS)	116
サブスクリプションの際のデータのサブセット化	117

第 7 章 SQL レプリケーション環境におけるデータ操作 119

ストアード・プロシージャまたは SQL ステートメントを使用したデータ拡張	120
名前が異なるソース列とターゲット列のマッピング	121
算出列の作成	121

第 8 章 SQL レプリケーション用のレプリケーション SQL スクリプトのカスタマイズおよび実行 123

第 9 章 SQL レプリケーションに関するキャプチャー・プログラムの操作 . . . 125

キャプチャー・プログラムのデフォルトの稼働パラメーター	125
キャプチャー・プログラムの稼働パラメーターの変更	127
キャプチャー・プログラムの始動 (Linux、UNIX、Windows、z/OS)	129
add_partition (Linux、UNIX、Windows)	131
autoprun (Linux、UNIX、Windows、z/OS)	131
autostop (Linux、UNIX、Windows、z/OS)	132
capture_path (Linux、UNIX、Windows、z/OS)	132
capture_schema (Linux、UNIX、Windows、z/OS)	133
capture_server (Linux、UNIX、Windows、z/OS)	133
commit_interval (Linux、UNIX、Windows、z/OS)	134

lag_limit (Linux, UNIX, Windows, z/OS) . . .	134
logreuselimit (Linux, UNIX, Windows, z/OS) . . .	134
logstdout (Linux, UNIX, Windows, z/OS) . . .	135
memory_limit (Linux, UNIX, Windows, z/OS) . . .	135
monitor_interval (Linux, UNIX, Windows, z/OS) . . .	136
monitor_limit (Linux, UNIX, Windows, z/OS) . . .	136
prune_interval (Linux, UNIX, Windows, z/OS) . . .	136
retention_limit (Linux, UNIX, Windows, z/OS) . . .	137
sleep_interval (Linux, UNIX, Windows, z/OS) . . .	138
startmode (Linux, UNIX, Windows, z/OS) . . .	138
term (Linux, UNIX, Windows, z/OS) . . .	139
trace_limit (Linux, UNIX, Windows, z/OS) . . .	139
キャプチャー・プログラムの始動 (OS/400) . . .	140
実行中のキャプチャー・プログラムの動作の変更	141
キャプチャー・パラメーター表の稼働パラメーターの変更	142
キャプチャー・プログラムの停止	143
キャプチャーの中断 (Linux, UNIX, Windows, z/OS)	144
キャプチャーの再開 (Linux, UNIX, Windows, z/OS)	144
キャプチャーの再初期化	145
第 10 章 SQL レプリケーションに関するアプライ・プログラムの操作 147	
アプライ・プログラムのデフォルトの稼働パラメーター	147
アプライ・プログラムの稼働パラメーターの変更	148
アプライ・プログラムの始動 (Linux, UNIX, Windows, z/OS)	149
apply_path (Linux, UNIX, Windows, z/OS) . . .	151
apply_qual (Linux, UNIX, Windows, z/OS) . . .	151
control_server (Linux, UNIX, Windows, z/OS) . . .	152
copyonce (Linux, UNIX, Windows, z/OS) . . .	152
db2_subsystem (z/OS)	153
delay (Linux, UNIX, Windows, z/OS)	153
errwait (Linux, UNIX, Windows, z/OS)	153
inamsg (Linux, UNIX, Windows, z/OS)	154
loadxit (Linux, UNIX, Windows, z/OS)	154
logreuselimit (Linux, UNIX, Windows, z/OS)	154
logstdout (Linux, UNIX, Windows, z/OS)	155
notify (Linux, UNIX, Windows, z/OS)	155
opt4one (Linux, UNIX, Windows, z/OS)	155
pwdfile (Linux, UNIX, Windows)	156
sleep (Linux, UNIX, Windows, z/OS)	156
spillfile (Linux, UNIX, Windows, z/OS)	157
sqlerrcontinue (Linux, UNIX, Windows, z/OS)	157
term (Linux, UNIX, Windows, z/OS)	158
trireuse (Linux, UNIX, Windows, z/OS)	158
アプライ・プログラムの始動 (OS/400)	159
アプライ・パラメーター表での稼働パラメーターの変更 (Linux, UNIX, Windows, z/OS)	160
アプライ・プログラムの停止	161
ASNDONE 出口ルーチンの変更 (Linux, UNIX, Windows, z/OS)	161
ASNDONE 出口ルーチンの変更 (OS/400)	162

ASNLLOAD 出口ルーチンを使ったターゲット表のリフレッシュ	164
ASNLLOAD 出口ルーチンを使ったターゲット表のリフレッシュ (Linux, UNIX, Windows)	165
ASNLLOAD 出口ルーチンを使ったターゲット表のリフレッシュ (z/OS)	167
ASNLLOAD 出口の動作のカスタマイズ (Linux, UNIX, Windows, z/OS)	168
ASNLLOAD 出口ルーチンを使ったターゲット表のリフレッシュ (OS/400)	170

第 11 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 173

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	173
レプリケーション・アラート・モニター	173
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	176
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	176
レプリケーション・アラート・モニターのアラート条件	176
レプリケーション・アラート条件の E メール通知	180
レプリケーション中のアラートを送信する ASNLMAIL 出口ルーチン	181
レプリケーション・アラート・モニターのセットアップ	182
レプリケーション・アラート・モニターのセットアップ	182
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	183
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	184
レプリケーション・アラート・モニターの操作	185
レプリケーション・アラート・モニターの操作	185
レプリケーションまたは公開用のモニターの作成	186
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	187
モニターの開始	188
モニターの再初期化	189
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	190
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	190
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	193
モニターの停止	196

第 12 章 SQL レプリケーションの要求時報告 199

レプリケーション・プログラムの現在の状況のチェック (Linux, UNIX, Windows, z/OS)	199
--	-----

キャプチャー・プログラムおよびアプライ・プログラムのジャーナル・ジョブの状況のチェック (OS/400)	201
傾向の履歴データの検討	201
キャプチャー・プログラムのメッセージの検討	203
キャプチャー・プログラムのスループットの検査	203
キャプチャー・プログラムによって処理されるデータの待ち時間の表示	204
アプライ・プログラムのメッセージの検討	204
アプライ・プログラムのスループットの検査	205
トランザクションのレプリケーションに要した平均時間の表示	205
モニター・プログラムのメッセージの検討	206
キャプチャー・プログラムの進行のモニター (OS/400)	206

第 13 章 SQL レプリケーション環境の変更 209

新規オブジェクトの登録	209
登録済みオブジェクトの登録属性の変更	210
ソース表への列の追加	211
登録済みオブジェクトの変更のキャプチャーの停止	214
登録の再活動化	215
登録の除去	216
キャプチャー・スキーマの変更	217
新規サブスクリプション・セットの作成	220
既存のサブスクリプション・セットに新しいサブスクリプション・セット・メンバーを追加する	220
既存のサブスクリプション・セットでサブスクリプション・セット・メンバーを使用不可にする	221
既存のサブスクリプション・セットに対してサブスクリプション・セット・メンバーを使用可能にする	221
サブスクリプション・セットの属性の変更	222
サブスクリプション・セット名の変更	222
サブスクリプション・セットの分割	224
サブスクリプション・セットのマージ	229
サブスクリプション・セットのアプライ修飾子の変更	232
サブスクリプション・セットの非活動化	235
サブスクリプション・セットの除去	236
データベース・アプリケーション・イベントとレプリケーション・イベントの調整	237
USER タイプ・シグナルを使用したイベント	
END_SYNCHPOINT の設定	237
リモート・ジャーナリング用にジャーナル・シグナル表を作成する	238
キャプチャー CMD STOP シグナルの使用法	239
アプライ・プログラム外部の CAPSTART ハンドシェイク・シグナルの実行	242
CAPSTOP シグナルの実行	244
別のシステムへのレプリケーション構成のプロモート	245

第 14 章 SQL レプリケーション環境の保守 247

ソース・システムの保守	247
-----------------------	-----

ソース・オブジェクトの保守	247
ソース・ログおよびジャーナル・レシーバーの保守および保存	248
コントロール表の保守	252
RUNSTATS ユーティリティの使用法 (Linux、UNIX、Windows、z/OS)	253
パッケージおよびプランの再バインド (Linux、UNIX、Windows、z/OS)	253
コントロール表の再編成	254
コントロール表の整理	255
レプリケーションの失敗の防止およびエラーからのリカバリー	259
ターゲット表の保守	261

第 2 部 レプリケーション・センタ 263

第 15 章 SQL レプリケーションの場合のレプリケーション・センターの使用法 . 265

レプリケーション・センターの前提条件	267
ホスト RDBMS 用のレプリケーション・センターの構成	267
レプリケーション・センターの始動	268
SQL レプリケーションの場合のレプリケーション・センター・ランチパッドの使用	269
レプリケーション・センターのユーザー ID およびパスワードの管理	271
レプリケーション・プロファイルの作成	272
コントロール表プロファイルの作成	272
ソース・オブジェクト・プロファイルの作成	273
ターゲット・オブジェクト・プロファイルの作成	274
レプリケーション・コントロール表の作成	275
キャプチャー・コントロール表の作成	275
アプライ・コントロール表の作成	276
モニター・コントロール表の作成	277
レプリケーション・センターへのサーバーの追加	278
データベースの変更キャプチャーの使用可能化 (UNIX および Windows).	280
ソースの登録	281
サブスクリプション・セットの作成	282
サブスクリプション・セットの情報の定義	283
ソースからターゲットへのマッピング	284
サブスクリプション・セットのスケジューリング	285
サブスクリプション・セットへの SQL ステートメントまたはストアド・プロシージャの追加	286
サブスクリプション・セットの活動化または非活動化	286
レプリケーション・オブジェクトのプロモート	287
登録済みの表またはビューのプロモート	287
サブスクリプション・セットのプロモート	288
ターゲット表の強制フル・リフレッシュ	289
レプリケーション定義の除去または削除	290
キャプチャー・プログラムの操作	290
アプライ・プログラムの操作	291
レプリケーション・アラート・モニターの操作	292

第 16 章 基本的な SQL レプリケーションのシナリオ: DB2 for Windows . . .	295
開始する前に	295
このシナリオの計画	296
レプリケーション・ソース	296
レプリケーション・ターゲット	296
レプリケーション・オプション	297
このシナリオ用のレプリケーション環境のセットアップ	298
ステップ 1: キャプチャー・プログラムのレプリケーション・コントロール表を作成する	298
ステップ 2: ソース・データベースのレプリケーションを使用可能にする	299
ステップ 3: レプリケーション・ソースを登録する	299
ステップ 4: アプライ・プログラムのレプリケーション・コントロール表を作成する	302
ステップ 5: 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	303
ステップ 6: アプライ・パスワード・ファイルを作成する	309
ステップ 7: シナリオのデータを複製する	310
レプリケーション環境における操作	312
ステップ 1: ソース表を更新する	312
ステップ 2: キャプチャー・プログラムの状況を表示する	313
ステップ 3: アプライ・プログラムの状況を表示する	314
ステップ 4: キャプチャー・プログラムとアプライ・プログラムを停止する	315
レプリケーションのモニター	316
ステップ 1: モニター・プログラムのレプリケーション・コントロール表を作成する	316
ステップ 2: レプリケーション・アラートの連絡先を作成する	318
ステップ 3: キャプチャー・プログラムのアラート条件を選択する	318
ステップ 4: アプライ・プログラムのアラート条件を選択する	320
ステップ 5: モニター修飾子のレプリケーション・アラート・モニターを開始する	321
<hr/>	
第 3 部 レプリケーションのリファレンス	325
第 17 章 SQL レプリケーション・オブジェクトの命名規則	327
第 18 章 SQL レプリケーション用のシステム・コマンド (Linux、UNIX、Windows、z/OS)	329
asnacmd: アプライの操作	330
asnanalyze: アナライザーの操作	331
asnapply: アプライの始動	334
asncap: キャプチャーの始動	342

asnccmd: キャプチャーの操作	349
asnmcmd: 実行中のレプリケーション・アラート・モニターの処理	354
asnmon: レプリケーション・アラート・モニターの始動	357
asnpwd: パスワード・ファイルの作成および保守	362
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	366
asnndrop: DB2 レプリケーション・サービスのドロップ	370
asnslst: DB2 レプリケーション・サービスのリスト	371
asntdiff: ソース表とターゲット表とのデータの比較	372
asntrc: レプリケーション・トレース機能の操作	374
asntrep: ソース表とターゲット表の間の違いの修復	382

第 19 章 SQL レプリケーション用のシステム・コマンド (OS/400)	385
ADDDPRREG: DPR 登録の追加 (OS/400)	386
ADDDPRSUB: DPR サブスクリプション・セットの追加 (OS/400)	395
ADDDPRSUBM: DPR サブスクリプション・セットのメンバーの追加 (OS/400)	411
ANZDPR: アナライザーの操作 (OS/400)	422
CHGDPRCAPA: DPR キャプチャー属性の変更 (OS/400)	425
CRTDPRBL: レプリケーション・コントロール表の作成 (OS/400)	430
ENDDPRAPY: アプライ・プログラムの停止 (OS/400)	431
ENDDPRCAP: キャプチャー・プログラムの停止 (OS/400)	434
GRTDPRAUT: ユーザーの許可 (OS/400)	436
INZDPRCAP: DPR キャプチャー・プログラムの再初期化 (OS/400)	446
OVRDPRCAPA: DPR キャプチャー属性のオーバーライド (OS/400)	448
RMVDPRREG: DPR 登録の除去 (OS/400)	453
RMVDPRSUB: DPR サブスクリプション・セットの除去 (OS/400)	454
RMVDPRSUBM: DPR サブスクリプション・セット・メンバーの除去 (OS/400)	456
RVKDPRAUT: 権限の取り消し (OS/400)	458
STRDPRAPY: アプライ・プログラムの始動 (OS/400)	460
STRDPRCAP: キャプチャー・プログラムの始動 (OS/400)	468
WRKDPRTRC: DPR トレース機能の使用方法 (OS/400)	476

第 20 章 SQL レプリケーション・プログラムの操作 (z/OS)	483
レプリケーション・プログラムの操作に JCL またはシステム開始タスクを使用する方法 (z/OS)	483
JCL を使用したレプリケーション・プログラムの操作	483

システム開始タスク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	485
レプリケーション・プログラムを自動的に再始動するために MVS 自動リスタート・マネージャー (ARM) を使用する方法 (z/OS)	486
データ共有モードへのレプリケーション環境の移行 (z/OS)	487

第 21 章 Windows Service Control Manager を使用して SQL レプリケーションのシステム・コマンドを発行する方法 (Windows)

レプリケーション・サービスの作成	489
レプリケーション・サービスの操作	490
レプリケーション・サービスのドロップ	491

第 22 章 各種オペレーティング・システムでの SQL レプリケーション・プログラムのスケジューリング

UNIX および Linux 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	493
Windows 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	493
z/OS 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	494
OS/400 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	494

第 23 章 SQL レプリケーション・コンポーネントの通信方法

レプリケーション・センター、キャプチャー・プログラムまたはトリガー、およびアプライ・プログラム	495
キャプチャー・プログラムおよびアプライ・プログラム	496
キャプチャー・トリガーおよびアプライ・プログラム	498
レプリケーション・センターおよびレプリケーション・アラート・モニター	499
レプリケーション・アラート・モニター、キャプチャー・プログラム、およびアプライ・プログラム	499

第 24 章 SQL レプリケーションの表構造

表の概観	501
キャプチャー・コントロール・サーバーで使用される表のリスト	508
アプライ・コントロール・サーバーで使用される表のリスト	511
モニター・コントロール・サーバーのコントロール表のリスト	512
ターゲット・サーバーで使用される表のリスト	514
キャプチャー・コントロール・サーバー上の表、およびその列に関する記述	514
ASN.IBMSNAP_CAPSCHEMAS	515

schema.IBMSNAP_AUTHTKN (OS/400)	515
schema.IBMSNAP_CAPENQ (UNIX、Windows、z/OS)	516
schema.IBMSNAP_CAPMON	517
schema.IBMSNAP_CAPPARMS	519
schema.IBMSNAP_CAPTRACE (DB2 のみ)	522
schema.CCD_table (DB2 以外)	523
schema.CD_table	525
schema.IBMSNAP_PARTITIONINFO	526
schema.IBMSNAP_PRUNCNTL	527
schema.IBMSNAP_PRUNE_LOCK	529
schema.IBMSNAP_PRUNE_SET	529
schema.IBMSNAP_REG_EXT (OS/400)	530
schema.IBMSNAP_REGISTER	532
schema.IBMSNAP_REG_SYNCH (DB2 以外のレリショナル)	539
schema.IBMSNAP_RESTART	540
schema.IBMSNAP_SEQTABLE (Informix)	542
schema.IBMSNAP_SIGNAL	543
schema.IBMSNAP_UOW	546
アプライ・コントロール・サーバー上の表、およびその列に関する記述	548
ASN.IBMSNAP_APPENQ	548
ASN.IBMSNAP_APPLY_JOB (OS/400)	549
ASN.IBMSNAP_APPPARMS	550
ASN.IBMSNAP_APPLYTRACE	554
ASN.IBMSNAP_APPLYTRAIL	554
ASN.IBMSNAP_SUBS_COLS	560
ASN.IBMSNAP_SUBS_EVENT	562
ASN.IBMSNAP_SUBS_MEMBR	563
ASN.IBMSNAP_SUBS_SET	568
ASN.IBMSNAP_SUBS_STMTS	574
モニター・コントロール・サーバー上の表、およびその列に関する記述	576
IBMSNAP_ALERTS 表	576
IBMSNAP_CONDITIONS 表	578
IBMSNAP_CONTACTGRP 表	584
IBMSNAP_CONTACTS 表	585
IBMSNAP_MONENQ 表	586
IBMSNAP_GROUPS 表	586
IBMSNAP_MONPARMS 表	586
IBMSNAP_MONSERVERS 表	588
IBMSNAP_MONTRACE 表	590
IBMSNAP_MONTRAIL 表	591
ターゲット・サーバー上の表、およびその列に関する記述	592
基礎集約表	592
変更集約表	593
整合変更データ (CCD) 表	594
ポイント・イン・タイム表	596
レプリカ表	597
ユーザー・コピー表	597

付録 A. SQL レプリケーション用の Unicode および ASCII のコード化スキーム (z/OS)

コード化スキームの選択	599	ユーザー補助	639
コード化スキームの設定	600	キーボード入力およびナビゲーション	639
付録 B. SQL レプリケーションの場合		利用可能なモニター	639
にキャプチャー・プログラムがジャーナル		フォントの設定	639
項目タイプを処理する方法 (iSeries) . 601		色の非依存	639
		代替アラート・キュー	639
付録 C. アプリケーション		支援テクノロジーとの互換性	640
(Linux、UNIX、Windows) 内部からの		利用しやすい資料	640
SQL レプリケーション・プログラムの			
始動 605		特記事項 641	
		商標	643
用語集 607		IBM と連絡を取る 645	
用語集	607	製品情報	645
		資料についてのコメント	645
索引 621			

本書について

本書は、DB2 データ・レプリケーションに焦点を当て、DB2 データ・レプリケーションを使用してデータ・レプリケーション環境を計画、セットアップ、保守、およびモニターする方法について説明します。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 SQL レプリケーション・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。

SQL レプリケーションとは、DB2 レプリケーションとも呼ばれ、SQL でシステム間でデータを複製するのに使用されるレプリケーションのタイプのことです。この用語は、Q レプリケーションのレプリケーション・タイプと区別するために使用されます。Q レプリケーションは、メッセージ・キューを使用してデータを複製します。

本書の対象読者

本書は、データベース管理者、LAN 管理者、および SQL 環境中のデータ・レプリケーション環境のセットアップおよび保守担当者を対象にしています。読者は、標準 SQL データベース用語を知り、オペレーティング・システム (レプリケーションも含むようになる) の作業知識を持ち、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、サーバー接続、およびネットワークングでの経験を持つ必要があります。また、ユーザー環境のアプリケーションについて理解し、そのアプリケーションが SQL の照会とコマンドを使用して複製したいデータをどのように操作するのかを知っている必要があります。さらに、レプリケーションの概念とコンポーネントについても熟知している必要があります。

本書の使用方法

本書のほとんどのセクションは、すべてのオペレーティング・システム環境での SQL レプリケーション関数に関するものです。中には、オペレーティング・システム固有の情報を含むセクションもあります。

本書の編成と内容は、最終リリース以降、変更されています。以下の 3 部構成になっています。

- 1 ページの『第 1 部 レプリケーションのガイド』では、ユーザーのレプリケーション環境をプランニング、セットアップ、実行、および保守する方法について説明します。以下の章があります。
 - 3 ページの『第 1 章 SQL レプリケーションの計画』では、レプリケーション環境を計画および設計する方法について説明します。
 - 17 ページの『第 2 章 SQL レプリケーション用のサーバーの構成』では、レプリケーション用に環境を設定する方法について説明します。
 - 43 ページの『第 3 章 表およびビューを SQL レプリケーション・ソースとして登録する』では、レプリケーション・ソースを登録するために必要な情報について説明します。

- 69 ページの『第 4 章 SQL レプリケーションのソースのサブスクリプト』では、サブスクリプション・セットの作成方法と、メンバーをサブスクリプション・セットへ追加する方法について説明します。
- 105 ページの『第 5 章 SQL レプリケーションでの特殊なデータ・タイプのレプリケーション』では、ソース表の LOB 値および DATALINK 値のためのレプリケーション・オプションについて説明します。
- 115 ページの『第 6 章 SQL レプリケーション環境におけるデータのサブセット化』では、どのようにデータがターゲットへアプライされるのかを説明する他、ターゲットへキャプチャーおよびアプライされるデータをカスタマイズする方法についても説明します。
- 119 ページの『第 7 章 SQL レプリケーション環境におけるデータ操作』では、ソース・データを操作するための、キャプチャー・プログラムまたはアプライ・プログラムの使用方法について説明します。
- 123 ページの『第 8 章 SQL レプリケーション用のレプリケーション SQL スクリプトのカスタマイズおよび実行』では、レプリケーション環境で SQL を実行する方法について説明します。
- 125 ページの『第 9 章 SQL レプリケーションに関するキャプチャー・プログラムの操作』では、すべてのオペレーティング・システム環境について、キャプチャー・プログラムを操作する方法を説明します。
- 147 ページの『第 10 章 SQL レプリケーションに関するアプライ・プログラムの操作』では、すべてのオペレーティング・システム環境について、アプライ・プログラムを操作する方法を説明します。
- 173 ページの『第 11 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、レプリケーション・アラート・モニターを使用してレプリケーション環境をモニターする方法について説明します。
- 199 ページの『第 12 章 SQL レプリケーションの要求時報告』では、要求時にレプリケーション環境に関するレポートを生成して表示する方法について説明します。
- 209 ページの『第 13 章 SQL レプリケーション環境の変更』では、レプリケーション環境で日常の変更を行う方法について説明します。
- 247 ページの『第 14 章 SQL レプリケーション環境の保守』では、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方法について説明します。
- 263 ページの『第 2 部 レプリケーション・センター』では、レプリケーションのためのグラフィカル・ユーザー・インターフェースについて説明します。以下の章があります。
 - 265 ページの『第 15 章 SQL レプリケーションの場合のレプリケーション・センターの使用法』では、レプリケーション・センターについて説明します。
 - 295 ページの『第 16 章 基本的な SQL レプリケーションのシナリオ: DB2 for Windows』では、レプリケーション・センターを使用して、サンプル・データを使って簡単なレプリケーション・シナリオを実行する方法について説明します。

- 325 ページの『第 3 部 レプリケーションのリファレンス』では、レプリケーション・コマンドとレプリケーション表の構造について説明します。以下の章があります。
 - 327 ページの『第 17 章 SQL レプリケーション・オブジェクトの命名規則』では、レプリケーション・オブジェクトに有効な名前の指定方法について説明します。
 - 329 ページの『第 18 章 SQL レプリケーション用のシステム・コマンド (Linux、UNIX、Windows、z/OS)』では、DB2 レプリケーションの経験のあるユーザーが、Linux、UNIX、Windows、および z/OS オペレーティング・システムでレプリケーション操作をするために、レプリケーション・センターの代わりに使用することができるコマンドについて説明します。
 - 385 ページの『第 19 章 SQL レプリケーション用のシステム・コマンド (OS/400)』では、OS/400 オペレーティング・システムで、ローカルに、レプリケーションをセットアップ、管理、および保守するためのコマンドについて説明します。
 - 483 ページの『第 20 章 SQL レプリケーション・プログラムの操作 (z/OS)』では、z/OS オペレーティング・システムで JCL やシステム起動タスクを使用して、レプリケーション・プログラムを開始および操作する方法について説明します。
 - 489 ページの『第 21 章 Windows Service Control Manager を使用して SQL レプリケーションのシステム・コマンドを発行する方法 (Windows)』では、Windows オペレーティング・システムで、レプリケーション・プログラムを操作するためのサービスを作成する方法について説明します。
 - 493 ページの『第 22 章 各種オペレーティング・システムでの SQL レプリケーション・プログラムのスケジューリング』では、さまざまなオペレーティング・システムで、キャプチャー、アプライ、およびレプリケーション・アラート・モニター・プログラムをスケジューリングする方法について説明します。
 - 495 ページの『第 23 章 SQL レプリケーション・コンポーネントの通信方法』では、レプリケーション・コンポーネントが相互通信のためにコントロール表を使用する方法について説明します。
 - 501 ページの『第 24 章 SQL レプリケーションの表構造』では、さまざまな SQL レプリケーション・サーバー上にある SQL レプリケーション表の構造について説明します。
- 付録には、役に立つ補足情報があります。

本書で使用される規則と用語

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、示された通りに正確に入力するテキストの例を表します。
- **イタリック** は、値に置き換える変数を表します。また、本の表題や語の強調にも使用されます。

本書では、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。本書で使用するレプリケーション概念はすべて、用語集に定義されています。

特に指定しない限り、以下を意味するものとします。

UNIX UNIX とは、すべての UNIX オペレーティング・システム (UNIX、HP UX、AIX など) 用の DB2 Universal Database を指します。

Linux Linux とは、DB2 Universal Database for Linux を指します。

Windows

Windows とは、DB2 Universal Database for Windows を指します。

OS/400

OS/400 とは、DB2 DataPropagator for iSeries を指します。

z/OS z/OS とは、DB2 Universal Database for z/OS and OS/390 を指します。z/OS は、OS/390 の次世代オペレーティング・システムで、z/OS 上の UNIX System Services (USS) も含まれます。

iSeries

iSeries とは、AS/400 と iSeries サーバーの両方を指します。iSeries は、AS/400 の次世代サーバーです。OS/400 オペレーティング・システムは、AS/400 および iSeries サーバーの両方で稼働します。

SQL レプリケーション

SQL レプリケーションとは、DB2 レプリケーションとも呼ばれ、DB2 用に開発された 2 種類のデータ・レプリケーションのうちの 1 つです。この用語は、Q レプリケーションの SQL を使用したレプリケーションと区別するために使用されます。Q レプリケーションとは、メッセージ・キューを使用したレプリケーションのことです。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。このプログラムはトランザクションをステージング表に保管し、この表はアプライ・プログラムによって並行して読み取られてターゲットに適用されません。

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並行して読み取られてターゲットに適用されます。

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。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。

たとえば、アプライ・プログラムの始動 (*Linux*、*UNIX*、*Windows*、*z/OS*) というセクションでは、DB2 Universal Database (*Linux* 用、すべての UNIX オペレーテ

ィング・システム用、Windows 用、z/OS および OS/390 用の各版) からアプライ・プログラムを始動する方法について説明しています。また、アプライ・プログラムの始動 (OS/400) というセクションでは、DB2 DataPropagator for iSeries を使用する際のアプライ・プログラムの始動方法について説明しています。

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 \blacktriangleright — は、ステートメントの先頭を示します。

記号 — \blacktriangleright 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 \blacktriangleright — 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 — \blacktriangleleft 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 \blacktriangleright — で始まり、記号 — \blacktriangleleft 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、区切りの句読点が無ければ、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。

\blacktriangleright —*required_item*— \blacktriangleleft

- 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。

\blacktriangleright —*required_item*— $\boxed{\text{default_value}}$ — \blacktriangleleft

- 任意指定の項目は、メイン・パスの下に表示されます。

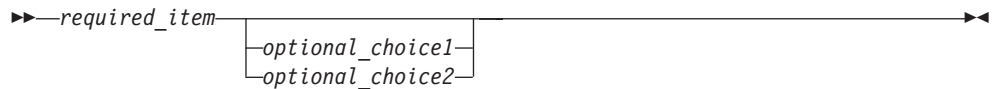
\blacktriangleright —*required_item*— $\boxed{\text{optional_item}}$ — \blacktriangleleft

- 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) 表示されず。

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。

\blacktriangleright —*required_item*— $\boxed{\text{required_choice1}}$
— $\boxed{\text{required_choice2}}$ — \blacktriangleleft

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。



ロード・マップ

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。

表 1. 情報のロードマップ

目的	参照先
DataPropagator に関する情報へアクセス する	DB2 DataPropagator については、 www.ibm.com/software/data/dpropr/
Q レプリケーション、SQL レプリケー ション、およびイベント発行について 学ぶ	「 <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」
Q レプリケーションおよびイベント発 行をセットアップして管理する	「 <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。
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/
IBM 研修サービスのクラスの空き状況 を調べる	www.ibm.com/services/learning/
旧バージョンの SQL レプリケーション からバージョン 8 の SQL レプリケー ションにマイグレーションする	SQL レプリケーション製品資料については、 www.ibm.com/software/data/dpropr/library.html
バージョン 8 のレプリケーションのチ ューニング・パフォーマンス	DB2 レプリケーションのチューニング製品資料については、 www.ibm.com/software/data/dpropr/library.html
エラー・メッセージをデバッグする	Linux、UNIX、Windows、z/OS では、メッセージ・リファレンス 第 1 巻 を参照。 OS/400 では、エラー・メッセージを受け取ったら <F1> を押す。
その他の DB2 および Information Integrator の情報を得る	製品 Web ページは www.ibm.com/software/data/

DB2 レプリケーション バージョン 8 での変更点

このセクションでは、DB2 レプリケーション・バージョン 7 以降の主な変更を要約しています。ここでは、基本バージョン 8 製品における新機能、およびそれ以降の修正パッケージでリリースされた拡張機能について紹介しています。これらの変更には、ユーザビリティの改善、パフォーマンスの向上、新機能、保守容易性の向上、レプリケーション・システム・コマンドの変更、コントロール表の変更、サポートされなくなった機能などがあります。それらについて、これから詳しく述べていきます。

- 『バージョン 8.2 での新機能』
- xviii ページの『バージョン 8.1.4 での新機能』
- xviii ページの『バージョン 8 フィックスパック 2 での新機能』
- xix ページの『バージョン 8.1 での新機能』
- xxix ページの『サポートされなくなった機能』

バージョン 8.2 での新機能

以下のレプリケーションの拡張機能がバージョン 8.2 で追加されました。

新しいレプリケーションの解決策

Q レプリケーションは、バージョン 8.2 で使用可能で、メッセージ・キュー上に複製することにより Websphere MQ の能力と柔軟性を利用するという、新しいレプリケーションの解決策を備えています。対等レプリケーションを含めて、単一方向レプリケーションから複数方向レプリケーションに至るまで、多くの種類の構成を使用できます。

イベント発行という Q レプリケーションの機能は、ソースの変更内容を XML メッセージに変換し、WebSphere MQ メッセージ・キューを使用してこのメッセージを独自のユーザー・アプリケーションやサード・パーティーのユーザー・アプリケーションに送信します。

本書は、Q レプリケーションにとって固有の問題を扱ってはいませんが、レプリケーション・アラート・モニターなどの一部の機能 (SQL レプリケーションと Q レプリケーションで共有される) について扱っています。

新機能

ソース表とターゲット表の比較: SQL レプリケーションと Q レプリケーションの両方の場合に、**asntdiff** コマンドを使用し、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します。

ソース表とターゲット表の同期化: SQL レプリケーションと Q レプリケーションの両方の場合に、**asntrep** コマンドを使用し、ソース表とターゲット表の間の違いを修復して、2 つの表を同期化します。

モニターの更新: モニター機能が拡張され、Q レプリケーションが組み込まれました。

DB2 レプリケーション・サービスのリスト: Windows の **asnslist** コマンドを使用すると、Windows の Service Control Manager (SCM) 中の SQL と Q のレプリケーション・サービスをリストできます。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。

フィックスパック 5 以降、競合検出が使用不可の場合に、Update-anywhere シナリオとレプリカ・シナリオ中に LOB 列を組み込むことに関する制約事項が除去されました。

バージョン 8.1.4 での新機能

以下のレプリケーションの拡張機能がバージョン 8.1.4 で追加されました。

新機能

DB2 for z/OS でのさらに長いオブジェクト名のサポート : DB2 UDB for z/OS バージョン 8 が新機能モードで実行されている時、レプリケーションは、z/OS で 128 バイトまでのスキーマ名および表名をサポートするようになりました。

パスワード・ファイルにある別名およびユーザー ID のリスト表示 : **asnpwd** コマンドで、パスワード・ファイルに含まれている別名およびユーザー ID をリストできます。さらに、**asnpwd** コマンドの **encrypt** パラメーターを使用して、ファイル内のすべての項目またはファイル内のパスワード項目のみのいずれかを暗号化できます。

パフォーマンスの向上

Oracle ソースでのデータの可用性の向上 : Oracle ソースでの CCD 表に対して、アプライ・プログラムが LOCK TABLE ステートメントを発行する必要はなくなりました。この改善された機能を利用するには、「IBM DB2 Information Integrator 移行ガイド: SQL レプリケーション バージョン 8 への移行」にある指示に従い、Oracle ソース用のすべての既存の登録およびサブスクリプションを移行する必要があります。

バージョン 8 フィックスパック 2 での新機能

以下のレプリケーションの拡張機能がバージョン 8 フィックスパック 2 で追加されました。

ユーザビリティの改善

パフォーマンスおよび統計データの表示 : キャプチャー、アプライ、およびモニター・プログラムに関するパフォーマンスおよび統計データを表示できます。レプリケーション・センターを使用して、データを照会し、それをファイルに保管したり印刷したりできます。

レプリケーションのスケジューリング：バージョン 8 では、レプリケーション・センターを使用して、マスターからレプリカへのレプリケーションおよびレプリカからマスターへのレプリケーションを何度もスケジュールできます。

アプライおよびモニター・プログラムにより生成されたメッセージの表示。レプリケーション・センターを使用して、アプライおよびモニター・プログラム (APPLYTRACE および MONTRACE) により生成されたメッセージを表示できます。

パフォーマンスの向上

IASP サポート：iSeries では、基本の補助記憶域プール (ASP) または独立補助記憶域プール (IASP) から使用可能なデータベースをカタログすることができます。

新機能

キャプチャー・プログラムでマルチパーティション表から変更をキャプチャーできる：DB2 Enterprise Server Edition を実行している場合、複数のパーティション表にまたがって広がるソース表への変更をキャプチャーできます。

メンバー単独でのフル・リフレッシュ：すべてのメンバーのフル・リフレッシュを実行せずに、既存のサブスクリプション・セットに 1 つ以上のメンバーを追加できます。サブスクリプション・セットの個々のメンバーを使用不可にすることもできます。

コントロール表に追加の履歴データが提供される：DB2 レプリケーションでは、レプリケーション活動を記述する追加の履歴データを、コントロール表に提供します。そのようなデータは 3 つの新しい表、すなわち、アプライ・トレース (IBMSNAP_APPLYTRACE) 表、キャプチャー・モニター (IBMSNAP_CAPMON) 表、およびモニター・トレース (IBMSNAP_MONTRACE) 表に入れます。それらのデータを照会するには、レプリケーション・センターを使用できます。

トランザクションのモードの処理がサポートされる：COMMIT_COUNT 列が IBMSNAP_SUBS_SET 表に追加され、CCD 表でのトランザクションのモードの処理がサポートされます。この列を使用して、このターゲット表のデータに変更がコミットされる頻度を制御することもできます。

コントロール表の変更

IBMSNAP_PARTITIONINFO 表が追加されました。そこには、必要とされる中で一番古いログ・シーケンス番号からキャプチャー・プログラムを再始動できるようにするための情報が含まれます。

バージョン 8.1 での新機能

ユーザビリティの改善

キャプチャー・プログラムとアプライ・プログラム間でのハンドシェーク・メカニズムが拡張された：ハンドシェークとは、レプリケーション・ソース用データのキャプチャーを開始することを、キャプチャー・プログラムへ知らせるために、アプライ・プログラムが使用するメカニズムです。バージョン 8 では、このメカニズム

が変更、拡張されました。アプライ・プログラムは、新規のシグナル (IBMSNAP_SIGNAL) 表にシグナルを挿入し、キャプチャー・プログラムがソース用データのキャプチャーを開始する必要がある場合をコントロールします。

キャプチャー・プログラムとアプライ・プログラムは、どの順にでも開始できる：バージョン 8 では、キャプチャー・プログラムをアプライ・プログラムの開始後に開始することも、逆にアプライ・プログラムをキャプチャー・プログラム開始後に開始することもできます。バージョン 7 では、キャプチャー・プログラムはアプライ・プログラムの開始前に開始する必要がありました。

キャプチャー・プログラムの実行中に、登録やサブスクリプション・セットを追加できる：キャプチャー・プログラムを再初期設定、停止、または再始動しなくても、新規のレプリケーション・ソースの登録、既存の登録の更新、新規のサブスクリプション・セットの追加、または既存のサブスクリプション・セットの更新が可能です。

各登録ごとにキャプチャーするものをコントロールできる：レプリケーションのために表を登録する際、キャプチャー・プログラムに行の変更をキャプチャーさせるのは、表のどの 列が変更されても必ず行うのか、または登録された列が変更される時のみ行うかを指定することができます。前バージョンでは、キャプチャー・プログラムで始動パラメーターを使用してキャプチャーしたものがコントロール対象で、つまり、すべての表は同一に扱われていました。バージョン 8 では、各登録ごとにキャプチャーするものをコントロールできるため、始動パラメーターは使用できません。

レプリカからのデータの再キャプチャーをコントロールできる：ソースを登録する際、いくつかの表から再キャプチャーされた変更のみを必要とし、それ以外は必要としないかどうかを指定することができます。デフォルトは以下のとおりです。

- 変更はレプリカ表から再キャプチャーされず、別のレプリカ表へ転送される。
- Update-anywhere レプリケーションでのマスター表の変更は再キャプチャーされ、レプリカ表へ送られる。

プログラムごとに 1 つの Windows サービスが作成できる：バージョン 7 では、キャプチャー・プログラムやアプライ・プログラムをすべて操作するのに、1 つの Windows サービスしか作成できませんでした。バージョン 8 では、レプリケーション・アラート・モニターの他、キャプチャー・プログラムやアプライ・プログラムごとに、別々のサービスを作成することができます。各サービスを使用して、レプリケーションを開始または停止することができます。レプリケーション・センターを使用するか、レプリケーション・プログラム用のサービスを作成 (`asnscri` コマンド) またはドロップ (`asnsdrop` コマンド) する新規のコマンドを使用することができます。

キャプチャー・プログラムおよびアプライ・プログラムに ARM がサポートされる：z/OS 環境では、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・モニターは、MVS 自動リスタート・マネージャー (ARM) を使用できます。ARM とは MVS のリカバリー機能で、特定のバッチ・ジョブや開始済みタスクの可用性を向上させることができます。ジョブまたはタスクが失敗するか、ジョブやタスクを実行しているシステムに障害が発生した場合、ARM はオペレーターの介入なしに、ジョブまたはタスクを再始動できます。ARM

は、処理対象の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、ARM を使用できるアプリケーションはそれぞれ、ARM との通信に使用するユニークなエレメント名を使用します。レプリケーション用のエレメント名として、キャプチャー・プログラムには ASNTCxxxxyyyy、アプライ・プログラムには ASNTAxxxxyyyy、レプリケーション・アラート・モニターには ASNAMxxxxyyyy を使用します。

メッセージが改善された：既存のメッセージが改善され、新規のメッセージが追加されました。解説およびユーザー応答のセクションは更新されました。

パフォーマンスの向上

レプリケーション表間の結合が減少した：バージョン 8 の場合、状態によっては、結合が削除されます。多くの場合、アプライ・プログラムは、ユーザーがコピーしたターゲット表を転送するのに、CD と UOW 表を結合する必要はありません。また、整理でも、CD と UOW 表を結合する必要はありません。

キャプチャー整理と、DB2 ログの読み取りを並行して稼働できる

(UNIX、Windows、z/OS)：キャプチャー・プログラムは、表の整理中に DB2 ログを読み取ります。したがって、整理はキャプチャー待ち時間に影響を与えません。バージョン 7 では、キャプチャー・プログラムはこれらのタスクを、並行してではなく、順に実行していました。また、バージョン 8 では、キャプチャー・プログラムは、新規シグナル (IBMSNAP_SIGNAL) 表やモニター (IBMSNAP_CAPMON) 表の他に、UOW 表、CD 表、トレース表も整理します。

ターゲット表のフル・リフレッシュがより高速に行える (UNIX、Windows、z/OS)：DB2 レプリケーションでは、ターゲット表をより高速にフル・リフレッシュできるよう、以下の DB2 製品でロード・ユーティリティーが改善されました。

- DB2 Universal Database for Windows and UNIX (バージョン 8)
- DB2 Universal Database for z/OS and OS/390 (バージョン 7 以降)

アプライ・プログラムがサブスクリプション・セットを 1 つだけ持つ場合、処理を最適化する：バージョン 8 では、単一サブスクリプション・セットに関する情報をキャッシュして再利用するように、アプライ・プログラムを開始できます。新規のキーワード **opt4one** を使用すると、CPU 使用率やスループットを改善できます。

複数メンバーを持つサブスクリプション・セットの更新回数が減少した：DB2 レプリケーションの以前のバージョンと比較すると、バージョン 8 では、複数メンバーを持つサブスクリプション・セットのコントロール表を更新する回数が少なくなりました。

新規のユーザー・インターフェース

バージョン 8 では、レプリケーション環境のセットアップや保守、管理ツールを使用したキャプチャー・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・モニターの操作が可能です。新しい DB2 レプリケーション・センターはグラフィカル・ツールで、DB2 - DB2 間レプリケーション環境の管理や、DB2 - 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の管理をサポートします。

DB2 レプリケーション・センターは、DB2 コントロール・センターのツールのセットの一部で、その他の DB2 センターと似たところがあります。レプリケーショ

ン・センターには、DB2 コントロール・センターや DB2 DataJoiner レプリケーション管理 (DJRA) ツールの、以前から使用可能なレプリケーション関数のすべてが含まれます。また、レプリケーション・センターにはランチパッドがあり、これを使用して DB2 レプリケーション環境のセットアップに必要な基本関数を実行することができます。ランチパッドは、さまざまなステップがお互いにどのように関係しているかをグラフィカルに表示します。

レプリケーション・センターを使用して、以下を行うことができます。

- コントロール表、ソース・オブジェクト、およびターゲット・オブジェクト用プロファイルのデフォルトの定義
- レプリケーション・コントロール表の作成
- レプリケーション・ソースの登録
- サブスクリプション・セットの作成と、サブスクリプション・セットへのサブスクリプション・セット・メンバーの追加
- キャプチャー・プログラムの操作
- アプライ・プログラムの操作
- レプリケーション処理のモニター
- レプリケーション用基本トラブルシューティングの実行
- LOADX オプションの指定
- アプライおよびモニター・プログラム (APPLYTRACE および MONTRACE) により生成されたメッセージの表示
- パフォーマンスおよび統計データの表示

さらに、レプリケーション・センターを使用して、その他多くのレプリケーション管理タスクを実行することができます。

新機能

複数のキャプチャー・プログラムが、同じ DB2 ログまたはジャーナルを並行して読み取ることができる：複数のキャプチャー・プログラムを単一 DB2 ログ (DB2 カタログ) またはジャーナルに対して稼働することができます。z/OS データ共有グループの場合、複数のキャプチャー・プログラムがデータ共有グループのログを読み取ることができます。各キャプチャー・プログラムは互いに独立しています。必要な場合、単一ソース表を複数のキャプチャー・プログラムへ登録することができます。したがって、待ち時間が短い表の場合、その各表に専用のキャプチャー・プログラムを持たせ、それぞれ異なるランタイム優先順位と異なるキャプチャー特性 (整理インターバルなど) を設定することができます。また、別の編成が、それぞれ独自のレプリケーション環境を、同じソース・データと異なるキャプチャー・プログラムを使用して保守することも可能です。z/OS オペレーティング・システムでは、複数のキャプチャー・プログラムを使用して、単一 DB2 サブシステム内に ASCII、EBCDIC、および UNICODE ソース表を混在させることもできます。

フェデレーテッド・データベースごとに複数の非 DB2 リレーショナル・ソースを定義できる：非 DB2 ソースを含むレプリケーション環境では、単一フェデレーテッド・データベース内に複数の非 DB2 リレーショナル・ソースを定義することができます。

自動モニター：新しいレプリケーション・アラート・モニターは、連続的に実行し、キャプチャーおよびアプライ・プログラムをモニターします。ユーザーは、モニターする基準にしきい値を定義し、そのしきい値に達するか、またはそれを越えた場合に、自動的に電子メールで知らせる先を指定します。レプリケーション・センターまたは 2 つの新規コマンド (**asnmn** と **asnmcmd**) を使用して、レプリケーション・アラート・モニターを構成、操作することができます。

オンデマンド・モニター：キャプチャー、アプライ、およびモニター・プログラムの状況を **asnccmd**、**asnacmd**、**asnmcmd** 状況コマンドを使用して照会することができます。

暗号化されたパスワード・ファイル (UNIX、Windows)：バージョン 7 では、アプライ・プログラムやレプリケーション・アナライザーが使用するパスワード・ファイルには、暗号化された情報ではなく、プレーン・テキストが含まれていました。バージョン 8 では、パスワード・ファイル内のパスワードは暗号化されています。プレーン・テキストにはパスワードは保管されません。新規のコマンド (**asnpwd**) を使用して、パスワード・ファイルを作成、保守することができます。

改善された ASNLOAD 出口ルーチン (UNIX、Windows、z/OS)：ASNLOAD 出口ルーチンは、ソース・フォーマット (C) とコンパイル済みフォーマットの両方で、サンプルの出口ルーチンとして提供されます。サンプルの出口ルーチンは DB2 プラットフォームによって異なり、そのプラットフォームで提供されるユーティリティ・オプションを利用します。サンプルのコンパイル済みプログラム出口ルーチンを提供されたまま使用したり、レプリケーション構成をカスタマイズして動作を変更したり、出口ルーチンのコード自体をカスタマイズすることができます。

コールド・スタートをより幅広くコントロールできる (UNIX、Windows、および z/OS)：**warm** 始動パラメーターは以下の 2 つのパラメーターに置き換えられ、コールド・スタートをより幅広くコントロールできるようになりました。

warmsi

ウォーム・スタート情報が使用可能な場合、キャプチャー・プログラムは前回の実行で終了したところから処理を再開します。これが初回で、キャプチャー・プログラムの開始中、または新規の再始動 (IBMSNAP_RESTART) が空の場合は、キャプチャー・プログラムはコールド・スタートに切り替えます。これは、バージョン 8 のデフォルトの始動パラメーターです。

warmsa

ウォーム・スタート情報が使用可能な場合、キャプチャー・プログラムは前回の実行で終了したところから処理を再開します。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。

アプライ・プログラムによるコミットがより頻繁になる：ユーザー・コピー、ポイント・イン・タイム、CCD、またはレプリカ・ターゲット表がサブスクリプション・セットにある場合は、アプライ・プログラムが、指定したトランザクション数を処理した後にその作業をコミットするよう指定することができます。これを行うには、アプライ・プログラムをトランザクション・モードで実行する必要があります。

参照保全性をより多くのタイプのターゲット表に：多くの場合、アプライ・プログラムがトランザクション・モードで作業をコミットするよう開始すると、ユーザー・コピーやポイント・イン・タイムのターゲット表に参照保全性を持たせることができます。

キャプチャー・プログラムの操作パラメーターを設定する方法が増えた：キャプチャー・プログラムを操作するために、提供されたデフォルトを使用することができます。また、キャプチャー・パラメーター (IBMSNAP_CAPPARMS) を使用して、ユーザーのレプリケーション環境に合うよう新しいデフォルトを作成することもできます。さらに、あるセッションではデフォルトを使用したくない場合、プログラムの開始時に、キャプチャー・プログラムの操作パラメーターを提供することもできます。キャプチャー・プログラムの実行中に、レプリケーション・センター、**asnccmd** コマンドの **chgparms** キーワード (UNIX、Windows、z/OS)、または **OVRDPRCAPA** コマンド (iSeries) を使用して、操作パラメーターを変更することができます。これらの変更は、セッションを終了するか、または別の変更コマンドを出すまで、有効です。

ターゲット・キー列への変更を複製するための新規オプション：バージョン 7 では、キー列への変更がターゲット表へ正しく複製されたことを、ソース表を登録して、更新を削除/挿入ペアとしてキャプチャーすることで確かめることができました。バージョン 8 では、サブスクリプション・セット・メンバーの定義時に、アプライ・プログラムが **WHERE** 文節をその述部に主キー列を使用して組み立てる際に、変更前イメージ値か、変更後イメージ値のいずれを使用するかを指定することができます。変更前イメージ値を使用すると、更新を挿入へ変換されることを回避できます。登録が更新に削除/挿入ペアを使用するか、サブスクリプション・セット・メンバーが **Apply WHERE** 文節の述部に変更前イメージ値を使用するかを指定することができます。

キャプチャー・プログラムで整理できる表が多くなる：キャプチャー・プログラムでは、新規のシグナル (IBMSNAP_SIGNAL) 表やモニター (IBMSNAP_CAPMON) 表の他に、CD 表、UOW 表、トレース (IBMSNAP_CAPTRACE) 表も整理します。

より長い表名や列名をサポートする：現在 DB2 レプリケーションは、ソース表やターゲット表の名前は最長 128 文字まで、列名は 30 文字まで (ロング・ネームをサポートするデータベースのため) サポートします。

キャプチャー・プログラムの実行中に、ソース表や CD 表へ列を追加できる：キャプチャー・プログラムを再初期設定したり、停止および再始動しなくても、レプリケーション・ソース表へ列を追加することができます。UNIX、Windows、および z/OS では、キャプチャー・プログラムの実行中に、CD 表を変更することもできます。

新規のシグナルでキャプチャー・プログラムをコントロールする：シグナル (IBMSNAP_SIGNAL) 表に書き込まれるシグナルによって、キャプチャー・プログラムをコントロールすることができるようになりました。シグナル表は、ログ・レコードを介してキャプチャー・プログラムと通信する方法を提供します。キャプチャー・プログラムは、以下の状態の場合に、シグナルを使用します。

- 特定の表に対する変更のキャプチャーを開始する時を決定するため。
- 終了する時を決定するため。

- Update-anywhere レプリケーションを実行する必要があるかどうか。
- アプライ・イベントに正確なエンドポイントを設定するためのログ・シーケンス番号を提供するため。

シグナル表は、アプライ・プログラムがキャプチャー・プログラムヘデータのキャプチャー開始時を知らせるようにするだけでなく、ログ・レコードの読み取りを正確に終了したり、ユーザー定義シグナルをログ・レコードに入れることもできます。

データ・リンク値の複製 (AIX、Solaris オペレーティング環境、Windows、iSeries) :

- 外部ファイルをポイントする DATALINK 値がある場合、列が RECOVERY YES で定義されているならば、統合したバージョンのファイルを検索することができます。過去のリリースでは、DB2 がファイルの最新コピーを複製しても、その複製ファイルが、複製されたデータベース・データ値と矛盾しないことを保証するものではありませんでした。
- ソース・データベース内の複数の変更について、同じターゲット・ファイルを保守することができます。
- AIX、Windows、および Solaris オペレーティング環境では、DB2 Data Links Manager レプリケーション・デーモン (DLFM_ASNCPYD) へ接続し、レプリケーション用のデータ・リンク・ファイルを検索したり、保管することができます。以前のリリースのように、別の ASNDLCOPYD デーモンを開始および保守する必要はありません。OS/400 では、まだ、別の ASNDLCOPYD デーモンを開始および保守する必要があります。

ユニコード・エンコード方式が追加された (z/OS) : DB2 DataPropagator for z/OS バージョン 8 は、UNICODE および ASCII エンコード方式をサポートします。この機能は DB2 DataPropagator for OS/390 バージョン 7 で登場しました。

64 ビット・サポートが追加された (Windows、UNIX、z/OS) : バージョン 8 では、DB2 が 64 ビット・サポートを提供するオペレーティング・システムで複製することができます。アプリケーションを 64 ビットのオペレーティング・システムで稼働すれば、これらのシステムが提供するメモリー・アドレス・スペースが増大したことによる利益を得ることができます。

移行ユーティリティー : 新規のレプリケーション移行ユーティリティー (**asnmig8**) は、バージョン 5、バージョン 6、またはバージョン 7 のレプリケーション表をすべて、バージョン 8 のフォーマットに変換するために使用できる、移行スクリプトのセットで構成されています。

保守容易性の向上

新規のトレース機能 (UNIX、Windows、z/OS) : 新規のレプリケーション・トレース機能 (**asntrc**) は、DB2 トレース機能と似ています。キャプチャーおよびアプライ・プログラムを停止、再始動しなくても、トレース機能を開始または停止することができます。さらに、トレース出力は簡潔で、通常、以前のリリースで生成されていたトレース・ファイルよりも小さくなり、DB2 トレース・フォーマットと整合しています。

レプリケーション・アナライザー・プログラムが更新された：新規の V8 機能を分析するために、レプリケーション・アナライザー・プログラムは変更されました。アナライザーは、特定のシステムにあるレプリケーション・コントロール表の状態について、レポートを生成します。これらのレポートは、ユーザーのレプリケーション環境の検査および調整、または問題の診断に使用することができます。アナライザーおよびそのドキュメンテーションは、Web からダウンロードすることができます。

新規および更新されたエラー・メッセージ：新機能に対する新規エラー・メッセージが追加されました。既存のメッセージは、読み易さを改善するために更新されました。

レプリケーション・システム・コマンドの変更

新規および変更されたレプリケーション・システム・コマンド

(UNIX、Windows、z/OS)：Windows、UNIX、および z/OS の既存のシステム・コマンド構文が変更されました。以下の変更も行われました。

- キャプチャー・コマンド行 (**asncmd**) は、**asnccmd** へ名前が変更になりました。これは、アプライ・プログラムを操作するために使用する、新規のアプライ・コマンド行 (**asnacmd**) や、モニター・プログラムを操作するために使用する、新規のモニター・コマンド行 (**asnmcmd**) と整合した名前になりました。
- キャプチャー・プログラムを開始する **asnccp** コマンドの名前が、**asnccap** へ変更されました。

UNIX、Windows、および z/OS オペレーティング・システムで稼働する、以下の新規システム・コマンドが追加されました。

- **asnacmd** (アプライ・コマンド行) は、アプライ・プログラムを操作および停止します。
- **asnmon** (モニター・コマンド) は、レプリケーション・アラート・モニターを開始します。
- **asnmcmd** (モニター・コマンド行) は、レプリケーション・アラート・モニターを操作および停止します。
- **asnanalyze** (アナライザー・コマンド) は、レプリケーション・コントロール表の状態についてのレポートを生成します。
- **asnpwd** (パスワード・コマンド) は、分散レプリケーション環境に必要なパスワード・ファイルを作成および保守します。
- **asntrc** (トレース機能) は、キャプチャーおよびアプライ・プログラム用トレースを生成するために、始動オプションを置き換えます。

新規および変更されたレプリケーション・システム・コマンド (OS/400 オペレーティング・システム (iSeries))：OS/400 システムで稼働する、以下の新規システム・コマンドが追加されました。

- **ADDDPRREG** (DPR 登録の追加) は、ユーザー表をレプリケーション用に登録します。
- **RMVDPREG** (DPR 登録の除去) は、レプリケーションで使用できるソース表のリストから、ユーザー表を除去します。

- **ADDDPRSUB** (DPR サブスクリプション・セットの追加) は、空のサブスクリプション・セット、またはメンバーが 1 つだけのサブスクリプション・セットを作成します。
- **RMVDPRSUB** (DPR サブスクリプション・セットの除去) は、空のセットか、またはセットの全メンバーを除去します。
- **ADDDPRSUBM** (DPR サブスクリプション・セット・メンバーの追加) は、メンバーを既存のサブスクリプション・セットへ追加します。
- **RMVDPRSUBM** (DPR サブスクリプション・セット・メンバーの除去) は、サブスクリプション・セットからサブスクリプション・セット・メンバーを 1 つ除去します。
- **OVRDPRCAPA** (DPR キャプチャー属性のオーバーライド) は、現在実行中のキャプチャー・プログラムの属性を変更します。
- **ANZDPR** (アナライザー) は、特定のシステムにあるレプリケーション・コントロール表の状態についてのレポートを生成します。これらのレポートは、ユーザーのレプリケーション環境の検査および調整、または問題の診断に使用することができます。
- **WRKDPRTRC** (トレース・オプション) は、Dump などの各種のトレース・オプションを操作します。

変更された既存のシステム・コマンド (OS/400 システム)

- **DPRVSN** (DataPropagator バージョン) パラメーターは、すべてのシステム・コマンドから取り除かれました。
- **CAPCTLLIB** (キャプチャー・コントロール・ライブラリー) パラメーターが、Capture コマンドに追加されました。
- 新規のトレース関数やモニター機能を利用するために、新規のパラメーターが、**CHGDPRCAPA** (DPR キャプチャー属性の変更) および **STRDPRCAP** (DPR キャプチャーの開始) コマンドに追加されました。
- スペースの再利用のために CD 表や UOW 表を自動的に再編成するよう、新規のパラメーターが **ENDDPRCAP** (DPR キャプチャーの終了) コマンドに追加されました。
- アプライ・プログラムを 1 度だけ実行し、アプライ・トレール (IBMSNAP_APPLYTRAIL) 表をクリーンアップし、単一サブスクリプション・セットの処理を最適化するための、新規のパラメーターが **STRDPRAPY** (DPR アプライの開始) コマンドに追加されました。

コントロール表の変更

新機能のサポートとユーザビリティの改善のため、バージョン 8 では、コントロール表へ重要な変更が行われました。新規の表が追加され、それによって、必要でなくなった表もあります。また、変更された既存の表もいくつかあります。

以下の新規の表が追加された：

- **IBMSNAP_APPENQ** は、単一アプライ修飾子に対して 1 つのアプライ・プログラムだけが実行されることを保証するものです。
- **IBMSNAP_APPLYTRACE** は、アプライ・プログラムからの重要なメッセージを保持します。

- IBMSNAP_APPPARMS には、アプライ・プログラムの動作を制御するために変更できるパラメーターが含まれています。
- IBMSNAP_CAPENQ は、単一キャプチャー・スキーマに対して 1 つのキャプチャー・プログラムだけが実行されることを保証するものです。
- IBMSNAP_CAPMON は、キャプチャー・プログラムの進行状況をモニターするための操作統計を保持します。
- IBMSNAP_CAPSCHEMAS は、すべてのキャプチャー・スキーマの名前を保持します。
- IBMSNAP_PRUNE_SET は、CD 表の整理を調整します。
- IBMSNAP_RESTART は、キャプチャー・プログラムがログやジャーナル内の正しい時点からキャプチャーを再開できるようにします。
- IBMSNAP_SIGNAL は、キャプチャー・プログラムをコントロールするために使用するシグナルを保持します。

レプリケーション・アラート・モニター用に、以下の新規の表が追加された：

- IBMSNAP_ALERTS 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履歴を保持します。
- IBMSNAP_CONDITIONS は、モニターされる各サーバーのアラート条件を保持します。
- IBMSNAP_CONTACTGRP はグループとの連絡先をマップします。
- IBMSNAP_CONTACTS は、連絡先名とアドレスを保持します。
- IBMSNAP_GROUPS は、連絡先グループを保持します。
- IBMSNAP_MONENQ は、単一モニター修飾子に対して 1 つのモニター処理だけが実行されることを保証するものです。
- IBMSNAP_MONPARMS には、レプリケーション・アラート・モニター・プログラムの動作を制御するために変更できるパラメーターが含まれています。
- IBMSNAP_MONSERVICES 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・サーバーまたはアプライ・コントロール・サーバーを最後にモニターした時刻を保持します。
- IBMSNAP_MONTRACE は、レプリケーション・アラート・モニターの活動をトレースします。
- IBMSNAP_MONTRAIL は、モニター・サイクルごとにモニターの活動の履歴を保持します。

以下の表が変更された：

- IBMSNAP_APPLYTRAIL
- IBMSNAP_AUTHTKN (OS/400 のみ)
- IBMSNAP_CAPPARMS (以前は IBMSNAP_CCPPARMS)
- IBMSNAP_CAPTRACE (以前は IBMSNAP_TRACE)
- IBMSNAP_PRUNCNTL
- IBMSNAP_REG_EXT (OS/400 のみ)
- IBMSNAP_REGISTER
- IBMSNAP_SUBS_COLS

- IBMSNAP_SUBS_EVENT
- IBMSNAP_SUBS_MEMBR
- IBMSNAP_SUBS_SET
- IBMSNAP_UOW

CD 表も変更されました。

DB2 レプリケーションの以前のバージョンの表で、以下のものは現在は使用されません：

- IBMSNAP_CRITSEC は IBMSNAP_SIGNAL に置き換えられました。
- IBMSNAP_WARMSTART は IBMSNAP_RESTART に置き換えられました。

サポートされなくなった機能

Windows System Services は、DB2 レプリケーション・センターではサポートされなくなりました。Windows コマンドと Windows サービスのダイアログは、両方とも Windows サービスにアクセス操作できます。

DB2 DataJoiner レプリケーション管理 (DJRA) ツールは、バージョン 8 ではサポートされません。したがって、DJRA を使用して、バージョン 8 のレプリケーション・コントロール表を作成したり、バージョン 8 のコントロール表を使用する、ソースの登録や、サブスクリプション・セットの定義はできません。バージョン 7 のレプリケーション環境では、DJRA は引き続きサポートされます。バージョン 8 のレプリケーション環境用レプリケーション・センターを使用してください。

DB2 コントロール・センターは、バージョン 8 のレプリケーション・コントロール表をサポートしません。したがって、コントロール・センターを使用して、バージョン 8 のコントロール表を使用する、ソースの登録や、サブスクリプション・セットの定義はできません。バージョン 7 のレプリケーション環境用コントロール・センターは使用できます。バージョン 8 のレプリケーション環境用レプリケーション・センターを使用してください。

ASNSAT コマンドは、もう使用できません。また、レプリケーション・サブスクリプションの汎用化や、DB2 サテライト・レプリケーション環境のセットアップは、サテライト管理センターから行えなくなりました。モバイル作業フォースのためにデータをレプリケーションする必要がある場合は、サテライト DB2 データベースを、DB2 Everyplace バージョン 8 へ移行することをお勧めします。詳細は、IBM 担当者へお尋ねください。

第 1 部 レプリケーションのガイド

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 SQL レプリケーションの計画』では、レプリケーション環境を計画する方法について説明します。

17 ページの『第 2 章 SQL レプリケーション用のサーバーの構成』では、レプリケーション用に環境を設定する方法について説明します。

43 ページの『第 3 章 表およびビューを SQL レプリケーション・ソースとして登録する』では、レプリケーション・ソースを登録するために必要な情報について説明します。

69 ページの『第 4 章 SQL レプリケーションのソースのサブスクリプション』では、サブスクリプション・セットの作成方法と、メンバーをサブスクリプション・セットへ追加する方法について説明します。

105 ページの『第 5 章 SQL レプリケーションでの特殊なデータ・タイプのレプリケーション』では、ソース表の LOB 値および DATALINK 値のためのレプリケーション・オプションについて説明します。

115 ページの『第 6 章 SQL レプリケーション環境におけるデータのサブセット化』では、どのようにデータがターゲットへアプライされるのかを説明する他、ターゲットへキャプチャーおよびアプライされるデータをカスタマイズする方法についても説明します。

119 ページの『第 7 章 SQL レプリケーション環境におけるデータ操作』では、ソース・データを操作するための、キャプチャー・プログラムまたはアプライ・プログラムの使用方法について説明します。

123 ページの『第 8 章 SQL レプリケーション用のレプリケーション SQL スクリプトのカスタマイズおよび実行』では、レプリケーション環境で SQL を実行する方法について説明します。

125 ページの『第 9 章 SQL レプリケーションに関するキャプチャー・プログラムの操作』では、すべてのオペレーティング・システム環境について、キャプチャー・プログラムを操作する方法を説明します。

147 ページの『第 10 章 SQL レプリケーションに関するアプライ・プログラムの操作』では、すべてのオペレーティング・システム環境について、アプライ・プログラムを操作する方法を説明します。

173 ページの『第 11 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、レプリケーション・アラート・モニターを使用してレプリケーション環境をモニターする方法について説明します。

| 199 ページの『第 12 章 SQL レプリケーションの要求時報告』では、要求時にレ
| プリケーション環境に関するレポートを生成して表示する方法について説明しま
| す。

209 ページの『第 13 章 SQL レプリケーション環境の変更』では、レプリケーシ
ョン環境を変更する方法について説明します。

247 ページの『第 14 章 SQL レプリケーション環境の保守』では、ソース表、コ
ントロール表、およびターゲット表を保守する方法について説明します。

第 1 章 SQL レプリケーションの計画

この章では、レプリケーション環境を計画する方法を説明しています。この章には、以下のセクションがあります。

- 『移行計画』
- 『メモリーの計画』
- 6 ページの『ストレージの計画』
- 11 ページの『競合検出の計画』
- 12 ページの『非 DB2 リレーショナル・ソースの計画』
- 13 ページの『コード・ページ・トランザクションの計画』
- 15 ページの『DB2 UDB for z/OS のレプリケーションの計画』
- 16 ページの『パフォーマンス・チューニング』

移行計画

既存のレプリケーション環境から移行している場合、特定の移行に関連する問題を考慮する必要があります。「*Migration Guide: Migrating to DB2 Replication*」は、既存の DB2 レプリケーション環境からバージョン 8 レプリケーションに移行する方法を説明しています。また、現在 DB2 DataJoiner[®] を使ってデータを複製しているレプリケーション環境を、非 DB2 リレーショナル・サーバーに、またはその逆に移行する方法も説明します。このマニュアルは、
www.ibm.com/software/data/dpropr/library.html でオンラインで参照できます。

メモリーの計画

DB2 レプリケーションに必要なメモリーの量を計画する必要があります。DB2 レプリケーションでは、必要なメモリーだけしか使用されません。必要なメモリー量は、ソースから複製されているデータの量とトランザクションの並行性に直接比例します。基本的に、複製されているデータの量が多ければ多いほど、また、同時実行されているトランザクションの数が多ければ多いほど、より多くのメモリーがレプリケーションに必要となります。

キャプチャー・プログラムとアプライ・プログラムを実行すると、かなり大量のメモリー・リソースが消費される可能性があります。

キャプチャー・プログラムによって使用されるメモリー

キャプチャー・プログラムが DB2 ログを読み取る際には、関連したコミット・レコードまたはアボート・レコードを読み取るまで、キャプチャー・プログラムは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。アボ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 CD 表および UOW 表に書き込まれます。コミットされたトランザクションは、キャプチャー・プログラムがコミット・インターバルに達して作業をコミットするまでメモリー内に留まります。

キャプチャー・プログラムが使用しているメモリー量をモニターするには、キャプチャー・モニター (IBMSNAP_CAPMON) 表の CURRENT_MEMORY 列の値を調べてください。

キャプチャー・プログラムの始動時に **memory_limit** パラメーターを設定すると、トランザクションと関連付けられているストレージとして、キャプチャー・プログラムが確実に指定された量のメモリーを使用するようにできます。このパラメーターによってその他のストレージ使用が限定されることはありません。また、**memory_limit** パラメーターは、キャプチャー・プログラムの実行中でも変更できます。メモリー限度に達すると、キャプチャーは一部のトランザクションを予備ファイルに書き込みます。予備ファイルのストレージ要件については、10 ページの『キャプチャー・プログラム用予備ファイルのスペース所要量の計画』を参照してください。キャプチャー・プログラムによって使用されるメモリー・リソースについては、このプログラムのストレージ・スペース要件と関連付けて検討する必要があります。

また、キャプチャー・プログラムのメモリー要件について計画する際には、ユーザー・トランザクションのサイズとコミット・インターバルも検討する必要があります。キャプチャー・プログラムの実行時にバッチ・ジョブを一時コミットなしで長時間実行すると、大量のメモリーが消費されます。一般に、コミット・インターバルの値が小さければ小さいほど、キャプチャー・プログラムが必要とするメモリーは少なくなります。

登録に関する情報の読み取り: アクティブ登録に関する情報は、キャプチャー・プログラムを開始するとき、およびキャプチャー・プログラムの実行中に登録が動的に追加されるときに、読み取られてメモリーに保管されます。

ログ・レコードの読み取り (Linux、UNIX、Windows、z/OS): DB2 レプリケーションは、ログ・レコードを読み取る際にメモリー・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 4 KB (K バイト) ページ x 50 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 1 KB ページ x 66 で、これは ECSA (拡張共通サービス域) ストレージです。レプリケーションが ECSA を使用するのはこの場合だけです。

OS/400 上で使用されるメモリー: CURRENT_MEMORY は、アクティブ CD 表のために標準入出力バッファーによって使用されるメモリーに入らないトランザクション・レコードを保留するために割り振られた追加のメモリーの最新のアカウントです。これは、多数のトランザクションを保留するために使用されている追加のメモリーの量を示す指標です。これは、特定のジャーナル・ジョブによって使用されているメモリーすべての正確な合計ではありません。

キャプチャー・モニター (IBMSNAP_CAPMON) 表に保管されている情報により、メモリーの使用を調整するのに役立つ稼働統計が提供されます。この表内の値は特定のキャプチャー・モニター・インターバルに関するものであり、各モニター・インターバルにわたった累積的な値ではないことに注意してください。

CURRENT_MEMORY 列のデータには、加算されるカウントは含まれません。レコードの作成時に、この値にはモニター・インターバルが終了した時点で使用中であったメモリーが反映されます。キャプチャー・モニター・インターバルは、キャプ

チャー・プログラムがこの表にデータを挿入する頻度を決定します。以下の方法のいずれかを使用して、キャプチャー・プログラムによって使用されているメモリーの量を調整してください。

予備の分も考慮したメモリー限度のチューニング:

1. キャプチャー・プログラムの始動時に、デフォルトのメモリー限度を使用する。
2. キャプチャー・モニター (IBMSNAP_CAPMON) 表の TRANS_SPILLED 列を調べて、データがメモリーから一時ファイルにあふれ出たかどうかをチェックする。この列は、メモリー制限のために特定のキャプチャー・モニター・インターバルの間にディスクにあふれたソース・システム・トランザクションの数を示しています。
3. データがメモリーからあふれた場合は、引き上げたメモリー限度または値を低くしたコミット・インターバルのいずれかを使用する。

あふれるのを防ぐためのメモリー限度のチューニング:

1. キャプチャー・プログラムの始動時に、メモリー限度を高く設定する (どのくらい高くするのは、システム・リソースに応じて決まります)。
2. キャプチャー・モニター (IBMSNAP_CAPMON) 表の CURRENT_MEMORY 列を調べて、使用中のメモリー量をチェックする。この列は、キャプチャー・プログラムが特定のキャプチャー・モニター・インターバルの間に使用したメモリーの量を (バイト単位で) 示しています。
3. メモリー限度として指定した量よりもかなり少ないメモリーしか使用されていない場合は、メモリー限度としてもう少し下げた値を設定する。

アプライ・プログラムによって使用されるメモリー

アプライ・プログラムがデータを取り出す場合、通常個々の行を取り出すために使用されるメモリーは少量です。使用されるメモリー量は、表の列のサイズおよび一度に取り出される行の数に比例します。たとえば、アプライ・プログラムが LOB 列を取り出す場合は、このプログラムは 2 G バイトのメモリーを潜在的に使用する可能性があります。

アクティブ・サブスクリプション・セットに関する情報は、アプライ・プログラムの実行中に読み取られてメモリーに保管されます。通常、アプライ・プログラムによって一度に使用されるメモリー量は、ほとんどのメンバーが含まれているサブスクリプション・セットを処理するのに必要なメモリー量に比例します。

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

メモリーは、定義を保管するため、および通知として送信する前にアラートをメモリー内に保持するために使用されます。定義用に必要なメモリーの量は、定義の数に直接比例します。レプリケーション・アラート・モニターは、アラート通知の保管用として 32 KB のメモリーを予約します。それより多いメモリーが、必要に応じて要求されたり、不要になった場合にはリリースされたりします。

ストレージの計画

DB2 が必要とするストレージに加えて、以下の項目のレプリケーションのためにストレージが確実に使用可能であるようにしておく必要があります。

データベース・ログおよびジャーナル・データ

データのレプリケーションをサポートするために記録される追加データ。詳細については、『ログの影響の計画』を参照してください。

ターゲット表およびコントロール表

複製されるデータおよびコントロール表 (CD 表を含む)。詳細については、8 ページの『ターゲット表およびコントロール表のストレージ要件の計画』を参照してください。

一時ファイル

レプリケーション・プログラムによって予備ファイルに保管されるデータ、および診断ログ・ファイル (たとえば、*CAP.log および *APP.log)。詳細については、9 ページの『一時ファイルのストレージ要件の計画』を参照してください。

OS/400: キャプチャー用の現行レシーバー・サイズ

これからキャプチャーされることになっている登録済みソース表の場合は、ジャーナル項目はまだレシーバーの現行チェーン内に残っていなければなりません。詳しくは、40 ページの『ジャーナル・レシーバー削除出口ルーチンの使用』を参照してください。

以下の部分で指定されているサイズはすべて目安にすぎません。実動可能システムを準備して設計するには、障害防止などの要因も考慮に入れなければなりません。たとえば、ネットワークが停止する可能性を考慮に入れて、データを保留する期間 (8 ページの『ターゲット表およびコントロール表のストレージ要件の計画』で説明されています) を延長しなければならない場合があります。

ヒント: ストレージ見積もりが不当に高いと思われる場合は、アプライ・プログラムがサブスクリプション・セットを実行する頻度とレプリケーション表の整理の頻度を再検討してください。ストレージの使用量、最大限の障害許容度、および CPU オーバーヘッドの間のトレードオフについて検討する必要があります。

ログの影響の計画

レプリケーション・サーバーに対するログの影響を計画する必要があります。DB2 レプリケーションでは、ソース表とターゲット表の両方をログに記録する (ジャーナリングする) 必要があります。

DB2 ソース・サーバーの場合のログの影響の計画

一般に、レプリケーションに関係するすべての表のために、現行ログの 3 倍の量のログが追加が必要となります。基本的に、ソース表用、CD 表用、およびレプリケーション・コントロール表用のログ・スペースが必要です。このセクションでは、使用しているレプリケーション環境で予測可能なログの影響をより正確に見積もるのに役立つ可能性のあるその他の要因を示します。

アプリケーションによってソース・データベースに加えられる更新と、レプリケーション要件を検討してください。たとえば、更新アプリケーションが一般に表の列

の 60% を更新する場合、レプリケーション要件によっては、ログ・レコードの大きさが、類似の表が複製されない場合に比べて半分以上も増加する可能性があります。

Linux、UNIX、Windows、および z/OS:

- DB2 は UPDATE ステートメントごとに完全な行イメージをログに記録します。これが発生するのは、DATA CAPTURE CHANGES キーワードを使用して表を作成 (または変更) してからでなければ表をレプリケーションできないからです。
- ログへの追加量が最大になるレプリケーション要件の 1 つとして、変更前イメージと変更後イメージのキャプチャー (Update-anywhere レプリケーション・シナリオにおけるレプリケーション・ターゲット表の例が当てはまる) が挙げられます。ログの量を少なくする方法の 1 つとして、レプリケーション・ソースに対して定義されている列数を少なくすることができます。たとえば、必要でない場合は変更前イメージをキャプチャーしないでください。

OS/400:

- DB2 は UPDATE ステートメントごとに完全な行イメージをログに記録します。ログの量を少なくする方法の 1 つは、レプリケーション・ソースに対して定義されている列数を少なくすることであり、たとえば、必要でない場合は変更前イメージをキャプチャーしないことです。
- CD 表用および UOW 表用として使用されるストレージの量を最小化するには、これらの表を頻繁に再編成する。なぜなら、整理ではユーザーのために DASD を回復するようなことはないからです。ENDDPRCAP コマンドでキーワード RGZCTLTLBL (コントロール表の再編成) を使用すれば、コントロール表を再編成できます。通常の操作条件で DASD の使用パターンを監視して、DASD 使用の予測と管理に役立ててください。ジャーナリングがオンになっている場合は、UOW 表および CD 表に対する DB2 ログの挿入および削除が行われるにつれてログまたはジャーナルの量が増えることも考慮に入れてください。
- 現行レシーバーがいっぱいになると、システムは新しいレシーバーに切り替える。複製する必要がなくなった古いレシーバーについては、保管するか削除するかをオプションで指定できます。たくさんのトランザクションを扱うシステムの場合、キャプチャー・プログラムは場合によって遅れることがあります。キャプチャーが頻繁に遅れる場合は、ソース表を複数のジャーナルに分離して、ワークロードをキャプチャー・プログラムの複数インスタンスに分散できます。

ターゲット・サーバーの場合のログの影響の計画

ソース・データベースのロギングに加えて、行が適用されるターゲット・データベースでもロギングされます。ログへの影響は、アプライ・プログラム用に選択したコミット・モードによって異なります。

表モード

表モード処理では、アプライ・プログラムはフェッチしたデータがすべて適用された後で単一コミットを発行します。アプライ・プログラムは一時チェックポイントを発行しません。このケースでは、アプライ・プログラムが 1 回のインターバルで処理する最大データ量を見積もり、そのデータ量を格納できるようにログ・スペースを調整する必要があります。

トランザクション・モード

トランザクション・モード処理では、アプライ・プログラムはそれぞれの更

新をソース・トランザクションの順序どおりにすべてターゲット表にコピーし、それらの変更を 1 回のインターバルでトランザクション境界に達したときにまとめてコミットします。一時コミットのインターバルを設定するには、サブスクリプション・セット・オプション (**commit_count(x)**) の値 x を設定します。すべての応答セットを取り出すと、アプライ・プログラムは予備ファイルの内容をコミット・シーケンスの順序に従って適用します。このタイプの処理では、すべての予備ファイルを同時に開いて処理できます。たとえば、コミット・カウントを 1 に設定と、アプライ・プログラムは 1 つのトランザクションが終わるたびにコミットします。コミット・カウントを 2 に設定した場合は、アプライ・プログラムは 2 つのトランザクションのセットが 1 つ終わるたびにコミットします。

OS/400: ターゲット・オペレーティング・システムが OS/400 である場合は、ターゲット表のログ・スペース (ジャーナル・レシーバー・スペース) も検討する必要があります。OS/400 では、ターゲット表用のジャーナル・レシーバーを **MNGRCV(*SYSTEM)** および **DLTRCV(*YES)** パラメーターで作成でき、さらにジャーナリングする必要があるのは変更後イメージ列だけであるため、以下の公式を使用して、ターゲット表用ジャーナル・レシーバーの大きさを見積もってください。

`journal_receiver_volume=target_table_row_length X journal_receiver_threshold`

ターゲット表およびコントロール表のストレージ要件の計画

新しいターゲット表の大きさを見積もる必要があります。ターゲット表に必要なスペースはソース表のスペース以下であるのが普通ですが、ターゲット表が非正規化されたり、その中に変更前イメージ (変更後イメージに加えて) または履歴データが含まれていると、ソース表のスペースよりもずっと大きくなる場合があります。ターゲット表のサイズは、複製する際に何をを選択するかによって異なります。選択するものとしては、たとえば、複製するソース表のパーセント、複製する列のデータ・タイプ、変更前および変更後のイメージを複製するかどうか、算出列を追加するかどうか、行をサブセット化するかどうか、レプリケーション時になんらかのトランスフォーメーションを行うかどうかなどが挙げられます。

CD 表および一部のレプリケーション・コントロール表 (**IBMSNAP_UOW**、**IBMSNAP_CAPTRACE**、**IBMSNAP_APPLYTRACE**、**IBMSNAP_APPLYTRAIL**、**IBMSNAP_CAPMON**、**IBMSNAP_ALERTS**) も、DB2 ソース・データベース用として必要なディスク・スペースに影響を与えます。レプリケーション環境のセットアップの仕方によっては、これらの表が非常に大きくなる可能性があります。上記以外のレプリケーション・コントロール表の場合は、必要なスペースは通常少なく、かつ静的です。

CD 表のサイズは、ソース表に変更が加えられるたびに大きくなり、やがてキャプチャー・プログラムによって整理されます。CD 表に必要なスペースを見積もるには、データをどのくらい保持してから整理するのかについて、その期間を最初に決定した後、キャプチャー・プログラムがこれらの表を自動的に整理する頻度、またはユーザー自身がコマンドを使用してこれらの表を整理する頻度を指定します。

レプリケーション済みデータのバイト数を計算する場合は、キャプチャー・プログラムが CD 表に追加したオーバーヘッド・データの分として 21 バイトをそれぞれの行ごとに組み込む必要があります。キャプチャー・プログラムが CD 表にデータ

をキャプチャーし続けられる期間 (たとえばネットワークが停止するなどしてデータが適用不能なときも含む) を決定してください。上記の偶発的な障害が発生した期間内にソース表で通常キャプチャーされる可能性のある挿入、更新、および削除の数を見積もってください。

CD 表の推奨サイズを判別するために、次のガイドラインを使用してください。

```
recommended_CD_size =  
  ( (21 bytes) + sum(length of all registered columns) ) X  
  (number of inserts, updates, and deletes to source table  
  during the contingency period)
```

例: CD 表の行の長さが 100 バイト (これにオーバーヘッドの分の 21 バイトを加える) で、24 時間にわたる偶発的障害の期間内に 100,000 個の更新がキャプチャーされるとすると、CD 表に必要なストレージは約 12 MB となります。

この公式では、登録済み列には変更前イメージと変更後イメージの両方が含まれるものとしています。更新が INSERT 操作と DELETE 操作の対に変換されるようになっていた場合は、挿入、更新、および削除の合計数を判別するときにそれらを勘定に入れてください。たとえば、ソース表に対する 1 つの更新は、それぞれ CD 表の 2 つの行としてカウントしてください。

UOW 表は、特定のコミット・インターバルの間にキャプチャー・プログラムによって挿入される行数、および整理される行数に基づいて、拡大および縮小します。アプリケーション・トランザクションが COMMIT を発行するたび、およびトランザクションが登録済みのレプリケーション・ソース表に対して INSERT、DELETE、または UPDATE 操作を実行するたびに、行が 1 つ UOW 表に挿入されます。当初はこの表に必要なスペースを多めに見積もっておき、実際に使用されるスペースをモニターして、回収できるスペースがあるかどうかを判別してください。

一時ファイルのストレージ要件の計画

予備ファイルおよび診断ログ・ファイルのストレージ要件を計画する必要があります。

診断ログ・ファイルのスペース所要量の計画 (Linux、UNIX、Windows、z/OS)

診断ログ・ファイルには、レプリケーション・プログラムのアクティビティに関する情報 (プログラムの始動時刻と停止時刻、およびプログラムが発行したその他の情報メッセージまたはエラー・メッセージなど) が保管されます。デフォルトでは、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、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・プログラムのそれぞれの始動時に始動パラメーター **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・ログを再利用するオプションがあります。これを使用すると、プログラムは始動するたびにログを削除して再作成します。プログラムの始動時に、ログを再利用するかどうかを指定できます。

キャプチャー・プログラム用予備ファイルのスペース所要量の計画

十分なメモリーがない場合、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 (つまり、メモリーに入りきらないトランザクションをここに入れます)。キャプチャー・プログラムは、最大のトランザクションを書き込みます。しかし、最大のトランザクションがメモリー限度を超える原因となったトランザクションであるとは限りません。

- **Linux、UNIX、Windows:** Linux、UNIX および Windows では、予備ファイルは常にディスク上にある。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。
- **OS/400:** OS/400 では、予備ファイルはライブラリー QTEMP の中に作成される。予備ファイルが必要な各登録ごとに 1 つの予備ファイルが作成されます。
- **z/OS:** z/OS では、予備ファイルは仮想入出力 (VIO) に送られる。

キャプチャー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

memory_limit 稼働パラメーターを使用して、キャプチャー・プログラムが使用できるメモリー量を指定します。許可するメモリーが多ければ多いほど、キャプチャー・プログラムが入りきらない分をファイルに入れる可能性が少なくなります。

トランザクションのサイズ

トランザクションのサイズが大きいと、入りきらないためにファイルに入れる必要性が高まります。

同時実行されるトランザクションの数

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増えた場合、またはキャプチャー・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、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。

コミット・インターバル

通常、コミット・インターバルの値が小さければ小さいほど、必要とされるストレージは少なくなります。なぜなら、キャプチャーが情報をコミットする前にその情報をメモリーに保管しておく期間が短くなるからです。

アプライ・プログラム用予備ファイルのスペース所要量の計画

アプライ・プログラムには、データを保管するための一時的なスペースが必要です。ASNLOAD ユーティリティーを使用する場合は、ロード予備ファイルではなくロード入力ファイルになります。アプライ・プログラムは、更新をターゲット表に適用するまでの間それらの更新を保留するために予備ファイルを使用します。一般に、予備ファイルはディスク・ファイルです。ただし、z/OS オペレーティング・システムでは、入りきらないデータをメモリーに入れるように指定できます。仮想メモリーの制約がない限り、予備ファイルはディスク上ではなく仮想メモリーに保管してください。

予備ファイルのサイズは、各レプリケーション・インターバル内で複製するよう選択されたデータのサイズに比例します。予備ファイルのサイズは、通常はこのデータのサイズのおよそ 2 倍です。予備ファイルのサイズを見積もるには、アプライ・

プログラム用に計画した頻度インターバル (またはデータ・ブロック値) と、同じ期間内 (または変更のピーク時) になされた変更の量を比較します。

OS/400 では、予備ファイルの行サイズは定数で 32 KB です。

Linux、UNIX、Windows、および z/OS では、予備ファイルの行サイズはターゲット行のサイズであり、これにはあらゆるレプリケーション・オーバーヘッド列が含まれます。この行サイズは、DB2 パック内部フォーマットでのサイズではなく、拡張された解釈済みの文字フォーマット (SELECT から取り出された状態) でのサイズです。行には、行の長さ、個々の列ストリング上の NULL 終止符も含まれます。以下の例では、複製するよう選択されたデータに必要な予備ファイルのサイズを見積もります。この例では、その予備ファイルに保管される他のデータに必要な追加のスペースは勘定に入っていません。

例: 変更量のピークが時間あたり 12,000 件の場合に、アプライ・プログラムの頻度が 1 時間インターバルで計画されていると、予備ファイルには 1 時間分の更新件数、つまり 12,000 件を保留しなければなりません。1 件の更新が 100 バイトのデータを表すとすれば、予備ファイルは最小限でも約 1.2 MB になります。これに加え、この予備ファイルに保管される他のデータ用のスペースも必要です。

競合検出の計画

標準または拡張競合検出を使用している場合は、レプリカ・ターゲット表用の CD (または CCD) 表に変更前イメージを保管する必要があります。また、参照保全規則に制限があります。対等シナリオおよび Update-anywhere シナリオの場合、またはアプライ・プログラムがトランザクション・モード処理を使用する場合は、ソース規則に合わせた参照保全規則を定義する必要があります。

対等シナリオまたは Update-anywhere シナリオを使用し、かつ競合検出をオンにしない場合は、更新競合が防止されるようなアプリケーション環境を設計する必要があります。使用するアプリケーション環境では競合が起こり得ないのであれば、競合検出を使用しないことで処理サイクルを節約できます。

以下の方法のいずれかを使用して、対等レプリケーションおよび Update-anywhere レプリケーションでの競合を防止してください。

キー別のフラグメント化

レプリケーション・ソースが、特定のサイトのキー範囲別にレプリカによって更新されるようにアプリケーションを設計します。たとえば、ニューヨークにあるサイトでは、米国東部のセールス・レコード (キー範囲として 49999 以下の ZIP コード¹を使用) だけを更新できます。ただし、すべてのセールス・レコードの読み取りは行うことができます。

時間別のフラグメント化

特定のサイトで特定の期間においてだけ表を更新できるようにアプリケーションを設計します。それらの期間は、互いに十分隔たっていなければなりません。なぜなら、これからマスター・バージョンになるサイトに対して加え

1. 米国の郵便番号

られた、すべてのペンディング中の変更をレプリケーションできるようにするためです。夏時間 (またはサマータイム) などの時間の変更や、時間帯の相違にも対応します。

非 DB2 リレーショナル・ソースの計画

非 DB2 リレーショナル・データベースからのレプリケーションの場合は、キャプチャー・プログラムの代わりにキャプチャー・トリガーが使用されます。これらのトリガーは、変更済みのデータを非 DB2 リレーショナル・ソース表からキャプチャーし、その変更済みのデータを CCD 表にコミットします。キャプチャー・トリガーは、トランザクションのスループット率とログ・スペース所要量に影響します。また、使用している環境に既存のトリガーがある場合は、それらを新しいキャプチャー・トリガーとマージしなければならない場合があります。詳細については、以下のセクションを参照してください。

- 『キャプチャー・トリガーの場合のトランザクション・スループット率の計画』
- 『非 DB2 リレーショナル・ソース・サーバーの場合のログの影響の計画』
- 13 ページの『Oracle ソース・サーバーの場合のロックの計画』
- 13 ページの『既存のトリガーとキャプチャー・トリガーとの共存の計画』

キャプチャー・トリガーの場合のトランザクション・スループット率の計画

トリガー・ベースの変更キャプチャーはトランザクション・スループット率に影響を与えるため、ソース・システムのトランザクション・ワークロードが増加します。また、キャプチャー・トリガーは更新トランザクションの場合の応答時間も増加させます。この影響が最大となるのは、複製されるアプリケーション・ソース表をきわめて頻繁に更新するトランザクションの場合です。

非 DB2 リレーショナル・ソース・サーバーの場合のログの影響の計画

非 DB2 リレーショナル・ソース・サーバーの場合、ソース・アプリケーションはより多くのアクティブ・ログ・スペースを必要とします。なぜなら、ログの量が複製されるソース表のおよそ 3 倍になるからです。変更は、トリガーによってソース表からキャプチャーされて、CCD 表に保管されます。変更済みのデータは変更中のソース表と同じコミット有効範囲内で書き込まれ、データは後でトリガー・ベースの整理機構によって削除されます。したがって、1 回のソース INSERT、UPDATE、または DELETE 操作は、それぞれ 1 回の INSERT、UPDATE、または DELETE 操作に INSERT 操作と DELETE 操作をさらに 1 回ずつ加えたものとなります。更新を DELETE 操作と INSERT 操作の対に変更する場合は、ログの量はさらに増えます。

ログ・スペースを使い尽くしてしまい、キャプチャー・トリガーがレコードを CCD 表に挿入できなくなると、ユーザーまたはアプリケーション・プログラムによって試行されたトランザクションは正常に完了しません。

Oracle ソース・サーバーの場合のロックの計画

現在 Oracle ソースを更新中のアプリケーションがすべて終了してからでなければ、アプライ・プログラムはデータの適用を開始できません。データを処理して同期点を設定できるように、アプライ・プログラムは CCD 表をロックする必要があります。CCD 表に対するロックが保留されるのは、アプライ・サイクル全体を通してではなく、アプライ・プログラムが同期点を設定するまでの間だけです。ソース表を更新する必要のあるアプリケーションは、アプライ・プログラムが CCD 表をアンロックするまで待たなければなりません。

既存のトリガーとキャプチャー・トリガーとの共存の計画

キャプチャー・トリガー・ロジックは、ソースの登録時にレプリケーション・センターによって生成される SQL スクリプトに含まれています。デフォルトでは、INSERT トリガー、UPDATE トリガー、および DELETE トリガーは、これらのタイプの変更 (挿入、更新、および削除) をソース表からレプリケーションできるように作成されます。トリガー名は、CCD 表の名前と、トリガーのタイプを記述する先行文字 (I は INSERT、U は UPDATE、D は DELETE) から構成されます。たとえば、CCD 表の名前が undjr02.ccd001 である場合は、生成される DELETE トリガーの名前は undjr02.dccd001 です。スクリプト内に生成されたトリガー名を変更してはなりません。

レプリケーション用に登録する表にトリガーがすでに存在しており、かつそのトリガーの名前が生成されたスクリプト内の名前と同じである場合は、そのスクリプトの生成時に警告を受け取ります。その生成されたスクリプトを実行しないでください。なぜなら、既存のトリガーが RDBMS によって上書きされる可能性があるからです。既存のトリガーと新しいトリガーとをどのようにマージするかを決定し、既存のロジックとレプリケーション・センターによって生成されたトリガー・ロジックとをマージするスクリプトを作成してください。

作成するトリガー・タイプがレプリケーション用に登録する表にすでに存在しており、かつ RDBMS がそのタイプのトリガーを 1 つの表当たり 1 つだけしか許可しない場合は、生成されたスクリプトを実行する前にロジックをマージする必要があります。

コード・ページ・トランザクションの計画

レプリケーション・コンポーネントは、各種のオペレーティング・システムでデータのコード・ページ・トランザクションの処理を DB2 データベースに依頼するデータベース・アプリケーションです。これらは SQL SELECT、INSERT、UPDATE、および DELETE ステートメントを使用してデータを処理します。

互換コード・ページを持つデータベース間でのデータのレプリケーション

使用するレプリケーション構成では SQL ステートメントが必要で、さらに異なるコード・ページを使用するシステム間でデータをやり取りする必要がある場合は、DRDA などの下層の DB2 プロトコルがコード・ページ・トランザクションを処理します。また、データが DB2 リレーショナル・データベースと非 DB2 リレーシ

ナル・データベース間で渡される場合は、DB2 レプリケーションは必要なコード・ページ・トランザクションの処理をすべて下層のデータベース製品に依頼します。

異なるコード・ページを使用するデータベース間でデータを複製する予定である場合は、「DB2 管理ガイド」をチェックして、持っているコード・ページが互換性のあるものかどうかを判別してください。たとえば、DB2 for Linux、DB2 for UNIX または DB2 for Windows を使用している場合は、文字データの変換に関するセクションを参照してください。

使用しているデータベースが互換コード・ページを持っていることを検査したら、次にそれらのデータベース間でコード・ページの使用方法に違いがあるかどうかを判別してください。たとえば、あるデータベース製品では 1 つの表内のそれぞれの列ごとに異なるコード・ページを指定できるとします。一方、別のデータベース製品では、列単位での異なるコード・ページの指定は許可されず、コード・ページはデータベース・レベルでのみ指定する必要があるとします。最初の製品の複数のコード・ページが指定されている表は、2 番目の製品の単一データベースにはレプリケーションできません。したがって、各データベースによるコード・ページの処理方法は、使用している環境内の各種データベース間で必ず正常にデータがレプリケーションされるようにするためにはどのようにレプリケーションをセットアップする必要があるのかということに影響します。

レプリケーションに合わせた各国語サポート (NLS) の構成

レプリケーション用の NLS 構成は、システム間のデータベース接続をセットアップする際に定義されます。ただし、キャプチャー・プログラムを Linux、UNIX または Windows オペレーティング・システムで実行している場合は、キャプチャー・プログラムはデータをそこから取り込んでいるデータベースと同じコード・ページを使用する必要があります。キャプチャー・プログラムがその同じコード・ページを使用していない場合は、DB2CODEPAGE と呼ばれる DB2 環境変数または登録変数を設定する必要があります。

コード・ページ変数の設定

DB2 は、アプリケーションのコード・ページをそのアプリケーションが実行されているアクティブ環境から導き出します。通常、DB2CODEPAGE 変数が設定されていない場合は、コード・ページはオペレーティング・システムによって指定される言語 ID から導き出されます。ほとんどの場合、データベースの作成時にデフォルトのコード・ページを使用したのであれば、この値はキャプチャー・プログラムにとって正しい値です。しかし、データベースの作成時にデフォルトのコード・ページ以外のコード・ページを明示的に指定した場合は、キャプチャー・プログラムのために DB2CODEPAGE 変数を設定する必要があります。そうしない場合、キャプチャー・プログラムがデータを CD 表に挿入する際にデータが正しく変換されない可能性があります。DB2CODEPAGE 変数に対して使用する値は、CREATE DATABASE ステートメントで指定する値と同じでなければなりません。DB2CODEPAGE 変数の設定については、「DB2 管理ガイド」を参照してください。

コード・ページからのレプリケーション

1 バイト文字セット (SBCS) コード・ページを使ったソース・データを Unicode UTF-8 を使ったターゲットに複製している場合、ソース・データベースの中のいく

つかの 1 バイト文字は、DB2 によって、ターゲット・データベースで複数のバイトに変換される場合があります。16 進値が 0x80 から 0xff であるすべての 1 バイト文字は、2 バイトの 1208 等価値に変換されます。つまり、場合によってはターゲット列はソース列よりも大きくなければならず、そうでない場合アプライ・プログラムが DB2 から SQL エラーを受け取ることがあります。

いくつかのデータベース製品のインプリメント・コード・ページのサポートは、他のコード・ページのサポートと異なることがあります、そのことがレプリケーション構成に影響する場合があります。たとえば、iSeries (OS/400) 上の現在の DB2 では、コード・ページの列レベルでの指定を許可していますが、DB2 (Linux、UNIX、および Windows 版) ではデータベース・レベルでの指定しか許可されていません。したがって、異なるコード・ページを使用する複数の列を持つ OS/400 表がある場合、すべてのコード・ページに互換性がある場合を除き、これらの列を単一の DB2 (Linux、UNIX、および Windows 版) データベースに対して複製することはできません。

LANG 変数の設定

Linux または UNIX システム上でキャプチャーおよびアプライ・プログラムを実行している場合、LANG 環境変数を設定する必要があるかもしれません。キャプチャーおよびアプライ・プログラムは、この環境変数の内容を使って、使用されている言語のメッセージ・ライブラリーを検索します。たとえば、LANG 環境変数が en_US に設定されると、キャプチャー・プログラムは DB2 インスタンスの /sqlib/msg/en_US サブディレクトリーにある英語のメッセージ・ライブラリーを検索します。キャプチャーがメッセージ・ライブラリーを見つけられなかった場合、キャプチャー・トレース表 (ASN_IBMSNAP_TRACE) に書き込まれるすべてのメッセージは ASN0000S です。

DB2 UDB for z/OS のレプリケーションの計画

DB2 DataPropagator for z/OS バージョン 8 は、スキーマ名と表名を 128 バイトまでサポートします。長い名前をサポートするには、次のようにします。

- DB2 UDB for z/OS バージョン 8 の下、新規関数モードで、キャプチャー・コントロール表、アプライ・コントロール表、およびモニター・コントロール表を作成する。
- DB2 UDB for z/OS バージョン 8 の下、新規関数モードで、キャプチャー・サーバー、アプライ・サーバー、およびモニター・サーバーを実行する。

制約事項: DB2 UDB for z/OS の新規関数モードのサブシステムと、DB2 UDB for Linux、DB2 UDB for Unix、DB2 UDB for Windows、または DB2 UDB for iSeries との間で複製したい場合は、30 バイト以下のスキーマ名を使用する必要があります。DB2 UDB for z/OS バージョン 8 の新規関数モードで 30 文字を超えるスキーマ名を使用すると、そのプラットフォームと DB2 UDB for Linux、DB2 UDB for UNIX、DB2 UDB for Windows、または DB2 UDB for iSeries との間で複製を行えません。

パフォーマンス・チューニング

パフォーマンスを最適にするため、レプリケーション環境を調整する必要があるかもしれません。「*Tuning for Replication Performance*」は、DB2 レプリケーション環境の主なコンポーネントを調整して、最適なパフォーマンスを得る方法を説明しています。このマニュアルは、www.ibm.com/software/data/dpropr/library.html でオンラインで参照できます。

第 2 章 SQL レプリケーション用のサーバーの構成

データを複製するためには、その前に環境をセットアップする必要があります。

この章には、以下のセクションがあります。

- 『レプリケーション・サーバーに対するアクセスのコントロール』
- 19 ページの『レプリケーション用のユーザー ID の許可』
- 26 ページの『レプリケーション用のユーザー ID およびパスワードの保管 (Linux、UNIX、Windows)』
- 26 ページの『レプリケーション・コントロール表のセットアップ』
- 29 ページの『レプリケーション・プログラムのセットアップ』
- 36 ページの『ジャーナルのセットアップ (OS/400)』

レプリケーション・サーバーに対するアクセスのコントロール

ほとんどのレプリケーション環境では、データは複数のサーバーに分散されています。このような環境を使用している場合は、レプリケーション・プログラムがすべてのサーバーに確実に接続できるようにしておく必要があります。サーバー間の接続を実現するには正しいソフトウェアがインストールされていなければならない、さらにサーバー間の接続を構成する必要があります。非 DB2 リレーショナル・データベースへのレプリケーションを行う場合は、フェデレーテッド・サーバーおよび関連付けられた接続も構成する必要があります。

レプリケーションのための接続要件

アプライ・プログラム、レプリケーション・センター、またはレプリケーション・コマンドを実行するワークステーションは、必ずソース・サーバー、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、およびターゲット・サーバーのデータベースに接続できなければなりません。

レプリケーション・アラート・モニターを使用する場合、これを実行するワークステーションは、モニター・コントロール・サーバーとこのモニターがモニターするあらゆるサーバーに接続できなければなりません。レプリケーション・センターを使用してモニターをセットアップする場合は、レプリケーション・センターがモニター・コントロール・サーバーに接続できることを確認してください。

レプリケーションの設計で、ソース・データベースとは異なるサーバーでデータのステージングが関係しているなら、さまざまなサーバー間の通信について慎重に考慮する必要があります。エミュレーション層の数、LAN ブリッジの数、およびルーター・リンクの数については、どれもレプリケーションのパフォーマンスに影響を与えるものなので、必ず制限してください。

データベースがネットワークに接続される場合のコネクティビティーは、接続されるオペレーティング・システムによって異なります。

非 DB2 リレーショナル・サーバーへの接続

非 DB2 リレーショナル・サーバーとの間で相互にデータのレプリケーションを行う場合は、その非 DB2 リレーショナル・サーバーにアクセスして接続できなければなりません。

非 DB2 リレーショナル・ソース・サーバーからのレプリケーションを試行する前に、使用するフェデレーテッド・サーバーとフェデレーテッド・データベースをセットアップする必要があります。セットアップには以下の 3 つのメイン・ステップがあります。

1. ラッパーを定義して、DB2 データベースが他の非 DB2 リレーショナル・データベースにアクセスできるようにする。
2. サーバー・マッピングを使用して非 DB2 リレーショナル・データベースを定義する。
3. DB2 データベースへの接続に使用されるユーザー ID とパスワードの組み合わせが非 DB2 リレーショナル・データベースへのアクセスに使用されるものとは異なる場合は、ユーザー・マッピングを作成する必要がある。

「DB2 フェデレーテッド・システム・ガイド (GC88-9170)」の説明に従って、使用する環境が正しく構成されていることを確認してください。

Linux、UNIX、または Windows サーバーからの z/OS または iSeries サーバーへの接続

すべてのリモート・サーバーに接続できることを確認してください。z/OS または OS/400 システムと Windows、Linux、または UNIX システムとの間の接続を構成する方法は、「DB2 Connect 概説およびインストール」を参照してください。

前提条件:

iSeries サーバーに接続する前には、以下の条件が満たされていなければなりません。

- DB2 Universal Database または DB2 Connect がワークステーションにインストールされていなければならない。
- TCP/IP がワークステーションにセットアップされている必要があること。

手順 (iSeries 用):

DB2 for Windows ワークステーションから iSeries サーバーに接続する場合

1. iSeries サーバーにログオンして、リレーショナル・データベースを探します。
 - a. 接続先となる iSeries サーバーにログオンします。
 - b. **dsprdbdire** コマンドをサブミットしてから、*LOCAL に local と指定します。
 - c. 出力の中から、リレーショナル・データベースの名前を探します。たとえば、以下の出力では、データベースの名前は DB2400E です。

```
MYDBOS2          9.112.14.67
RCHASDPD         RCHASDPD
DB2400E          *LOCAL
RCHASLJN         RCHASLJN
```

2. OS/400 データベースを DB2 for Windows のカタログに入れる。

- a. Windows ワークステーションで、「スタート」 → 「プログラム」 → 「IBM DB2」 → 「コマンド・ウィンドウ (Command Window)」をクリックします。DB2 CLP コマンド・ウィンドウが開きます。
- b. コマンド・ウィンドウで、以下の 3 つのコマンドを正確な順序で入力します。

```
db2 catalog tcpip node server_name remote server_name server 446 system
server_name ostype OS400
```

```
db2 catalog dcs database rdb_name AS rdb_name
```

```
db2 catalog database rdb_name AS rdb_name at node server_name
authentication dcs
```

ここで、*server_name* は iSeries システムの TCP/IP ホスト名で、*rdb_name* はステップ 1 (18 ページ)で説明されている iSeries リレーショナル・データベースの名前です。

3. コマンド・ウィンドウで、以下のコマンドを発行します。

```
db2 terminate
```

4. iSeries システムにログオンするために使用する iSeries ユーザー・プロファイルが CCSID37 を使用していることを確認します。

- a. iSeries システムにログオンします。

- b. 以下のコマンドを入力します。ここで、*user* はユーザー・プロファイルです。

```
CHGUSRPRF USRPRF (user) CCSID(37)
```

- c. 次のように入力して、DDM サーバーが iSeries システムで始動していることを確認します。

```
STRTCPSVR SERVER(*DDM)
```

5. DB2 for Windows および DB2 for iSeries が接続されていることを確認する。

```
db2 connect to rdb_name user user_name using password
```

レプリケーション用のユーザー ID の許可

DB2 および非 DB2 のリレーショナル・サーバーに置かれているデータにアクセスする必要がある場合は、以下の許可要件が満たされていることを確認してください。

- 『管理の許可要件』
- 21 ページの『キャプチャー・プログラムの許可要件』
- 23 ページの『非 DB2 リレーショナル・データベースでのキャプチャー・トリガーの場合の許可要件』
- 23 ページの『アプライ・プログラムの許可要件』
- 25 ページの『レプリケーション・アラート・モニターの場合の許可要件』

管理の許可要件

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を管理します。(詳しくは、265 ページの『第 15 章 SQL レプリケーションの場合のレプリケーション・センターの使用法』を参照してください。) レプリケーション環境が OS/400 オペレーティング・システム上にのみある場合は、OS/400 システム・コマンドを使用し

レプリケーションを管理できます。(詳しくは、385 ページの『第 19 章 SQL レプリケーション用のシステム・コマンド (OS/400)』を参照してください。) レプリケーションを管理するためには、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 にはレプリケーションをセットアップする権限がなければなりません。使用するユーザー ID は、すべてのシステムで同じである必要はありません。ただし、同じであった方が楽です。レプリケーションのセットアップには、オブジェクト (コントロール表や表スペースなど) の作成、プランのバインディング (Linux、UNIX、Windows、および z/OS の場合)、SQL パッケージの作成 (OS/400 の場合)、ならびに表、登録、およびサブスクリプション・セットを作成するための生成済み SQL の実行が含まれます。使用するレプリケーション環境内のすべてのサーバーに対して同一の許可ユーザー ID を使用するか、あるいはそれぞれのサーバーごとに別々の許可ユーザー ID を使用するかは任意です。

Linux、UNIX、Windows、および z/OS の場合の要件

レプリケーションのセットアップに使用するユーザー ID で以下のタスクを実行できることを確認してください。

- すべてのサーバー (ソース・サーバー、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、モニター・コントロール・サーバー、およびターゲット・サーバー) への接続
- ソース・サーバー、キャプチャー・コントロール・サーバー、モニター・コントロール・サーバー、およびターゲット・サーバーに置かれているカタログ表からの選択
- ソース・サーバー、モニター・コントロール・サーバー、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・サーバーでの表 (レプリケーション・コントロール表も含む)、表スペース、およびビューの作成
- ターゲット・サーバーでの表および表スペースの作成 (新しいターゲット表の作成に DB2 レプリケーション・プログラムを使用する場合) (既存の表をターゲットとして使用する場合は必須ではありません)
- レプリケーションに関与するそれぞれの DB2 データベース (ソース・サーバー、ターゲット・サーバー、モニター・コントロール・サーバー、およびアプライ・コントロール・サーバーも含む) でのプランのバインドまたはパッケージの作成
- 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 (Linux、UNIX と Windows のみ)

非 DB2 リレーショナル・データベースの場合は、使用するユーザー ID で以下のアクションを実行できなければなりません。

- 表を作成します。
- ソース表およびコントロール表に対するキャプチャー・トリガーの作成
- プロシージャの作成
- DB2 フェデレーテッド・データベースでのニックネームの作成
- シーケンスの作成 (Oracle データベースの場合のみ)
- カタログ表からの選択

ほとんどのレプリケーション管理者は DBADM 特権または SYSADM 特権を持っています。DB2 for z/OS の場合、レプリケーション管理者は、少なくともカタログからの選択を許可されている必要があり、さらに索引作成特権も含め、ASN スキーマを使用した表の作成、およびソース表の特性を持つ CD 表とターゲット表の作成を行うために必要なすべての特権を持っている必要があります。

OS/400 の場合の要件

レプリケーションのセットアップに使用するユーザー ID で以下のタスクを実行できることを確認してください。

- すべてのサーバー (ソース・サーバー、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、モニター・コントロール・サーバー、およびターゲット・サーバー) への接続
- ソース・サーバー、キャプチャー・コントロール・サーバー、モニター・コントロール・サーバー、およびターゲット・サーバーに置かれているカタログ表からの選択
- ソース・サーバー、モニター・コントロール・サーバー、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・サーバーでの表 (レプリケーション・コントロール表も含む) とビューの作成
- ターゲット・サーバーでの表の作成 (新しいターゲット表の作成に DB2 レプリケーション・プログラムを使用する場合) (既存の表をターゲットとして使用する場合は必須ではありません)
- レプリケーションに関与するそれぞれの DB2 データベース (ソース・サーバー、ターゲット・サーバー、モニター・コントロール・サーバー、およびアプライ・コントロール・サーバーも含む) でのプランのバインドまたはパッケージの作成

ほとんどのレプリケーション管理者は DBADM 特権または SYSADM 特権を持っています。

DPR 権限付与 (**GRTDPRAUT**) コマンドは、ユーザーがソースの登録、それらのソースへのサブスクライブ、およびコントロール表の作成を行うのを許可するために使用します。OS/400 システム間でのみレプリケーションを行う場合は、すべてのサーバーに対して同じユーザー ID を使用する必要があります。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、436 ページの『GRTDPRAUT: ユーザーの許可 (OS/400)』を参照してください。

DPR 権限付与 (**GRTDPRAUT**) コマンドがマシンにインストールされていない場合は、オブジェクト権限付与 (**GRTOBJAUT**) コマンドを使用する必要があります。

キャプチャー・プログラムの許可要件

キャプチャー・プログラムを実行するユーザー ID は、DB2 システム・カタログにアクセスできなければなりません。また、キャプチャー・コントロール・サーバー上のすべてのレプリケーション・コントロール表へのアクセスと更新が可能で、さらにキャプチャー・プログラム・パッケージを実行できなければなりません。レプリケーション管理者のユーザー ID を使用してキャプチャー・プログラムを実行できますが、これは要件ではありません。

Linux、UNIX、Windows の場合の要件

キャプチャー・プログラムを実行するユーザー ID が以下の権限および特権を持っていることを確認してください。

- DBADM または SYSADM 権限。
- キャプチャー・パス・ディレクトリーに対する WRITE 特権。これは、キャプチャー・プログラムはキャプチャー・プログラムの始動時に指定された **capture_path** ディレクトリーに診断ファイルを作成するためです。

z/OS の場合の要件

キャプチャー・プログラムを実行するために使用するユーザー ID は、USS にアクセスできるものとして登録する必要があります。これは、z/OS UNIX または OS/390 UNIX (OMVS セグメントを持っていないといけない) を使用するためのユーザー ID を定義する必要があることを意味します。

また、キャプチャー・ロード・ライブラリーに APF 許可が与えられていること、およびキャプチャー・プログラムを実行するユーザー ID が以下の特権を持っていることも確認してください。

- 一時ディレクトリー (/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) への WRITE アクセス権。
- キャプチャー CONTROL サーバー上のすべてのレプリケーション表に対する SELECT、UPDATE、INSERT、および DELETE 特権。(これらの表のリストについては、508 ページの『キャプチャー・コントロール・サーバーで使用される表のリスト』を参照。)
- DB2 カタログ (SYSIBM.SYSTABLES および SYSIBM.SYSCOLUMNS) に対する SELECT 特権。
- TRACE 特権。
- MONITOR1 および MONITOR2 特権。
- キャプチャー・プログラム・パッケージに対する EXECUTE 特権。

また、そのユーザー ID にキャプチャー・パス・ディレクトリーに対する WRITE アクセス権があること (USS)、または上位修飾子が付けられていること (z/OS) を確認してください。キャプチャー・プログラムを USS シェルで実行するためには、STEPLIB システム変数が設定されていて、さらにこの変数にキャプチャー・ロード・ライブラリーが組み込まれている必要があります。PATH には HFS パス (/usr/lpp/db2repl_08_01/bin) が含まれていなければなりません。

OS/400 の場合の要件

DPR 権限付与 (**GRTDPRAUT**) コマンドは、ユーザーにローカル・システムでのキャプチャー・プログラムの実行を許可するために使用します。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、436 ページの

『GRTDPRAUT: ユーザーの許可 (OS/400)』を参照してください。OS/400 システム間でのみレプリケーションを行う場合は、すべてのサーバーに対して同じユーザー ID を使用する必要があります。**GRTDPRAUT** コマンドがマシンにインストールされていない場合は、オブジェクト権限付与 (**GRTOBJAUT**) コマンドを使用する必要があります。

非 DB2 リレーショナル・データベースでのキャプチャー・トリガーの場合の許可要件

非 DB2 RDBMS から複製する場合は、ソースからの変更のキャプチャーにはキャプチャー・トリガーが使用されます。リモート・ソース表を変更するリモート・ユーザー ID (たとえば、ユーザー・アプリケーションからの) には、CCD 表への挿入を行う権限が必要です。ほとんどの場合、INSERT、UPDATE、または DELETE トリガーを実行するのに明示的な権限は必要ありません。なぜなら、表に対してトリガーが定義されると、それ以降トリガーは INSERT、UPDATE、または DELETE を実行中のアプリケーションにとって透過的に実行されるからです。Informix データベースの場合は、登録済みのソース表に対して INSERT、UPDATE、および DELETE アクションを実行するリモート・ユーザー ID には EXECUTE PROCEDURE 特権が必要です。

アプライ・プログラムの許可要件

アプライ・プログラムを実行するユーザー ID は、DB2 システム・カタログにアクセスできなければなりません。また、キャプチャー・コントロールおよびターゲット・サーバー上のすべてのレプリケーション・コントロール表へのアクセスと更新が可能で、さらにアプライ・プログラム・パッケージを実行できなければなりません。レプリケーション管理者のユーザー ID を使用してアプライ・プログラムを実行できますが、これは要件ではありません。

Linux、UNIX、Windows の場合の要件

アプライ・プログラムを実行するユーザー ID が以下の権限および特権を持っていることを確認してください。

- アプライ・パス・ディレクトリーに対する書き込み特権
- レプリケーション・ソース表 (関連した CD 表および CCD 表も含む) へのアクセス特権
- レプリケーション・ターゲット表に対するアクセスおよび更新特権
- DB2 レプリケーション・プログラムによって生成されて、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・サーバーでビルドされたすべてのコントロール表に対するアクセスと更新特権
- アプライ・プログラムによって使用される任意のパスワード・ファイルの読み取り特権

注: 使用するソース表が非 DB2 リレーショナル・データベース管理システム上にある場合は、ユーザー ID は DB2 フェデレーテッド・データベースおよび非 DB2 リレーショナル・データベースの両方において、フェデレーテッド・データベースで定義されているニックネームを使用してソース表にアクセスできるだけの十分な特権を持っている必要があります。

z/OS の場合の要件

アプライ・プログラムを実行するユーザー ID が以下の権限および特権を持っていることを確認してください。

- 一時ディレクトリー (/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) への WRITE アクセス権。

- アプライ CONTROL サーバー上のすべてのレプリケーション表に対する SELECT、UPDATE、INSERT、および DELETE 特権。(これらの表のリストについては、511 ページの『アプライ・コントロール・サーバーで使用される表のリスト』を参照。)
- DB2 カタログ (SYSIBM.SYSTABLES および SYSIBM.SYSCOLUMNS) に対する SELECT 権限。

注: アプライ・プログラムを実行するために使用するユーザー ID は、USS にアクセスできるものとして登録する必要があります。これは、z/OS UNIX または OS/390 UNIX (OMVS セグメントを持っていないなければならない) を使用するためのユーザー ID を定義する必要があることを意味します。ロード・ライブラリーに APF 許可が必要となるのは、アプライ・プログラムが ARM の指定付きで登録される場合だけです。アプライ・プログラムを USS シェルで実行するためには、STEPLIB システム変数が設定されていて、さらにこの変数にアプライ・ロード・ライブラリーが組み込まれている必要があります。PATH には HFS パス (/usr/lpp/db2repl_08_01/bin) が含まれていなければなりません。

非 DB2 リレーショナル・データベース管理システムの要件

コントロール表が非 DB2 リレーショナル・データベース管理システム上にある場合は、変更されたデータを非 DB2 リレーショナル・ターゲットにプッシュしたりそこからデータをプルするユーザー ID は、DB2 フェデレーテッド・データベースおよび非 DB2 リレーショナル・データベースにおいて十分な特権を持っている必要があります。

非 DB2 リレーショナル・ターゲットの場合、アプライ・プログラムを実行中のユーザー ID には、DB2 フェデレーテッド・データベース上のニックネームに対して WRITE を実行できる特権と、ユーザー・マッピングを介して実際の非 DB2 ターゲットへの WRITE を実行できる特権が必要です。

非 DB2 リレーショナル・ソースの場合は、アプライ・プログラムを実行中の ID には以下の特権が必要です。

- DB2 フェデレーテッド・データベース上のニックネームに対して READ と WRITE を実行できる特権、およびユーザー・マッピングを介してキャプチャー CONTROL 表に対し READ と WRITE を実行できる特権
- DB2 フェデレーテッド・データベース上のニックネームからの READ を実行できる特権、およびユーザー・マッピングを介して非 DB2 サーバー上の実際の CCD 表からの READ を実行できる特権
- DB2 フェデレーテッド・データベース上のニックネームからの READ を実行できる特権、およびユーザー・マッピングを介して非 DB2 サーバー上の実際のソース表からの READ を実行できる特権

OS/400 の場合の要件

DPR 権限付与 (GRTPRAUT) コマンドは、ユーザーにローカル・システムでのアプライ・プログラムの実行を許可するために使用します。OS/400 システム間でのみレプリケーションを行う場合は、すべてのサーバーに対して同じユーザー ID を使用する必要があります。GRTPRAUT コマンドがマシンにインストールされていない場合は、オブジェクト権限付与 (GRTOBJAUT) コマンドを使用する必要があります。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、436 ページの『GRTDPRAUT: ユーザーの許可 (OS/400)』を参照してください。

使用するレプリケーション環境内のそれぞれのサーバーごとに異なるユーザー ID を使用できます。

レプリケーション・アラート・モニターの場合の許可要件

モニター・プログラムを実行するユーザー ID は、モニター・コントロール・サーバー上のすべてのレプリケーション・コントロール表へのアクセスと更新が可能で、かつモニター・プログラム・パッケージを実行できなければなりません。レプリケーション管理者のユーザー ID を使用してモニター・プログラムを実行できますが、これは要件ではありません。

Linux、UNIX、Windows の場合の要件

レプリケーション・アラート・モニターを始動するユーザー ID が、モニター・コントロール表が置かれているモニター・コントロール・サーバー、およびモニターしているコントロール表を含んでいるサーバーでの有効なログイン ID であることを確認してください。また、レプリケーション・アラート・モニターを実行するユーザー ID が以下の権限および特権を持っていることも確認してください。

- モニター CONTROL サーバー上のモニター CONTROL 表に対する SELECT、UPDATE、INSERT、および DELETE 特権 (これらの表のリストについては、512 ページの『モニター・コントロール・サーバーのコントロール表のリスト』を参照。)
- モニターするサーバー上にあるキャプチャー・コントロール表およびアプライ・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モニター・パス・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 READ アクセス

z/OS の場合の要件

モニター・プログラムを実行するユーザー ID が以下の権限および特権を持っていることを確認してください。

- 一時ディレクトリー (/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) への WRITE アクセス権。
- モニター・コントロール・サーバー上のすべてのレプリケーション表に対する SELECT、UPDATE、INSERT、および DELETE 特権。
- DB2 カタログ (SYSIBM.SYSTABLES および SYSIBM.SYSCOLUMNS) に対する SELECT 権限。

注: モニター・プログラムを実行するために使用するユーザー ID は、USS にアクセスできるものとして登録する必要があります。これは、z/OS UNIX または OS/390 UNIX (OMVS セグメントを持っていないなければならない) を

使用するためのユーザー ID を定義する必要があることを意味します。ロード・ライブラリーに APF 許可が必要となるのは、モニター・プログラムが ARM の指定付きで登録される場合だけです。モニター・プログラムを USS シェルで実行するためには、STEPLIB システム変数が設定されていて、さらにこの変数にキャプチャー・ロード・ライブラリーが組み込まれている必要があります。PATH には HFS パス (/usr/lpp/db2repl_08_01/bin) が含まれていなければなりません。

レプリケーション用のユーザー ID およびパスワードの保管 (Linux、UNIX、Windows)

レプリケーション環境が複数のサーバーに分散されていない場合は、ユーザー ID とパスワードを保管する必要はありません。しかし、ほとんどのレプリケーション環境では、データは複数のサーバーに分散されています。そのような環境がある場合は、データベースへの接続を試行する時に有効なユーザー ID とパスワードを入力して、DB2 がユーザーの ID を検証できるようにする必要があります。レプリケーション・センターの場合とそれ以外のレプリケーション・プログラムの場合とでは、パスワード情報の保管方法が異なります。

asnpwd コマンドは、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・アナライザーがリモート・サーバー上のデータにアクセスできるように、パスワード・ファイルを作成して保守するために使用します。(キャプチャー・プログラムはパスワード・ファイルを必要としません。)パスワード・ファイル内の情報は、機密性を確保するために暗号化されます。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、362 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。

レプリケーション・センターの場合のパスワード要件は、レプリケーション・センターのヘルプおよび 271 ページの『レプリケーション・センターのユーザー ID およびパスワードの管理』を参照してください。

レプリケーション・コントロール表のセットアップ

レプリケーションに使用されるコントロール表を作成できます。

- 『コントロール表の作成 (Linux、UNIX、Windows)』
- 27 ページの『コントロール表の作成 (z/OS)』
- 27 ページの『コントロール表の作成 (OS/400)』
- 28 ページの『非 DB2 リレーショナル・ソース用のコントロール表の作成』
- 28 ページの『キャプチャー・コントロール表のセットを複数作成する』
- 29 ページの『複数のデータベース・パーティション上でのコントロール表の作成』

コントロール表の作成 (Linux、UNIX、Windows)

レプリケーション・センターを使用して、Linux、UNIX および Windows でのキャプチャー・プログラム用とアプライ・プログラム用のレプリケーション・コントロール表を作成します。レプリケーション・コントロール表の作成時にコントロール表の作成方法をカスタマイズしない場合は、2 つの表スペース (1 つは UOW 表

用、もう 1 つはその他のコントロール表用) が作成されます。それらのデフォルトのレプリケーション表スペースを使用したくない場合は、既存の表スペースを指定するか、新しい表スペースを作成するか、または現行の DB2 デフォルト表スペースを使用できます。レプリケーション・コントロール表の作成の詳細については、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。

複数データベース・パーティション環境でキャプチャーを開始する場合、キャプチャーは `IBMSNAP_RESTART` 表と同じ表スペースに追加のコントロール表 (`IBMSNAP_PARTITIONINFO`) を作成します。

コントロール表の作成 (z/OS)

レプリケーション・センターを使用して、z/OS でのレプリケーション・コントロール表を作成します。z/OS オペレーティング・システム用のプロファイルを作成して、このタイプのシステム用のコントロール表を作成するときに使用すべきデフォルトが識別されるようにできます。それらのコントロール表に対してプロファイルを設定してしまえば、それ以降はコントロール表のセットを作成するたびにプロファイルを設定する必要がなくなります。ただし、それらのデフォルトはコントロール表の作成時にオーバーライドできます。また、プロファイルもいつでも変更できます。ただし、その変更の影響を受けるのは、プロファイルの変更後に作成したコントロール表だけです。レプリケーション・コントロール表の作成の詳細については、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。

コントロール表の作成 (OS/400)

レプリケーション・コントロール表は、DB2 DataPropagator for iSeries のインストール時に自動的に作成されます。それらの表は、DataPropagator (DPROPR) のデフォルト・スキーマ (ASN と呼ばれる) 内に作成されます (まだ存在していない場合)。

新しいキャプチャー・スキーマを使用してキャプチャー・コントロール表の新しいセットを作成できます。スキーマは、最大 25 個まで作成できます。28 ページの『キャプチャー・コントロール表のセットを複数作成する』の説明に従って、DPR 表の作成 (`CRTDPRTBL`) コマンドを使用してください。また、`CRTDPRTBL` コマンドは、レプリケーション・コントロール表が誤って削除または破壊された場合にも使用できます。このコマンドの詳細については、430 ページの『`CRTDPRTBL`: レプリケーション・コントロール表の作成 (OS/400)』を参照してください。

重要: OS/400 では、コントロール表は必ず `CRTDPRTBL` コマンドを使用して作成するようにしてください。レプリケーション・センターは OS/400 版コントロール表の作成をサポートしていません。

ユーザー定義のファイル・システムでは、レプリケーション・コントロール表を基本補助記憶域プール (ASP) または独立補助記憶域プール (IASP) グループに作成できますが、両方に作成することはできません。コントロール表を IASP グループに作成する場合は、まず最初にすべてのキャプチャーおよびアプライ・コントロール表を基本 ASP から除去する必要があります。キャプチャーまたはアプライ・プログラムを開始する前に、ASN ライブラリー (またはキャプチャー・スキーマ用のライブラリー) を含む ASP グループに対して `SETASPGRP` コマンドを発行します。

非 DB2 リレーショナル・ソース用のコントロール表の作成

Informix のような非 DB2 RDBMS から のレプリケーションを行う場合は、DB2 からのレプリケーションの場合とまったく同じように、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必要があります。このようなタイプのソースの場合、レプリケーション・センターは非 DB2 リレーショナル・データベースに以下のキャプチャー・コントロール表を作成します。

- 整理コントロール表 (IBMSNAP_PRUNCNTL)
- 整理セット表 (IBMSNAP_PRUNE_SET)
- 登録同期表 (IBMSNAP_REG_SYNCH)
- 登録表 (IBMSNAP_REGISTER)
- 順序付け表 (IBMSNAP_SEQTABLE) (Informix の場合のみ)
- シグナル表 (IBMSNAP_SIGNAL)

フェデレーテッド・データベース内に順序付け表 (IBMSNAP_SEQTABLE) 以外のすべての表のニックネームが作成されます。(この順序付け表は、Informix トリガーだけが使用します。アプライ・プログラムはこれを使用しません。) シグナル表 (IBMSNAP_SIGNAL) および登録同期表 (IBMSNAP_REG_SYNCH) にトリガーが自動的に作成されます。

重要: IBMSNAP_SIGNAL 表および IBMSNAP_REG_SYNCH 表に作成されたトリガーは、除去または変更をしないでください。

キャプチャー・コントロール表のセットを複数作成する

1 台のサーバー上で複数のキャプチャー・プログラムを使用する場合は、キャプチャー・コントロール表のセットを複数作成して、それぞれの表セットが必ずユニークなキャプチャー・スキーマを持つようにする必要があります。このスキーマにより、ある表のセットを使用するキャプチャー・プログラムが識別されます。複数のキャプチャー・スキーマを使用すると、複数のキャプチャー・プログラムを並行実行できます。

次のような状況で、複数のキャプチャー・プログラムを実行したい場合があります:

- 待ち時間が短い表を他の表と異なる方法で扱うことにより、パフォーマンスを最適化するため。待ち時間が短い表がある場合は、それらの表のレプリケーションはそれら専用のキャプチャー・プログラムを使用して行うのが望ましいことがあります。この方法を使用すると、それらに異なるランタイム優先順位を与えることができます。また、キャプチャー・プログラムのパラメーター (整理インターバルやモニター・インターバルなど) を、それらの表の短い待ち時間に合うように設定することも可能です。
- より高いキャプチャー・スループットの可能性を提供するため。これは、複数の CPU を使用しているソース環境において、多大な利点が得られることがあります。スループット増加のトレードオフは、ログ・リーダーが複数存在することに関連した CPU オーバーヘッドの増加です。

同じフェデレーテッド・データベース内の複数の非 DB2 ソース・データベースから複製したい場合は、キャプチャー・コントロール表の複数のセットを作成し、各セットは独自のスキーマを持つ必要があります。または、希望する場合は、別のフ

エデレーテッド・データベースを使用することもできます。この場合、各サーバー上のキャプチャー・コントロール表はデフォルトの ASN スキーマを使用できません。

z/OS システムでは、UNICODE コード化スキームと EBCDIC コード化スキームを別々に使用して作業したい場合、または 1 つのサブシステムでキャプチャー・プログラムのインスタンスを複数実行したい場合は、複数のキャプチャー・スキーマを使用できます。コントロール表の作成については、27 ページの『コントロール表の作成 (z/OS)』を参照してください。

OS/400 システムでは、DPR 表の作成 (CRTDPRTBL) コマンドを使用してキャプチャー・コントロール表の追加のセットを作成します。その際には、**CAPCTLLIB** パラメーターを使用してスキーマ名を指定します。このコマンドの詳細については、430 ページの『CRTDPRTBL: レプリケーション・コントロール表の作成 (OS/400)』を参照してください。

複数のデータベース・パーティション上でのコントロール表の作成

複数のパーティション・データベースにキャプチャー・コントロール表を作成する場合、これらのコントロール表によって使用される表スペースすべてはカタログ・ノード上に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・ノード上にある必要があります。

キャプチャー・プログラムを初めて始動し、WARMSI スタート・モードを選択する場合は、IBMSNAP_PARTITIONINFO 表はまだ存在していません。キャプチャー・プログラムはこの表および表のユニーク索引を、IBMSNAP_RESTART 表があるのと同じ表スペースに作成します。IBMSNAP_PARTITIONINFO 表の作成後、キャプチャー・プログラムは、各データベース・パーティションについて表の中に 1 つの行を挿入します。

キャプチャー・プログラムを開始するのが初めてではなく、いずれかのウォーム・スタート・モードを選択する場合は、IBMSNAP_PARTITIONINFO 表はすでに存在しています。「キャプチャーを最後に実行した後で 1 つまたは複数のパーティションが追加された」というチェック・ボックスを選択している場合は、キャプチャー・プログラムは、キャプチャー・プログラムが最後に実行された後で追加された各データベース・パーティションについて、IBMSNAP_PARTITIONINFO 表内に 1 つの行を挿入します。レプリケーション・センターから複数のデータベース・パーティション用にキャプチャー・コントロール表を作成する方法については、レプリケーション・センターのヘルプを参照してください。

レプリケーション・プログラムのセットアップ

以下のセクションでは、環境内のサーバーに対してレプリケーション・プログラムをセットアップするステップについて説明しています。

- 30 ページの『レプリケーション・プログラムのセットアップ (Linux、UNIX、Windows)』
- 34 ページの『キャプチャー・プログラムおよびアプライ・プログラムのセットアップ (OS/400)』
- 36 ページの『レプリケーション・プログラムのセットアップ (z/OS)』

- 36 ページの『複数データベース・パーティションのキャプチャー』

レプリケーション・プログラムのセットアップ (Linux、UNIX、Windows)

以下の指示に従って、レプリケーション・プログラムをセットアップしてください。

- 『レプリケーション・プログラムのための環境変数の設定 (Linux、UNIX、Windows)』
- 31 ページの『キャプチャー・プログラムを実行するための DB2 データベースの準備 (Linux、UNIX、Windows)』
- 31 ページの『オプション：キャプチャー・プログラム・パッケージのバインディング (Linux、UNIX、Windows)』
- 32 ページの『オプション：アプライ・プログラム・パッケージのバインディング (Linux、UNIX、Windows)』
- 33 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・プログラム・パッケージのバインディング (Linux、UNIX、Windows)』

レプリケーション・プログラムのための環境変数の設定 (Linux、UNIX、Windows)

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・プログラムを始動する前と停止する前、およびレプリケーション・センターまたはレプリケーション・システム・コマンドを使用する前に、環境変数を設定する必要があります。

手順:

環境変数を設定するには、以下のようになります。

1. 以下のように入力して、DB2 インスタンス名 (DB2INSTANCE) の環境変数を設定します。

Windows の場合

```
SET DB2INSTANCE=db2_instance_name
```

Linux および UNIX の場合

```
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・ページに設定します。14 ページの『レプリケーションに合わせた各国語サポート (NLS) の構成』を参照してください。²
3. オプション: 環境変数 DB2DBDFT をソース・サーバーに設定します。

2. キャプチャーは、データを取り込んでいるデータベースのコード・ページと同じコード・ページで実行されている必要があります。DB2 は、キャプチャーが実行されているアクティブ環境からキャプチャー・コード・ページを導き出します。DB2CODEPAGE が設定されていない場合は、DB2 はコード・ページ値をオペレーティング・システムから取得します。データベースの作成時にデフォルトのコード・ページを使用したのであれば、オペレーティング・システムから取得した値はキャプチャーにとって正しいものです。

4. **Linux および UNIX の場合:** 使用しているシステムに固有のライブラリー・パス・システム変数および実行可能モジュール・パス・システム変数に、レプリケーション用のライブラリーと実行可能モジュールがインストールされているディレクトリーが組み込まれていることを確認します。

キャプチャー・プログラムを実行するための DB2 データベースの準備 (Linux、UNIX、Windows)

手順:

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

1. 以下のように入力して、キャプチャー・コントロール・サーバーのデータベースに接続します。

```
db2 connect to database
```

ここで、*database* はキャプチャー・コントロール・サーバーのデータベースです。

2. キャプチャー・コントロール・サーバーのデータベースを、ロールフォワード・リカバリーのために準備します。これを行うには、**update database configuration** (データベース構成の更新) コマンド (ログ保存リカバリー) と **backup database** (データベースのバックアップ) コマンドを発行します。インストール要件に基づいて構成値を大きくする必要がある場合があります。多数の行または非常に大きな行を使用するトランザクションの場合は、CAPPARMS メモリー限度パラメーターの値を大きくすることが推奨されます。

複数データベース・パーティション環境の場合、キャプチャー・コントロール・サーバー・データベースがある各ノードに対してロールフォワード・リカバリーができるように各パーティションを設定しなければなりません。

多くの大規模ワークステーション・シナリオの場合、適切なデータベース構成値は以下のとおりです: APPLHEAPSZ 1000、LOGFILSIZ 4000、LOGPRIMARY 8、LOGSECOND 40、DBHEAP 1000、LOGBUFSZ 16、MAXAPPLS 200。

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

以下のステップはオプションです。なぜなら、Linux、UNIX、および Windows では、キャプチャー・プログラムは実行時に自動的にバインドされるからです。

手順:

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、以下のようにします。

1. 以下のように入力して、キャプチャー・コントロール・サーバーのデータベースに接続します。

```
db2 connect to database
```

ここで、*database* はキャプチャー・コントロール・サーバーのデータベースです。

2. キャプチャー・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します。

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。

3. 以下のようなコマンドを入力して、キャプチャー・プログラム・パッケージを作成してソース・サーバー・データベースにバインドします。

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

ただし *ur* は、パフォーマンスを向上するための非コミット読み取り形式のリストを指定します。

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、*capture.lst* ファイル内で見つけることができます。

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

Linux、UNIX および Windows では、アプライ・プログラムは実行時に自動的にバインドされます。したがって、これらのオペレーティング・システムでは、以下のステップはオプションです。

手順:

アプライ・プログラム・パッケージをバインドするには、以下のようにします。

1. 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します。

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. アプライ・プログラムが接続するそれぞれのソース・サーバー、ターゲット・サーバー、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・サーバーごとに、以下のステップを実行します。

- a. 以下のように入力して、データベースに接続します。

```
db2 connect to database
```

ここで、*database* はソース・サーバー、ターゲット・サーバー、キャプチャー・コントロール・サーバー、またはアプライ・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。例:

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


- b. 次の 2 つのコマンドを両方入力して、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します。

```
db2 bind @applycs.lst isolation cs blocking all grant public
db2 bind @applyur.lst isolation ur blocking all grant public
```

なお、cs はカーソル固定形式でリストを指定するのに対し、ur は非コミット読み取り形式でリストを指定します。

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、applycs.lst および applyur.lst ファイル内で見つけることができます。

オプション：レプリケーション・アラート・モニター・プログラム・パッケージのバインディング (Linux、UNIX、Windows)

パッケージを BIND するための以下のステップはオプションです。レプリケーション・アラート・モニター・パッケージは、実行時に自動的にバインドされます。オプションを指定したい場合、またはすべての BIND 処理が正常に完了したことをチェックしたい場合は、以下のステップを実行してください。

手順:

レプリケーション・アラート・モニター・プログラム・パッケージをバインドするには、以下のようにします。

1. レプリケーション・アラート・モニター・プログラムのバインド・ファイルが配置されているディレクトリーに変更します。

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. 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します。

- 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します。

```
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。例:

```
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・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します。

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

なお、cs はカーソル固定形式でリストを指定するのに対し、ur は非コミット読み取り形式でリストを指定します。

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・プログラム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します。

- a. 以下のように入力して、データベースに接続します。

```
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。例:

```
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・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 @asnmonit.lst isolation ur blocking all grant public
```

なお、ur は、非コミット読み取り形式のリストを指定します。

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。

キャプチャー・プログラムおよびアプライ・プログラムのセットアップ (OS/400)

アプライ・プログラムを他の非 OS/400 オペレーティング・システムのリモート・システムで使用する場合は、環境をセットアップする必要があります。以下のセクションでは、レプリケーション環境をセットアップするステップについて説明しています。

- 『リモート・システムで使用するための SQL パッケージの作成 (OS/400)』
- 35 ページの『SQL パッケージへの特権の付与』

リモート・システムで使用するための SQL パッケージの作成 (OS/400)

以下のケースでは、**CRTSQLPKG** コマンドを使用してパッケージを作成する必要があります。

- リモート・ジャーナリングを使用する場合。キャプチャー・プログラムを実行中のシステムで **CRTSQLPKG** コマンドを実行し、ソース表が配置されているシステムを指示してください。
- **ADDDPRSUB** または **ADDDPRSUBM** コマンドを使用してサブスクリプション・セットまたはサブスクリプション・セットのメンバーを追加する前。次のようにして、ターゲット・サーバーで **CRTSQLPKG** コマンドを実行してください。
 - ソース表が別のマシン上にある場合は、そのソース表が配置されているシステムを指示する。
 - アプライ・コントロール・サーバーが別のマシン上にある場合は、そのアプライ・コントロール・サーバーを指示する。

SQL パッケージを使用すると、レプリケーション・プログラムが分散レプリケーション環境で稼働できるようになります。その場合、レプリケーションが OS/400 システム間で行われる環境であっても、OS/400 システムと他のいずれかのオペレーティング・システム (Linux、UNIX や Windows など) との間で行われる環境であっても構いません。

CRTSQLPKG コマンドの使用法については、「DB2 UDB for iSeries SQL プログラミング」を参照してください。

パッケージは ASN 修飾子を使って作成されます。OS/400 では、パッケージは ASN ライブラリーに作成されます。他のオペレーティング・システムでは、パッケージは ASN スキーマに作成されます。

アプライ・プログラム用の SQL パッケージの作成: SQL パッケージを作成して、接続する必要のあるすべてのリモート・サーバーとアプライ・プログラムが対話できるようにする必要があります。たとえば、アプライ・プログラムがリモート・システムに接続できるようにするためには、アプライ・プログラムが実行されているシステムで以下のコマンドを実行してください。

```
CRTSQLPKG PGM(QDP4/QZSNAPV2) RDB(remote_system)
```

remote_system は、アプライ・プログラムが接続する必要のあるリモート・システムでのリレーショナル・データベース入り口名です。

レプリケーション・アナライザー用の SQL パッケージの作成: SQL パッケージを作成して、キャプチャー・コントロール・サーバーやターゲット・サーバーといった、分析の対象としているサーバーとレプリケーション・アナライザーが対話できるようにする必要があります。レプリケーション・アナライザーが実行されているシステムで以下のコマンドを実行してください。

```
CRTSQLPKG PGM(QDP4/QZSNANZR) RDB(remote_system)
```

ここで、*remote_system* は分析の対象としているシステムの名前です。

レプリケーション管理コマンド用の SQL パッケージの作成: OS/400 システム間のレプリケーションでは、リモート・ジャーナルを使用する場合、このコマンドを使用してキャプチャー・プログラム用およびレプリケーション管理コマンド用のパッケージを作成する必要があります。キャプチャーが実行されているシステムで以下のコマンドを実行してください。

```
CRTSQLPKG PGM(QDP4/QZSNSQLF) RDB(source_system) OBJTYPE(*SRVPGM)
```

ここで、*source_system* はソース表が実際に存在するシステムの名前です。

SQL パッケージへの特権の付与

パッケージの作成後には、ソース・データベースに登録されているファイルにサブスクライブするすべてのユーザーに対して *EXECUTE 特権を付与する必要があります。ソース・データベースが常駐している OS/400 システムにログオンし、以下の方法のいずれかを使用してください。

- 以下のオブジェクト権限付与 (**GRTOBJAUT**) コマンドを使用する。

```
GRTOBJAUT OBJ(ASN/package_name) OBJTYPE(*SQLPKG)  
USER(subscriber_name) AUT(*OBJOPR *EXECUTE)
```

- SQL を使用してソース・データベースに接続し、以下の GRANT SQL ステートメントを実行する。

```
CONNECT TO data_server_RDB_name  
GRANT EXECUTE ON PACKAGE ASN/package_name TO subscriber_name
```

- **GRTDPRAUT** コマンドを使用する (このコマンドがローカル・システムにインストールされている場合)。コマンド構文およびパラメーターの詳細については、436 ページの『GRTDPRAUT: ユーザーの許可 (OS/400)』を参照してください。

レプリケーション・プログラムのセットアップ (z/OS)

IBM DB2 DataPropagator 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。「*Program Directory for IBM DB2 DataPropagator for z/OS*」の指示を参照してください。

複数データベース・パーティションのキャプチャー

DB2 Enterprise Server Edition でデータを複製している場合、複数のデータベース・パーティションにまたがって広がるソース表への変更をキャプチャーできます。キャプチャー・プログラムは、同じパーティション・グループに属するデータベース・パーティションのリストを IBMSNAP_PARTITIONINFO 表に保持します。この表は、キャプチャー・プログラムが初めて開始したときに同じパーティション・グループに 1 つ以上のデータベース・パーティションがあることを検出したときに、キャプチャー・プログラムによって作成されます。

キャプチャー・プログラムがウォーム・スタートされるたびに、キャプチャーはそのコントロール表があるパーティション・グループのデータベース・パーティションのリストを読み取ります。キャプチャーは DB2 に認識されているデータベース・パーティションの数と、IBMSNAP_PARTITIONINFO 表にリストされているデータベース・パーティションの数を比較します。IBMSNAP_PARTITIONINFO 表にリストされているデータベース・パーティションの数が DB2 に認識されている数と一致していなければなりません。一致していない場合、キャプチャー・プログラムは実行しません。

最後にキャプチャー・プログラムを実行した後で 1 つ以上のデータベース・パーティションを追加した場合には、新規データベース・パーティションをキャプチャー・プログラムが認識するようにならなければなりません。これはレプリケーション・センターで、「キャプチャーの開始」ウィンドウ上でいずれかのウォーム・スタート・モードに対して STARTMODE オプションを設定する際に「**キャプチャーを最後に実行した後で 1 つまたは複数のパーティションが追加された**」チェック・ボックスを選択することで行えます。レプリケーション・センターから複数のデータベース・パーティション用にキャプチャーを設定する方法については、レプリケーション・センターのヘルプを参照してください。

ジャーナルのセットアップ (OS/400)

DB2 DataPropagator for iSeries は、データに対する変更についてジャーナルから受け取る情報を使用することによって、レプリケーションのための CD および UOW 表にデータを入れます。

DB2 DataPropagator for iSeries は、大部分の操作のコミットメント・コントロール下で実行するため、コントロール表のジャーナリングを必要とします。(QSQRN ジャーナルは、**CRTDPRTBL** コマンドが集合を作成するときに作成されます。)

管理者は、ソース表、CD 表、およびターゲット表が含まれているライブラリーにジャーナルが含まれていることを確認する必要があります。すべてのソース表のジャーナル処理が正しく行われるのを見届ける責任もあります。

OS/400 でレプリケーション用の表を登録する前に、その表が変更前イメージと変更後イメージの両方についてジャーナリングされている必要があります。

以下のセクションでは、レプリケーションに必要なジャーナル・セットアップについて説明しています。

- 『ソース表用ジャーナルの作成 (OS/400)』
- 38 ページの『ジャーナルおよびジャーナル・レシーバーの管理 (OS/400)』

ソース表用ジャーナルの作成 (OS/400)

ソース表のジャーナルをセットアップするには、ソース表のジャーナルおよびジャーナル・レシーバーを定義するために作成する権限を所有している必要があります。(使用するソース表がすでにジャーナリングされている場合は、このセクションをスキップしてください。)

重要: ソース表のために使用するジャーナルは、DB2 DataPropagator for iSeries によって ASN (またはこれ以外のキャプチャー・スキーマ) ライブラリーに作成されたジャーナル以外のものにしてください。

手順:

ソース表のジャーナルを作成するには、以下のように入力してください。

1. ジャーナル・レシーバーの作成 (**CRTJNRRCV**) コマンドを使用して、選択したライブラリーにジャーナル・レシーバーを作成します。定期的に保管されるライブラリーにジャーナル・レシーバーを置きます。将来ジャーナル・レシーバーの命名規則を作成するのに使用できるようなジャーナル・レシーバー名 (**RCV0001** など) を選択します。***GEN** オプションを使用して、ジャーナル・レシーバーを変更するときに命名規則を継続することができます。このタイプの命名規則は、ジャーナル・レシーバーの変更をシステム管理に許可する場合にも役立ちます。以下の例では、ジャーナル・レシーバー用に **JRNLIB** という名前のライブラリーを使用します。

```
CRTJNRRCV JRNRCV(JRNLIB/RCV0001)
          THRESHOLD(100000)
          TEXT('DataPropagator Journal Receiver')
```

2. ジャーナルの作成 (**CRTJRN**) コマンドを使用して、ジャーナルを作成します。

```
CRTJRN JRN(JRNLIB/DJRN1)
       JRNRCV(JRNLIB/RCV0001)
       MNGRCV(*SYSTEM) DLTRCV(*YES)
       TEXT('DataPropagator Journal')
```

- ステップ 1 で作成したジャーナル・レシーバーの名前を指定します。
- アタッチされたレシーバーが大きくなりすぎる場合、システムがジャーナル・レシーバーを変更して、新しいレシーバーをアタッチするために、レシーバーの管理 (**MNGRCV**) パラメーターを使用します。このオプションを選択する

と、**CRTJRN** コマンドを使って、手動でレシーバーを切り離したり、新しいレシーバーを作成およびアタッチしたりする必要がなくなります。

- デフォルトの属性 **MINENTDTA(*NONE)** を使用します。このキーワードには、他の値は無効です。
- オーバーライドする理由がある場合 (たとえば、リカバリーさせる目的でこれらのジャーナル・レシーバーを保管する必要がある場合) にのみ、**DLTRCV(*NO)** を指定します。 **DLTRCV(*YES)** を指定すると、これらのレシーバーは、保管する機会もなく削除されてしまう可能性があります。

CRTJRN コマンドの **RCVSIZEOPT** パラメーターに 2 つの値 (***RMVINTENT** および ***MINFIXLEN**) を使用して、ストレージの可用性やシステム・パフォーマンスを最適化することができます。詳細については、「*OS/400 Programming: Performance Tools Guide*」を参照してください。

3. 以下の例で示す方法で、物理ファイルのジャーナル開始 (**STRJRNPf**) コマンドを使って、ソース表のジャーナリングを開始します。

```
STRJRNPf FILE(library/file)
          JRN(JRNLIB/DJRN1)
          OMTJRNE(*OPNCLO)
          IMAGES(*BOTH)
```

ステップ 2 (37 ページ) で作成したジャーナルの名前を指定します。キャプチャー・プログラムには、**IMAGES** パラメーターの ***BOTH** という値が必要です。

4. ソース表のジャーナリングのセットアップを以下のように変更します。
 - a. **IMAGES(*BOTH)** を使用して、ソース表に対するジャーナリングが確実に変更前イメージと変更後イメージの両方について行われるようにする。
 - b. そのジャーナルに属性 **MNGRCV(*SYSTEM)** および **DLTRCV(*YES)** が指定されていることを確認する。
 - c. そのジャーナルに **MINENTDTA(*NONE)** 属性が指定されていることを確認する。
 - d. リモート・システムのジャーナルの場合、ソース・ジャーナルに **MNGRCV(*SYSTEM)**、**DLTRCV(*YES)**、および **MINENTDTA(*NONE)** 属性を指定する。リモート・ジャーナルを定義するには、**ADDRMTJRN** コマンドに **DLTRCV(*YES)** 属性を指定する。

ジャーナルおよびジャーナル・レシーバーの管理 (OS/400)

キャプチャー・プログラムは、ジャーナル項目の受信 (**RCVJRNE**) コマンドを使って、ジャーナルを受信します。以下のセクションでは、レプリケーション環境でジャーナルおよびジャーナル・レシーバーを管理する方法が説明されています。

- 『ジャーナル・レシーバーのシステム管理の指定 (OS/400)』
- 39 ページの『実行管理機能オブジェクトの定義の変更 (OS/400)』
- 39 ページの『ジャーナル・レシーバーのユーザー管理の指定』
- 40 ページの『ジャーナル・レシーバー削除出口ルーチンの使用』

ジャーナル・レシーバーのシステム管理の指定 (OS/400)

推奨: OS/400 システムにジャーナル・レシーバーの変更を管理させてください。これは、システム変更ジャーナル管理 と呼ばれています。ジャーナルを作成するとき、またはジャーナルをその値に変更するときは、**MNGRCV(*SYSTEM)** を指定し

てください。システム変更ジャーナル管理を使用する場合、システムに変更させるジャーナル・レシーバーのしきい値を指定するジャーナル・レシーバーを作成する必要があります。このしきい値は、最低でも 5,000 KB にし、システムでのトランザクションの数に基づいて決定する必要があります。システムは、レシーバーがしきい値のサイズに達して、新しいジャーナル・レシーバーが作成およびアタッチされると (可能な場合)、そのレシーバーを自動的に切り離します。

制約事項: RTVJRNE コマンドを使用してジャーナル項目を検索する場合、同一のジャーナルおよびキャプチャー・スキーマを使用できるのは、最大 299 個のソース物理ファイルまでです。同じジャーナルに 299 個以上のファイルを登録することが必要な場合は、ソースの登録を複数のキャプチャー・スキーマに分割してください。

実行管理機能オブジェクトの定義の変更 (OS/400)

DB2 DataPropagator for iSeries のインストール時に、このライブラリー用の SQL ジャーナルと SQL ジャーナル・レシーバー、および実行管理機能オブジェクトがインストール・プログラムによって作成されます。表 2 に、作成される実行管理機能オブジェクトをリストします。

表 2. 実行管理機能オブジェクト

説明	オブジェクト・タイプ	名前
サブシステム記述	*SBSD	QDP4/QZSNDPR
ジョブ・キュー	*JOBQ	QDP4/QZSNDPR
ジョブ記述	*JOB	QDP4/QZSNDPR

上記の 3 つのタイプの実行管理機能オブジェクトについて、そのデフォルトの定義を変更しても構いませんが、独自の定義も提供できます。独自のサブシステム記述を作成する場合は、サブシステムの名前として QZSNDPR を指定して、そのサブシステムを QDP4 以外のライブラリーに作成する必要があります。これらの定義の変更についての詳細は、「iSeries 実行管理の手引き (SD88-5009)」を参照してください。

ジャーナル・レシーバーのユーザー管理の指定

ジャーナルの作成時に MNGRCV(*USER) を指定すると (独自にジャーナル・レシーバーの変更管理を行いたい場合)、ジャーナル・レシーバーがストレージのしきい値 (レシーバーに指定した場合) に達すると、ジャーナルのメッセージ・キューにメッセージが送信されます。

CHGJRN コマンドを使用して、古いジャーナル・レシーバーを切り離し、新しいレシーバーをアタッチします。このコマンドを使用すると、ジャーナルされない項目のエラー状態を避け、ジャーナルが使用するストレージ・スペースの量を制限することができます。パフォーマンスに影響が出ないようにするため、システムが最大使用回数に達していないときにこのことを行ってください。

CHGJRN MNGRCV(*SYSTEM) を指定すると、ジャーナル・レシーバーの管理をシステムが行うよう切り替えることができます。

次の 2 つの理由で、現行のジャーナル・レシーバーを定期的に切り離し、新しいレシーバーをアタッチする必要があります。

- 各ジャーナル・レシーバーに特定の管理可能な期間が含まれていると、ジャーナル項目のアナライズがより簡単になるから。
- 大きなジャーナル・レシーバーは、システム・パフォーマンスに影響を及ぼし、補助記憶域の貴重なスペースをふさいでしまう場合があるから。

ジャーナルのデフォルトのメッセージ・キューは QSYSOPR です。QSYSOPR メッセージ・キューに大きなボリュームのメッセージがあると、別のメッセージ・キュー (DPRUSRMSG など) とジャーナルを関連付けることができます。メッセージ処理プログラムを使って、DPRUSRMSG メッセージ・キューをモニターすることができます。ジャーナル・メッセージ・キューに送信できるメッセージに関する説明は、「OS/400 バックアップおよびリカバリーの手引き」を参照してください。

ジャーナル・レシーバー削除出口ルーチンの使用

DB2 DataPropagator for iSeries をインストールすると、ジャーナル・レシーバー削除 出口ルーチン (DLTJRNRCV) が自動的に登録されます。この出口ルーチンは、ジャーナル・レシーバーが削除されると、そのレシーバーがソース表のジャーナリングに使用されているかどうかにかかわらず、常に呼び出されます。この出口ルーチンは、ジャーナル・レシーバーを削除できるかどうかを決定します。

ジャーナル・レシーバー削除出口ルーチンを使用して、ジャーナル管理をシステムに任せるには、**CHGJRN** または **CRTJRN** コマンドに対して DLTRCV(*YES) および MNGRCV(*SYSTEM) を指定します。

重要: ジャーナル・レシーバー削除出口ルーチン用の登録を除去する場合は、ソース表用に使用されるすべてのジャーナルを変更して、それらが DLTRCV(*NO) 属性を持つようにする必要があります。

レシーバーが関連付けられているジャーナルがどのソース表にも関連付けられていない場合、この出口ルーチンはそのレシーバーの削除を承認 します。

ジャーナル・レシーバーが 1 つかそれ以上のソース表で使われている場合、この出口ルーチンは、キャプチャー・プログラムによって処理されなかった項目が、削除されているレシーバーに含まれていないことを確認します。キャプチャー・プログラムがそのレシーバーの項目をさらに処理する必要がある場合には、出口ルーチンはレシーバーの削除を承認 しません。

ジャーナル・レシーバーを削除する必要があるのに、ジャーナル・レシーバー出口ルーチンが削除を承認しない場合には、DLTJRNRCV DLT OPT(*IGNEXITPGM) を指定して、その出口ルーチンをオーバーライド します。

ジャーナル・レシーバー削除出口ルーチンの削除: ジャーナル・レシーバーの削除を手動で処理したい場合、以下のコマンドを使用して、ジャーナル・レシーバー削除出口ルーチンを削除 することができます。

```
RMVEXITPGM EXITPNT (QIBM_QJO_DLT_JRNRCV)
                FORMAT (DRCV0100)
                PGMNBR (value)
```

手順:

RMVEXITPGM コマンドの PGMNBR 値を決定するには、以下のようになります。

1. **WRKREGINF** コマンドを発行 します。

2. 「登録情報の処理」ウィンドウで、出口点の項目 QIBM_QJO_DLT_JRNRCV を検索します。「Opt」フィールドに 8 と入力します。
3. 「出口プログラムの処理」ウィンドウで、ライブラリー QDP4 にある出口プログラムの項目 QZSNDREP を検索します。必要な数値は、出口プログラム番号の見出しの下にあります。

ジャーナル・レシーバー削除出口ルーチンの登録: 出口点を除去した後でそれを元に戻す必要が生じた場合は **ADDEXITPGM** コマンドを使用してください。その出口ルーチンは以下のコマンドを使用して登録する必要があります。

```
ADDEXITPGM EXITPNT(QIBM_QJO_DLT_JRNRCV)
          FORMAT(DRCV0100)
          PGM(QDP4/QZSNDREP)
          PGMNBR(*LOW)
          CRTEXITPNT(*NO)
          PGMDTA(65535 10 QSYS)
```

第 3 章 表およびビューを SQL レプリケーション・ソースとして登録する

DB2 レプリケーションの場合、レプリケーション・ソースとして使用する表とビューは、それらを登録するという方法で示します。レプリケーション用として特定の表またはビューを登録する際には、後でさまざまな目的のために異なるターゲットに対して使用できる使用可能なデータのソースを作成します。この章で説明されている管理タスクは、レプリケーション・ゴールに基づいてデータが各ソースからどのようにキャプチャーされるのかを定義するコントロール情報をセットアップするのに役立ちます。

ソースの登録時には、レプリケーション・ソースとして使用する表またはビュー、表内でレプリケーションのために使用可能にする列、および DB2 レプリケーションがソースからデータと変更をキャプチャーする方法に関するプロパティを示してください。

DB2 レプリケーションでは、以下のオブジェクトをソースとして登録できます。

- DB2 表
- 非 DB2 リレーショナル表 (ニックネームを使用)
- 表内のデータのサブセット (DB2 または非 DB2 リレーショナル)
- 単一表のビュー (DB2)
- 複数の表の内部結合を表すビュー (DB2)

この章には、以下のセクションがあります。

- 『DB2 表をソースとして登録する』
- 45 ページの『非 DB2 リレーショナル表をソースとして登録する』
- 47 ページの『ソース表用の登録オプション』
- 62 ページの『レプリケーション・ソースとしてのビューの動作』
- 65 ページの『表のビューをソースとして登録する』
- 66 ページの『CCD 表をソースとして保守する (IMS)』

DB2 表をソースとして登録する

このセクションでは、DB2 表をレプリケーション・ソースとして登録する方法を説明します。DB2 レプリケーションでは、次のタイプの DB2 表がソースとしてサポートされています。

Linux、UNIX、および Windows の場合

- ユーザー・アプリケーションが保守する DB2 表
- カタログ表 (フル・リフレッシュのみのレプリケーション)
- 自動サマリー表
- 外部 CCD 表

z/OS の場合

- ユーザー・アプリケーションが保守する DB2 表
- カタログ表
- 外部 CCD 表

OS/400 の場合

- ユーザー・アプリケーションが保守する DB2 表 (ローカルまたはリモート側でジャーナリングされる)
- 外部 CCD 表

OS/400 の場合を除くすべての DB2 ソースについては、ソース表 DDL は DATA CAPTURE CHANGES オプションを必要とします。このオプションをソースから除去しないでください。

表をソースとして登録すると、CD (変更データ) 表が作成されます。登録された表に関連付けられているキャプチャー・プログラムは、そのソース用のログを読み取って、登録された列に対して発生した未完了の変更を、トランザクションがコミットまたはロールバックされるまでメモリーに保管します。ロールバックの場合は、それらの変更はメモリーから削除されます。コミットの場合は、それらの変更はキャプチャー・プログラムがコミット・ログ・レコードを読み取ると同時に CD 表に挿入されます。これらの変更は、キャプチャー・プログラムが各キャプチャー・サイクルの後でコミットするまでメモリー内に残されます。キャプチャー・プログラムは、CAPSTART シグナルがユーザーまたはアプライ・プログラムによって発行されるまで、DB2 ソース表のデータのキャプチャーを開始しません。

非リレーショナル・ソース表の場合の注: IMS などの非リレーショナル・データベース管理システムからのデータを含む DB2 表を登録できます。これを行うには、IMS DataPropagator や Data Refresher などといった、非リレーショナル・データベースからのデータを含む CCD 表を移植するためのアプリケーションが必要です。このアプリケーションは、IMS データベース内の非リレーショナル・セグメントへの変更をキャプチャーし、CCD 表を移植します。この CCD 表はコンプリートでなければなりません。コンデンスされていてもコンデンスされていなくても構いません。他の CCD ソースと同様に、CCD ソース表には関連付けられているキャプチャー・プログラムがありません。なぜなら、この表には非リレーショナル・ソース表からの変更済みデータがすでに保管されているからです。IMS DataPropagator および Data Refresher 製品は、登録 (IBMSNAP_REGISTER) 表内の値を保守して、アプライ・プログラムがこのソース表からの読み取りを正しく行えるようにします。上記の製品のいずれかを使用してこれらのタイプの CCD 表を保守するのではなく、自分自身でそれらの表を保守する場合は、66 ページの『CCD 表をソースとして保守する (IMS)』を参照してください。

前提条件:

ソースとして登録する表を処理させるキャプチャー・コントロール・サーバー上に、キャプチャー・コントロール表がすでに存在していなければなりません。キャプチャー・コントロール表を作成する必要がある場合は、26 ページの『レプリケーション・コントロール表のセットアップ』を参照してください。

制約事項 (OS/400):

- SQL ステートメントは長さが 32,000 文字までに制限されているため、1 つの表当たりおよそ 2000 列までしか登録できない。正確な列数は、列名の長さによって決まります。
- 単一キャプチャー・スキーマに対し、同じジャーナルを使用するソース表を 300 個より多く登録してはならない。
- ソース表、CD 表およびソース表のジャーナルは、すべてこれらのソース表の登録情報を含むキャプチャー・コントロール表と同じ補助記憶域プール (ASP) に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、DB2 表を登録してください。

レプリケーション・センター

「表の登録」ウィンドウを使用します。詳しくは、レプリケーション・センターのヘルプを参照してください。

ヒント: 登録の際の時間を節約するために、キャプチャー・コントロール・サーバーに対してあらかじめソース・オブジェクト・プロファイルを設定アップしておけます。そのようにした場合は、表の登録時に、レプリケーション・センターはレプリケーション・センターのデフォルトではなく、そのソース・オブジェクト・プロファイルに定義されたデフォルトを使用します。これによって登録の際の時間を節約できます。なぜなら、それぞれの表を一度に 1 つずつ選択してデフォルト設定を手動で変更する代わりに、デフォルトを一度に上書きできるからです。

レプリケーション用のシステム・コマンド (OS/400)

ADDDPRREG システム・コマンドを使用します。このコマンドの構文およびパラメーターについての記述は、386 ページの『ADDDPRREG: DPR 登録の追加 (OS/400)』を参照してください。

DB2 表の登録時には、ソース・サーバー、ソース表の名前、およびキャプチャー・スキーマを指定して、どの表を登録するのかを示してください。別々のキャプチャー・スキーマを使用して、同じ表を複数回登録できます。登録用のデフォルト設定を使用するか、あるいはレプリケーションに関するニーズに合わせて登録オプションを変更するかは任意です。登録オプションとそれらのデフォルトの完全なリスト、およびこれらのデフォルトを使用した方がよい場合と変更した方がよい場合についての説明は、47 ページの『ソース表用の登録オプション』を参照してください。

非 DB2 リレーショナル表をソースとして登録する

このセクションでは、非 DB2 リレーショナル表をレプリケーション・ソースとして登録する方法を説明します。DB2 レプリケーションは、ニックネームを使用して非 DB2 リレーショナル表にアクセスします。

非 DB2 リレーショナル表をソースとして登録すると、CCD (整合変更データ) 表が作成されます。登録済みの非 DB2 リレーショナル表に対する変更が発生すると、キャプチャー・トリガーがキャプチャー・プログラムをシミュレートして、その変

更を CCD 表に挿入します。キャプチャー・トリガーは、非 DB2 リレーショナル・ソース表への変更のキャプチャーを、そのソースが登録された時点から開始します。

デフォルトでは、CCD 所有者はソース表のスキーマ名から導き出されます。 CCD 所有者をスキーマ名と一致しないように変更する場合は、ソース表の所有者が CCD 表への書き込みを許可されていることを確認してください。ソース表の所有者が CCD 表を更新できない場合は、ソース表のトリガーは変更を CCD 表に書き込めません。

前提条件:

このソースを処理させるキャプチャー・コントロール・サーバー上に、キャプチャー・コントロール表がすでに存在していなければなりません。キャプチャー・コントロール表を作成する必要がある場合は、28 ページの『非 DB2 リレーショナル・ソース用のコントロール表の作成』を参照してください。

制約事項:

- 複数の非 DB2 リレーショナル・ソース・サーバーにアクセスするのに単一のフェデレーテッド DB2 データベースを使用している場合は、その単一フェデレーテッド・データベースではそれぞれの非 DB2 リレーショナル・ソース・サーバーごとに異なるキャプチャー・スキーマを使用する必要があります。どれも同じであってはなりません。各非 DB2 リレーショナル表は、それぞれ 1 つのキャプチャー・スキーマの下にしか登録できません。
- データ・タイプ LOB または DATALINK を持つ非 DB2 リレーショナル表には列を登録できない。これらのデータ・タイプを含む表を登録する場合は、列のサブセットを登録する必要があります。列のサブセットだけを登録する方法についての詳細は、47 ページの『列のサブセットの登録 (垂直方向のサブセット化)』を参照してください。

手順:

以下に、非 DB2 リレーショナル表を登録する方法を示します。

レプリケーション・センター

「ニックネームの登録」ウィンドウを使用します。詳しくは、レプリケーション・センターのヘルプを参照してください。

ヒント: 登録の際の時間を節約するために、キャプチャー・コントロール・サーバーに対してあらかじめソース・オブジェクト・プロファイルを設定アップしておけます。そのようにした場合は、表の登録時に、レプリケーション・センターはレプリケーション・センターのデフォルトではなく、CCD 表用のソース・オブジェクト・プロファイルに定義されたデフォルトと CCD 表のニックネームを使用します。これによって登録の際の時間を節約できます。なぜなら、それぞれの表を一度に 1 つずつ選択してデフォルト設定を手動で変更する代わりに、デフォルトを一度に上書きできるからです。

非 DB2 リレーショナル表の登録時には、ソース表のニックネームを指定して、どの表を登録するのかを示してください。登録用のデフォルト設定を使用するか、あるいはレプリケーションに関するニーズに合わせて登録オプションを変更するかは任意です。登録オプションとそれらのデフォルトの完全なリスト、およびこれらの

デフォルトを使用した方がよい場合と変更した方がよい場合についての説明は、『ソース表用の登録オプション』を参照してください。

ソース表用の登録オプション

このセクションでは、表をレプリケーション・ソースとして登録する際に使用できるさまざまなオプションについて説明します。これらのオプションは、表の登録に関するタスクのうち、より大きな方のタスクの一部です。各表の登録方法の参照先を以下に示します。

- DB2 表、43 ページの『DB2 表をソースとして登録する』を参照してください。
- 非 DB2 リレーショナル表、45 ページの『非 DB2 リレーショナル表をソースとして登録する』を参照してください。

表のビューを作成してそれらのビューをソースとして登録するときには、ビューの登録オプションは基本表の登録定義によって決定されます。ビューが基本表からどの特性を継承するのか、およびビューが基礎登録定義に従ってどのように動作するのかについての詳細は、62 ページの『レプリケーション・ソースとしてのビューの動作』を参照してください。

登録する表の選択を済ませたら、レプリケーションのためにどの列を使用可能にするのかを示して、このソースから登録されたデータの処理方法と保管方法を決定するプロパティを定義できます。また、キャプチャー・プログラムによる CD 表へのソース・データの保管方法 (またはキャプチャー・トリガーによる CCD 表へのデータの保管方法) などといった、他の登録オプションも指定できます。このセクションでは、表をソースとして登録する際に指定可能な以下のオプションについて説明します。

- 『列のサブセットの登録 (垂直方向のサブセット化)』
- 48 ページの『フル・リフレッシュ・コピーと変更キャプチャー・レプリケーション』
- 50 ページの『変更後イメージ列と変更前イメージ列』
- 52 ページの『変更前イメージ接頭部』
- 53 ページの『エラー発生時におけるキャプチャー・プログラムの停止』
- 54 ページの『キャプチャー・プログラムが更新を保存する方法』
- 55 ページの『変更の再キャプチャーの防止 (Update-anywhere レプリケーション)』
- 59 ページの『競合検出の設定 (Update-anywhere レプリケーション)』
- 62 ページの『主キーの代わりに相対レコード番号 (RRN) を使用する (OS/400)』

列のサブセットの登録 (垂直方向のサブセット化)

デフォルト: すべての列がレプリケーションのために登録されます

レプリケーション用のソース表を定義するときには、その表内のすべての列をレプリケーションのために登録しなければならないわけではなく、ソース表の列のサブセットを登録できます。この垂直サブセットは、サブスクライブするターゲットからすべての列を使用可能にするのは望ましくない場合に有用です。また、このオプ

ションは、このソースのターゲット表ではソース表に定義されているデータ・タイプのすべてはサポートされていない場合にも選択するとよいでしょう。

列のサブセットを登録するには、ターゲット表へのレプリケーションのために使用可能にしたい列だけを選択します。選択しない列は、どのターゲット表へのレプリケーションにも使用できません。CD (および CCD) 表には一部のタイプのターゲット表にとって十分なキー・データ (ポイント・イン・タイムなど) が含まれていないため、使用するサブセットにターゲットに対してキー列 (主キーまたはユニーク索引) として機能する列が含まれていることを確認してください。

ヒント: ソース表内の列のサブセットを登録するのは、未登録の列は決して複製しないことを確信している場合だけにしてください。ソースの列のサブセットを登録し、後になって登録しなかった列を複製する必要がある場合は、登録を変更して未登録の列を追加しなければなりません (非 DB2 リレーショナル・ソースの場合は、1 つの登録に新しい列を追加するのにすべての登録をまとめて再定義する必要があります)。このソースに関連した内部 CCD を持つ予定である場合は、後で列を追加するのはさらに困難になる可能性があります。なぜなら、新しい列を登録すると、それらは内部 CCD ではなく、CD 表に追加されるからです。これらの問題を回避するためには、ソースのすべての列を登録し、代わりにアプライ・プログラムを使用して、ターゲットに複製する列をサブセット化するという方法もあります。ソースでなくターゲットでサブセット化を行う方法についての詳細は、97 ページの『ターゲットに適用するソース列』を参照してください。

フル・リフレッシュ・コピーと変更キャプチャー・レプリケーション

デフォルト: 変更キャプチャー・レプリケーション

レプリケーション・サイクルで毎回ソース表内のすべてのデータをターゲットに複製する (フル・リフレッシュのみのレプリケーション) か、ターゲットが最後にリフレッシュされた時点以降に発生した変更のみをレプリケーション (変更キャプチャー・レプリケーション) するかを選択できます。

フル・リフレッシュのみのレプリケーション

ターゲットがフル・リフレッシュのみのレプリケーション用に登録されているソースに対してサブスクライブしている場合、アプライ・プログラムは毎回のレプリケーション・サイクルで、ターゲット表からすべてのデータを削除し、ソースの登録済み列に含まれているデータをコピーして、ソース・データをターゲットに移植します。キャプチャー・プログラムは関与せず、CD 表もありません。アプライ・プログラムは直接ソース表からデータを読み取ります。

表が小さい場合のヒント: コピーするのに時間もリソースもあまり要しない非常に小さなソース表を使用している場合は、フル・リフレッシュのみのレプリケーションを選択するとよいでしょう。

表が大きい場合のヒント: 表が比較的大きく、かつフル・リフレッシュのみのレプリケーションを使用したい場合は、表をより迅速にロードするために ASNLOAD 出口ルーチンを使用するとよいでしょう。詳細については、164 ページの『ASNLOAD 出口ルーチンを使ったターゲット表のリフレッシュ』を参照してください。

制約事項: このソースにサブスクライブするのをコンデンスされたターゲット表にする予定であり、さらにそのターゲット表のユニーク索引は用意できない場合は、ソースをフル・リフレッシュのみのレプリケーション用として登録する必要があります。

変更キャプチャー・レプリケーション

デフォルト: すべての行に対する変更がキャプチャーされます

変更キャプチャー・レプリケーション時には、変更されたデータだけがターゲット表に複製されます。このソース用に選択したターゲット表のタイプによっては、この表の初期ロードを実行する必要があります。ほとんどの場合、アプライ・プログラムは初期フル・リフレッシュを実行してから、継続して変更キャプチャー・レプリケーションを実行をします。

ターゲット表に対するフル・リフレッシュを許可しないことを選択すると、ソース表とターゲット表の再同期化が必要になった場合にターゲット表を手動で再ロードする必要があります。ターゲットが初期ソース・データを含んでいる状態でロードされた後、キャプチャー・プログラムはソースで発生した変更をキャプチャー CD 表に保管します。非 DB2 リレーショナル・ソースの場合の変更キャプチャー・レプリケーションでは、ソースで発生した変更をキャプチャー・トリガーがキャプチャー CCD 表に保管します。アプライ・プログラムは、CD 表または CCD 表から変更を読み取って、その登録されているソースにサブスクライブしているターゲットにそれらの変更を適用します。

DB2 ソース表を変更キャプチャー・レプリケーション用として定義する場合、ソースで発生した変更をすべて CD 表に保管するのは望ましくないことがあります。行(水平方向)サブセットを登録して変更をフィルター操作し、実際にソースで発生している変更より少ない変更が CD 表にキャプチャーされるようにできます。以下の 2 つの行キャプチャー規則のいずれかを選択して、キャプチャー・プログラムがソース表からのどの変更済み行を CD 表に記録するのかを決定できます。

- すべての行に対する変更をキャプチャーする
- 変更が登録済みの列に発生した場合にのみ変更をキャプチャーする (DB2 のみ)

デフォルトでは、ソース表で行のいずれかの列 (登録済みでも未登録でも) が更新されるたびに、必ず変更がキャプチャーされます。列のサブセットだけを登録した場合は、ソース表に対して変更が発生するたびに、キャプチャー・プログラムは登録済み列の値を行単位で CD 表に記録します。これは、変更された列が登録済み列とは異なる列であった場合でも実行されます。このデフォルト・オプションは、ソース表へのすべての変更の履歴を保持したい場合に使用してください。非 DB2 リレーショナル・ソースの場合に使用可能なオプションはこれだけです。キャプチャー・トリガーは、変更が未登録の列に対して発生した場合であっても、ソースで変更のあったすべての行をキャプチャーします。

例: 表には 100 個の列があり、それらの列うち 50 個をレプリケーション用に登録しているとします。デフォルトでは、表内の 100 個の列のどれか 1 つにでも変更が加えられると、いつでもキャプチャー・プログラムによって行が CD 表に書き込まれます (あるいは、キャプチャー・トリガーによって行が CCD 表に書き込まれます)。

DB2 ソースがある場合は、登録済みの列への変更だけをキャプチャー・プログラムにキャプチャーしたいことがあります。このケースでは、キャプチャー・プログラムは登録済みの列に対する変更が発生したときだけ行を CD 表に書き込みます。

提案: 監査を目的とする情報が必要な場合、または表ではほとんどいつも登録済みの列に対してのみ変更が発生している場合は、すべての行への変更のキャプチャーを選択してください。未登録の列にのみ影響する変更が頻繁に発生している場合は、登録済みの列だけに対する変更のキャプチャーを選択してください。このオプションは、ソース表への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。

変更後イメージ列と変更前イメージ列

デフォルト: 変更後イメージ列のみ

ソースを変更キャプチャー・レプリケーション用に登録する際には、キャプチャー・プログラムに変更後イメージ値 (変更後の列内の値) だけをキャプチャーするか、あるいは変更後イメージ値と変更前イメージ値 (変更前に列内に入っていた値) の両方をキャプチャーするのを選択できます。Linux、UNIX、Windows、および z/OS の場合は、変更前イメージ値を取り組むかどうかを表内の列ごとに選択できます。OS/400 の場合は、変更前イメージのキャプチャーを表内のすべての列について行うか、あるいはどの列に対しても行わないかのどちらかを選択できます。このオプションは、個々の列ごとには選択できません。以下の各セクションでは、どのような場合に各オプションを選択する必要があるのかについて説明します。

以下に示すように、いくつかの非 DB2 リレーショナル・ソース表の場合、CCD 表には変更後イメージ値だけを含めるようにする必要があります。

- Sybase または Microsoft SQL Server の表には、タイプが TIMESTAMP の列は 1 つしか含められない。データ・ソースが Sybase または Microsoft SQL Server であり、かつソース表にタイプが TIMESTAMP の列があるときは、この列をレプリケーション・ソースの一部として定義する場合、この列については変更後イメージのみを選択してください。

以下の特定のデータ・タイプを指定された列の場合は、CD 表に変更前イメージ値を含められません。

- LOB データ・タイプの列
- DATALINK データ・タイプの列

変更後イメージ値のみのキャプチャー

変更キャプチャー・レプリケーション用に登録するそれぞれの列ごとに、変更が行われるたびにキャプチャー・プログラムまたはキャプチャー・トリガーに対して、変更後イメージ値だけを記録させることを選択することができます。変更後イメージ値だけを取り組むことを選択した場合は、CD (または CCD) 表にはそれぞれの変更された値ごとに 1 つの列が含められ、その列には変更発生後のソース列の値が保管されます。

このソースに対しては基礎集約ターゲット表タイプおよび変更集約ターゲット表タイプだけを使用する予定である場合は、変更前イメージは必要ありません。ターゲット表を計算済みの値のために使用する予定である場合は、変更前イメージ列は無

意味です。なぜなら、計算された列には変更前イメージがないからです。他のすべてのターゲット表タイプでは、変更前イメージ列を利用することができます。集約ターゲット表についての詳細は、88 ページの『ソースにおけるデータまたは変更の算出サマリー』を参照してください。

変更前イメージ値と変更後イメージ値のキャプチャー

変更キャプチャー・レプリケーション用に登録するそれぞれの列ごとに、変更が行われるたびにキャプチャー・プログラムまたはキャプチャー・トリガーに対して、変更前イメージ値と変更後イメージ値の両方を記録させることを選択することができます。変更前イメージ値と変更後イメージ値をキャプチャーすることを選択した場合は、CD (または CCD) 表にはそれぞれの変更された値ごとに 2 つの列が含まれます。一方の列は変更が発生する前にソース列に入っていた値用で、もう一方の列は変更発生後の値用です。

変更前イメージと変更後イメージの両方を CD (または CCD) 表に保管することを選択した場合は、以下に示すように、変更前イメージ列と変更後イメージ列には、ソース表に対して実行されたそれぞれ異なるアクションの種類に応じて異なる値が含まれます。

アクション 列値

挿入 変更前イメージ列には NULL 値が含まれます。変更後イメージ列には挿入された値が含まれます。

更新 変更前イメージ列には、変更が発生する前の列値が含まれます。変更後イメージ列には、変更が発生した後の列値が含まれます。

更新を削除と挿入の対としてキャプチャーすることを選択した場合は、削除行ではその行の変更前イメージ列と変更後イメージ列の両方に更新の変更前イメージが含まれ、挿入行では変更前イメージ列には NULL 値が、変更後イメージ列には変更後イメージが含まれます。このオプションの詳細については、54 ページの『キャプチャー・プログラムが更新を保存する方法』を参照してください。

削除 変更前イメージ列と変更後イメージ列に、変更が発生する前の列値が含まれます。

Linux、UNIX、Windows、および OS/400 の場合の重要事項: 定義によって変更前イメージが含まれている列の場合、DB2 レプリケーションでは列名が 29 文字までに制限されます。なぜなら、列名全体で許容される文字数が 30 文字だけだからです。列名がそれより長いと、左方から切り捨てるようにプロファイルを設定していないかぎり、デフォルトでは余分な文字が DB2 レプリケーションによって右方から切り捨てられます。DB2 レプリケーションによってターゲット列に変更前イメージ列 ID (通常は X) が追加される上に、各列名は必ずユニークでなければならぬため、29 文字より長い列名は使用できません。複製するつもりがない表にはさらに長い列名を使用できますが、その列を将来複製する可能性がある場合は 29 文字の名前を使用することを検討してください。

z/OS の場合の重要事項: DB2 for z/OS の表であれば 18 文字の列名を使用できますが、DB2 DataPropagator は 18 番目の文字をターゲット表では変更前イメージ列 ID に置き換えるため、列名の最初の 17 文字が必ずユニークになる必要があります。

以下のセクションでは、変更前イメージ値のキャプチャーが必要となる可能性のあるケースについて説明しています。

- 『ソース・データの履歴を保持する場合』
- 『競合検出を使用する Update-anywhere 構成の場合』
- 『ターゲットでキー列が更新の対象となっている場合』

ソース・データの履歴を保持する場合: 監査目的でデータを保持する必要がある場合は、ある期間内にデータがどのように変更されたかについてのレコードを持てるように、変更前イメージと変更後イメージの両方を選択します。監査やアプリケーション・ロールバック機能を必要とする業界では、変更前イメージと変更後イメージのコピーのセットが役立ちます。

競合検出を使用する Update-anywhere 構成の場合: レプリカ表 (競合検出が None 以外に設定されているもの) の間で競合が起こり得る Update-anywhere 構成では、レプリカの CD 表に変更後イメージ列と変更前イメージ列の両方を登録して、競合が発生した場合には変更をロールバックできるようにする必要があります。

ターゲットでキー列が更新の対象となっている場合: ソースを登録する際には、この表をソースとして使用して定義する可能性のあるターゲット表があるかどうかを検討してください。通常ターゲット表はコンデンスされており、そのターゲット表内の各行をユニークにする列または列のセットを必要とします。それらのユニーク列によっていわゆるターゲット・キーが構成されます。これらのターゲット・キー列のいずれかがソースで更新される可能性がある場合は、DB2 レプリケーションではターゲット表で必ず正しい行が更新されるようにするための特殊な処理が必要となります。DB2 レプリケーションがターゲット表で必ず正しい行を新しいキー値で更新するようにするために、ターゲット・キーを構成する列については変更後イメージと変更前イメージの両方をキャプチャーすることを選択できます。これらの登録済み列の変更前イメージ値は、アプライ・プログラムが非キー・ソース列の変更をターゲット表のターゲット・キー列に適用するとき必要となります。変更を適用するときには、アプライ・プログラムはこの行をターゲット表でソースの CD (または CCD) 表内の変更前イメージ値と一致するターゲット・キー値を探すという方法で検索し、次にそのターゲット行をソースの CD (または CCD) 表内の変更後イメージ値で更新します。

これらの変更前イメージ値をソース表またはビューの登録時に登録しても、DB2 レプリケーションはユーザー・アプリケーションがターゲット・キーに対して更新を行うことを知りません。後で (サブスクリプション・セットを作成して) どのターゲットをこのソースにサブスクライブするのかを定義するときに、アプライ・プログラムに対し、変更をソースの非キー列からターゲットのキー列に適用する際に特殊な更新を実行するよう指定できます。詳細については、101 ページの『アプライ・プログラムがターゲット・キー変更オプションを使用してターゲット・キー列を更新する方法』を参照してください。

変更前イメージ接頭部

デフォルト (レプリケーション・センター): X

デフォルト (OS/400 システム・コマンド): @

CD (または CCD) 表の変更後イメージ列と変更前イメージ列の両方をキャプチャーすることを選択した場合は、変更後イメージ列名はソース表での列名となり、変更前イメージ列名は先頭に 1 文字の接頭部を追加されたソース表での列名となります。変更前イメージ列名用のデフォルトの 1 文字接頭部は変更可能です。変更前イメージ接頭部と CD (または CCD) 列名の組み合わせは、あいまいさのないものでなければなりません。これは、接頭部列名は CD (または CCD) 表内の現行列名または潜在的な列名と同じであってはならないという意味です。

制限: 変更前イメージ接頭部に、ブランク文字は使用できません。

例: 変更前イメージ接頭部として X を使用していて、さらに COL という名前のソース列を登録している場合は、XCOL という名前の列は登録できません。なぜなら、XCOL が別のソース列の実際の列名なのか、それとも COL という列名と変更前イメージ接頭部 X を持つ変更前イメージ列の名前なのかがはっきりしないからです。

表の変更前イメージ列を一切複製していない場合は、変更前イメージ接頭部を持たないことを選択して、このプロパティを NULL に設定できます。

エラー発生時におけるキャプチャー・プログラムの停止

デフォルト: キャプチャー・プログラムは特定のエラーを検出したときに停止します

登録の処理中に特定の問題を検出すると、キャプチャー・プログラムは停止するか、または実行を継続します。以下のオプションのいずれかを選択して、登録されたソースの処理中に特定のエラーを検出した場合のキャプチャー・プログラムの反応を決定できます。

エラー発生時にキャプチャーを停止

キャプチャー・プログラムは、キャプチャー・トレース (IBMSNAP_CAPTRACE) 表にエラー・メッセージを書き込んで終了します。

このオプションは、以下の致命的エラーが発生したときにキャプチャー・プログラムを停止します。

- CD 表スペースがいっぱいになった。
- SQLCODE-911 エラーが 1 行で 10 回発生した。
- 予期しない SQL エラーが発生した。

このオプションは、以下のような特定の非致命的エラーが発生してもキャプチャー・プログラムを停止しません。

- SQLCODES が無効なデータ長を示す。
- z/OS の下で実行されるキャプチャー・プログラムの場合、コンプレッション・ディクショナリーが存在しない。

これらの非致命的エラーが発生すると、キャプチャー・プログラムは登録を無効にし、実行を継続します。

エラー発生時にキャプチャーを停止しない

特定のエラーが発生した場合、キャプチャー・プログラムは実行を継続します。ソース処理の初回の試行中にエラーを検出した場合は、キャプチャー・プログラムは登録を活動化しません。登録されたソースがすでにアクティブ

化されていた場合は、登録の処理を停止します。いずれかの場合に登録は停止します。停止された登録は、登録 (IBMSNAP_REGISTER) コントロール表の STATE 列に "S" (停止された) の値を取ります。

このオプションは、以下の非致命的エラーが発生してもキャプチャー・プログラムを停止しません。

- 登録が正しく定義されていない。
- キャプチャー・プログラムが変更済みデータの行の挿入を試行したときに、CD 表が検出されなかった。
- キャプチャー・プログラムが始動または再初期化されたときに、(非 OS/400) ソース表の DATA CAPTURE CHANGES オプションがオフにされた状態で検出された。

エラーのためにサブスクリプション・セット・メンバーの登録状態が停止状態にあると、アプライ・プログラムはサブスクリプション・セットを処理できません。

キャプチャー・プログラムが更新を保存する方法

デフォルト: 更新は CD 表の単一行に保管されます

ソースの更新を CD (または CCD) 表に保管する方法を選択できます。ソース表に対する更新を取り込んだ場合、キャプチャー・トリガーまたはキャプチャー・プログラムは、更新された値を CD 表の単一行に保管するか、あるいは CD (または CCD) 表の 2 つの行を使用して、削除を一方の行に、挿入をもう一方の行に保管するかのいずれかを実行できます。デフォルトでは、更新は単一行に保管されます。このデフォルトを使用すると、ストレージが削減され、パフォーマンスが向上します。なぜなら、CD (または CCD) 表には 1 行だけが保管され、さらにその行は変更が行われるたびにアプライ・プログラムによって読み取られるからです。ただし、キャプチャー・プログラムまたはキャプチャー・トリガーに対してソース表への更新を DELETE と INSERT の対としてキャプチャーするよう指示する必要があるシナリオがいくつかあります。

ソース・アプリケーションがサブスクリプション・セット・メンバーの述部で参照される列を 1 つ以上更新する場合は、更新を DELETE ステートメントと INSERT ステートメントとしてキャプチャーする必要があります。ソース・データだけにサブスクライブするターゲットを、特定の列値を基にした述部を使用して定義する (たとえば、WHERE DEPT = 'J35') 予定であるとし、その列を変更すると (たとえば、DEPT='FFK' に変更)、キャプチャーされた変更は、ターゲットに複製する対象として選択されません。なぜなら、その列は述部の基準に合致していないからです。つまり、新しい FFK 部門は、サブスクリプション・セット・メンバーが部門 J35 に基づいているため複製されません。更新を DELETE および INSERT の対に変換すると、そのターゲット表行は確実に削除されます。

キャプチャーされたそれぞれの更新は、CD (または CCD) 表の 2 つの行に変換されます。この変換は、すべての列について実行されます。この取り込んだデータの増加に合わせて CD (または CCD) 表のスペース割り振りの調節が必要になる場合もあります。

DATALINK 値の場合の重要事項: DATALINK 列が ON UNLINK DELETE として定義されている場合、DELETE と INSERT の対が同じトランザクション内で処理されるため、このリンク解除は無視されます。外部ファイルは削除されませんが、更新は行われます。

変更の再キャプチャーの防止 (Update-anywhere レプリケーション)

新しいソース表の場合のデフォルト: 変更が再キャプチャーされます

新しいレプリカ表の場合のデフォルト: 変更は再キャプチャーされません

制約事項: 非 DB2 リレーショナル・データベースからの表は、Update-anywhere にかかわれません。したがって、このオプションは DB2 ソース専用です。

Update-anywhere レプリケーションの場合は、変更はマスター表または関連したレプリカ表で発生する可能性があります。Update-anywhere レプリケーションで使用する予定の表を登録する場合、DB2 レプリケーションはその表がユーザーの構成の中でマスター表となることを前提とします。再キャプチャー・オプションを使用すると、あるサイトで発生して 2 番目のサイトに複製された変更を、その 2 番目のサイトで再キャプチャーし、それによってその変更を追加の各サイトへのレプリケーションに使用できるようにするかどうかをコントロールできます。登録時に、このオプションをマスター表に対して設定してください。その後、そのマスター・ソース表をレプリカ・ターゲットにマップする際に、レプリカの変更を再キャプチャーして他の表に転送するかどうかを設定できます。(マスターをレプリカにマップする方法についての詳細は、94 ページの『読み取り/書き込みターゲットの定義 (Update-anywhere)』を参照してください。)

Update-anywhere 構成でマスターとして機能するソース表を登録する場合は、以下の 2 つのオプションのうち、どちらかを選択できます。

マスターで変更を再キャプチャーする

あるレプリカで発生したマスターに対する更新は、マスターで再キャプチャーされて、他のレプリカに転送されます。

マスターで変更を再キャプチャーしない

あるレプリカで発生してマスターに加えられた更新は、マスターで再キャプチャーされず、他のレプリカに転送されません。

Update-anywhere 構成のレプリカ表を登録する場合は、以下の 2 つのオプションのうち、どちらかを選択できます。

レプリカで変更を再キャプチャーする

マスターで発生してレプリカに加えられた更新は、そのレプリカで再キャプチャーされて、このレプリカにサブスクライブしている他のレプリカに転送されます。

レプリカで変更を再キャプチャーしない

マスターで発生してレプリカに加えられた更新は、そのレプリカで再キャプチャーされず、このレプリカにサブスクライブしている他のレプリカに転送されません。

変更の再キャプチャーを防止すると、パフォーマンスを向上させ、さらにストレージ・コストを削減できます。なぜなら、キャプチャー・プログラムが同じ変更をそれぞれのレプリカごとに再キャプチャーしないからです。

以下のセクションでは、Update-anywhere 構成に基づいて変更を再キャプチャーするかどうかを決定する方法について説明しています。

- 『レプリカを 1 つだけ持つマスターの場合』
- 『マスターの相互に排他的なパーティションである複数のレプリカの場合』
- 57 ページの『変更を複数のレプリカに複製するマスターの場合』
- 58 ページの『他のレプリカに変更を複製するレプリカの場合 (multi-tier)』

レプリカを 1 つだけ持つマスターの場合

マスター: 変更をマスターで再キャプチャーしません

単一レプリカ: 変更をレプリカで再キャプチャーしません

Update-anywhere 構成内にレプリカを 1 つしか持たない予定である場合は、マスター表でもレプリカ表でも変更が再キャプチャーされないようにしたいことがあります。マスター表が他のレプリカ表にとってのソースではなく、レプリカが他のレプリカにとってのソースではない場合 (multi-tier 構成において) は、これが最良の設定です。関与するのがこの 2 つの表だけである場合は、レプリカで発生した変更をマスターで再キャプチャーする必要は無く、またマスターで発生したいかなる変更も単一レプリカで再キャプチャーする必要がありません。

マスターの相互に排他的なパーティションである複数のレプリカの場合

マスター: 変更をマスターで再キャプチャーしません

レプリカ: 変更をレプリカで再キャプチャーしません

マスター表のパーティションである複数のレプリカを持つ予定である場合は、マスター表および各レプリカ表の両方で変更が再キャプチャーされないようにすることができます。どのレプリカも他のレプリカ表にとってのソースではない場合は、これが最良の設定です。レプリカがマスターのパーティションである場合は、複数のレプリカがマスターの同じデータにサブスクライブすることがあってはなりません。したがって、どのレプリカで発生したどの変更もマスターで再キャプチャーする必要は無く、他のレプリカに転送する必要もありません。なぜなら、そのソース・データにサブスクライブしているのはその変更が発生したレプリカだけだからです。

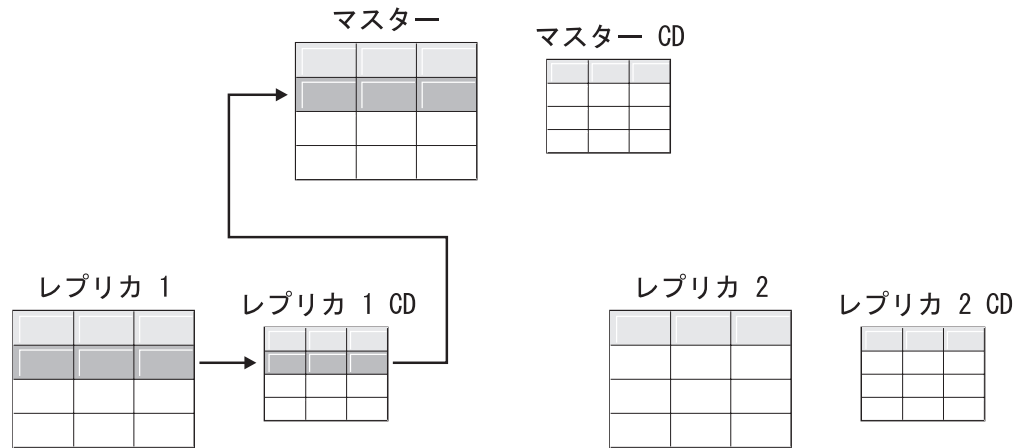


図1. マスターの相互に排他的なパーティションであるレプリカ用の再キャプチャー・オプション： マスターの同じデータにサブスクライブしていない複数のレプリカを持っている場合は、どの表に対してもこの再キャプチャー・オプションを使用する必要はありません。

変更を複数のレプリカに複製するマスターの場合

マスター: 変更をマスターで再キャプチャーします

レプリカ: 変更をレプリカで再キャプチャーしません

マスター表の同じデータにサブスクライブする複数のレプリカを持つ予定である場合は、キャプチャー・プログラムに変更をマスターで再キャプチャーさせる必要がある場合があります。そのようにすると、レプリカで発生した変更はマスターで再キャプチャーされて、その更新されたマスター・データにサブスクライブしている他のレプリカへと複製されます。

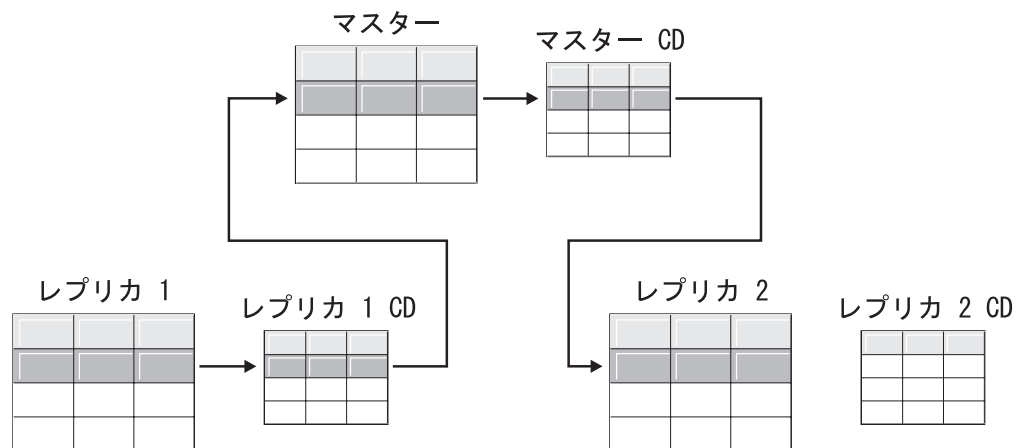


図2. 変更を複数のレプリカに複製するマスター用の再キャプチャー・オプション： マスターの同じデータにサブスクライブしている複数のレプリカを持っている場合は、マスターで再キャプチャー・オプションを使用して、あるレプリカで発生した変更がマスターで再キャプチャーされて他のレプリカ表に転送されるようにできます。

他のレプリカに変更を複製するレプリカの場合 (multi-tier)

マスター: 変更をマスターで再キャプチャーしません

レプリカ: 変更をレプリカで再キャプチャーします

マスター (層 1) があるレプリカ (層 2) に対してソースとして機能し、次にそのレプリカが同様に別のレプリカ (層 3) に対してソースとして機能する multi-tier 構成を持てます。このタイプの構成を持つ予定である場合は、キャプチャー・プログラムに中間のレプリカ (層 2) で変更を再キャプチャーさせて、マスターで発生した変更がその次のレプリカ (層 3) に転送されるようにしたい場合があります。

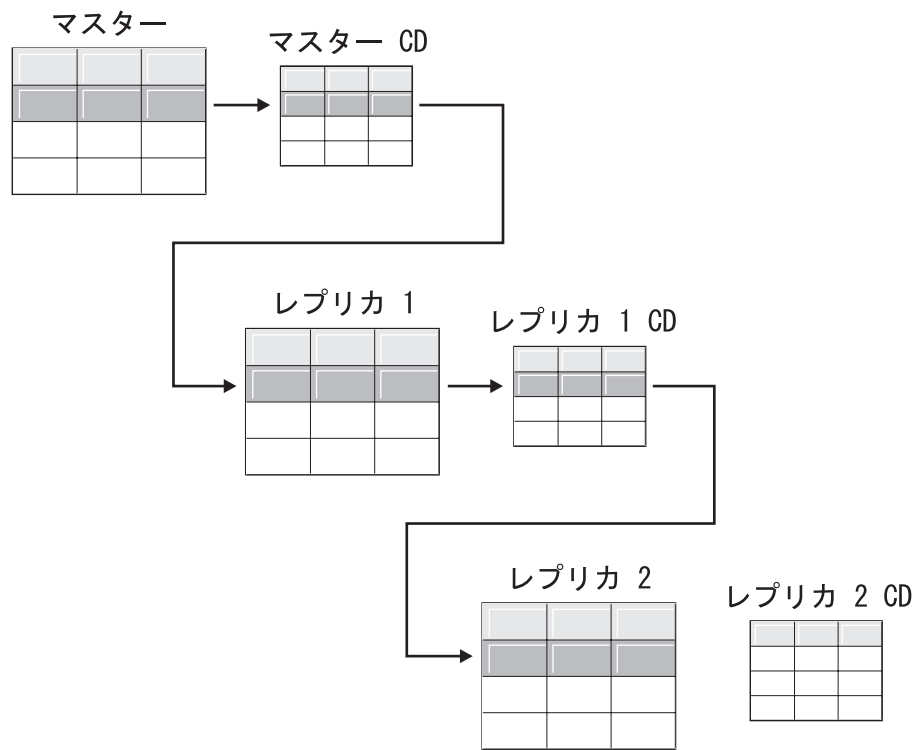


図3. 層 2 で再キャプチャー・オプションを使用すると、層 1 での変更が層 3 にまで複製されるようになります。multi-tier 構成で中間層として機能するレプリカ表がある場合は、そのレプリカで再キャプチャー・オプションを使用して、マスターで発生した変更が中間層のレプリカで再キャプチャーされて下層のレプリカに転送されるようになります。

また、中間のレプリカ (層 2) に再キャプチャーを設定してある場合は、最終のレプリカ (層 3) で発生した変更が中間のレプリカ (層 2) で再キャプチャーされてマスター (層 1) に転送されます。

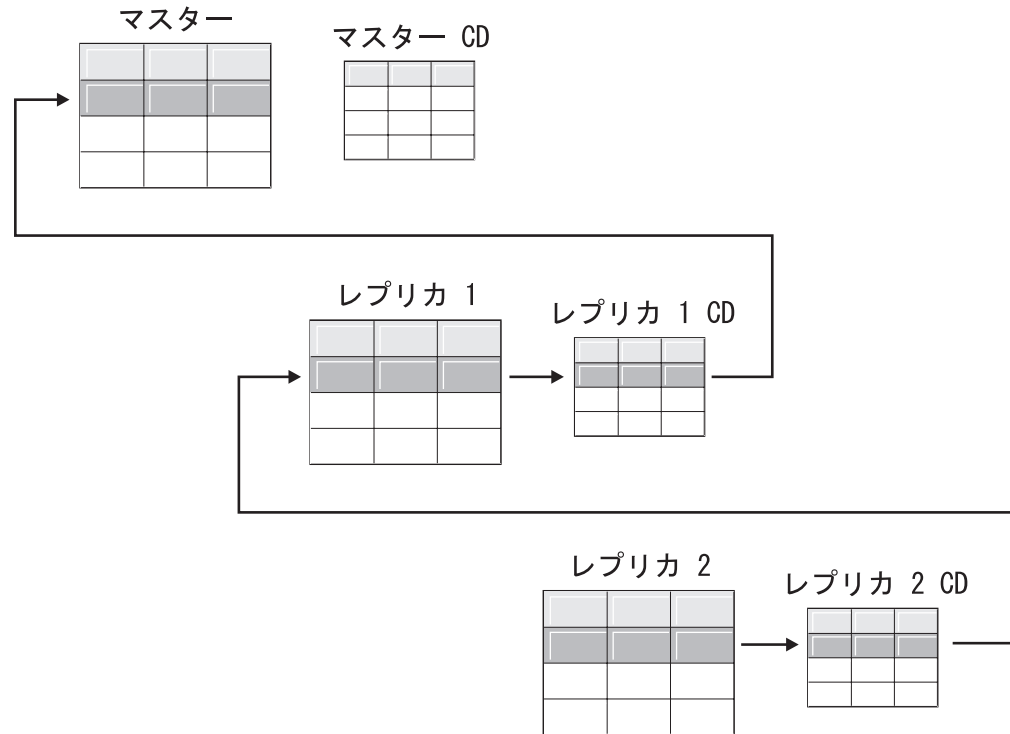


図4. 層 2 で再キャプチャー・オプションを使用すると、層 3 での変更が層 1 にまで複製されるようになります。multi-tier 構成で中間層として機能するレプリカ表がある場合は、そのレプリカで再キャプチャー・オプションを使用して、下層のレプリカで発生した変更が中間層のレプリカで再キャプチャーされてマスターに転送されるようになります。

競合検出の設定 (Update-anywhere レプリケーション)

デフォルト: 競合検出なし

制約事項:

- 非 DB2 リレーショナル・データベースからの表は Update-anywhere にかかわれない。したがって、非 DB2 リレーショナル・ソースでは競合検出は行われません。
- DATALINK 列を含む Update-anywhere 構成がある場合は、競合検出レベルに None を指定する必要がある。DB2 は DATALINK 列が参照する外部ファイルの更新競合はチェックしません。
- LOB 列を含む Update-anywhere 構成がある場合は、競合検出レベルに None を指定する必要がある。

Update-anywhere 構成では、マスターとそのレプリカの間で時々競合が発生する可能性があります。以下に、競合が起こる可能性のある場合を示します。

- マスター表である行が更新されたが、1 つ以上のレプリカ表で同じ行にそれとは異なる更新が加えられ、さらにアプライ・プログラムがそれらの競合する変更を同一サイクルの間に処理した。
- 制約に違反した。

競合検出レベルを個々のレプリケーション・ソースに設定した場合でも、アプライ・プログラムはサブスクリプション・セットのすべてのメンバーのレベルとして、サブスクリプション・セット・メンバーの競合検出レベルのうち最も高いものを使用します。

DB2 レプリケーションでは、3 つのレベルの競合検出 (検出なし、標準検出、および拡張検出) が提供されます。逸失またはリジェクトされたトランザクションの許容度とパフォーマンス要件に基づいて、どのタイプの検出を使用するかを決定できます。Update-anywhere レプリケーションに使用する予定のソースの登録時に、以下の競合検出レベルの中からいずれかを選択できます。

なし 競合検出を行いません。マスター表とレプリカ表の間で競合している更新は検出されません。このオプションは、Update-anywhere レプリケーションではお勧めしません。

標準 適度な競合検出。

各アプライ・サイクルの間に、アプライ・プログラムはマスターの CD 表のキー値とレプリカの CD 表のキー値を比較します。両方の CD 表に同じキー値が存在している場合、それは競合です。競合が発生すると、アプライ・プログラムは直前にレプリカでコミットされたトランザクションを取り消します。これは、レプリカの CD 表からの読み取りを実行し、マスターで発生した変更だけを保持するという方法で行われます。

拡張 マスターとそのレプリカとの間での最良のデータ保全性を実現する競合検出。

標準検出の場合と同様に、アプライ・プログラムは各アプライ・サイクルの間にマスターの CD 表のキー値とレプリカの CD 表のキー値を比較します。両方の CD 表に同じキー値が存在している場合、それは競合です。ただし拡張検出では、競合をチェックする前に、アプライ・プログラムはすべての未完了トランザクションがコミットされるまで待ちます。すべての未完了トランザクションを確実に捕らえられるように、アプライ・プログラムはこれから処理されるトランザクションに対してサブスクリプション・セット内のすべてのターゲット表をロックし、CD 表内のすべての変更がキャプチャーされた後で競合検出を開始します。競合が発生すると、アプライ・プログラムは直前にレプリカでコミットされたトランザクションを取り消します。これは、レプリカの CD からの読み取りを実行し、マスターで発生した変更だけを保持するという方法で行われます。

制約事項: 拡張競合検出を指定しても、アプライ・プログラムが不定期接続環境で実行されている (COPYONCE キーワードを使用して始動された) 場合は、アプライ・プログラムは標準競合検出を使用します。

アプライ・プログラムは、読み取り従属関係を検出することはできません。たとえば、後で (DELETE ステートメントによって、またはロールバック・トランザクションによって) 除去されることになる情報をアプリケーションが読み取る場合、アプライ・プログラムは従属関係を検出することはできません。

競合が発生し得るレプリケーション構成を (検出なしまたは標準検出のいずれかを選択して) セットアップする場合は、発生したあらゆる競合を識別して処理するための方法を組み込む必要があります。競合のあったトランザクション更新はレプリケーションのインフラストラクチャーによってすでに検出されてバックアウトされ

ているとしても、アプリケーション・デザイナーは、いったんコミットされたが現在はバックアウト済みのトランザクションをどのように処理するのかを決定する必要があります。ASNDONE 出口ルーチンは各サブスクリプション・サイクルの最後で実行されるため、アプリケーション・デザイナーはこのルーチンを上記のようなアプリケーション固有のロジックのためのランチ点として使用できます。バックアウト済みの競合する更新に関する情報は、保持制限整理の対象として適格となるまで CD 表および UOW 表に残されます。このアプリケーション固有のロジックを追加する方法についての詳細は、使用しているプラットフォームの種類に応じて 161 ページの『ASNDONE 出口ルーチンの変更 (Linux、UNIX、Windows、z/OS)』または 162 ページの『ASNDONE 出口ルーチンの変更 (OS/400)』を参照してください。

リモート・ジャーナリングを使用する表の登録 (OS/400)

デフォルト: リモート・ジャーナルはソースとして使用されません

リモート・ジャーナリングを使用する OS/400 表を登録する場合、DB2 レプリケーションではレプリケーション・ソースとしてローカル・ジャーナルの代わりにリモート・ジャーナルを使用することを定義できます。レプリケーション用のリモート・ジャーナリング・オプションを選択して、CD 表、キャプチャー・プログラム、およびキャプチャー・コントロール表を、ソース表が置かれている OS/400 サーバーとは別個の OS/400 データベース・サーバーに移動してください。

OS/400 で表をソースとして登録する場合、デフォルトではリモート・ジャーナリングを使用しないことが前提とされます。

推奨: ある OS/400 表から別の OS/400 表にデータを複製する場合、セットアップ済みのリモート・ジャーナルがあるときは常に、登録時にリモート・ジャーナリング関数を使用することを強く推奨します。レプリケーションにリモート・ジャーナリングを使用すると、パフォーマンスが大幅に向上します。リモート・ジャーナル関数を使用すると、登録、キャプチャー・プログラム、およびキャプチャー・コントロール表をソース表が置かれているシステムから遠く離れたところへ移動できるため、そのシステムにはより多くのリソースが使用可能な状態で残されます。これにより、プロセッサ使用量が削減され、ディスク・スペースが節約されます。また、ターゲット・サーバーにあるリモート・ジャーナルを使用する場合は CD 表がターゲット表と同じシステム上に存在するため、アプライ・プログラムは予備ファイルを使用せずに直接 CD 表からターゲット表に変更を適用できます。予備ファイルを使用しないことで、アプライ・プログラムによって使用されるリソースの量が削減されます。

推奨: リモート・ジャーナルを使用する表をソースとして登録するのは、登録がレプリケーション・ターゲットと同じ OS/400 システム上にある場合だけにしてください。DB2 レプリケーションを使用すると、登録がターゲットと同じ OS/400 システム上になくてもリモート・ジャーナルをソースとして登録できますが、その場合にはジャーナルをターゲット・システムに置くことで得られるパフォーマンス上の利点を得られません。

リモート・ジャーナリングを使用する OS/400 表を登録する前に、使用するリモート・ジャーナルがアクティブ状態になっていることを確認してください。

制約事項: リモート・ジャーナル構成ではレプリカ・ターゲット表タイプはサポートされません。

リモート・ジャーナル関数についての詳細は、「バックアップおよびリカバリーの
手引き (SD88-5008)」および「OS/400 Remote Journal Function for High
Availability and Data Replication (SG24-5189)」を参照してください。

主キーの代わりに相対レコード番号 (RRN) を使用する (OS/400)

通常、あるソース用のターゲット表はそのソース内の主キー列と同じキー列を使用します。アプライ・プログラムはこのキー値を使用して、自分がどのデータをソースの CD 表からターゲットに複製したのかをトラッキングします。主キー、ユニーク索引、またはユニーク索引として使用できる列の組み合わせを含んでいない OS/400 表を登録する場合は、その表を相対レコード番号 (RRN) を使用して登録する必要があります。RRN を使用して複製することを選択すると、CD 表とターゲット表の両方に、それぞれの行ごとにユニークな値を含む INTEGER 型の列 (IBMQSQ_RRN) が追加されます。この列に含まれているのは、ソース表の各行に対応する RRN です。

ソース表が再編成されないかぎり、RRN はソース表の行に対する主キーとして使用されます。ソース表が再編成されると、ソース表の各行の RRN が変更されます。したがって、CD 表およびターゲット表の行に含まれている RRN の値は、当該の行のソース表内での新しい位置を反映した正しい値ではなくなります。ソース表を (たとえば、削除された行を圧縮するために) 再編成すると、必ず DB2 DataPropagator for iSeries はそのソース表のセットになっているすべてのターゲット表に対してフル・リフレッシュを実行します。この理由から、RRN を主キーとして使用するターゲット表をやはり RRN を使用する他のターゲットと一緒にサブスクリプション・セットに入れて、RRN 以外で一意性を表すなんらかの因子を使用する表とはセットにしないでください。

レプリケーション・ソースとしてのビューの動作

レプリケーション用のビューを登録すると、それらは登録オプションを基本表の登録定義から継承します。最も重要なことは、ビューが変更キャプチャー・レプリケーション用として登録されるか、あるいはフル・リフレッシュのみとして登録されるかが、ビューの基本表によって決定されることです。以下のセクションでは、登録されたビューがさまざまなシナリオのレプリケーションにおいてどのように動作するのか説明されています。

- 『単一の表に対するビュー』
- 63 ページの『複数の表の結合に対するビュー』

単一の表に対するビュー

基本表がレプリケーション用に登録されている場合は、単一表に対するビューを登録できます。単一の登録済みの表に対するビューを登録すると、ビューはその基本表が持っているレプリケーション・タイプを継承します。基本表がフル・リフレッシュのみのレプリケーション用に登録されている場合は、ビューはフル・リフレッシュのみのレプリケーションを持ちます。そのビューを変更キャプチャー・レプリケーション用に登録することはできません。なぜなら、その基本表は変更をトラッ

キングするための関連した CD 表を持っていないからです。基本表が変更キャプチャー・レプリケーション用に登録されている場合は、ビューは変更キャプチャー・レプリケーションを持つため、フル・リフレッシュのみのレプリケーション用に登録することはできません。

変更キャプチャー・レプリケーション用に登録されている表に対するビューを登録すると、基本表の CD 表に対するビューが作成されます。この CD ビューには、登録したビューから参照される列だけが含まれています。

ビュー内の列のサブセットは登録できません。なぜなら、ビュー内のすべての列が自動的に登録されるからです。

複数の表の結合に対するビュー

複数の表の結合に対するビューを登録する場合は、その結合内の表が 1 つでも登録されているかぎり、基本表は登録済みの表であっても未登録の表であっても構いません。また、ソースとして登録されている CCD 表の内部結合も持てます。

結合をレプリケーション・ソースとして登録すると、DB2 レプリケーションは同一の SOURCE_OWNER 値と SOURCE_TABLE 値が含まれている複数の行を登録 (IBMSNAP_REGISTER) 表に追加します。これらの行は、各自の SOURCE_VIEW_QUAL 値によって区別されます。これらの各項目によって結合のコンポーネントが識別されます。

制約事項: CCD 表が組み込まれた結合を定義する場合は、その結合内の他のすべての表が CCD 表である必要があります。

ある結合ビューがレプリケーション・ソースとして存続できるようにするためには、その結合ビューを相関 ID を使用して作成する必要があります (単一表に対するビューの場合には相関 ID は不要です)。

例:

```
create view REGRES1.VW000 (c000,c1001,c2001,c2002,c1003) as
  select a.c000,a.c001,b.c001,b.c002,a.c003
  from REGRES1.SRC001 a, REGRES1.SRC005 b
  where a.c000=b.c000;
```

ここで、VW000 はビューの名前です。SRC001 および SRC005 は、ビューの一部である表です。C000、C001、C002、および C003 は、両方の表 (SRC001 と SRC005) の C000 列が等しいという条件の下でビューの一部となる列です。

ビューが継承するレプリケーションのタイプは、そのビューの基本表の組み合わせによって決まります。各基本表は次のいずれかです。

- 変更キャプチャー・レプリケーション用に登録されているもの
- フル・リフレッシュのみのレプリケーション用に登録されているもの
- 未登録のもの

64 ページの表 3 には、基本表のさまざまな組み合わせと、それぞれの組み合わせからソース・ビューと CD ビューが結果的にどのタイプになるのかが示されています。

表 3. ビューの場合の基本表の組み合わせ

表 1	表 2	結合ビューおよび CD ビューの記述
変更キャプチャー用に登録されているもの	変更キャプチャー用に登録されているもの	このビューは、変更キャプチャー・レプリケーション用に登録されたものである。この CD ビューには、表 1 の CD 表と表 2 の CD 表から参照される列が含まれている。
変更キャプチャー用に登録されているもの	フル・リフレッシュのみに登録されているもの	このビューは、変更キャプチャー・レプリケーション用に登録されたものである。この CD ビューには、表 1 の CD 表から参照される列と、表 2 から参照される列が含まれている。各レプリケーション・サイクルでは、表 1 に入っている列に対する変更だけが登録済みビューのターゲットに複製される。
フル・リフレッシュのみに登録されているもの	フル・リフレッシュのみに登録されているもの	このビューは、フル・リフレッシュのみのレプリケーション用に登録されたものである。CD ビューはない。
フル・リフレッシュのみに登録されているもの	未登録のもの	このビューは、フル・リフレッシュのみのレプリケーション用に登録されたものである。CD ビューはない。
変更キャプチャー用に登録されているもの	未登録のもの	このビューは、変更キャプチャー・レプリケーション用に登録されたものである。この CD ビューには、表 1 の CD 表から参照される列と、表 2 から参照される列が含まれている。各レプリケーション・サイクルでは、表 1 に入っている列に対する変更だけが登録済みビューのターゲットに複製される。
未登録のもの	未登録のもの	このビューは有効なレプリケーション・ソースではないため登録できない。

複数のソース表がレプリケーション・ソースとして組み込まれているビューを定義する場合は、二重削除を防ぐための配慮が必要です。二重削除は、同一レプリケーション・サイクルの間に両方とも 1 つのビューの一部である表から行を削除した場合に発生します。たとえば、CUSTOMERS 表と CONTRACTS 表を含むビューを作成したとします。二重削除は、同一のレプリケーション・サイクルで CUSTOMERS 表から 1 つの行を削除し、同様に (ビューの結合点から) それに対応する行を CONTRACTS 表からも削除した場合に発生します。ここで問題となるのは、その行が結合の 2 つのソース表から削除されているために、その行はビューに (基本ビューにも CD 表ビューにも) 表示されず、したがってこの二重削除がターゲットに複製できないことです。

二重削除を防ぐためには、結合内のソース表のいずれか 1 つに CCD 表を定義する必要があります。この CCD 表は、コンデンスされた非コンプリートの表でなければならず、さらにターゲット・サーバー上になければなりません。結合内のいずれかのソース表に対し、コンデンスされた非コンプリートの CCD 表を定義すれば、ほとんどの場合の二重削除問題は解決されます。なぜなら、この CCD 表の IBMSNAP_OPERATION 列を使用すれば削除を検出できるからです。サブスクリプション・サイクルの後 に実行するサブスクリプション・セットの定義に、単に SQL ステートメントを追加してください。この SQL ステートメントは、CCD 表内の IBMSNAP_OPERATION が “D” と等しくなるターゲット表からすべての行を除去します。

同一アプライ・サイクルで、CCD を持つソース表のある行が更新されたが、結合内のもう一方の表ではそれに対応する行が削除されたという場合には、更新と削除に関する問題がまだ発生する可能性があります。その結果、アプライ・プログラムは結合された表でその対応する行を検出できず、更新された値を複製できません。

表のビューをソースとして登録する

このセクションでは、DB2 表のビューをレプリケーション・ソースとして登録する方法を説明します。

前提条件:

- ソースとして登録するビューを処理させるキャプチャー・コントロール・サーバー上に、キャプチャー・コントロール表がすでに存在していなければならない。キャプチャー・コントロール表を作成する必要がある場合は、26 ページの『レプリケーション・コントロール表のセットアップ』を参照してください。
- ソース・ビューの名前は DB2 表の命名規則に従う必要がある。
- ビューの基礎となる基本表を少なくとも 1 つソースとして登録しなければならない。基本表を登録するときは、ビューの登録時に使用するのと同じキャプチャー・スキーマを使用します。表の登録方法は、43 ページの『DB2 表をソースとして登録する』を参照してください。

制約事項:

- 非 DB2 リレーショナル表のビューは登録できない。
- 別のビューの上層のビューは登録できない。
- OS/400 では、SQL ステートメントは長さが 32,000 文字までに制限されているため、1 つのビュー当たりおよそ 2000 列までしか登録できない。正確な列数は、列名の長さによって決まります。
- ビューを定義されているすべての CCD 表は、レプリケーション・ソースとして登録するためにはコンプリートでコンデンスされたものでなければならない。

手順:

以下の方法のいずれかを使用して、ビューを登録してください。

レプリケーション・センター

「ビューの登録」ウィンドウを使用します。詳しくは、レプリケーション・センターのヘルプを参照してください。

レプリケーション用のシステム・コマンド (OS/400)

ADDDPRREG システム・コマンドを使用します。このコマンドの構文およびパラメーターについての記述は、386 ページの『ADDDPRREG: DPR 登録の追加 (OS/400)』を参照してください。

ビュー用の登録オプションは、それらのビューが定義されているソース表の登録定義から導き出されます。登録オプションとそれらのデフォルトの完全なリスト、およびこれらのデフォルトを使用した方がよい場合と変更した方がよい場合についての説明は、47 ページの『ソース表用の登録オプション』を参照してください。基本表に基づいてビューがどのレプリケーション・タイプ (変更キャプチャーまたはフル・リフレッシュのみ) を継承するのかについては、62 ページの『レプリケーション・ソースとしてのビューの動作』を参照してください。

CCD 表をソースとして保守する (IMS)

アプライ・プログラムによって移植されたのではなく外部的に移植された CCD 表、または IMS DataPropagator や DataRefresher などのプログラムによって保守されている CCD 表がある場合は、アプライ・プログラムがそれらの CCD 表をソースとして読み取れるように、また正しく機能できるように、それらの表を保守する必要があります。このセクションでは、CCD 表をレプリケーション・ソースとして保守する方法を説明します。

外部ツールによって移植された CCD 表を保守するためには、登録 (IBMSNAP_REGISTER) 表の 3 つの列 (CCD_OLD_SYNCHPOINT、SYNCHPOINT、および SYNCHTIME) を更新する必要があります。(登録表のこれらの列についての詳細は、532 ページの『*schema.IBMSNAP_REGISTER*』を参照してください。) これらの 3 つの列は、以下のタイプのイベントが発生するたびに更新する必要があります。

- はじめてフル・リフレッシュまたは CCD 表のロードが行われたとき
 - CCD_OLD_SYNCHPOINT を CCD 表の IBMSNAP_COMMITSEQ の最小値を表す値に設定する。
 - SYNCHPOINT を CCD 表の IBMSNAP_COMMITSEQ の最大値を表す値に設定する。SYNCHPOINT を 0 に設定しないでください。順序付け用に独自の値を作成する場合は、初回用の SYNCHPOINT 値は 1 にしてください。
 - SYNCHTIME を CCD 表の IBMSNAP_LOGMARKER の最大タイム・スタンプ値を表す値に設定する。
- フル・リフレッシュ後またはロード後に CCD 表が更新されたとき
 - CCD_OLD_SYNCHPOINT 値を変更してはならない。
 - SYNCHPOINT を CCD 表の IBMSNAP_COMMITSEQ の新しい最大値を表す値に設定する。
 - SYNCHTIME を CCD 表の IBMSNAP_LOGMARKER の新しい最大タイム・スタンプ値を表す値に設定する。
- 2 回目以降のフル・リフレッシュまたは CCD 表のロードが行われたとき
 - CCD_OLD_SYNCHPOINT を CCD 表の IBMSNAP_COMMITSEQ の最小値を表す値に設定する。
 - SYNCHPOINT を CCD 表の IBMSNAP_COMMITSEQ の最大値を表す値に設定する。
 - SYNCHTIME を CCD 表の IBMSNAP_LOGMARKER の最大タイム・スタンプ値を表す値に設定する。

重要: 上記では、CCD 表で IBMSNAP_COMMITSEQ および IBMSNAP_LOGMARKER に使用されている値は常に増えてゆく値であることを前提としています。アプライ・プログラムは、CCD_OLD_SYNCHPOINT 値が最新に適用された SYNCHPOINT 値より大きくなならないかぎり、ソース CCD 表でフル・リフレッシュが実行されたことを検出しません。

関連概念:

- 265 ページの『第 15 章 SQL レプリケーションの場合のレプリケーション・センターの使用法』

関連タスク:

- 69 ページの『第 4 章 SQL レプリケーションのソースのサブスクライブ』

関連資料:

- 386 ページの『ADDDPRREG: DPR 登録の追加 (OS/400)』

第 4 章 SQL レプリケーションのソースのサブスクライブ

レプリケーション・ソースとして使用する表とビューを登録後、ターゲット表またはビューがソースからのソース・データと変更内容を受け取れるように、ターゲット表またはビューのサブスクリプションを定義できます。この章で説明する管理タスクは、キャプチャー・プログラムとアプライ・プログラムが、ソース・データをコピーするため、あるいは変更データを取り込んでそれをターゲット表に適切なインターバルで複製するために使用するコントロール情報のセットアップを支援します。

この章は、以下のセクションで構成されています。

- 『ソースおよびターゲットの分類方法の計画』
- 72 ページの『サブスクリプション・セットの作成』
- 74 ページの『サブスクリプション・セットのオプション処理』
- 82 ページの『サブスクリプション・セット内におけるソース表およびビューのターゲット表およびビューへのマッピング』
- 85 ページの『ターゲット・タイプの選択』
- 97 ページの『すべてのターゲット表タイプに共通のプロパティ』

ソースおよびターゲットの分類方法の計画

どのターゲットがどのソースをサブスクライブするかを定義する前に、ソースとターゲットの分類方法を計画する必要があります。DB2 レプリケーションは、グループ単位でソースからターゲットへのマッピングを処理します。これらのグループは、同一のキャプチャー・プログラムによって処理される 1 つ以上のソースと、ソース・データのすべてまたは一部をサブスクライブし、同一のアプライ・プログラムによって処理される 1 つ以上のターゲットで構成されています。これらのグループを、サブスクリプション・セットと呼び、ソースからターゲットへのマッピングをサブスクリプション・セット・メンバーと呼びます。

サブスクリプション・セットを計画するときは、以下の規則と制約に注意してください。

- サブスクリプション・セットは、ソース・サーバーをターゲット・サーバーにマップする。サブスクリプション・セット・メンバーは、ソース表またはビューを、ターゲット表またはビューにマップします。サブスクリプション・セットとそのメンバーは、アプライ・コントロール・サーバーに保管されます。
- アプライ・プログラムは、サブスクリプション・セットのすべてのメンバーを単一グループとして処理する。このため、サブスクリプション・セットのいずれかのメンバーで、何らかの理由でフル・リフレッシュ・コピーが必要な場合、セット全体のすべてのメンバーがリフレッシュされます。
- 1 つのセットのメンバーにおいて、すべてのソース表およびビューのキャプチャー・スキーマは同じでなければならない。
- OS/400 システムでは、1 つのサブスクリプション・セットのメンバーにおいて、すべてのソース表は同じジャーナルに記録されなければならない。

- IMS DataPropagator で作成され、サブスクリプション・セットのメンバーである外部 CCD 表すべてのキャプチャー・スキーマは同じでなければならない。

ユニークなアプライ修飾子を持つ単一のアプライ・プログラムは、サブスクリプション・セットを 1 つでもあるいは多数でも処理できます。単一の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・メンバーを 1 つでもあるいは多数でも入れることができます。以下のセクションでは、アプライ・プログラムごとに少数または多数のセットを保有する場合と、サブスクリプション・セットごとに少数または多数のサブスクリプション・セット・メンバーを保有する場合のトレードオフについて説明します。

サブスクリプション・セット・メンバー数の計画

サブスクリプション・セットにメンバーを追加するとき、ソースとターゲットのすべてのペア (サブスクリプション・セット・メンバー) を 1 つのサブスクリプション・セットにまとめるか、それぞれのペアごとに別々のサブスクリプション・セットを作成するか、あるいはそれぞれにかなりの数のペアが含まれた、少数のサブスクリプション・セットを作成するかを決定しなければなりません。

アプライ・プログラムは、サブスクリプション・セットのメンバーを 1 つの (論理) トランザクションで複製するため、以下のどちらの状態においても、複数のメンバーを 1 つのサブスクリプション・セットにまとめる必要があります。

- ソース表に相互に論理的な関連がある場合。
- ターゲット表に参照保全制約がある場合。

複数のメンバーを 1 つのサブスクリプション・セットにまとめると、すべてのメンバーのレプリケーションを必ず同時に開始することができます。さらに、サブスクリプション・セットを処理するのに必要なデータベース接続の数と、レプリケーション環境を保守するための管理オーバーヘッドが削減されます。サブスクリプション・セットに SQL ステートメントまたはストアド・プロシージャが含まれている場合、これらのステートメントまたはプロシージャを使用して、そのサブスクリプション・セットのすべてのメンバーを処理できます。

サブスクリプション・セットの表の間に、論理的または参照保全のリレーションシップがない場合、その表を 1 つまたはいくつかのサブスクリプション・セットにまとめることができます。サブスクリプション・セットの数を制限する主な理由は、レプリケーション環境の管理を簡単にすることです。ただし、サブスクリプション・セットの数を増やすと、レプリケーションの失敗による影響は最小になります。

アプライ・プログラムが失敗する原因となるエラーをより簡単に突き止められるようにするには、少数のメンバーしかサブスクリプション・セットに追加しないようにします。メンバーが少数なら、多数のメンバーがセットに含まれている場合より迅速に問題のソースを見つけることができます。サブスクリプション・セットの 1 つのメンバーで障害が起こった場合、そのセットの他のメンバーに適用されたデータはすべてロールバックされます。そのため、すべてのメンバーがサイクルを完了しない限り、どのメンバーも正常にサイクルを完了できません。アプライ・プログラムは、失敗したサブスクリプション・セットをその最後の正常なコミット・ポイントまでロールバックします。このコミット・ポイントは、アプライ・プログラム

を始動したときに `commit_count` キーワードを指定していれば、現行のアプリ・サイクル内にある可能性があります。

アプリ修飾子ごとのサブスクリプション・セット数の計画

サブスクリプション・セットを定義するとき、そのサブスクリプション・セット用のアプリ修飾子を指定します。アプリ修飾子は、アプリ・プログラムのインスタンスを 1 つ以上のサブスクリプション・セットに関連付けます。それぞれのサブスクリプション・セットを処理するのは、ただ 1 つのアプリ・プログラムですが、アプリ・プログラムはそれぞれ 1 つ以上のサブスクリプション・セットを、1 つ 1 つのアプリ・サイクル内で処理できます。

アプリ・プログラムのインスタンス (それぞれに独自のアプリ修飾子があります) は必要な数だけ実行でき、アプリ・プログラムはそれぞれサブスクリプション・セットを必要な数だけ処理できます。基本オプションが 2 つあります。

- アプリ修飾子をそれぞれ 1 つのサブスクリプション・セットに関連付ける (個々のアプリ・プログラムが、厳密に 1 つのサブスクリプション・セットを処理する)。

速度が重要な場合は、いくつかのアプリ修飾子間にセットを分散でき、こうすることで、アプリ・プログラムのいくつかのインスタンスを同時に実行できます。1 つのアプリ・プログラム・インスタンス・プロセスで 1 つのサブスクリプション・セットを処理するように決めていると、このアプリ・プログラムの `OPT4ONE` 始動オプションを使用できます。このオプションは、サブスクリプション・セットのコントロール表情報をメモリーにロードするものです。このオプションを使用すると、アプリ・プログラムは、サブスクリプション・セット情報のコントロール表をアプリ・サイクルごとには読み取りません。したがって、アプリ・プログラムのパフォーマンスが向上します。ただし、実行するアプリ・プログラム・インスタンスが多くなれば、それらが使用するシステム・リソースも多くなり、全体としてのパフォーマンスは低下する可能性があります。

- アプリ修飾子をそれぞれ複数のサブスクリプション・セットに関連付ける (個々のアプリ・プログラムが、多数のサブスクリプション・セットを処理する)。

複数のアプリ修飾子を使用すれば、シングル・ユーザー ID からアプリ・プログラムの複数のインスタンスを実行できます。

アプリ・プログラムでは、指定されたアプリ修飾子に対応するセットすべてを、可能な限り現状のまま保持しようとしています。アプリ・サイクルが開始すると、アプリ・プログラムは含まれる現行データが最も少ないサブスクリプション・セットを判別し、まずそのセットの処理を開始します。

速度が主要なゴールでない場合は、1 つのアプリ修飾子が指定された膨大な数のサブスクリプション・セットを複製することがあります。たとえば、営業時間後まで待ってから複製する場合に、これは大変適切なオプションということもあります。

1 つのアプライ・プログラムで複数のサブスクリプション・セットを処理する欠点は、そのアプライ・プログラムがサブスクリプション・セットを順次処理することです。したがって、全体のレプリケーション待ち時間は増加する可能性があります。

一部のサブスクリプション・セットに対して特定の要件がある場合、この 2 つのオプションを組み合わせることができます。たとえば、1 つのアプライ・プログラムにほとんどのサブスクリプション・セットを処理させることができるので、関連するサブスクリプション・セットをまとめて処理すれば、アプライ・プログラムを有効に利用できるでしょう。そして、別のアプライ・プログラムに単一のサブスクリプション・セットを処理させることができるので、そのサブスクリプション・セットのレプリケーション待ち時間を確実に最小にできます。さらに、アプライ・プログラムの 2 つのインスタンスを使用すれば、サブスクリプション・セットの並列処理全体が強化されます。

サブスクリプション・セットの作成

登録済みのソースからデータを複製する前に、アプライ・プログラムが 1 つの集合として処理するサブスクリプション・セット・メンバー (ソースからターゲットへのマッピング) のコレクションである、サブスクリプション・セットを作成しなければなりません。このセクションでは、それぞれのサブスクリプション・セットごとに定義するプロパティについて説明します。セットに追加するメンバーごとに適用されるこれらのプロパティは、データのレプリケーション先とレプリケーション元のサーバーを定義し、使用するキャプチャー・プログラム (登録済みソースのキャプチャー・スキーマを基にしています) とアプライ・プログラム、およびアプライ・プログラムがセットを処理する時期と方法も定義します。

サブスクリプション・セット・メンバーをサブスクリプション・セットに追加する必要はありません。代わりに、ソースからターゲットへのマッピングを含まない空のセットを作成できます。以下のような理由から、空のセットの作成が必要になることがあります。

- 後でメンバーをセットに追加する計画があり、メンバーを追加するまではサブスクリプション・セットをアクティブにする計画はない。
- 空のサブスクリプション・セットが処理に適格であるとき、いつでも SQL ステートメントまたはストアド・プロシージャを呼び出すために、アプライ・プログラムでそのセットを処理する。

前提条件:

1. サブスクリプション・セット用に、アプライ・コントロール表をアプライ・コントロール・サーバーで作成しなければならない。
2. サブスクリプション・セットにサブスクリプション・セット・メンバーを追加する前に、ソースとして使用する表またはビューを登録しなければならない。レプリケーションのためにソースを登録する必要がある場合は、43 ページの『第 3 章 表およびビューを SQL レプリケーション・ソースとして登録する』の説明を読んでください。セットの分類方法について検討する必要もあります。セットを計画する必要がある場合、詳細については、69 ページの『ソースおよびターゲットの分類方法の計画』を参照してください。

手順:

サブスクリプション・セットを作成するには、以下の 2 つの方法のどちらかを使用できます。

レプリケーション・センター

「サブスクリプション・セットの作成 (Create subscription sets)」ノートブックを使用します。詳しくは、レプリケーション・センターのヘルプを参照してください。

レプリケーション用のシステム・コマンド (OS/400)

ADDDPRSUB システム・コマンドを使用します。このコマンドの構文およびパラメーターについての記述は、395 ページの『ADDDPRSUB: DPR サブスクリプション・セットの追加 (OS/400)』を参照してください。

サブスクリプション・セットを作成するには、以下の基本的な特性を指定します。

アプライ・コントロール・サーバーの別名

サブスクリプション・セットを処理するアプライ・プログラム用のコントロール表を含むサーバーのローカルの別名。どのデータベースにあるアプライ・コントロール・サーバーにも、同一の別名を定義します。このサーバーからレプリケーション・センターまたはアプライ・プログラムを実行して、レプリケーション・センターがアプライ・コントロール表を適切に移植するようにし、さらにどのアプライ・プログラムも標準の別名を使用して適切なサーバーに接続するようにします。

サブスクリプション・セット名

サブスクリプション・セットの名前。サブスクリプション・セットを処理するアプライ・コントロール・サーバーにおいて、セット名は指定されたアプライ修飾子に対してユニークでなければなりません。名前の長さは 18 文字まで可能です。

アプライ修飾子

新規または既存のアプライ修飾子の名前。サブスクリプション・セットを処理するアプライ・プログラムを示します。同じアプライ修飾子を使用して、複数のサブスクリプション・セットを処理できます。同じアプライ修飾子を持つサブスクリプション・セットは、同じアプライ・コントロール・サーバーで定義しなければなりません。新規のアプライ修飾子を作成する場合は、アプライ修飾子の命名方法の規則について、327 ページの『第 17 章 SQL レプリケーション・オブジェクトの命名規則』を参照してください。

キャプチャー・コントロール・サーバーの別名

サブスクリプション・セットに登録済みのソースを処理するキャプチャー・プログラム用のコントロール表を含むサーバーの別名。どのデータベースにあるキャプチャー・コントロール・サーバーにも、同一の別名を定義します。このサーバーからレプリケーション・センターまたはアプライ・プログラムを実行して、レプリケーション・センターがキャプチャー・コントロール表とアプライ・コントロール表を適切に移植するようにし、さらにどのアプライ・プログラムも標準の別名を使用して適切なサーバーに接続するようにします。

キャプチャー・スキーマ

キャプチャー・スキーマの名前。サブスクリプション・セットに登録済みのソースを定義するキャプチャー・コントロール表のセットを示します。サブスクリプション・セットのすべてのソース表は、同じサーバーに常駐していなければならない、1つのキャプチャー・プログラムだけがソース表への変更をキャプチャーすることができます。

ターゲット・サーバーの別名

アプライ・プログラムがソースからの変更を複製する先の表またはビューが含まれるターゲット・サーバーの名前。どのデータベースにあるターゲット・サーバーにも、同一の別名を定義します。このサーバーからレプリケーション・センターまたはアプライ・プログラムを実行して、レプリケーション・センターがアプライ・コントロール表を適切に移植するようにし、さらにどのアプライ・プログラムも標準の別名を使用して適切なサーバーに接続するようにします。

サブスクリプション・セットを作成するときは、アプライ・プログラムがセットを処理する方法について、デフォルトの設定値を使用できます。あるいは、サブスクリプションのプロパティを、レプリケーションの要求を満たすように変更できます。サブスクリプション・セット用の処理オプションとそのデフォルト、およびデフォルトの使用または変更が必要な場合についての説明の完全なリストは、『サブスクリプション・セットのオプション処理』を参照してください。

サブスクリプション・セットのオプション処理

このセクションでは、アプライ・プログラムがサブスクリプション・セットを処理する方法を指定するために定義可能なプロパティについて説明します。さらに、レプリケーションの要求に基づいて選択すべき設定値を決定する際にも、このセクションは役立ちます。

- 『セットがアクティブかどうかの指定』
- 75 ページの『アプライ・プログラムが取り出すデータに相当する分数の指定』
- 78 ページの『アプライ・プログラムがセットのメンバーへの変更を複製する方法の指定』
- 79 ページの『サブスクリプション・セット用の SQL ステートメントまたはストアド・プロシージャの定義』
- 80 ページの『サブスクリプション・セットのレプリケーションのスケジューリング』

セットがアクティブかどうかの指定

デフォルト: 非アクティブ

アプライ・プログラムがサブスクリプション・セットの処理を開始するかどうかを指定できます。サブスクリプション・セットを活動化すると、アプライ・プログラムは、そのセットのフル・リフレッシュを開始します。活動化レベルは以下の 3 つから選択します。

アクティブ

アプライ・プログラムは、その次のサイクル内でセットを処理します。アプライ・プログラムに次の実行時でセットを処理させる場合は、セットを活

動化します。後になっても、メンバーをセットに追加できます。セットをアクティブ化すると、ユーザーがセットを非アクティブ化するまで、セットはアクティブのまま、アプライ・プログラムはその処理を続けます。

非アクティブ

アプライ・プログラムはセットを処理しません。アプライ・プログラムでセットを処理する準備ができていない場合は、セットを非アクティブにしておきます。

1 回のみアクティブ

アプライ・プログラムは次のサイクル内でセットを処理し、その後セットを非活動化します。セットを 1 回だけ実行したい場合は、このオプションを指定します。このオプションを選択する前に、必ずすべてのサブスクリプション・セット・メンバーを追加してください。その理由は、サブスクリプション・セットを再活動化しない限り、アプライ・プログラムは後で追加したメンバーを処理しないためです。

アプライ・プログラムが取り出すデータに相当する分数の指定

デフォルト: 20 分

アプライ・プログラムが各アプライ・サイクル内でレプリケーション・ソースから取り出すデータを、それに相当するおおよその分数で指定できます。この指定が役に立つ状態がいくつかあります。

- 1 回のサブスクリプション・セット・サイクル内で処理されるデータ量が大量の場合。

1 回のアプライ・サイクルで大量の変更ブロックを複製するサブスクリプション・セットは、予備ファイルまたはログ (ターゲット・データベースの場合) でのオーバーフローの原因となります。たとえば、アプライ・プログラムをバッチ処理するシナリオでは、レプリケーションを必要とするエンキューされたトランザクションのバックログが大量に生成される可能性があります。

- ネットワークの停止が長引くと、大量のデータ・ブロックが CD 表に累積され、アプライ・プログラムの予備ファイルとターゲットのログがオーバーフローする可能性がある。

指定する分数を、データ・ブロックと呼びます。指定するデータ・ブロックの値は、サブスクリプション・セット (IBMSNAP_SUBS_SET) の表で MAX_SYNCH_MINUTES 列に保管されます。データの累積がデータ・ブロックのサイズより大きいと、アプライ・プログラムは 1 つのアプライ・サイクルをいくつかのミニサイクルに変換します。リソースが、指定されたブロックング因数をハンドルするにはまだ不十分な場合、アプライ・プログラムはデータ・ブロックのサイズを削減し、使用可能なシステム・リソースに一致させます。より小さいデータ集合を取り出すことにより、アプライ・プログラムはネットワーク負荷と取り出されたデータ用に一時的に必要なスペースの両方を減少させます。

各アプライ・サイクル中に、サブスクリプション・セットの MAX_SYNCH_MINUTES 値が NULL であるか、または 1 未満の数値に設定された場合、アプライ・プログラムはその設定に適格なすべてのデータを 1 つのアプライ・サイクルで処理します。CD 表と UOW 表に大容量のデータが含まれている場合は、その状況のためにデータベース・トランザクション・ログがいっぱいになっ

たり、または予備ファイルがオーバーフローするという問題が起きることがあります。以下のガイドラインを使用して、MAX_SYNC_MINUTES を NULL 以外の値に変更することができます。

- ASN.IBMSNAP_SUBS_SET 表の SLEEP_MINUTES 列が特定のサブスクリプション・セットに対して 5 分 (またはそれ未満) に設定されている場合は、MAX_SYNC_MINUTES を 5 分に設定します。
- 特定のサブスクリプション・セットに対して、SLEEP_MINUTES が 30 分 (またはそれ以上) に設定されている場合は、MAX_SYNC_MINUTES を 60 分に設定します。
- SLEEP_MINUTES が 5 ~ 30 分の間設定されている場合は、MAX_SYNC_MINUTES を SLEEP_MINUTES と等しい値に設定します。

レプリケーション環境をモニターして、必要に応じて MAX_SYNC_MINUTES を調整します。MAX_SYNC_MINUTES の数値がゼロより大きいことを確認してください。

例: アプライ・プログラムがミニサイクル当たりせいぜい 10 分相当のデータしか取り出さないように指定すると、最後のミニサイクルの約 10 分間以内で、アプライ・プログラムはかなりの量のコミット済みデータをソースの CD 表から取り出すこととなります。

ログ・ファイルと予備ファイルをオーバーフローさせない点に加えて、ミニサイクルには、他にもいくつかの利点があります。レプリケーション・サイクルでエラーが発生している場合、アプライ・プログラムは、失敗したミニサイクルで行った変更のみをロールバックしなければなりません。ミニサイクルでレプリケーションが失敗した場合、アプライ・プログラムは、最後に正常に実行されたミニサイクルからサブスクリプション・セットを処理しようとはしますが、これで大量の変更データを処理できる場合は、かなりの時間を節約できます。77 ページの図 5 に、変更データが変更のサブセットに分割される仕組みを示します。

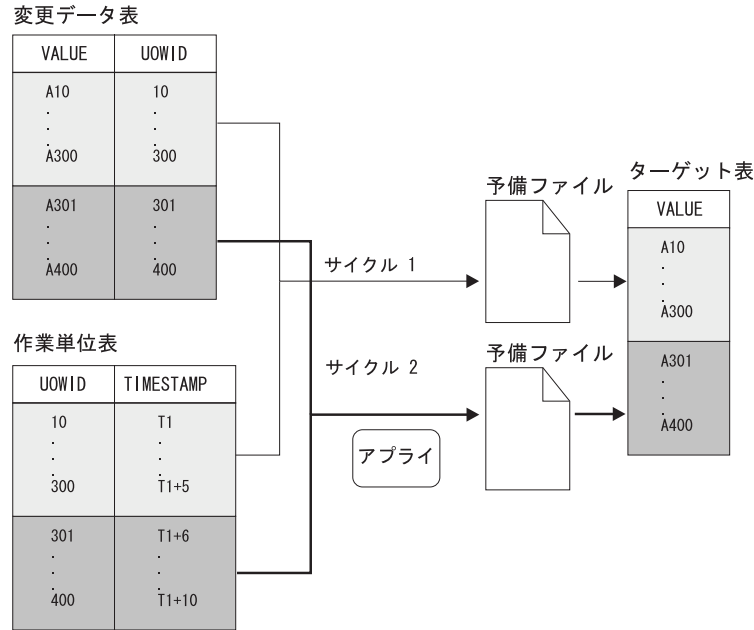


図5. データ・ブロックング： ネットワークのトラフィック量を少なくするには、データ・ブロック値を指定します。

設定する分数が小さければ、インターバル内に生じるサブスクリプション・セットのトランザクションをすべてコピーし、ミニサイクル内での予備ファイルやログのオーバーフローを防ぐことができます。

データを処理するとき、アプライ・プログラムは以下のどのアクションも行いません。

- 作業単位の分割 (実行時間の長い、コミットなしのバッチ・ジョブが、データ・ブロック係数で分割されることはありません)
- 前にコミット済みの小サブスクリプション・サイクルのロールバック
- フル・リフレッシュ中の、データ・ブロック係数の使用

アプライ・プログラムが参照保全性のあるターゲット表をロードする方法の決定

セットのターゲット表の間で参照保全が要求される場合、ターゲット表の初期ロード時にアプライ・プログラムがソース・データを処理する方法を選択しなければなりません。デフォルトでは、アプライ・プログラムはソースのすべての行を読み取り、それをメモリーに保管し、その後それらの行をターゲット表に挿入することによって、ターゲットのフル・リフレッシュを実行します。しかし、ターゲット表を初期ロードする方法には他のオプションがあります。サブスクリプション・セットを作成するとき、あるいはソースからターゲットへのマッピング (メンバー) をセットごとに定義するとき、これらを決定することはありません。すなわち、ターゲットのロード方法は、アプライ・プログラムの始動パラメーターを設定するとき決定します。サブスクリプション・セット・メンバーを定義するとき、それぞれのメンバーごとにレプリケーション要件を考慮する必要があり、それぞれのメンバーを処理するアプライ・プログラムで使用する始動オプションを決定できます。

これらの参照保全リレーションシップを作成する時期について、以下の 2 つの選択肢のどちらかを検討してください。

- ターゲット表にデータが移植される前。

この場合は、ターゲット表の抽出およびロードの全段階を通じて、ソース表への変更が行われないことが必要です。また、この初期の移植中にロード処理の速度を確保し、参照制約チェックをバイパスするため、LOADX 始動オプションを使用してアプライ・プログラムを始動しなければなりません。アプライ始動オプション LOADX を使用しない場合、ターゲット表への挿入が失敗する可能性があります。

- アプライ・プログラムがターゲット表に完全にデータを移植し、それらの表の該当するセットに変更を適用する 1 つのサイクルを正常に完了した後。

参照保全制約をこれらの表に追加する時期を待つ場合、ターゲット表のロード中に、ソース表でさらに変更を行うことができるという利点があります。バイパスする必要があるという制約がないため、始動オプション LOADX を使用しなくても、アプライ・プログラムを始動できます。始動オプション LOADX を使用するよりも、一般的にフル・リフレッシュはかなり速度があります。ターゲット表の初期の移植中、ターゲットは参照保全リレーションシップの点で相互に同期がとれていないことがあります。しかし、ロード中に、セットの変更はすべてキャプチャーされます。アプライ・プログラムが変更の入った最初のセットを複製した後、すべてのターゲット表には同じトランザクションが入り、参照保全性を持つようになります。この時点でセットを非活動化して参照保全制約を追加し、その後セットを再び活動化できます。

ターゲット表の初期ロード方法のために提供されている始動オプションについての詳細は、164 ページの『ASNLOAD 出口ルーチンを使ったターゲット表のリフレッシュ』を参照してください。

アプライ・プログラムがセットのメンバーへの変更を複製する方法の指定

サブスクリプション・セットに変更キャプチャー・レプリケーションが含まれるとき、アプライ・プログラムがセットのソースからターゲットへのすべてのマッピングへの変更を複製する方法を決定できます。ターゲット表の初期ロード後、アプライ・プログラムは CD (または CCD) 表の読み取りを開始し、変更を予備ファイルに収集します。それぞれの CD (または CCD) 表ごとに、アプライ・プログラムは別々の予備ファイルを作成します。そこでアプライ・プログラムは予備ファイルから変更を読み取り、変更をターゲット表に適用します。以下の 3 つの方法のいずれかで、これを実行します。

- 表モード処理の使用。
- トランザクション・モード処理の使用。
- 表モードとトランザクション・モードが混合した処理の使用。サブスクリプション・セットのターゲット表タイプに依存します。

サブスクリプション・セットの処理タイプを指定すると、ターゲット表またはビューへの変更をアプライ・プログラムがコミットする頻度をコントロールできます。アプライ・プログラムは、それぞれのサブスクリプション・セット・メンバーごと

に 1 回、あるいは相当数のトランザクションをアプライした後にコミットできません。コミットが 1 回の場合、サブスクリプション・セットの待ち時間を削減できますが、コミットが複数の場合は、アプライ・プログラムがオリジナルのコミット・シーケンスのデータを適用できます。

表モード

アプライ・プログラムは CD (または CCD) 表の予備ファイルからすべての変更を読み取り、対応するターゲット表に変更を適用してから、次の CD (または CCD) 表の処理を開始します。セット内のすべての CD (または CCD) 表からの変更の読み取りと適用が完了したら、アプライ・プログラムは DB2 コミットを発行して、サブスクリプション・セット内のすべてのターゲット表への変更をすべてコミットします。

トランザクション・モード

アプライ・プログラムは、すべての予備ファイルを一度にオープンし、それらの予備ファイルからの変更を同時に処理します。変更は、ソース表でトランザクションが発生した順番にターゲット表に適用されます。IBMSNAP_SUBS_SET 表の COMMIT_COUNT 列は、そのサブスクリプション・セットのすべてのターゲット表に変更が適用されコミットされる方法を制御します。サブスクリプション・セットのターゲット表に参照保全制約がある場合は、このトランザクション・モード処理を使用してください。

どのサブスクリプション・セットの場合でも、アプライ・プログラムがトランザクション・モード処理を使用するように指定できます。しかし、その指定で変更されるのは、ユーザー・コピーおよびポイント・イン・タイムのターゲット表がセットに含まれる場合のアプライ・プログラムの動作のみで、以下のタイプのターゲット表がセットに含まれる場合の動作は変更されません。

- CCD ターゲット表。CCD 表がソースとして含まれるセットは、常に表モードで処理されます。
- ソース表が CCD 表である場合、それに対応するターゲット表。CCD 表を含むセットは常に表モードで処理されます。
- レプリカ・ターゲット表。レプリカ表を含むセットは常にトランザクション・モードで処理されます。

サブスクリプション・セット用の SQL ステートメントまたはストアード・プロシージャの定義

アプライ・プログラムがサブスクリプション・セットを処理するたびに実行される SQL ステートメントまたはストアード・プロシージャを定義できます。これらのステートメントは、CCD 表の整理、またはターゲットに適用される前のソース・データの取り扱いに役立ちます。SQL ステートメントまたはストアード・プロシージャの実行時期と実行場所を次のように指定できます。

- アプライ・プログラムがデータを適用する前にキャプチャー・コントロール・サーバーで。
- アプライ・プログラムがデータを適用する前にターゲット・サーバーで。
- アプライ・プログラムがデータを適用した後にターゲット・サーバーで。

アプライ・プログラムは、ステートメントまたはプロシーチャーを上記にリストした順序で処理します。

レプリケーション・センターを使用して SQL ステートメントをサブスクリプション・セットに追加するときは、「SQL ステートメントまたはプロシーチャー呼び出しの追加」ウィンドウで「ステートメントの準備」をクリックし、ステートメントの構文を検証します。

サブスクリプション・セットのレプリケーションのスケジューリング

アプライ・プログラムがサブスクリプション・セットを処理する頻度をコントロールできるので、その結果、ターゲット表のデータをどのくらいで最新にするかをコントロールできるようになります。時間に基づくスケジューリングまたはイベントに基づくスケジューリングを使用して、どのくらいの頻度でサブスクリプション・セットを処理対象として適格とするかをコントロールでき、あるいはこれらのスケジューリング・オプションを一緒に使用することもできます。アプライ・プログラムは、サブスクリプション・セットが処理対象として適格になると、その処理を開始します。たとえば、インターバルを 1 日に設定し、さらにサブスクリプション・サイクルを起動するイベントを指定することもできます。これらのスケジューリング・オプションの両方を使用する場合、サブスクリプション・セットはスケジュールされた時刻とイベント発生時刻の両方で処理対象として適格になります。

Update-anywhere レプリケーションでは、master-to-replica サブスクリプション・セットと replica-to-master サブスクリプション・セットに使用するタイミングは同じであっても異なっても構いません。

アプライ・プログラムが処理するサブスクリプション・セットのインターバル内またはイベントとイベントの間で、複製されるデータ量が膨大な場合、特定のサブスクリプション・セットを処理対象として適格にすることは可能です。しかし、アプライ・プログラムがそれを処理できるのは、先行するインターバルまたはイベントで、すべてのサブスクリプション・セットへのデータの適用が終了してからです。サブスクリプション・セットの処理が終了すると、アプライ・プログラムは直ちに次の処理に適格なサブスクリプション・セットの処理を開始します。このケースでは、レプリケーション待ち時間が予想とは異なるかもしれませんが、失われるデータはありません。

手順:

サブスクリプション・セットを指定するには、以下の 2 つの方法のどちらかを使用できます。

レプリケーション・センター

以下のノートブックのいずれかを使用します。

- サブスクリプション・セットの作成。「スケジュール」ページを使用して、スケジューリング・オプションを選択します。
- サブスクリプション・セット・プロパティ。すでにサブスクリプション・セットを作成済みで、サブスクリプション・セットのスケジューリングを変更したい場合は、このノートブックを使用します。

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。

レプリケーション用のシステム・コマンド (OS/400)

ADDDPRSUB システム・コマンドを使用します。このコマンドの構文およびパラメーターについての記述は、395 ページの『ADDDPRSUB: DPR サブスクリプション・セットの追加 (OS/400)』を参照してください。

時間に基づくスケジューリング

セットを処理する時期をコントロールする最も単純な方法は、時間に基づくスケジューリング (相対タイミングまたはインターバル・タイミングとしても知られています) を使用することです。特定の開始日付、時間、およびインターバルを決定します。インターバルは、特定の値 (1 分 ~ 1 年) または連続した値にすることができますが、時間間隔はおおよそその値になります。アプライ・プログラムは、ワークロードとリソースの可用性に基づいて、可能な限りすみやかにサブスクリプション・セットの処理を開始します。ある時間間隔を選択しても、レプリケーションが正確にその頻度で行われるとは限りません。連続タイミングを指定すると、アプライ・プログラムは可能な限り頻繁にデータを複製します。

イベントに基づくスケジューリング

イベントに基づくスケジューリング (イベント・タイミングとしても知られています) を使用してデータを複製するために、サブスクリプション・セットを定義するときにイベント名を指定できます。イベントの発生時に、アプライ・プログラムにそのイベントを認識させるには、イベント名に対応するタイム・スタンプをサブスクリプション・イベント (IBMSNAP_SUBS_EVENT) 表に移植することも必要です。アプライ・プログラムはイベントを検出するとレプリケーションを開始します。

サブスクリプション・イベント表には、表 4 で示されているように、4 つの列があります。

表 4. サブスクリプション・イベント表

EVENT_NAME	EVENT_TIME	END_OF_PERIOD	END_SYNCHPOINT
END_OF_DAY	2002-05-01- 17.00.00.000000	2002-05-01- 15.00.00.000000	

EVENT_NAME は、サブスクリプション・セットの定義時に指定するイベントです。EVENT_TIME は、アプライ・プログラムがセットの処理を開始する時刻を示すタイム・スタンプです。END_OF_PERIOD は、該当時刻より後の更新が後のイベントまたは時刻まで据え置かれることを指定するオプション値です。

END_SYNCHPOINT は、該当するログ・シーケンス番号より後の更新が後のイベントまたは時刻まで据え置かれることを指定するオプション値です。

END_OF_PERIOD と END_SYNCHPOINT の両方の値を指定すると、END_SYNCHPOINT の値が優先されます。EVENT_TIME の値はアプライ・コントロール・サーバーのクロックにより設定されますが、END_OF_PERIOD の値はソース・サーバーのクロックにより設定されます。2 つのサーバーが別の時間帯にある場合、この区別は重要です。

81 ページの表 4 によると、END_OF_DAY というイベントの場合、EVENT_TIME のタイム・スタンプ値 (2002-05-01-17.00.00.000000) は、アプライ・プログラムがサブスクリプション・セットの処理を開始する時刻です。END_OF_PERIOD のタイム・スタンプ値 (2000-05-01-15.00.00.000000) は、更新が複製されなかった後、次の日のサイクルで複製される時刻です。つまり、イベントは 3 時前に作成されたすべての未確定の更新を複製して、それに続くすべての更新を延期します。

ユーザーまたはアプリケーションは、行を表に挿入する SQL の INSERT ステートメントを使用してイベントをサブスクリプション・イベント (IBMSNAP_SUBS_EVENT) 表に通知し、イベントを活動化しなければなりません。たとえば、現行のタイム・スタンプに 1 分加算した値を使用して、EVENT_NAME に指定したイベントを起動します。このイベントに結び付けられたサブスクリプション・セットはすべて、1 分以内で実行するのに適格なものとなります。フル・リフレッシュと変更キャプチャー・レプリケーションの両方について、イベントを手動で通知しなければなりません。

イベントは、翌週、翌年、または毎週土曜日のように、前もって通知することができます。アプライ・プログラムが実行されている場合、アプライ・プログラムは指定されたおおよその時刻に処理を開始します。アプライ・プログラムは、指定された時刻に停止しており、後で再始動されると、サブスクリプション・イベント表をチェックして、通知されたイベントのサブスクリプション・セットの処理を開始します。

アプライ・プログラムはこの表を整理しません。ユーザーはこの表にデータを移植して保守しなければなりません。さらに、レプリケーション・センターを使用してサブスクリプション・イベント表を更新することはできません。この表にイベントを追加するには、SQL ステートメントを発行するか自動手順を定義しなければなりません。

例:

```
INSERT INTO ASN.IBMSNAP_SUBS_EVENT
    (EVENT_NAME, EVENT_TIME)
VALUES ('EVENT01', CURRENT_TIMESTAMP + 1 MINUTES)
```

アプライ・プログラムがサブスクリプション・セットを処理した最新の時点 (サブスクリプション・セットのコントロール表の LASTRUN 列にある値で指定) より前に発生するイベントは、期限切れのイベントと見なされて無視されます。そのため、アプライ・プログラムが実行中である場合は、期限切れのイベントを通知することを避けるため、時間的にわずかに先のイベントを通知しなければなりません。

サブスクリプション・セット内におけるソース表およびビューのターゲット表およびビューへのマッピング

サブスクリプション・セット内で、アプライ・プログラムがそのセットの処理中にグループとして処理するソースからターゲットへのマッピングを追加できます。ソースからターゲットへのこれらのマッピングを、サブスクリプション・セット・メンバーと呼びます。サブスクリプション・セット・メンバーを定義するとき、ソース・データをサブスクライブするターゲット表またはビューを指定し、複製されたデータのターゲットでの表示方法を定義できます。

前提条件:

ソースでの変更をサブスクライブするターゲットをセットアップする前に、ソースとして使用する表またはビューを登録しなければなりません。レプリケーション用のソースをまだ登録していない場合は、43 ページの『第 3 章 表およびビューを SQL レプリケーション・ソースとして登録する』の指示に従ってください。さらに、サブスクリプション・セットを作成したり、セットに追加するメンバーの数を計画する必要もあります。サブスクリプション・セットの作成が必要な場合は、72 ページの『サブスクリプション・セットの作成』を参照してください。サブスクリプション・セット・メンバーの計画が必要な場合は、70 ページの『サブスクリプション・セット・メンバー数の計画』を参照してください。

制約事項:

- DB2 レプリケーションは非 DB2 リレーショナル表のビューをソースとしてサポートしない。
- ターゲット・ビューを定義する場合、そのビューは挿入可能なビューでなければならない。すなわち、ビューのすべての列は更新可能でなければならない。ビューの全選択にキーワード UNION ALL を組み込むことができません。
- レプリケーション・センターを使用する場合、該当する列がターゲット表にまだ存在していなければ、サブスクリプション・セット・メンバーにその列を追加できない。
- **Windows、Linux、UNIX、z/OS の場合:** それぞれのサブスクリプション・セットごとに、最大 200 のメンバーを定義できます。
- **OS/400 の場合:** それぞれのサブスクリプション・セットごとに、最大 78 のメンバーを定義できます。

手順:

サブスクリプション・セット・メンバーを追加するのに、以下の 2 つの方法のどちらかを使用できます。

レプリケーション・センター

以下のノートブックのいずれかを使用します。

- サブスクリプション・セットの作成。サブスクリプション・セットを作成するときは、このノートブックを使用します。
- サブスクリプション・セット・プロパティ。すでにサブスクリプション・セットを作成済みで、このセットに 1 つ以上のサブスクリプション・セット・メンバーを追加する場合は、このノートブックを使用します。
- サブスクリプション・セットにメンバーを追加する。1 つのメンバーを複数のサブスクリプション・セットに追加するには、このノートブックを使用します。たとえば、このノートブックを開いて 4 つのサブスクリプション・セットを選択した場合、1 つのメンバーをそれぞれのセットに追加できます。各メンバーは同じソースを使用する必要があります。

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。

レプリケーション用のシステム・コマンド (OS/400)

ADDDPRSUBM システム・コマンドを使用します。このコマンドの構文お

よびパラメーターについての記述は、411 ページの『ADDDPRSUBM: DPR サブスクリプション・セットのメンバーの追加 (OS/400)』を参照してください。

ソースをターゲットにマップするには、ソースとして使用する登録済みの表またはビューについて、以下の情報を指定します。

- ソース表またはビュー、およびターゲット表またはビュー (ターゲット表の表スペースと索引も含まれる)。
- ターゲット表のタイプ。
- ターゲット表に複製するソース表の登録済みの列。

レプリケーション・センターを使用してソースをターゲットにマップするとき、LOB 列と DATALINK 列は列マッピングに自動的に組み込まれません。これらの列は明示的に選択しなければなりません。

- ターゲット表に複製するソース表の行 (行を指定するには WHERE 文節を組み込む)。

選択したソースを DB2 ターゲットにマップするには、ターゲット表またはビューについて、以下の情報を指定します。

- ターゲット表またはビューのスキーマ。
- ターゲットとして使用する表またはビューの名前。

デフォルト: デフォルト名は、ターゲット・サーバーのターゲット・オブジェクト・プロファイルがあれば、それからもたらされます。このプロファイルを設定していないと、デフォルトは TG にソース表またはビューの名前が続いたものになります。(たとえば、ソース表の名前が EMPLOYEE なら、ターゲット表のデフォルト名は TGEMPLOYEE になります。)

- ターゲット表のタイプ

デフォルト: ユーザー・コピー

指定したターゲット表が存在しない場合、レプリケーション・センターまたは **ADDDPRSUBM** システム・コマンドを使用して作成します。

選択したソースを非 DB2 リレーショナル・ターゲットにマップするには、ターゲット表について、以下の情報を指定します。

- ターゲット表のニックネームのスキーマ
- ターゲット表のニックネーム
- リモート・スキーマ
- リモート表の名前

デフォルト: デフォルト名は、ターゲット・サーバーのターゲット・オブジェクト・プロファイルがあれば、それからもたらされます。このプロファイルを設定していないと、デフォルトは TG にソース表またはビューの名前が続いたものになります。(たとえば、ソース表の名前が EMPLOYEE なら、ターゲット表のデフォルト名は TGEMPLOYEE になります。)

- ターゲット表のタイプ

デフォルト: ユーザー・コピー

サブスクリプション・セット・メンバーを追加するとき、ターゲット表のデフォルト・タイプであるユーザー・コピーが使用できます。あるいはレプリケーションの要求を満たす別のターゲット表タイプを選択することができます。

まだ存在しないターゲット表のサブスクリプション・セット・メンバーを追加するときは、デフォルトの設定値を使用することができます。あるいは、メンバーのプロパティを、レプリケーションの要求を満たすように変更できます。まず、使用するターゲット表のタイプを選出してから、アプライ・プログラムがそのターゲットにデータを複製する方法に合うようにプロパティを設定することができます。多様なレプリケーションのシナリオの記述と、それぞれのケースで使用するターゲット表のタイプについては、『ターゲット・タイプの選択』を参照してください。このセクションは、レプリケーションのゴールに基づく設定値選択全般を通じてのガイドとしても役立ちます。選択するターゲット・タイプにかかわらず、すべてのメンバーが共有するプロパティの共通セットを変更できます。サブスクリプション・セット・メンバー用のオプションとそのデフォルト、およびデフォルトの使用または変更が必要な場合についての説明の完全なリストは、97 ページの『すべてのターゲット表タイプに共通のプロパティ』を参照してください。

ターゲット・タイプの選択

このセクションでは、選択可能なターゲット表の各タイプについて記述し、選択するターゲット表のタイプと、レプリケーションのゴールに基づいたターゲット表のプロパティの定義方法の決定を支援します。また、既存の表をターゲットとして使用する場合に必要なことについても説明します。必要なターゲット表のタイプは、ターゲットでのデータ表示方法と、使用しているレプリケーション構成に左右されます。既存の表をターゲットとして使用できますが、新規の表も作成できます。

すべての非 DB2 のリレーショナル・ターゲット表と索引の名前は、DB2 の表と索引の命名規則に従わなければなりません。

制約事項:

- 変更後イメージ・ターゲットの列の NULL 属性は、ソース表またはビューのその列の NULL 属性と互換性がなければならない。既存の列に互換性を持たせるには、SQL の COALESCE 式を使用します。
- 非 DB2 リレーショナル・データベースのソース表の場合、ターゲット表は以下のタイプのみ定義できる。
 - ユーザー・コピー表
 - ポイント・イン・タイム表
 - 外部 CCD 表
- RRN 列をキー列として使用する OS/400 システムのソース表の場合、ターゲット表は以下のタイプのみ定義できる。
 - ポイント・イン・タイム表
 - 外部 CCD 表

- z/OS サブシステムのソース表については、ユーザー・コピー表用のサブスクリプション・セットの `where` 文節を満たすために、アプライ・プログラムが CD 表および UOW 表を結合する場合、CD 表および UOW 表のコード化スキームは同じでなければなりません。コード化スキームについての詳細は、599 ページの『付録 A. SQL レプリケーション用の Unicode および ASCII のコード化スキーム (z/OS)』を参照してください。

ターゲット表のタイプは以下から選択できます。

ユーザー・コピー

サブスクリプション・セット・メンバーで定義された列のみが組み込まれる読み取り専用ターゲット表です。ユーザー・コピー表の構造はソース表と同じにすることができ、そうでなければ、ソース列のサブセットを組み込むことができます。変更前イメージまたは算出列が含まれる場合と含まれない場合があります。

DB2 レプリケーションは、ユーザー・コピー・ターゲット表へのアプリケーション書き込みのみを想定しています。エンド・ユーザーまたはアプリケーションがユーザー・コピー表を直接変更しても、DB2 レプリケーションによって上書きされることがあり、そのためにソース表とターゲット表のデータが一致なくなってしまうことが起こりえます。ソース表とターゲット表の両方を更新する必要がある場合は、Update-anywhere レプリケーションを使用することを考えてください。

ポイント・イン・タイム

サブスクリプション・セット・メンバーで定義された列とタイム・スタンプ列を含む読み取り専用ターゲット表です。ポイント・イン・タイム表の構造はソース表と同じにすることができ、そうでなければ、ソース列のサブセットを組み込むことができます。変更前イメージまたは算出列が含まれる場合と含まれない場合があります。

基礎集約

SQL 列関数 (SUM、AVG など) を使用して、ソース表の内容全体のサマリーを計算する読み取り専用ターゲット表です。

基礎集約表は、ソース表の内容を要約します。基礎集約表には、アプライ・プログラムが集約を実行したタイム・スタンプも組み込まれます。基礎集約表は、ソース表の状態を定期的にトラッキングする場合に使用します。

変更集約

SQL 列関数 (SUM、AVG など) を使用して、ソース表に加えられた最近の変更の内容 (CD 表または内部 CCD 表に保管されています) 全体に関するサマリーを計算する読み取り専用ターゲット表です。

変更集約表は、ソース表ではなく、CD 表または内部 CCD 表の内容を要約します。変更集約表には、変更がキャプチャーされた (CD 表または CCD 表に書き込まれた) 時間間隔にマークを付けるための 2 つのタイム・スタンプも組み込まれます。レプリケーション・サイクルの合間に行われた変更 (UPDATE、INSERT、および DELETE 操作) をトラッキングする場合は、変更集約表を使用します。

CCD (整合変更データ)

レプリケーション・コントロール情報用の列が追加された読み取り専用ター

ゲット表です。これらの列に組み込まれるのは、ログ・レコード番号 (またはジャーナル・レコード番号)、ソース表が SQL の INSERT、DELETE、または UPDATE ステートメントを使用して変更されたかどうかの標識、および挿入、削除、更新に関連したコミット・ステートメントのログ・レコード番号とタイム・スタンプです。オプションで、変更前イメージ列と UOW 表からの列を組み込むこともできます。

レプリカ

Update-anywhere レプリケーション用の読み取り/書き込みターゲット表です。ユーザー・アプリケーション・プログラムまたはユーザーが直接更新できるタイプのターゲット表は、レプリカ表のみです。したがって、レプリカ表は、マスター表から、そしてローカル・アプリケーション・プログラムまたはユーザーから変更を受け取ります。レプリカ表の構造はソース表と同じにすることができ、そうでなければ、ソース列のサブセットを組み込むことができます。しかし、追加のレプリケーション・コントロール列 (タイム・スタンプなど) は組み込みません。レプリカ表は DB2 データベースでのみサポートされています。

以下のセクションで、各ターゲット・タイプごとに可能な使用方法について説明します。各セクションでは、使用できるターゲット表のタイプを通じてのガイドと、レプリケーションの要求を満たすターゲット表のプロパティの設定方法のガイドを行います。

- 『読み取り専用ターゲット表の定義』
- 90 ページの『行の最終的な変更をターゲット表に複製する』
- 91 ページの『複数層構成における中間層の定義』
- 94 ページの『読み取り/書き込みターゲットの定義 (Update-anywhere)』
- 96 ページの『既存の表をターゲット表として使用する』

ターゲット表のタイプを選択したら、そのターゲット表のデフォルトの設定値を使用できます。あるいは、ターゲット表のプロパティを、レプリケーションの要求を満たすように変更できます。共通のターゲット表オプションとそのデフォルト、およびデフォルトの使用または変更が必要な場合についての説明の完全なリストは、97 ページの『すべてのターゲット表タイプに共通のプロパティ』を参照してください。

読み取り専用ターゲット表の定義

ターゲット表タイプ: ユーザー・コピー、ポイント・イン・タイム、基礎集約、変更集約、CCD

ソース・データをターゲットでどのように表示するかによって、読み取り専用ターゲット表に以下のものが含まれるように定義できます。

- 『ソース表またはビューのコピー』
- 89 ページの『変更の履歴または監査情報』
- 88 ページの『ソースにおけるデータまたは変更の算出サマリー』

ソース表またはビューのコピー

ターゲット表タイプ: ユーザー・コピー、ポイント・イン・タイム

ソース表のコピー: デフォルトでは、サブスクリプション・セット・メンバーを定義するときに、ユーザー・コピー表がターゲット・タイプとして作成されます。コピー時にターゲット表をソース表に突き合わせる場合は、このデフォルト・タイプを使用します。ユーザー・コピー表に追加のレプリケーション・コントロール列は含まれませんが、ソース表の行または列のサブセット、あるいは複製されない追加の列を入れることができます。

タイム・スタンプが含まれたソース表のコピー: ターゲットに変更が適用されたポイント・イン・タイムをトラッキングする場合は、ポイント・イン・タイムをターゲット・タイプとして選択します。ポイント・イン・タイム・ターゲットにはソース表と同じデータが含まれ、アプライ・プログラムがそれぞれの行をいつターゲットにコミットしたかを知らせるために追加されたタイム・スタンプ列もあります。タイム・スタンプ列は、当初は NULL です。ポイント・イン・タイム表には、ソース表の行または列のサブセット、あるいは複製されない追加の列を入れることができます。

制約事項: DB2 は、AS IDENTITY GENERATED ALWAYS と定義されている DB2 表の列には、値が挿入されないようにします。この制約を回避するには、次のようにすることができます。

- IDENTITY CLAUSE が指定されていないターゲット表を作成する
- AS IDENTITY GENERATED BY DEFAULT が指定された列を持つターゲット表を作成する

ソースにおけるデータまたは変更の算出サマリー

ターゲット表タイプ: 基礎集約、変更集約

制約事項: 非 DB2 リレーショナル・ターゲットを、集約ターゲット表タイプにすることはできません。非 DB2 リレーショナル・ソースのタイプを、集約ターゲット表にすることはできません。

ソース表の内容全体またはソース表データに加えられた最新の変更のサマリーを含むターゲット表を作成することができます。集約ターゲット表タイプでは、COUNT、SUM、MIN、MAX、AVG などの集約 SQL 列関数を使用して、ターゲットの列を定義できます。これらの列には、オリジナルのソース・データは含まれません。含まれるのは、定義した SQL 関数の算出値です。アプライ・プログラムはフル・リフレッシュの間、集約を作成しません。行は、アプライ・プログラムがセットを処理する経過において付加されます。集約表を使用すると、DB2 レプリケーションが 1 行 1 行ではなくサマリー情報だけをレプリケーションできるため、ネットワーク帯域幅とターゲット表内のスペース両方を節約できるという利点があります。

ソース表の内容の要約:

基礎集約ターゲット表を使用して、各レプリケーション・サイクル内でソース表の状態をトラッキングします。基礎集約ターゲット表では、アプライ・プログラムがソース表から集約します (読み取って計算を実行します)。基礎集約表には、アプライ・プログラムが集約を実行したタイム・スタンプも組み込まれます。

基礎集約表のみが登録済みソース表のターゲットである場合、ソース表への変更をキャプチャーする必要はありません。

例: 週ごとの平均カスタマー数を知りたいとします。ソース表にカスタマーごとの行があれば、アプライ・プログラムが週ごとにソース表の行数を合計し、結果を基礎集約表に保管します。集約を毎週実行すると、ターゲット表にはその年の週ごとのカスタマー数を示す 52 の項目が含まれるようになります。

CD 表または CCD 表の内容の要約:

レプリケーション・サイクルの合間にソース表で行われた変更 (UPDATE、INSERT、および DELETE 操作) をトラッキングする場合は、変更集約表を使用します。変更集約ターゲット表では、アプライ・プログラムが CD 表または内部 CCD 表から集約します (読み取って計算を実行します)。変更集約表には、キャプチャー・プログラムが CD 表または CCD 表に変更を挿入した時間間隔にマークを付けるための 2 つのタイム・スタンプも組み込まれます。

例: 毎週新たに獲得したカスタマー数 (INSERT) と、失った既存のカスタマー数 (DELETE) を知りたいとします。CD 表の挿入された行と削除された行の数を週ごとに数え、その数を変更集約表に保管します。

重要: サブスクリプション・セット・メンバーのソース表がフル・リフレッシュのみのレプリケーション用に登録済みの場合、ソースで CD 表または CCD 表が必須である変更集約ターゲット表を持つことはできません。

変更の履歴または監査情報

ターゲット表タイプ: CCD

ソース・データの監査、あるいはデータの使用状況の履歴を保持することが必要な場合があります。ターゲット・タイプとして CCD 表を使用すると、CCD 表をどのように定義するかによって決まる多様な方法で、ソースの変更履歴をトラッキングできます。たとえば、データの変更が発生したときの変更前後の比較や、ソース表への更新を行ったユーザー ID をトラッキングできます。

ソース表の履歴を保持する読み取り専用ターゲット表を定義するには、以下の属性を持つようにターゲット CCD 表を定義します。

非コンデンス

ソースの変更すべてに関するレコードを保持するには、CCD 表を非コンデンスに定義し、発生した変更ごとに 1 行が保管されるようにします。非コンデンスの表には同じキー値を持つ複数の行が含まれるため、ユニーク索引は定義しないでください。非コンデンス CCD 表は、UPDATE、INSERT、または DELETE 操作ごとに 1 行を保有することで、ソース表に対して実行された操作の履歴を保留します。UPDATE 操作を INSERT および DELETE 操作 (パーティション化キー列用) としてキャプチャーする場合、CCD 表は、それぞれの更新ごとに 2 つの行、すなわち DELETE に 1 行、INSERT に 1 行を指定します。

コンプリートまたは非コンプリート

CCD 表をコンプリートにするか、あるいは非コンプリートにするかを選択できます。未完成の CCD 表にはソース行のコンプリート・セットが最初含まれていないため、ソース表への更新 (アプライ・プログラムが CCD 表の移植を開始してからの更新) 履歴を保持する未完成の CCD 表を作成します。

UOW (作業単位) 列の組み込み

監査機能の改善のため、UOW 表からの追加の列を組み込んでください。ユーザー指向の識別がさらに必要であれば、UOW 表で、DB2 for z/OS の関連 ID に関する列、1 次許可 ID、または OS/400 のジョブ名およびユーザー・プロファイルを使用することができます。CCD 表に組み込み可能な UOW 列についての詳細は、594 ページの『整合変更データ (CCD) 表』を参照してください。

行の最終的な変更をターゲット表に複製する

ターゲット表タイプ: 内部 CCD

ソース表で変更が頻繁に発生する場合、最後のアプライ・サイクル以降にソースで発生したコミット済みの変更を要約するために、内部 CCD 表を作成できます。キャプチャー・プログラムがログからの変更を追加するとき、CD 表は常に流動的であるため、CCD のソース変更のローカル・キャッシュが、ターゲット用のより継続的なソースとして機能します。

オリジナルのソース表が更新される時、キャプチャー・プログラムはソースのログでの頻繁な変更を読み取り、その変更をソースの CD 表に追加します。その CD 表から、アプライ・プログラムは CD 表での変更を読み取り、内部 CCD 表に移植します。内部 CCD 表を、最終のサイクル内で発生した CD 表のそれぞれの行ごとの最新の変更のみが含まれるように定義できます。したがって、CCD 表はアプライ・サイクルの間では静的であるため (CD 表から CCD 表に複製するアプライ・プログラムの場合)、ターゲット用のより継続的なソースになります。ソースからのコンデンス変更を行うことで、同じ行の多数の変更をターゲット表に複製しなくなるため、レプリケーションのパフォーマンス全体が改善されます。

キャプチャー・プログラムは新しい変更を常に CD 表に追加しているため、2 番目のアプライ・プログラムは CD 表の代わりに内部 CCD 表から変更を読み取ります。そのため、様々な変更を様々なターゲットに複製することはなく、ターゲット相互の同期がとれた状態を保つことができます。2 番目のアプライ・プログラムは、オリジナルのソース表をフル・リフレッシュのために使用し、内部 CCD 表を変更キャプチャー・レプリケーションのために使用します。

推奨事項:

- ソース表と内部 CCD 表との間でサブスクリプション・セット・メンバーを定義するのは、ソース表と他のターゲット表との間で他のサブスクリプション・セット・メンバーを定義する前に行う。そうするとアプライ・プログラムは、ソース表から変更を複製するために CD 表ではなく 内部 CCD 表を使用するようになります。ソース表に内部 CCD 表を定義する前に他のサブスクリプション・セット・メンバーを定義し、それらのメンバーを使用してレプリケーションを開始すると、ソース表のすべてのターゲットに対してフル・リフレッシュの実行が必要になる可能性があります。
- すべての内部 CCD 表を 1 つのサブスクリプション・セットに結合して、ソース・データベースのすべてのターゲット表相互の同期が必ずとれるようにする。
- 頻繁に変化するソース列のサブセットを他のターゲットに適用するだけの場合でも、すべての登録済みソース列が内部 CCD に複製されるというデフォルトを使用する。そうすると、オリジナルのソース表の他の登録済み列のデータを今後必要とする可能性のあるターゲット表のために、内部 CCD 表をソースとして使用

できます。今後のどのようなターゲットへの変更キャプチャー・レプリケーションにおいても使用可能なのは、内部 CCD 表の列のみです。

内部 CCD 表は、レプリケーションにおける暗黙的なソースとして使用します。レプリケーション・ソースとして明示的に定義することはできません。サブスクリプション・セット・メンバーを追加する場合、オリジナルのソース表 (内部 CCD 表ではありません) をターゲット表にマップします。内部 CCD 表には以下の属性があります。

内部 CCD 表はソースの CD 表の代わりとして機能します。内部 CCD 表についての情報は、登録 (IBMSNAP_REGISTER) 表で、この内部 CCD 表のソース表と同じ列に保管されます。内部 CCD 表独自の行は、登録表内にはありません。アプライ・プログラムは、CD 表からではなく、内部 CCD 表 (存在する場合) から変更を自動的に複製します。それぞれのレプリケーション・ソースごとに、内部 CCD 表は 1 つだけ存在できます。

制約事項: ユーザー表には算出列は含まれないため、CCD サブスクリプションに算出列を組み込まないでください。

ローカル

CCD 表はソース表と同じデータベース内にあります。

非コンプリート

アプライ・プログラムは、フル・リフレッシュのために内部 CCD ではなくオリジナルのソース表を使用するため、後続ターゲットにすべてのソース行の初期コピーがすでに含まれるという理由から、CCD は非コンプリートです。

コンデンス

内部 CCD はコンデンスされています。すなわち、その表にはキー値ごとに 1 行が含まれ、アプライ・プログラムは変更ごとに 1 行をアプライする代わりに、CCD 表のそれぞれの行ごとに最新の変更をアプライするようになります。

UOW 列なし

内部 CCD 表では、UOW 表の列の追加はサポートされません。UOW 列を組み込んでいるターゲット CCD 表をすでに定義している場合は、内部 CCD 表を使用できません。

重要 (Update-anywhere の場合): 内部 CCD 表を定義すると、アプライ・プログラムはレプリカをターゲットとしてサブスクリプション・セットを処理するときにそれを無視します。マスター・ソースの CD 表からレプリカへの変更を適用します。

複数層構成における中間層の定義

ターゲット表タイプ: CCD

基本のレプリケーション・モデルは、ソースが 1 つでターゲットが 1 つ以上の 2 層モデルですが、3 (またはそれより多い) 層の構成もセットアップできます。複数層構成にはソース表とターゲット表が含まれ、そのターゲット表は他のターゲット表にはソースとして機能します。

複数層レプリケーション環境をセットアップする 1 つの理由は、第 3 層のターゲットに継続的なソースを提供することです。変更を層 1 から層 2 の CCD 表に収集できるため、各層への変更を複製する頻度をコントロールでき、ターゲット (層 3) に複製される変更の数を削減できます。しかも、ソース・システムへのデータベース接続の多くを回避できるので、接続コストを第 2 層に移動できます。

たとえば、3 層モデルで、第 1 層 (層 1) はソース・データベース、第 2 層 (層 2) は層 1 のターゲットです。しかも層 2 はターゲットを含む第 3 層 (層 3) のソースであり、変更を 1 つまたは多数の層 3 のデータベースに分散できます。レプリケーション構成内に 2 つより多い層が含まれる場合、ソースとターゲットの両方として機能する中間層は CCD 表です。

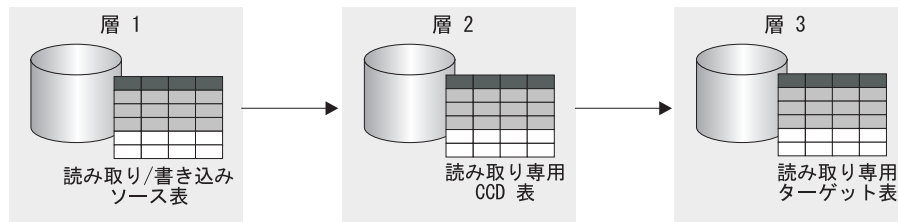


図 6. 3 層レプリケーションのモデル：データを、ソース表からターゲット表に、さらにその表から別のターゲット表にレプリケーションできます。

制約事項:

非 DB2 リレーショナル表または非 DB2 リレーショナル・データベースの CCD 表は、複数層構成の中間層としては使用できません。

手順:

この手順はレプリカ表にも適用されます。CCD 表は通常読み取り専用レプリケーションに使用されますが、レプリカ表は Update-anywhere レプリケーションに使用されます。

複数層レプリケーションをセットアップするには、ターゲット表が後続のターゲットへのソースとして機能するようにします。

1. ソース表 (層 1) をレプリケーション用に登録する。表をレプリケーションのために登録する方法については、43 ページの『DB2 表をソースとして登録する』を参照してください。

このソースにおいて、キャプチャー・プログラムは層 1 で発生する変更をキャプチャーし、それを層 1 の CD 表に保管します。

2. ソース・サーバーと (層 2 の) ターゲット・サーバーとの間でサブスクリプション・セットを作成する。サブスクリプション・セットの作成方法については、72 ページの『サブスクリプション・セットの作成』を参照してください。

このサブスクリプション・セットにおいてアプライ・プログラムは、層 1 から層 2 の CCD 表に変更を適用します。

3. ソース表 (層 1) と CCD ターゲット表 (層 2) をマッピングするサブスクリプション・セット・メンバーを定義する。サブスクリプション・セット・メンバー

の定義方法については、82 ページの『サブスクリプション・セット内におけるソース表およびビューのターゲット表およびビューへのマッピング』を参照してください。

このメンバーにターゲット表を定義するとき、以下の属性を持つ CCD 表になるようなターゲット表を選択します。

外部登録済みソース

外部ターゲット表として定義しなければならず、この表が後続の層のためのソースとして機能できるように登録しなければなりません。他の登録済みソースと同様に、外部 CCD 表にも独自の行が登録 (IBMSNAP_REGISTER) 表内にあります。ソースとしても機能可能な外部 CCD 表は、ただ 1 つのソース表によってのみ移植されます。

すべての外部 CCD 表は、同じキャプチャー・スキーマを使用してサブスクリプション・セットに登録しなければなりません。

コンプリート

後続の層においてアプライ・プログラムがフル・リフレッシュと変更キャプチャー・レプリケーションの両方の実行に使用するため、完全な CCD 表を使用しなければなりません。

コンデンス

コンデンスされた CCD を使用します。すなわち、この表にはキー値ごとに 1 行が含まれ、必ず最新の変更のみが後続の層に複製されます。アプライ・プログラムは変更ごとに 1 行を適用する代わりに、CCD 表のそれぞれの行ごとに最新の変更を適用します。コンデンスされた表はそれぞれの行ごとにユニーク・キー値が必須であるため、ユニーク索引を定義しなければなりません。

4. CCD 表が登録されるため、中間層のデータベースにキャプチャー・コントロール表がまだない場合は作成する。
5. 登録済み CCD 表が含まれる層 2 のサーバーと (層 3 の) 後続のターゲット・サーバーとの間で、サブスクリプション・セットを作成する。サブスクリプション・セットの作成方法については、72 ページの『サブスクリプション・セットの作成』を参照してください。

このセットにおいてアプライ・プログラムは、CCD 表から後続の層のターゲット表に変更を適用します。アプライ・プログラムは、CCD 表をフル・リフレッシュと変更キャプチャー・レプリケーションの両方に使用します。アプライ修飾子は通常、CCD にデータを入れるのに使った修飾子とは別のものを使用しますが、同一のものを使用することも可能です。

6. CCD ソース表 (層 2) と後続のターゲット表 (層 3) をマッピングするサブスクリプション・セット・メンバーを定義する。サブスクリプション・セット・メンバーの定義方法については、82 ページの『サブスクリプション・セット内におけるソース表およびビューのターゲット表およびビューへのマッピング』を参照してください。

この CCD ソース表をサブスクライブするターゲット表を組み込んで、複数のメンバーをセットアップできます。これが複数層構成の最終層であれば、ターゲット表のタイプは何でも可能です。ただし、計画している層が 3 つより多い場合

は、層 3 のターゲット表をステップ 3 で示されているように定義し、ステップ 4 から 5 を繰り返して後続の層を追加します。

重要: フル・リフレッシュが外部 CCD (中間層) で発生した場合、ソースとしてその外部 CCD を使用する、すべての後続の層におけるアプライ・プログラムはフル・リフレッシュを実行します。これをカスケード・フル・リフレッシュ と呼びます。

読み取り/書き込みターゲットの定義 (Update-anywhere)

ターゲット表タイプ: レプリカ

Update-anywhere レプリケーションでは、マスター・ソース表における変更はレプリカ・タイプの従属ターゲット表に複製され、さらにそのレプリカ表における変更はマスター・ソース表に逆に複製されることが可能です。Update-anywhere レプリケーションで、マスター表とそのレプリカのすべては、ソースとターゲットの両方として機能する読み取り/書き込み表です。

前提条件:

Update-anywhere レプリケーションに必要な条件は以下のとおりです。

- 単一アプリケーション・プログラムはマスター表とレプリカ表の両方を更新しないため、参照保全制約を宣言して使用しなければならない。参照保全違反はアプリケーションのロジックでは検出できません。
- 参照保全違反を防止するため、複数のマスター表を対象に存在するすべての参照制約をレプリカ表に含めなければならない。いくつかの参照制約を省略すると、レプリカ表に加えられた更新がマスター表に複製されるとき、参照保全違反を引き起こす可能性があります。管理ツールは、ソース表からターゲット表への参照制約定義のコピーを行わず、新しい制約を生成することもできません。
- フル・リフレッシュ中の参照保全チェックをバイパスするには、ASNLOAD 出口ルーチンを使用しなければならない。

制約事項:

- リモート・ジャーナル構成ではレプリカ・ターゲット表タイプはサポートされません。
- Update-anywhere レプリケーションで、CCD 表をソースまたはターゲットとして使用することはできない。
- LOB データ・タイプの列を Update-anywhere レプリケーションの対象にできるようにするには、登録表の CONFLICT_LEVEL を 0 に設定しなければならない。
- DATALINK データ・タイプの列は、ソース表を競合検出ないように登録する場合を除いて、Update-anywhere レプリケーションの対象にならない。
- 非 DB2 データベースのタイプをレプリカ・ターゲット表にすることはできない。そのため、Update-anywhere レプリケーションの対象にできません。

手順:

マスター表と 1 つ以上のレプリカ表との間で (それぞれのレプリカ表が別々のデータベースにある場合) Update-anywhere 構成をセットアップするには、以下のようになります。

1. キャプチャー・プログラムはそれぞれのレプリカ表ごとに変更をキャプチャーするため、レプリカ表を入れるデータベースそれぞれにキャプチャー・コントロール表がまだない場合は作成する。
2. ソース表 (マスター表) をレプリケーションのために登録する。表をレプリケーションのために登録する方法については、43 ページの『DB2 表をソースとして登録する』を参照してください。

このソースにおいて、キャプチャー・プログラムはマスター表での変更をキャプチャーし、それをマスターの CD 表に保管します。

3. 1 つ以上のレプリカを入れるマスター・データベースとターゲット・データベースとの間に、サブスクリプション・セットを作成する。サブスクリプション・セットの作成方法については、72 ページの『サブスクリプション・セットの作成』を参照してください。

すべてのレプリカ表が同じデータベースに、そしてすべてのマスター表が別のデータベースに含まれている場合、サブスクリプション・セットは 1 つだけ必要です。レプリカ表が複数のデータベースに含まれている場合、保有するレプリカ・データベースと同数のサブスクリプション・セットが必要です。

4. マスター表とそのおのおのに関連したレプリカ表との間のマッピングごとに、サブスクリプション・セット・メンバーを定義する。サブスクリプション・セット・メンバーの定義方法については、82 ページの『サブスクリプション・セット内におけるソース表およびビューのターゲット表およびビューへのマッピング』を参照してください。

この構成では、レプリカ表が含まれたサーバーで通常実行されるアプライ・プログラムは 1 つしかありません。このセットにおけるアプライ・プログラムは、マスターの CD 表から変更をプルして、それをレプリカ表に適用します。また、アプライ・プログラムはレプリカ表の CD 表から変更をプッシュして、それをマスター表に適用します。

重要: Update-anywhere 構成において、マスター表とレプリカ表はデータを往復させて互いに複製するため、レプリカ・ターゲット表にはソース表と同じ列が含まれている必要があります。脱落した列がマスター・サイトで NULL 可能、または NOT NULL WITH DEFAULT として定義されている場合に限り、マスター表の列のサブセットが含まれるレプリカ・ターゲットを作成できますが、レプリカで新しい列の追加または列の名前変更は行わないでください。

5. レプリカ表にソースのプロパティを定義する。

レプリカ表を組み込んでサブスクリプション・セット・メンバーを作成するとき、DB2 レプリケーションはそのレプリカ表をレプリケーション・ソースとして自動的に登録します。レプリカ・ターゲット表はソースとして機能するため、ターゲット表に共通するプロパティとは別に設定可能なプロパティがあります。これらのプロパティにより、キャプチャー・プログラムがレプリカへの変更をハンドルする方法が決定されます。ただし、マスター表から継承されても、レプリカ表に合うように変更できないプロパティが 2 つあります。競合検出レベルとフル・リフレッシュの可否のプロパティです。このソースにおいて、キャプチャー・プログラムはレプリカ表での変更をキャプチャーし、それをレプリカの CD 表に保管します。登録オプションとそれらのデフォルトの完全なり

スト、およびこれらのデフォルトを使用した方がよい場合と変更した方がよい場合についての説明は、47 ページの『ソース表用の登録オプション』を参照してください。

重要: マスターとレプリカがソースとターゲットの両方として機能するとしても、フル・リフレッシュ・コピーは、マスターからレプリカへのみ行われ、レプリカからマスターへは行われません。

競合を防止するため、レプリカ表のターゲット・キーを、マスター・ソース表の主キーまたはユニーク索引と同じにしなければなりません。マスター表はレプリカを更新でき、レプリカはマスターを更新できるため、アプライ・サイクルの間にマスター表のある行が更新され、1 つ以上のレプリカ表の同じ行にそれとは異なる更新が加えられた場合 (変更がマスター CD 表とレプリカ CD 表で行われます)、競合が発生する可能性があります。レプリカ表は競合検出レベルをマスター・ソース表またはビューから継承します。最善なのは、マスターからすべてのレプリカ表ヘデータを複製するときに、競合が決して発生しないようにアプリケーションを設計することです。マスター・ソースを登録したとき、競合検出のレベルを 3 つの選択肢から選びました。競合検出レベルの選択と、(標準または拡張競合検出を選択した場合に) 発生する競合への対応方法についての詳細は、59 ページの『競合検出の設定 (Update-anywhere レプリケーション)』を参照してください。

ソース表における参照保全制約を定義した場合、保全違反を防止するために、同じ参照保全制約をレプリカ表にも定義しなければなりません。参照保全違反が発生した場合、サブスクリプション・サイクルは自動的に再試行されます。

既存の表をターゲット表として使用する

事前定義した DB2 表を、サブスクリプション・セットでのターゲット表として使用できます。すなわち、DB2 レプリケーションの外部で定義されているターゲット表を組み込むサブスクリプション・セット・メンバーを定義することができます。このようなユーザー定義のターゲット表は、表の構造が有効である限り、レプリケーションで有効なターゲット表タイプ (ユーザー・コピー、ポイント・イン・タイム、基礎または変更集約、CCD、あるいはレプリカ) のいずれかになります。たとえば、ユーザー定義のポイント・イン・タイム表には、IBMSNAP_LOGMARKER という TIMESTAMP タイプの列が組み込まれなければなりません。

要件:

- サブスクリプション・セット・メンバーの定義に含まれる列が既存のターゲット表より少ない場合、レプリケーションに関連するターゲット表の列は NULL が許可されているか、あるいは NOT NULL WITH DEFAULT として定義されていないなければならない。
- ポイント・イン・タイム CCD 表、ユーザー・コピー CCD 表、レプリカ CCD 表、およびコンデンス CCD 表にはユニーク索引が必要である。既存のターゲット表を使用してサブスクリプション・セット・メンバーを定義するときは、既存のユニーク索引を使用するか、または新規のユニーク索引を指定することができます。

制約事項:

- サブスクリプション・セット・メンバーの定義に、既存のターゲット表より多くの列を含めることはできない。
- レプリケーション・センターを使用する場合、該当する列がターゲット表にまだ存在していなければ、サブスクリプション・セット・メンバーにその列を追加できない。

DB2 レプリケーションは、既存のターゲット表とサブスクリプション・セット・メンバー定義との間の矛盾をチェックします。

重要 (複数層の場合): ソース表が層 1、CCD 表が層 2、そして既存の表が層 3 であるような複数層構成をセットアップする場合、層 1 と層 2 との間でサブスクリプション・セット・メンバーを定義するときに、CCD 表を既存のターゲット表に指定されている属性に一致するように定義します。そして、この CCD 表をソース表とする既存のターゲット表のサブスクリプション・セット・メンバーを定義します。

すべてのターゲット表タイプに共通のプロパティ

このセクションでは、ターゲット表を作成するときにタイプに関係なく設定できる共通のプロパティについて説明します。必要なレプリケーションのタイプを基にして、使用するターゲット表またはビューに合うようにプロパティを変更することができます。以下のセクションでは、ソース・データをターゲット表にマップする方法に応じて定義できる共通な特性について説明します。

- 『ターゲットに適用するソース列』
- 98 ページの『ターゲットに適用するソース行』
- 99 ページの『ソース列からターゲット列にマップする方法』
- 100 ページの『ターゲット・キー』
- 101 ページの『アプライ・プログラムがターゲット・キー変更オプションを使用してターゲット・キー列を更新する方法』

ターゲットに適用するソース列

デフォルト: すべての登録済みソース列はターゲットに複製されます。

レプリケーションのシナリオによっては、すべての列をターゲット表に複製したいわけではない場合があります。また、ソース表に定義したすべてのデータ・タイプでターゲット表がサポートできないものがある場合もあります。ソース表より列数が少ない列 (垂直) のサブセットを定義することができます。

デフォルトでは、ターゲット表にはソース表の LOB 列と DATALINK 列を除くすべての登録済みの列が含まれます。ソース表に存在する列で、ターゲット表に入れたくないものがある場合は、ターゲット表に複製したいソース列のみを選択します。選択しないソース表の登録済みの列は、他のサブスクリプション・セット・メンバーに引き続き使用できますが、現行のソースからターゲットへのマッピングには組み込まれません。

算出列をターゲット表に追加することもできます。これらの列は、SUBSTR のような SQL スカラー関数で定義するか、あるいは列 A の値を列 B の値で除算する (colA/colB) ような派生列にすることができます。これらの算出列は、ソース表のすべての列を参照します。

ターゲットに適用するソース行

デフォルト: すべてのソース行はターゲットに複製されます。

デフォルトでは、ターゲット表にはソース表のすべての行が含まれます。レプリケーションのシナリオによっては、すべての行をソース表からターゲット表に複製したいわけではなかったり、あるいは異なる種類のデータを含むソース行を異なるターゲット表に複製したいという場合があります。一定の条件 (SQL WHERE 文節) と一致する行が含まれる行 (水平方向) サブセットを定義できます。SQL の述部には、通常 ID または区切り ID を入れることができます。WHERE 文節についての詳細は、「DB2 SQL リファレンス」を参照してください。

例:

- ターゲット表が、社内のある業務部門の業務データを格納したものであると想定します。WHERE 文節をサブスクリプション・セット・メンバーで定義すると、その部門 (またはその部門内のすべての部) のすべての行を、ソース表からターゲット表にレプリケーションできます。
- 同じデータベース内にいくつかのターゲット表があると想定する。WHERE 文節を 1 つのサブスクリプション・セット・メンバーで定義すると、すべての LOB 列 (主キー列も加えて) を 1 つのターゲット表にレプリケーションでき、WHERE 文節を別のサブスクリプション・セット・メンバーで定義すると、他の列すべてを別々のターゲット表にレプリケーションできます。したがって、ターゲット・データベースはソース表からのデータすべてを保有できますが、ターゲット・データベースのソース表を非正規化して、データウェアハウスへの照会パフォーマンスを調整します。

行述部の制約事項:

- この文節に WHERE を入力しない。これは暗黙で指定されます。副選択ステートメントの文節にのみ WHERE を入力してください。
- 文節をセミコロン (;) で終わらせない。
- 変更前イメージ列、算出列、または IBMSNAP 列を使用して、データをサブセット化またはフィルター操作する場合は、117 ページの『サブスクリプションの際のデータのサブセット化』を参照してください。
- WHERE 文節にブール式 OR を含める場合、たとえば、(COL1=X OR COL2=Y) のように、述部を括弧で囲む。
- ターゲット表が変更集約表であって、変更前イメージ列を含んでいる場合、変更前イメージ列を GROUP BY 文節に含めなければならない。

次に示す例には、ターゲット表の行をフィルターに掛けるために使用できる WHERE 文節が含まれます。これらの例は一般的であり、モデルとして使用するものです。

- 特定の値を持つ行を指定する WHERE 文節

特定の値 (たとえば、管理職にある社員を表す MGR) をもつ行だけをコピーするには、次のような WHERE 文節を使用します。

```
EMPLOYEE = 'MGR'
```

- ある範囲の値を持つ行を指定する WHERE 文節

ある範囲の値 (たとえば従業員番号 5000 ~ 7000) の行をターゲット表にコピーするには、以下のような WHERE 文節を使用します。

```
EMPID BETWEEN 5000 AND 7000
```

ソース列からターゲット列にマップする方法

デフォルト: ソース列名は同じターゲット列名にマップします (ターゲット表がまだ存在しない場合)。

デフォルトでは、ターゲット表 (存在していない場合) の列名はソース表の列名と一致し、ソース列のデータ値は同じ名前のターゲット列に複製されます。レプリケーション・コントロール列 (IBMSNAP または IBMQSQ で始まる) を除くターゲット表のすべての列の名前を変更できます。ターゲット表が存在していれば、レプリケーション・センターは列を名前でもマップします。

ターゲット表の列を、ソース列とは異なる長さにすることができます。ターゲット列がソース列より短い場合、サブスクリプション・セット・メンバーの式を使用して文字を長い列から短い列にマップしたり、その式を含むビューを登録できます。たとえば、ソース列が char(12) でターゲット列が char(4) である場合、次の式を使ってレプリケーション中に COL1 からの値を切り捨てることができます。

```
substr(col1, 1,4)
```

ターゲット列名が長い場合は、ターゲット列にブランクを埋め込みます。

DB2 表を、既存のニックネームを持つ非 DB2 リレーショナル表にマップする場合、一部の列のデータ・タイプに互換性がないことがあります。ソース列のデータ・タイプにターゲット列のデータ・タイプとの互換性がない場合、以下のようにターゲットでデータ・タイプを変更して、ソースとの互換性を持たせることができます。

- ソースのデータ・タイプがターゲットで必要とするデータ・タイプに一致するように調整するための算出列を追加できる。
- 非 DB2 リレーショナル・ターゲット表のニックネームを変更して、データ・タイプ変換を変更できる。

DB2 Universal Database の長可変文字 (LONG VARCHAR) から z/OS と OS/400 の両方へのマップには、いくらかの制約事項があります。データ・タイプの制約事項の詳細については、105 ページの『レプリケーションにおける一般的なデータの制約事項』を参照してください。

例: データ・タイプが DATE の DB2 列を含む DB2 ソース表から、データ・タイプが DATE の Oracle 列を含む Oracle ターゲット表にデータを複製するとします。

表 5. DB2 DATE 列の Oracle DATE 列へのマッピング

DB2 列	ニックネーム・データ・マッピング	Oracle 列
A_DATE DATE	A_DATE TIMESTAMP A_DATE DATE	A_DATE DATE

Oracle データ・タイプ DATE を含む Oracle ターゲット表が作成されます (日付データとタイム・スタンプ・データの両方が入ります)。フェデレーテッド・データベースにある Oracle の DATE データ・タイプの初期のニックネームは、DB2 データ・タイプを TIMESTAMP としてマップします。DB2 レプリケーション・センターとレプリケーション用の OS/400 システム・コマンドは、ニックネームのデータ・タイプを DATE に変更するので、DATE は Oracle に複製されますが、TIMESTAMP はされません。

レプリケーション・センターを使用してターゲット表を作成する際に、ターゲット表のタイプに関係なく、ターゲット表で列の名前を変更できます。さらに、列の属性 (データ・タイプ、長さ、位取り、精度、NULL 可能かどうか) に互換性があれば、それら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、既存のターゲット表の列を名前変更することはできません。ソース列とターゲット列が一致しない場合は、レプリケーション・センターを使用してソースからターゲットに列をマップするか、あるいはソースの列名に一致する名前を含むターゲット表のビューを作成するかのどちらかが可能です。

ターゲット・キー

デフォルト索引名: デフォルト名は、ターゲット・サーバーのターゲット・オブジェクト・プロファイルがあれば、それからもたらされます。このプロファイルを設定していないと、デフォルトは「IX + ターゲット表の名前」となります。たとえば、ターゲット表の名前が TGMEMPLOYEE なら、ターゲット表の索引のデフォルト名は IXTGMEMPLOYEE になります。

コンデンス・ターゲット表が変更キャプチャー・レプリケーションに関係している場合、アプライ・プログラムは、そのターゲット表にターゲット・キーと呼ばれる主キーまたはユニーク索引があることを必要とします。ターゲット表のユニーク索引として使用する列は選択できます。以下のタイプのターゲット表はコンデンスされており、ターゲット・キーが必須です。

- ユーザー・コピー
- ポイント・イン・タイム
- レプリカ
- コンデンス CCD

ターゲット表を新規に作成する場合は、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。

新しいターゲット表のユニーク索引を作成するためのオプションが 2 つあります。

- ターゲット表のユニーク索引として使用する列を指定する。
- DB2 レプリケーションでユーザー向けのユニーク索引を選択する。

ユニーク索引用の列を選択しない場合、DB2 レプリケーションではソース表に以下の定義のいずれか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

DB2 レプリケーションで、ソース表のこれらの定義のいずれかが検出され、関連した列が登録されていてターゲット表の一部であることが検出された場合、DB2 レプリケーションはソース表の主キー（またはユニーク索引か RRN）をターゲット・キーとして使用します。ユニーク制約の場合、DB2 レプリケーションは制約列を使用して、ターゲット表のユニーク索引を作成します。

主キーまたはユニーク索引を持たない OS/400 ソース表の場合は、相対レコード番号 (RRN) を一意性の因子として使用するよう表の登録を変更します。サブスクリプション・セット・メンバーを定義するときに、RRN 列をターゲット表のユニーク索引として指定します。OS/400 ソース表における RRN の定義についての詳細は、62 ページの『主キーの代わりに相対レコード番号 (RRN) を使用する (OS/400)』を参照してください。

ターゲット・キーとして RRN を使用する OS/400 システムのターゲット表の場合、これらのターゲット表を複製するには、アプライ・プログラムを OS/400 システムで実行します。

既存のターゲット表の場合、ユニーク索引を選択する必要があります。以下のオプションのいずれかを選択することができます。

- ターゲット表の既存の索引を使用する。

既存の索引を使用するには、レプリケーション・センターで、索引を表す列を選択します。レプリケーション・センターが厳密に一致するものを検出すれば、アプライ・プログラムが使用するようターゲット・キーを設定するだけですが、検出しなかった場合は、ユニーク索引を作成してから、ターゲット・キーをアプライ・プログラムが使用するよう設定します。

- ターゲット表の別の索引を作成する。

ユニーク索引がまだない場合は作成され、アプライ・プログラムが使用するようターゲット・キーが設定されます。

重要: ソース表で更新される可能性のある列が組み込まれたターゲット表のキーを選択する場合、ターゲット・キー列に特殊な更新を加えるようアプライ・プログラムに指示しなければなりません。詳細については、『アプライ・プログラムがターゲット・キー変更オプションを使用してターゲット・キー列を更新する方法』を参照してください。

アプライ・プログラムがターゲット・キー変更オプションを使用してターゲット・キー列を更新する方法

制約事項:

- 更新を削除および挿入のペアとしてキャプチャーするために登録されたソース表において、ターゲット・キー変更オプションを使用することはできません。

- アプライ・プログラムが、ターゲット・キー列の変更前イメージを基にして、ターゲット表を更新する場合、ソース表内の式を、ターゲット表内のキー列にマップすることはできません。(すなわち、IBMSNAP_SUBS_MEMBR 表の TARGET_KEY_CHG 列にそのターゲット表に対する値 Y がある場合です。)

サブスクリプション・セット・メンバーを定義するときにターゲット・キー変更オプションを選択すると、ターゲット・キーの変更時、アプライ・プログラムはターゲット・キー列に特殊な更新を加えます。アプライ・プログラムでこれらの特殊な更新を行うためには、ターゲット表のターゲット・キー列の一部になっているソース表の列を、CD (または CCD) 表の変更前イメージ列に登録する必要があります。ターゲット・キーを構成する列の変更前イメージ値をキャプチャーするためのソース登録を定義していない場合は、異なるキーを持つターゲット表をサブスクライブする前に、登録を変更して変更前イメージ値を組み込まなければなりません。

ターゲット・キー列の変更前イメージ値が CD (または CCD) 表にあることを確認後、ターゲット・キー列の更新時にアプライ・プログラムが変更前イメージ値を使用するように、サブスクリプション・セット・メンバーのオプションを選択します。

ターゲット・キー列の更新時にアプライ・プログラムが変更前イメージ値を使用するように指定しない場合、ターゲット・キーの一部となっているソース表の列を更新しても、DB2 レプリケーションはデータを正しく複製しません。アプライ・プログラムは、新しい値を使用してターゲット表の行の更新を試みますが、更新のための新しいキー値がターゲット表で検出されません。そこでアプライ・プログラムはこの更新を INSERT に変換して、新しいキー値をターゲット表に挿入します。この場合、古いキー値をもつ古い行はターゲット表内に残ります。これは不要な行です。ターゲット・キー列が変更前イメージ値を使用して処理されるような変更を指定すると、アプライ・プログラムは古いキー値が指定された行を検出でき、その行を新しい値で更新します。たとえば、*target_key_chg* 変数が **N** に設定されている場合、更新操作の SQL ステートメントは次のようになります。

```
UPDATE targettable SET <non-key columns>= after-image values
WHERE <key columns> = after-image values
```

target_key_chg 変数が **Y** に設定されている場合は、更新操作の SQL ステートメントは次のようになります。

```
UPDATE targettable SET <all columns> = after-image values
WHERE <key columns> = before-image values
```

関連概念:

- 265 ページの『第 15 章 SQL レプリケーションの場合のレプリケーション・センターの使用法』

関連タスク:

- 43 ページの『第 3 章 表およびビューを SQL レプリケーション・ソースとして登録する』
- 115 ページの『第 6 章 SQL レプリケーション環境におけるデータのサブセット化』
- 599 ページの『付録 A. SQL レプリケーション用の Unicode および ASCII のコード化スキーム (z/OS)』

関連資料:

- 411 ページの『ADDDPRSUBM: DPR サブスクリプション・セットのメンバーの追加 (OS/400)』
- 395 ページの『ADDDPRSUB: DPR サブスクリプション・セットの追加 (OS/400)』
- 594 ページの『整合変更データ (CCD) 表』

第 5 章 SQL レプリケーションでの特殊なデータ・タイプのレプリケーション

LOB、DATALINK、ROWID、または DB2 以外のデータ・タイプなど、特殊なデータ・タイプを複製するときには、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。場合によっては、DB2 レプリケーションでこれらのデータ・タイプを処理できるように追加のセットアップ・ステップを実行する必要があります。この章では、これらの条件および制約事項について説明しており、以下のセクションがあります。

- 『レプリケーションにおける一般的なデータの制約事項』
- 106 ページの『ラージ・オブジェクトのレプリケーション』
- 107 ページの『DATALINK 値のレプリケーション』

レプリケーションにおける一般的なデータの制約事項

DB2 レプリケーションでは現在のところ、特定のデータ・タイプに関して制約事項があります。

DB2 レプリケーションでは、いかなる状況でも以下のデータ・タイプを複製することはできません。

- DB2 以外のリレーショナル・ソースからの LOB 列
- 以下のいずれかのプロシージャが定義されている場所の列
 - EDITPROC
 - FIELDPROC
 - VALIDPROC

DB2 レプリケーションでは、特定の状況下で以下のデータ・タイプを複製することができません。

- ソース表およびターゲット表が DB2 for z/OS にある場合は、長い可変 GRAPHIC (LONG VARGRAPHIC) データ。
- 長い可変文字 (LONG VARCHAR) データでは、ソース・データベース表が DB2 for z/OS にあるか、またはソース表とターゲット表の両方が DB2 Universal Database (Windows、Linux、および UNIX 版) にあることが必要です。LONG VARCHAR データを使用可能にする方法については、「*DB2 Universal Database SQL* リファレンス」の「Alter Table」のセクションを参照してください。

DB2 レプリケーションでは、要約データ・タイプを含む表を複製することはできません。

DB2 では、空間データ・タイプ列を含む表を複製することはできますが、実際の空間データ・タイプ列を複製することはできません。

ユーザー定義のデータ・タイプ (DB2 Universal Database の特殊データ・タイプ) は、複製される前に変更データ (CD) 表で基本データ・タイプに変換されます。さ

らに、DB2 レプリケーションがターゲット表をサブスクリプション・セット・メンバー定義として作成した場合、ユーザー定義タイプは CD 表内と同様に、ターゲット表内で基本データ・タイプに変換されます。

ラージ・オブジェクトのレプリケーション

DB2 Universal Database はラージ・オブジェクト (LOB) のデータ・タイプをサポートします。このデータ・タイプには、バイナリー LOB (BLOB)、文字 LOB (CLOB)、2 バイト文字 LOB (DBCLOB) が含まれます。ここでは、ここに示したタイプの LOB データをすべて取り上げます。

キャプチャー・プログラムはログ・レコード内の LOB 記述子を読み、変更されているために複製する必要があるデータが LOB 列にあるかどうかを判別しますが、変更データ (CD) 表には LOB データをコピーしません。LOB 列が変更されると、キャプチャー・プログラムは CD 表に標識を設定します。アプライ・プログラムはこの標識を読み取ると、次に LOB 列全体 (LOB 列の変更部分だけではない) をソース表からターゲット表に直接コピーします。

LOB 列には最大 2 GB のデータを入れることができるので、アプライ・プログラムには十分なネットワーク帯域幅を指定できるようにしてください。同じように、ターゲット表にも LOB データが入るだけのディスク・スペースを指定する必要があります。

制約事項:

- アプライ・プログラムは、常に最新バージョンの LOB 列をソース表 (CD 表ではない) から直接コピーします。これはその列が CD 表の他の列よりも新しい場合にも当てはまります。このため、ターゲット行の LOB 列が変更された場合この LOB 列は、そのターゲット行の残りのデータと矛盾するものになる可能性があります。ターゲット行でデータの矛盾が発生する可能性を最小化するために、アプライ・サイクルのインターバルは、アプリケーションで実用上の問題が起きない範囲で、できるだけ短くしてください。
- 1 つの表でレプリケーションできる LOB 列は 10 以下です。10 よりも多い LOB 列を含む表を登録すると、アプライ・プログラムからエラー・メッセージが戻されます。1 つの表に 10 よりも多くの LOB 列を登録しようとする、レプリケーション・センターからエラー・メッセージが戻されます。
- 競合検出が使用不可の場合、LOB データをレプリカ表にコピーできます。
- DB2 for OS/390 バージョン 6 (またはそれ以降) と DB2 Universal Database (その他のオペレーティング・システム用) の間で LOB データをコピーするには、DB2 Connect 7 以降が必要です。
- ニックネームを使って LOB データを参照することはできません。
- LOB、DATALINK、または ROWID 列の変更前イメージ値はサポートされていません。
- 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、テキスト、音声、ビデオ、イメージ、その他のエクステンダーの場合、DB2 エクステンダー™におけるレプリケーションはサポートしていません。

- DB2 は、LOB 全体の複製しかできません。LOB の部分的なレプリケーションはできません。
- OS/400 上のレプリケーション環境でリモート・ジャーナル・セットアップを使用する場合は、LOB 列はレプリケーションできません。

DATALINK 値のレプリケーション

リモート・ネットワークを介して大きなファイル (マルチメディア・データなど) にアクセスするのは、非効率的で費用がかかります。外部ファイル・システムに保管されるデータを表す DATALINK データ・タイプを使用すると、構造化されていないファイルに、より迅速にアクセスし、ファイルを複製できます。

DB2 Universal Database は、アクセス・コントロール、参照保全、および構造化されていない大きなファイルのリカバリーをデータベースで管理できるようにする、DATALINK データ・タイプをサポートしています。DB2 Universal Database は、以下のオペレーティング・システムで DATALINK の値をサポートします。

- AIX
- Solaris™ オペレーティング環境
- Windows
- OS/400

DATALINK 列の値には、外部ファイルのロケーションを示す URL が入っています。DB2 レプリケーションは、DATALINK 列の値、およびここから参照されるファイルを複製するときに、以下のコンポーネントを使用します。

ASNDLCOPY 出口ルーチン

ソース・ファイル・システム上の URL をターゲット・ファイル・システム上の URL にマップした後、適切なファイル・コピー・デーモンに接続して、URL がポイントする外部ファイルを複製します。

Data Links Manager レプリケーション・デーモン (DLFM_ASNCOPYD)

ASNDLCOPY 出口ルーチンと共同で、DATALINK 列の値から参照されるファイルをコピーします。DLFM_ASNCOPYD デーモンは DB2 Data Links Manager バージョン 8 の一部です。このデーモンは、AIX、Solaris™ オペレーティング環境、および Windows オペレーティング・システムで使用できます。

ASNDLCOPYD デーモン

ASNDLCOPY 出口ルーチンと共同で、DATALINK 列の値から参照されるファイルをコピーします。このデーモンは DB2 for iSeries で提供されます。ASNDLCOPYD は OS/400 で、またオプションとして、他のオペレーティング・システムで使用してください。

アプライ・プログラムはデータ・タイプが DATALINK のデータを読み取ると、参照データを予備ファイルに入れるとともに、更新されたファイルの URL を入力ファイルに入れます。

アプライ・プログラムは続いて ASNDLCOPY 出口ルーチン呼び出します。この ASNDLCOPY 出口ルーチンは、ソース・ファイル・システム上に物理ファイルが存在することを確認し、ターゲット・ファイル・システム上の対応するファイルに

URL をマップし、このターゲット・ファイルのロケーションを結果ファイルに保管し、適切なファイル・コピー・デーモン (DLFM_ASNCOPYD、ASNDLCOPYD、または FTP) に接続し、ソース・ファイル・システムからターゲット・ファイル・システムへ外部ファイルをコピーします。

推奨: アプライ・プログラムは ASNDLCOPY ルーチンの処理が完了するのを待ってからサブスクリプション・セットのレプリケーションを完了させるため、DATALINK 列に対しては、別個のサブスクリプション・セットを使用してください。外部ファイルのコピー時にエラーが生じると、サブスクリプション・セット全体のレプリケーションが失敗します。サブスクリプション・セットが失敗した場合、アプライ・プログラムはサブスクリプション・セットを非活性化せずに、次のアプライ・サイクルでサブスクリプション・セットを再度処理します。

Linux、UNIX、および Windows の場合: `loadxit` パラメーターを `y` に設定して ASNLOAD 出口ルーチン呼び出して、アプライ・プログラムを始動してください。ASNLOAD 出口ルーチンはフル・リフレッシュ時に外部ファイル (DATALINK の値が指し示す) をコピーします。詳細については、164 ページの『ASNLOAD 出口ルーチンを使ったターゲット表のリフレッシュ』を参照してください。

OS/400 の場合: アプライ・プログラムがフル・リフレッシュ時に外部ファイルをコピーできるように、ASNDLCP 出口ルーチン呼び出すように ASNLOAD 出口ルーチンを変更してください。詳細については、164 ページの『ASNLOAD 出口ルーチンを使ったターゲット表のリフレッシュ』を参照してください。

重要: 外部ファイルは非常に大きい場合があるため、アプライ・プログラムとファイル転送メカニズムの両方について、これらのファイルをコピーするために使用する十分なネットワーク帯域幅を確保してください。同じように、ターゲット・システムにもこれらのファイルが入るだけのディスク・スペースを確保する必要があります。

制約事項:

- OS/400 上の DB2 データベースと他のオペレーティング・システム上の DB2 データベースの間で DATALINK 列を複製することはできません。
- OS/400 オペレーティング・システムの場合、DATALINK の値の "comment" 属性に対するサポートはありません。
- DATALINK 列で Update-anywhere レプリケーションを使用する場合は、競合検出レベルに **None** を指定して、DATALINK 列、および同じサブスクリプション・セット内のその他の列の両方について、競合検出をオフにする必要があります。DB2 レプリケーションは、DATALINK 列が参照する外部ファイルの更新競合はチェックしません。
- DATALINK 列における変更前イメージ値はサポートしていません。
- ターゲット表が基礎集約表または変更集約表である場合は、DATALINK 列がサポートできません。
- 整合変更データ (CCD) 表の中のデータを複製するときには、以下の制約事項が適用されます。
 - 内部 CCD 表は、DATALINK 標識 (関連付けられた URL に関する情報を含む VARCHAR タイプの文字ストリング) を含むことはできますが、

DATALINK の値を含むことはできません。アプライ・プログラムは、これらの表タイプでデータを複製するときには、ASNDLCOPY 出口ルーチン呼び出しません。

- コンデンスされた外部 CCD 表は DATALINK 列を含むことができます。
- 非コンデンス CCD ターゲット表は DATALINK 列を含むことができません。

以下のセクションでは、DATALINK の値および URL がポイントする外部ファイルの両方をターゲット・システムに複製するためにアプライ・プログラムが使用する、ユーザー出口ルーチンおよびファイル・コピー・デーモン (オペレーティング・システムによって異なる) について説明します。

- 『ASNDLCOPY 出口ルーチンの設定と使用方法』
- 111 ページの『DLFM_ASNCOPYD の設定と使用方法 (Linux、UNIX、Windows)』
- 112 ページの『ASNDLCOPYD の設定と使用方法 (OS/400)』

ASNDLCOPY 出口ルーチンの設定と使用方法

サブスクリプション・セットのレプリケーションの準備ができると、アプライ・プログラムは変更データ (CD) 表の中の対象の行を識別します。DATALINK 列の値が検出された場合、アプライ・プログラムは更新されたファイルの URL を入力ファイルに入れます。アプライ・プログラムは続いて ASNDLCOPY 出口ルーチン呼び出しします。この出口ルーチンはこの入力ファイルを読み取り、DATALINK ソース・ファイルの各ロケーションを対応するターゲット・ファイル・ロケーションにマップします。次に、ASNDLCOPY 出口ルーチンはファイル・コピー・デーモンに接続し、URL がポイントする外部ファイルを、ソース・ファイル・システムから新しくマップされたターゲット・ファイル・システム・ロケーションにマップします。

ASNDLCOPY ルーチンが完了した時点で、このルーチンはアプライ・プログラムに戻りコードを渡します。ゼロ以外の戻りコードは、1 つまたは複数のファイルでレプリケーションが失敗したことをアプライ・プログラムに知らせます。このような場合、アプライ・プログラムはメッセージを出し、そのサブスクリプションをスキップして次のサブスクリプションを処理します。ゼロの戻りコードは、レプリケーションが正常に行われたことをアプライ・プログラムに知らせます。

ASNDLCOPY 出口ルーチンのソース・コードを使用して、システムの要件に合わせてサンプル・プログラム (ASNDLCOPY.smp と呼ばれ、¥sqllib¥samples¥repl ディレクトリーに置かれている) を変更できます。サンプル・プログラムには、以下の構成ファイルが含まれます。

ASNDLSRVMAP

ソース URL をターゲット URL にマップします。

例: `http://source.com/file` to `http://target.com/file`

ASNDLUSER

ソースおよびターゲットのファイル・システムに接続するときを使用される、ログオンおよびアドレス・ロケーション情報が入っています。

ASNDLPARM

ASNDLCOPY 出口ルーチンの関数をコントロールするために使用される稼働パラメーターが入っています。これらのパラメーターには、ソース・ファイルを異なるターゲット・ファイル・ロケーションに複製するために使用される **REPLACE_FILE** パラメーター、そして複製するファイルの最後の変更時刻を保存するために使用される **PRESERVE_MODTIME** パラメーターが含まれます。ASNDLPARM は、Linux、UNIX、および Windows オペレーティング・システムでのみ使用される、オプションの構成ファイルです。

外部ファイルを複製するために、ユーザー自身の出口ルーチンを構成することができますが、プログラムの名前は ASNDLCOPY にする必要があります。構成ファイルは、アプライ・プログラムの現行実行パスに入れてください。

構成ファイルのセットアップ方法、およびこの出口ルーチンの変更方法については、`¥sqllib¥samples¥repl` ディレクトリーにあるサンプル・プログラムの PROLOG セクションを参照してください。

手順:

ASNDLCOPY 出口ルーチンを使用するには:

1. サイトの要件に合わせて、必要であれば ASNDLCOPY ルーチンを変更します。

アプライ・プログラムのトレース・オプションをオンにすると、ASNDLCOPY ルーチンはログ・ファイルとトレース・ファイルという 2 つのファイルを作成します。このログ・ファイルの名前は以下のとおりです。

```
ASNDLApplyQualSetNameSrcSrvrTgtSrvr.LOG
```

ApplyQual はアプライ修飾子、*SetName* はサブスクリプション・セットの名前、*SrcSrvr* はソース・サーバーの名前、そして *TgtSrvr* はターゲット・サーバーの名前です。ログ・ファイルには、ASNDLCOPY ルーチンが生成するすべてのメッセージが入れられます。トレース・ファイルの名前は以下のとおりです。

```
ASNDLApplyQualSetNameSrcSrvrTgtSrvr.TRC
```

トレース・ファイルには、ASNDLCOPY ルーチンが生成するトレース情報がすべて入れられます。

2. 必要に応じて ASNDLUSER、ASNDLSRVMAP、および ASNDLPARM 構成ファイルを構成します。

Linux、UNIX、および Windows の場合: ASNDLPARM ファイルの中で **REPLACE_FILE** パラメーターを YES (デフォルト) に設定した場合は、ターゲット・ディレクトリーにターゲット・ファイルがすでに存在するときには、ASNDLCOPY 出口ルーチンはソース・ファイルの内容を別のターゲット・システム・ファイルに複製します。ASNDLCOPY 出口ルーチンは、ソース・ファイルの内容を一時ファイルに直接コピーします。この一時ファイルの名前は、ソース・ファイルの名前と同じですが、*new* という接尾部が付けられます。(この接尾部は ASNDLPARM ファイルの中で変更できます。)アプライ・プログラムは次に、結果ファイルからオリジナルのターゲット・ファイルの URL と一時ファイルの URL を受け取ります。アプライ・プログラムが変更をターゲット表に伝搬すると、DB2 は、レプリケーション・トランザクションをコミットするときに、一時ファイルの名前を、オリジナルのターゲットの URL の中のファイル名に変更します。

- ASNDLCOPY 出口ルーチンを変更した場合には、プログラムをコンパイルし、実行可能プログラムを適切なディレクトリーに入れてください。

アプライ・プログラムは ASNDONE 出口ルーチンを、正常に処理されたかエラーが発生したかに関係なく、サブスクリプションの処理が完了した後に呼び出します。そのため、ASNDLCOPY ルーチンが外部ファイルのレプリケーションに失敗した場合は、ASNDONE 出口ルーチンを使って必要なクリーンアップを任意に実行できます。

DLFM_ASNCOPYD の設定と使用方法 (Linux、UNIX、Windows)

DB2 Data Links Manager バージョン 8 をインストールしてある場合は、Data Links Manager レプリケーション・デーモン (DLFM_ASNCOPYD) を使用して、DATALINK データ・タイプから参照されるファイルをコピーできます。

ASNDLCOPY 出口ルーチンはソースおよびターゲットの URL をマップした後、ファイルをコピーするためのデーモンに接続します。使用するファイル・コピー・デーモンに接続するようにアドレスとポート番号を指定して、ASNDLUSER 構成ファイルを構成できます。任意の FTP デーモン、または DLFM_ASNCOPYD ファイル・コピー・デーモンを使用できます。

FTP デーモンも DLFM_ASNCOPYD デーモンもソース・ファイル・システムからターゲット・ファイル・システムへ外部ファイルをコピーします。しかし、DLFM_ASNCOPYD ファイル・コピー・デーモンには追加の関数があります。

- RECOVERY YES と定義された DATALINK 列から参照される特定のバージョンのファイルを検索する機能。
- ユーザーのアクセス権に基づいて、READ PERMISSION DB と定義された DATALINK 列から参照されるファイルを検索する機能。
- 複製されたファイルの最終変更時刻を保存する機能。

DLFM_ASNCOPYD の制約事項:

複製したファイルを DLFM_ASNCOPYD を使用してコピーするには、DB2 Universal Database バージョン 8 で、DB2 Data Links Manager バージョン 8 を使用する必要があります。

DLFM_ASNCOPYD ファイル・コピー・デーモンは、以下のオペレーティング・システムでしか使用できません。AIX、Solaris™ オペレーティング環境、および Windows。

FTP の場合の Solaris™ オペレーティング環境の制約事項:

Solaris™ オペレーティング環境で DATALINK 列の値を複製する場合、FTP デーモンを使用してファイルをコピーするには、MDTM (modtime) コマンドをサポートする FTP デーモンを使用する必要があります。ソースおよびターゲットのファイル・システムで実行される FTP デーモンは、特定のファイルの最終変更時刻を表示する、MDTM をサポートする必要があります。Solaris™ オペレーティング環境のバージョン 2.6、または FTP による MDTM サポートを含まないその他のバージョンを使用する場合は、WU-FTPD などの追加ソフトウェアが必要です。

手順:

DLFM_ASNCOPYD ファイル・コピー・デーモンを設定するには:

1. このファイル・コピー・デーモンへの接続を必要とするユーザーを識別します。
2. これらのファイルが置かれているディレクトリーに基づいて、ファイルへのアクセス権をユーザーに付与します。
3. DLFM_ASNCOPYD デーモンが使用可能であり、正しいポート番号が指定されていることを確認します。

このポート番号は、ASNDLUSER 構成ファイルで指定されているポート番号と一致する必要があります。

詳細については、「DB2 Data Links Manager 概説およびインストール」および「DB2 Data Links Manager 管理ガイドおよびリファレンス」を参照してください。

データ・リンク・ファイル・マネージャーは、標準の SQL 操作によってアプリケーションがファイルにリンクするたびに、RECOVERY YES として定義された DATALINK 列で、ソース・ファイルの新しいバージョンをアーカイブします。キャプチャー・プログラムは、RECOVERY YES として定義された DATALINK 列で行の変更をキャプチャーすると、ファイルのバージョン番号を記録し、このバージョン番号を CD 表に入れます。アプライ・プログラムは、CD 表でデータの変更とバージョン番号を読み取り、新しい DATALINK 列の値の URL とバージョン番号を ASNDLCOPY 出口ルーチンに渡します。ASNDLCOPY 出口ルーチンが DLFM_ASNCOPYD デーモンに接続すると、このファイル・コピー・デーモンは外部ファイルの一貫性のあるバージョンをリトリートします。

ソース・システム上にファイルのより新しいバージョンが存在している場合も、データ・リンク・ファイル・マネージャーは、CD 表にキャプチャーされているバージョンと整合するバージョンのファイルを提供します。このため、ターゲット・サーバーは、キャプチャー・プログラムがまだログ内に取り込んでいないバージョンを受け取ることはできません。

ASNDLCOPYD の設定と使用方法 (OS/400)

ASNDLCOPYD は、ASNDLCOPY 出口ルーチンがソースとターゲットの URL をマップした後で、許可ユーザーが OS/400 ソース・サーバーから OS/400 ターゲット・サーバーにファイルをリトリートできるようにするデーモンです。

ASNDLCOPY 出口ルーチンはソースおよびターゲットの URL をマップした後、ファイルをリトリートするために ASNDLCOPYD デーモンに接続します。

ASNDLCOPYD ファイル・コピー・デーモンは FTP デーモンと似ていますが、DATALINK の値を複製するために以下の関数を提供します。

- ファイル情報 (ファイル・サイズや最終変更時刻など) を抽出するためのコマンド
- 特定ファイルの中身を取り出すコマンド

READ PERMISSION DB として定義されている DATALINK 列を複製するために、ASNDLCOPYD ファイル・コピー・デーモンに接続するように ASNDLCOPY 出口ルーチンを構成できます。

ASNDLCOPYD のサンプル・ファイルはライブラリー QDP4、ソース・ファイル QCSRC、メンバー ASNDLCPD に入っています。このサンプル・ファイルは 3 つのプログラムを構築します。

ASNDLCOPYD

メインの親プログラムとファイル・コピー・デーモン。

ASNCHILD

クライアントから ASNDLCOPYD デーモンへの接続を調整するプログラム。ASNCHILD は ASNDLCOPYD デーモンの一部です。クライアントからの要求のたびに、新しい ASNCHILD プロセスが作成されます。

ASNDLCFG

ユーザー ID の追加と除去、そしてユーザー ID のパスワードの変更のための構成プログラム。

注: OS/400 またはその他のオペレーティング・システムで、現在 DB2 バージョン 7 上で ASNDLCOPYD ファイル・コピー・デーモンを使用している場合は、DB2 バージョン 8 で引き続きこのデーモンを使用できます。

前提条件:

ASNDLCOPYD デーモンを実行するにはルート (管理者) 権限が必要です。

手順:

ASNDLCOPYD ファイル・コピー・デーモンを使用するには:

1. ライブラリー QDP4、ソース・ファイル QCSRC、メンバー ASNDLCPD の ASNDLCOPYD サンプル・プログラムにアクセスします。
2. サイトの要件に合わせてサンプル・プログラムを変更します。
3. プログラム・デーモンを構築します。

- a. 基本モジュールを構築します。

```
CRTCMOD MODULE(libraryname/ASNDLCPD) SRCFILE(QDP4/QCSRC)
                DBGVIEW(*SOURCE) SYSIFCOPT(*ALL)
```

- b. 子プログラム (ASNCHILD) を構築します。

```
CRTPGM PGM(libraryname/ASNCHILD) MODULE(libraryname/ASNDLCPD)
```

- c. 親プログラム (ASNDLCOPYD) を構築します。

```
CRTPGM PGM(libraryname/ASNDLCOPYD) MODULE(libraryname/ASNDLCPD)
```

- d. 構成プログラム (ASNDLCFG) を構築します。

```
CRTPGM PGM(libraryname/ASNDLCFG) MODULE(libraryname/ASNDLCPD)
```

この *libraryname* は既存のライブラリー名です。詳細については、サンプル・プログラムの PROLOG セクションを参照してください。

4. 実行可能プログラムを QDP4 ライブラリーに入れます。
5. サイトの要件に合わせて構成ファイルを変更します。
6. 管理者権限およびスーパーユーザー・アクセスを使用して ASNDLCOPYD デーモンを始動します。ポート番号、および構成ファイルを含むディレクトリーの両方を指定してください。

ASNDLCOPYD ファイル・コピー・デーモンは、ASNDLCOPYD プログラムが生成するすべてのメッセージのためのログ・ファイルを作成します。このログ・ファイルの名前は ASNDLCOPYDYMMDDHHMMSS.LOG (YYYYMMDDHHMMSS は、デーモンの実行開始時刻) になります。

OS/400 の場合、DB2 レプリケーションは必ず、DATALINK 列の値から参照される、最新のバージョンの外部ファイルを複製します。

関連タスク:

- 147 ページの『第 10 章 SQL レプリケーションに関するアプライ・プログラムの操作』

第 6 章 SQL レプリケーション環境におけるデータのサブセット化

通常、レプリケーションにはなんらかのサブセット化が関与します。レプリケーション・ソースの登録時には、ソース表から複製する列と行を選択します。サブスクリプション・セットの作成時には、各ターゲット表に複製する登録済み列を選択します。

基本的なサブセット化の方式は、43 ページの『第 3 章 表およびビューを SQL レプリケーション・ソースとして登録する』および 69 ページの『第 4 章 SQL レプリケーションのソースのサブスクリプト』で説明されています。この章では、データをサブセット化するのに使用できるいくつかの高度な技法を説明します。レプリケーション要件に応じて、ソースの場合は登録時に、ターゲットの場合はサブスクリプションの際に、以下の技法を使用してデータをサブセット化できます。

- ソースに対するターゲットが 1 つだけの場合、またはすべてのターゲットがまったく同じデータを必要とする場合は、ニーズがターゲットごとに異なる可能性について考慮する必要がないため、登録時のサブセット化または操作が可能である。
- 1 つのソースに対して複数のターゲットが存在し、アプライされるデータに関する要件がその複数のターゲット間でそれぞれ異なるときは、登録時のサブセット化は不可能な場合がある。このケースでは、データのサブセット化はサブスクリプションの際に行うこととなります。

レプリカ・ターゲット表に対するレプリケーションの場合は、これらの技法はどれも使用しないでください。Update-anywhere 構成では、マスター表と各レプリカ表は互いにデータを複製し合います。ソース表の使用されていない列が NULL 可能であるかぎり、レプリカ表はソース表の列のサブセットを持てます。そうでない場合は、レプリカ表はソース表と同じ列を含んでいなければならず、そのために列のサブセット化、新しい列の追加、または列の名前変更が不可能 となります。

この章には、以下のセクションがあります。

- 『登録時におけるデータのサブセット化』
- 117 ページの『サブスクリプションの際のデータのサブセット化』

登録時におけるデータのサブセット化

高度な技法を使用して、登録時にデータをサブセット化できます。これらの技法は、データの同じサブセットを 1 度キャプチャーし、そのサブセットを多数のターゲット表に複製する場合に特に便利です。データのサブセット化を、登録されているソースからそのデータがキャプチャーされる前と後のどちらに行うのかを選択できます。このセクションで示されている技法は、Update-anywhere レプリケーションまたは対等レプリケーション以外のすべてのレプリケーション構成で使用できます。

登録時にデータをサブセット化すると、レプリケーションのパフォーマンスが全体的に向上する可能性があります。なぜなら、キャプチャー・プログラムが CD 表に追加するデータの量が削減されて、アプライ・プログラムが読み取るデータ量が削減されるからです。また、CD 表内の行数が少なくなるため、ストレージも削減されます。

このセクションでは、登録時にデータをサブセット化できる以下の方法について説明します。

- 『ビューを使用したソース・データのサブセット化』
- 『特定の行のキャプチャーを防止するために CD 表にトリガーを定義する (Linux、UNIX、Windows、z/OS)』

ビューを使用したソース・データのサブセット化

ソースの登録時には、レプリケーションに使用できるようにする列を選択します。選択した列はレプリケーションのためにキャプチャーされます。いくつかのケースでは、変更レプリケーション用のソースを登録した後で、そのソースのビューを登録する必要があることがあります。

たとえば、人材部門では給与情報が組み込まれた人事データを含む表が保守されているとします。バックアップ・データベースを保守するために、この人事表全体がバックアップ・サイトに登録され、このサイトからサブスクライブされています。しかし、別のターゲット・サイトがこの人事表にサブスクライブする必要がある場合は、給与情報はこの 2 番目のサブスクライバーから隠蔽しなければならないことがあります。解決策は、この人事表に対するビューを登録し、2 番目のサブスクライバー用に登録されたビューだけにアクセス権を許可して、給与情報がアクセスから保護されるようにすることです。サブスクリプションは、この登録されたビューに対して作成できます。

また、複数のソース表が組み込まれたビューも登録できます。たとえば、カスタマー表と支店表がある場合、カスタマーを適切にサブセット化してターゲットに正しく複製する唯一の方法は、この 2 つの表を結合して、特定の支店のカスタマーだけが特定のターゲットに複製されるようにすることです。このケースでは、二重削除を防止するための配慮が必要です。

特定の行のキャプチャーを防止するために CD 表にトリガーを定義する (Linux、UNIX、Windows、z/OS)

ソースの登録時にレプリケーション・センターを使用すると、キャプチャーする列を選択できますが、それらの行における特定の変更がレプリケーションされるのを防止する機能は提供されません。レプリケーションのシナリオによっては、行内の特定の変更がキャプチャーされてターゲット表にレプリケーションされるのを防止したい場合があります。たとえば、ターゲット表にはすべての行が含まれているようにし、それらの表からはどの行も削除されないようにしたい場合は、ソースから削除が複製されないようにする必要があります。

特定の変更がキャプチャーされるのを抑制するには、CD 表にトリガーを定義してください。これらのトリガーにより、キャプチャー・プログラムが無視する必要のある変更と、CD 表に行を追加してはならないことが指定されます。これらのトリガーは、レプリケーション・センターでは作成できませんが、既存の CD 表に対し

て (つまり、ソースを登録した後で) 手動で作成できます。値が 99999 の SQLSTATE が表示されるトリガーの失敗はすべてキャプチャー・プログラムによって無視され、その行は CD 表に挿入されません。

例: 表 SAMPLE.TABLE (CD 表は SAMPLE.CD_TABLE) からのレプリケーション中はソース表でのすべての DELETE 操作を抑制したいとします。以下のトリガーは、DELETE 操作であるすべての行が CD 表に挿入されるのを抑制します。

```
CREATE TRIGGER SAMPLE.CD_TABLE_TRIGGER
NO CASCADE BEFORE INSERT ON SAMPLE.CD_TABLE
REFERENCING NEW AS CD
FOR EACH ROW MODE DB2SQL
WHEN (CD.IBMSNAP_OPERATION = 'D')
SIGNAL SQLSTATE '99999' ('CD INSERT FILTER')
```

登録時に生成された SQL にこの CREATE TRIGGER ステートメントを追加する場合は、登録を完了して CD 表にトリガーを作成するために、その変更した SQL を実行する必要があります。

これらのトリガーは、キャプチャー・プログラムが CD 表に対する行の挿入を試行するたびに実行されます。そのため、ここでのトリガーの使用がレプリケーション環境での最高のパフォーマンスをもたらすものなのかどうかについて検討する必要があります。CD 表にトリガーを追加すると、データ・スループットが上がる場合と下がる場合とがあります。CD 表のトリガーは、ソースで著しく多発する変更を抑制する場合に使用するようになっています。変更の大部分をキャプチャーする予定だが、そのうちの一部の変更については複製されるのを抑制したいという場合は、不要な行をサブスクリプションの際に抑制します。

サブスクリプションの際のデータのサブセット化

このセクションでは、サブスクリプションの際に述部を使用して行をサブセット化する方法を説明します。サブスクリプションの際にデータをサブセット化すると、レプリケーションのパフォーマンスが全体的に向上する可能性があります。なぜなら、アプライ・プログラムが取り出すデータの量が削減されるからです。また、ターゲット表内の行数が少なくなるため、ストレージも削減されます。

アプライ・プログラムは、フル・リフレッシュ・レプリケーション時および変更キャプチャー・レプリケーション時にどのデータをコピーするのかを述部を使用して判断します。レプリケーション・センターでは、フル・リフレッシュ・レプリケーション用および変更キャプチャー・レプリケーション用の述部値を指定できます。変更キャプチャー・レプリケーションの場合にのみ使用する補足的な述部情報を追加したいときがあります。なぜなら、この情報はフル・リフレッシュ時には使用不可であるからです。この補足的な述部情報は、ユーザーが提供する SQL を使用して、UOW_CD_PREDICATES 列のサブスクリプション・セット・メンバー (IBMSNAP_SUBS_MEMBR) 表に追加する必要があります。

たとえば、ALL.CUSTOMERS という名前の登録済みの表があり、それに関連した CD 表の名前が ALL.CD_CUSTOMERS であるとします。サブスクリプション・ターゲットには ALL.CUSTOMERS のサブセットだけを含めたいとします (ALL.CUSTOMERS の ACCT_BALANCE 列は 50000 を超えます)。さらに、ターゲット表の履歴データを保守する必要がある (つまり、ターゲット表内のデータはど

れも削除されたくない) とします。レプリケーション・センターを使用すれば、PREDICATES 値が 'ACCT_BALANCE > 50000' のサブスクリプション・セット・メンバーを作成できます。

レプリケーション・センターを使用する場合は、ターゲット表での削除は防止できません。なぜなら、操作のタイプに関する情報は CD 表に保管されるため、ソース表またはビューではこの情報が使用不可だからです。したがって、以下の情報が組み込まれた SQL ステートメントを使用して、追加の変更キャプチャー述部を生成する必要があります。³

```
UPDATE ASN.IBMSNAP_SUBS_MEMBR SET UOW_CD PREDICATES = 'IBMSNAP_OPERATION <>"D"'
WHERE APPLY_QUAL = 'apply_qual' AND SET_NAME = 'set_name' AND
SOURCE_OWNER = 'ALL' AND SOURCE_TABLE = 'CUSTOMERS'
```

CD 表内の変更前イメージ列、CD 表からのあらゆるオーバーヘッド列、または UOW 表からのあらゆる列などといった、フル・リフレッシュ時に使用不可である列を参照するサブスクリプション・セット・メンバー述部の場合は、必ず UOW_CD_PREDICATES 列を手動でセットアップする必要があります。

デフォルトでは、ターゲット表がユーザー・コピーである場合でも、アプライ・プログラムは UOW 表と CD 表を結合しません。このプログラムは CD 表から直接データをフェッチして適用します。述部が UOW 表を参照する必要があり、かつターゲット表がユーザー・コピーである場合は、サブスクリプション・メンバー (IBMSNAP_SUBS_MEMBR) 表の JOIN_UOW_CD 列の値を Y に設定する必要があります。このフラグを設定すれば、UOW 表と CD 表がアプライ・プログラムによって確実に結合されます。

行のサブセットに対して 1024 バイト (サブスクリプション・メンバー (IBMSNAP_SUBS_MEMBR) 表の PREDICATES 列の容量) を超える述部を指定する場合は、ソース・ビューを使用する必要があります。

サブスクリプション・セットに複合述部ステートメントを使用している場合は、式全体を括弧で囲みます。たとえば、述部ステートメントに AND および OR 文節を使用している場合には、式を次のように囲みます。

```
((TOSOURCE = 101 AND STATUS IN (202,108,109,180,21,29,32,42))
OR (SOURCE = 101))
```

3. シナリオによっては、この更新ステートメントに列を追加して、サブスクリプション・メンバー (IBMSNAP_SUBS_MEMBR) 表の単一行が確実に更新されるようにしなければならない場合があります。

第 7 章 SQL レプリケーション環境におけるデータ操作

ターゲット表のデータは、ソース表における表示ほど正確に表示される必要はありません。ソース・データは、ターゲット表に複製される前にトランスフォームまたは拡張することができます。たとえば、データ整理の実行、データ集約の実行、あるいはソースにないターゲット表の列の移植のようなデータ操作を行う場合があります。

この章では、データをトランスフォームするために使用できるいくつかの先進的な技法について説明します。

データ操作は、登録済みソースからのキャプチャー前、キャプチャー後のいずれかで行うことができます。データを 1 回操作して、トランスフォームされたデータを多数のターゲット表に複製する場合は、サブスクリプションではなく登録で操作します。すべてのソース・データをキャプチャーし、トランスフォームされたデータを選択して個々のターゲットに適用する場合は、登録ではなくサブスクリプション中にデータを操作します。

レプリケーションのシナリオによっては、CD 表に保管されたソース・データの内容を操作することになります。トリガー、サブスクリプション中の式、あるいはソース・ビューのどれを使用しても、同じジョブを実行できます。それぞれの方式に、良い点と悪い点があります。トリガーは、使用される CPU の点からコストがかかりすぎる可能性があります。関数をセットアップするのに、サブスクリプションは複数回必要ですが、ビューを使用すると 1 回でセットアップできます。

たとえば、ソース表で特定の値が脱落している場合、キャプチャー・プログラムで NULL 値をキャプチャーする必要のない場合があります。

CD 表に対してトリガーを使用して、データを CD 表に挿入するときにキャプチャー・プログラムがデータを拡張する条件を指定できます。この場合、キャプチャー・プログラムがソースで NULL 値を発見したら、デフォルト値を CD 表に挿入するように指定できます。データがソース表の更新から脱落している場合、以下のコードを使用して、確定したデフォルトを提供するトリガーを作成できます。

```
CREATE TRIGGER ENHANCECD
NO CASCADE BEFORE INSERT ON CD_TABLE
REFERENCING NEW AS CD
FOR EACH ROW MODE DB2SQL
WHEN (CD.COL1 IS NULL)
SET CD.COL1 = 'MISSING DATA'
END
```

トリガーの代わりに、登録済みソース・ビューまたはサブスクリプションの式で DB2 の COALESCE スカラー関数を使用できます。登録済みビューで、この合体関数が最初の非 NULL 値を戻します。

ソース・ビューの使用例の一部:

```
CREATE VIEW SAMPLE.SRCVIEW (columns) AS SELECT
... COALESCE(A.COL1, 'MISSING DATA') ...
FROM SAMPLE.TABLE A
```

式の使用例の一部:

```
COALESCE(CD.COL1, 'MISSING DATA')
```

アプライ・プログラムは、データをターゲットに適用する前または後どちらかに、以下の方法でデータを操作できます。

- 『ストアード・プロシージャまたは SQL ステートメントを使用したデータ拡張』
- 121 ページの『名前が異なるソース列とターゲット列のマッピング』
- 121 ページの『算出列の作成』

ストアード・プロシージャまたは SQL ステートメントを使用したデータ拡張

サブスクリプション・セット情報を定義するとき、特定のセットを処理するたびにアプライ・プログラムに実行させるランタイム処理ステートメントを、SQL ステートメントまたはストアード・プロシージャを使って定義できます。これらのランタイム処理により、レプリケーション中にデータを操作できるようになります。このようなステートメントは、CCD 表の整理やサブスクリプション・セット処理でのシーケンス・コントロールにも役立ちます。ランタイム処理ステートメントは、サブスクリプション・セットの処理前ならキャプチャー・コントロール・サーバーで、あるいはサブスクリプション・セットの処理前または後であればターゲット・サーバーで実行できます。たとえば、データを取り出す前、またはターゲット表にデータを複製した後、あるいはその両方の場合に SQL ステートメントを実行することができます。

ニックネームの制約事項: フェデレーテッド DB2 表 (ニックネームを使用する) は、通常単一の作業単位内で更新されます。アプライ・プログラムがすべてのデータをターゲットに適用した後、SQL ステートメントをサブスクリプション・セットに追加する場合、以下の 2 つの状況のいずれかで、SQL COMMIT ステートメントをその SQL ステートメントより先行させる必要があります。

- この SQL ステートメントは、サブスクリプション・セットのターゲット表またはターゲット・ニックネームがあるサーバー以外のサーバー上のニックネームに対して挿入、更新、または削除を行う。
- この SQL ステートメントは、アプライ・コントロール・サーバーに対してローカルの表に対して、挿入、更新、または削除を行うが、サブスクリプション・セットのターゲット・ニックネームはリモート・サーバーにある。

追加の COMMIT ステートメントは、追加された SQL ステートメントを処理する前に、アプライ・プログラムの作業をコミットします。

ストアード・プロシージャは、パラメーターなしの SQL CALL ステートメントを使用します。プロシージャ名の長さは、18 文字以下でなければなりません (OS/400 の場合は、最大 128 文字です)。ソースまたはターゲット表が非 DB2 リレーショナル・データベースにある場合、SQL ステートメントはフェデレーテッド DB2 データベースを対象として実行されます。SQL ステートメントが非 DB2 データベースを対象として実行されることは決してありません。各タイプのランタイ

ム・プロシージャーは単一トランザクションとして一緒に実行されます。さらに、それぞれのステートメントごとに受け入れ可能 `SQLSTATE` を定義することもできます。

(特定のセットの完了後ではなく) それぞれのセットの完了後にデータを操作する場合は、`ASNDONE` 出口ルーチンを使用します。

名前が異なるソース列とターゲット列のマッピング

レプリケーション・センターを使用してサブスクリプション・セット・メンバーを定義しており、かつ参照中のターゲット表が存在していない場合、ターゲット表のタイプに関係なく、ターゲット表で列を名前変更できます。さらに、互換性があれば、列の属性 (データ・タイプ、長さ、位取り、精度、NULL 可能かどうか) 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、既存のターゲット表の列を名前変更することはできません。サブスクリプション・セット・メンバーによって参照中のターゲット表がすでに存在している場合、レプリケーション・センターは名前による列のマッピングを試行します。ソース列とターゲット列が一致しない場合は、レプリケーション・センターを使用してソースからターゲットに列をマッピングするか、あるいはソースの列名に一致する名前を含むターゲット表のビューを作成するかのどちらかが可能です。

算出列の作成

既存のターゲット表の列名を変更することはできませんが、ソース列の式を変更して、既存のターゲット表の列に正しくマッピングするようにしたり、あるいはそれとの互換性を持たせることができます。SQL 式を使用すれば、既存のソース列から新しい列を派生させることもできます。集約ターゲット表タイプでは、`COUNT` や `SUM` といった集約関数を使って新しい列を定義できます。ターゲット表の他のタイプでは、式でスカラー関数を使用して新しい列を定義できます。ソース表とターゲット表の列の名前だけが違って、他は互換性がある場合、レプリケーション・センターを使用すれば列をもう一方の表の列にマッピングできます。

たとえば、既存のソース表 (`SRC.TABLE`) とターゲット表 (`TGT.TABLE`) があるとします。

```
CREATE TABLE SRC.TABLE (SRC_COL1 CHAR(12) NOT NULL, SRC_COL2 INTEGER,
                        SRC_COL3 DATE, SRC_COL4 TIME, SRC_COL5 VARCHAR(25))
```

```
CREATE TABLE TGT.TABLE (TGT_COL1 CHAR(12) NOT NULL,
                        TGT_COL2 INTEGER NOT NULL, TGT_COL3 TIMESTAMP, TGT_COL4 CHAR(5))
```

以下のステップで、サブスクリプション中に算出列を使用して必要なターゲット表をマッピングします。

1. レプリケーション・センターを使用して、ソース表の `SRC_COL1` をターゲット表の `TGT_COL1` にマッピングする。これらの列には互換性があるので、一方から他方にマッピングするのに式を使う必要はありません。
2. 式 `COALESCE(SRC_COL2, 0)` を使用して列値を計算し、`TGT_COL2` に提供するためにマッピングする。`SRC_COL2` は NULL 可能で `TGT_COL2` は NOT NULL のため、`TGT_COL2` の値が必ず非 NULL になるようにこのステップを実行する必要があります。

3. 式 `TIMESTAMP(CHAR(SRC_COL3) CONCAT CHAR(SRC_COL4))` を使用して列値の計算をし、`TGT_COL3` に提供するためにマップする。この列式はターゲット・データベースのタイム・スタンプ列にマップするデータを提供します。
4. 式 `SUBSTR(SRC_COL5, 1,5)` を使用して列値の計算をし、`TGT_COL4` に提供するためにマップする。

第 8 章 SQL レプリケーション用のレプリケーション SQL スクリプトのカスタマイズおよび実行

コントロール表の作成、ソース表の登録、およびサブスクリプション・セットの作成のためには、レプリケーション・センターで生成される SQL スクリプトを実行しなければなりません。SQL スクリプトは、レプリケーション・センターまたはタスク・センターを使用して、あるいは DB2 コマンド行から実行できます。SQL スクリプトは、必要に応じて、ユーザーの要求を満たすように変更できます。

レプリケーション・センターには、生成された SQL スクリプトをすぐに実行するためのオプション、あるいはそれをタスクとして保管するかまたはファイルに保管し、後でスクリプトを実行するためのオプションがあります。SQL をレプリケーション・センターから実行するように選択する場合でも、今後の参照用に、SQL をタスクとして保管するかまたはファイルに保管することが必要な場合もあります。たとえば、大規模なレプリケーション・サブスクリプション・セットの定義を SQL ファイルに保存しておけば、必要なときにその定義を再実行できます。

生成済みの SQL スクリプトを編集するときは、終了文字を変更しないように注意してください。また、ファイルに保管されるスクリプトが複数ある場合は、スクリプト区切り文字を変更しないでください。

以下のタスクを行うために、SQL スクリプトをユーザーの環境に合わせてカスタマイズすることがあります。

- 同一のレプリケーション・アクション (複数のサーバー用にカスタマイズされたもの) の複数のコピーを作成する。
- CD 表の表スペースまたはデータベースをサイズ変更する。
- サイト別の標準を定義する。
- 定義を結合してバッチ・ジョブとして実行する。
- 指定時刻までレプリケーション・アクションの実行を延期する。
- バックアップまたはサイト専用のカスタマイズしたい SQL スクリプトのライブラリーを作成したり、不定期接続の環境におけると同様に、分散サイトでスタンドアロン実行したりする。
- 表や索引の作成ステートメントを編集し、データベース・オブジェクトを表すようにする。
- Informix および非 DB2 リレーショナル・データベースの場合、目標の表スペースに表が確実に作成されるようにする。
- Microsoft SQL Server の場合、既存のセグメントにコントロール表を作成する。
- 複数のサブスクリプション・セットを同時に定義する方法として、サブスクリプション・セット・メンバーの述部を表示または編集する。述部に置換変数を使用して、その変数をプログラミング論理によって解決することができます。

SQL スクリプトを DB2 コマンド行から実行する場合は、SQL スクリプトの実行時にサーバーに手動で接続しなければなりません。生成されたスクリプトには、CONNECT ステートメントが含まれています。SQL スクリプトを実行する前に、

ユーザー ID とパスワードをサーバーに指定する SQL ステートメントを編集しなければなりません。たとえば、以下の例に似た行を探し、プレースホルダー (XXXX) を上書きして情報を追加します。

```
CONNECT TO srcdb USER XXXX USING XXXX ;
```

手順:

次の方法のどちらかを使用して、SQL スクリプトが含まれたファイルを DB2 コマンド行から実行します。

- SQL スクリプトにセミコロン (;) が終了文字として含まれている場合は、このコマンドを使用する。

```
db2 -tvf filename
```

- SQL スクリプトに区切り文字として他の文字が含まれている場合は (この例では、異機種のレプリケーションなどでポンド記号 (#) が終了文字)、このコマンドを使用する。

```
db2 -td# -vf filename
```

推奨事項: スクリプトの実行前には、常に管理ログ・ファイルを読んでください。

第 9 章 SQL レプリケーションに関するキャプチャー・プログラムの操作

この章は、DB2 データベースに対するログ・ベースのキャプチャーに関するものです。トリガー・ベースのキャプチャーを使用している場合は、登録時にトリガーが作成されるため、ユーザーはこの章で説明されている操作を実行しません。

この章には、以下のセクションがあります。

- ・ 『キャプチャー・プログラムのデフォルトの稼働パラメーター』
- ・ 127 ページの 『キャプチャー・プログラムの稼働パラメーターの変更』
- ・ 129 ページの 『キャプチャー・プログラムの始動 (Linux、UNIX、Windows、z/OS)』
- ・ 140 ページの 『キャプチャー・プログラムの始動 (OS/400)』
- ・ 141 ページの 『実行中のキャプチャー・プログラムの動作の変更』
- ・ 142 ページの 『キャプチャー・パラメーター表の稼働パラメーターの変更』
- ・ 143 ページの 『キャプチャー・プログラムの停止』
- ・ 144 ページの 『キャプチャーの中断 (Linux、UNIX、Windows、z/OS)』
- ・ 144 ページの 『キャプチャーの再開 (Linux、UNIX、Windows、z/OS)』
- ・ 145 ページの 『キャプチャーの再初期化』

重要: DB2 ユーティリティは、変更をキャプチャー・プログラムから見える方法ではログに記録しません。そのため、キャプチャー・プログラムは一部の DB2 ユーティリティによって行われた変更を一切キャプチャーしません。

キャプチャー・プログラムのデフォルトの稼働パラメーター

キャプチャーには、デフォルト値のあるパラメーターがいくつかあります。製品出荷時のデフォルト値は、表 6 および 126 ページの表 7 に示されています。ほとんどの稼働パラメーターのデフォルト値は、出荷時にはキャプチャー・パラメーター (IBMSNAP_CAPPARMS) 表に保管されています。これらのデフォルトをご使用のレプリケーション環境で使用し、必要に応じて 127 ページの 『キャプチャー・プログラムの稼働パラメーターの変更』 で説明されている方法のいずれかを使用して変更してください。

表 6. キャプチャー稼働パラメーターのデフォルト設定 (Linux、UNIX、Windows、z/OS)

稼働パラメーター	デフォルト値	IBMSNAP_CAPPARMS 表の列名
capture_server	DB2DBDFT ¹	該当せず
capture_schema	ASN ²	該当せず
add_partition	n ⁴	該当せず
retention_limit	10080 分	RETENTION_LIMIT
lag_limit	10080 分	LAG_LIMIT
commit_interval	30 秒	COMMIT_INTERVAL

表6. キャプチャー稼働パラメーターのデフォルト設定
(Linux、UNIX、Windows、z/OS) (続き)

稼働パラメーター	デフォルト値	IBMSNAP_CAPPARMS 表の列名
prune_interval	300 秒	PRUNE_INTERVAL
trace_limit	10080 分	TRACE_LIMIT
monitor_limit	10080 分	MONITOR_LIMIT
monitor_interval	300 秒	MONITOR_INTERVAL
memory_limit	32 MB	MEMORY_LIMIT
autoprun	y ³	AUTOPRUNE
term	y ³	TERM
autostop	n ⁴	AUTOSTOP
logreuse	n ⁴	LOGREUSE
logstdout	n ⁴	LOGSTDOUT
sleep_interval	5 秒	SLEEP
capture_path	キャプチャーが始動された ディレクトリー ⁵	CAPTURE_PATH
startmode	warmsi ⁶	STARTMODE

注:

1. キャプチャー・コントロール・サーバーは、Windows、Linux、および UNIX の場合の DB2DBDFT 環境変数の値である (変数が指定されている場合)。z/OS にはデフォルトはない。
2. キャプチャー・スキーマのデフォルトは変更できない。別のキャプチャー・スキーマを使用するには、**capture_schema** 始動パラメーターを使用する。
3. Yes
4. No
5. キャプチャーが Windows サービスとして始動する場合、そのキャプチャー・パスは %sqllib%bin である。
6. 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始動である場合にのみ、コールド・スタートに切り替わる。

これらの稼働パラメーターとそのデフォルトについての詳細は、129 ページの『キャプチャー・プログラムの始動 (Linux、UNIX、Windows、z/OS)』を参照してください。

表7. キャプチャー稼働パラメーターのデフォルト設定 (OS/400)

稼働パラメーター	デフォルト値	IBMSNAP_CAPPARMS 表の列名
CAPCTLLIB	ASN ¹	該当せず
JOB	*LIBL/QZSNDPR	該当せず
JRN	*ALL	該当せず
RETAIN	10080 分	RETENTION_LIMIT
LAG	10080 分	LAG_LIMIT
FRCFRQ	30 秒	COMMIT_INTERVAL
CLNUPITV	*IMMED ²	該当せず
CLNUPITV	86400 秒 ²	PRUNE_INTERVAL
CLNUPITV	*IMMED ²	該当せず

表 7. キャプチャー稼働パラメーターのデフォルト設定 (OS/400) (続き)

稼働パラメーター	デフォルト値	IBMSNAP_CAPPARMS 表の列名
TRCLMT	10080 分	TRACE_LIMIT
MONLMT	10080 分	MONITOR_LIMIT
MONITV	300 秒	MONITOR_INTERVAL
MEMLMT	32 MB	MEMORY_LIMIT
WAIT	120 秒	該当せず
RESTART	*YES ³	該当せず

注:

1. キャプチャー・スキーマのデフォルトは変更できない。別のキャプチャー・スキーマを使用するには、キャプチャー・プログラムの始動時に CAPCTLLIB パラメーターを指定する。他のほとんどの稼働パラメーターのデフォルト値は、出荷時にはキャプチャー・パラメーター (IBMSNAP_CAPPARMS) 表に保管されている。
2. CLNUPITV には 2 つのサブパラメーターがある。デフォルトでは、キャプチャー・プログラムは実行開始直後に整理を実行し、整理インターバルに達するたび (デフォルトでは 24 時間ごと) に再び整理を実行する。
3. デフォルトでは、キャプチャー・プログラムはウォーム・スタートする。

これらの稼働パラメーターとそのデフォルトについての詳細は、385 ページの『第 19 章 SQL レプリケーション用のシステム・コマンド (OS/400)』を参照してください。

キャプチャー・プログラムの稼働パラメーターの変更

稼働パラメーターのデフォルト値を、ご使用の環境で通常使用している値に変更できます。これらのデフォルト値は、キャプチャー・プログラムの始動時にオーバーライドできます。あるいは、キャプチャー・プログラムの実行中に変更できます。

IBMSNAP_CAPPARMS 表での新しいデフォルト値の設定

キャプチャー・パラメーター (IBMSNAP_CAPPARMS) 表には、キャプチャー・プログラムの動作をコントロールするために変更できるパラメーターが含まれています。この表のスキーマ名はキャプチャー・スキーマです。この表が作成されると、その中にはキャプチャー・プログラム用の出荷時のデフォルト値が入れられます。CAPPARMS 表の列値が設定されていない場合は、125 ページの表 6 および 126 ページの表 7 に示されているハードコーディングされたデフォルト値が使用されます。表内の値の変更方法についての詳細は、142 ページの『キャプチャー・パラメーター表の稼働パラメーターの変更』を参照してください。

キャプチャー・プログラムの始動時に行うパラメーター値の指定

キャプチャー・プログラムの始動時に、このプログラム用の値を指定できます。始動時に設定された値により、現行セッションの間のキャプチャーの動作がコントロールされ、デフォルトの稼働パラメーター値およびキャプチャー・パラメーター表内に存在する可能性のあるあらゆる値がオーバーライドされます。キャプチャー・パラメーター表内の値については、これらの値による更新は行われません。キャプチャー・プログラムを始動する前にキャプ

チャー・パラメーター表を変更せず、さらにキャプチャー・プログラムの始動時にパラメーターをまったく指定しなかった場合は、稼働パラメーターに対してデフォルト値が使用されます。

キャプチャー・プログラムの実行中に行うパラメーター値の変更

キャプチャーの実行中に、その稼働パラメーターを一時的に変更できます。キャプチャー・プログラムは、ユーザーが値を再び変更するまで、またはキャプチャー・プログラムを停止して再始動するまで、その新しい値を使用します。キャプチャー・パラメーターは、セッションの間必要なだけ何度でも変更できます。詳細については、141 ページの『実行中のキャプチャー・プログラムの動作の変更』を参照してください。

例 (Linux、UNIX、Windows): キャプチャー・スキーマ ASNPROD のキャプチャー・コミット・インターバルのデフォルト設定を使用しないとします。

1. ASNPROD キャプチャー・スキーマ用のキャプチャー・パラメーター表を更新する。コミット・インターバルを 60 秒に設定してください。これにより、今後のキャプチャー・プログラムの始動時にはデフォルト・コミット・インターバルが 60 秒に設定されます。

```
update asnprod.ibmsnap_capparms set commit_interval=60;
```

2. 場合によっては、なんらかのパフォーマンス・チューニングが必要となることがあります。そのために、比較的小さな値を指定したコミット・インターバルを使用してキャプチャーを始動してみるがあります。キャプチャー・パラメーター表内の値を変更する代わりに、単に 20 秒に設定したコミット・インターバルのパラメーターを使用してキャプチャー・プログラムを始動してください。キャプチャー・プログラムが 20 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。

```
asncap capture_server=srcdb1 capture_schema=asnprod commit_interval=20
```

3. 値をさらに小さくしたコミット・インターバルを試してみる。キャプチャー・プログラムを停止する代わりに、コミット・インターバルを 15 秒に設定するパラメーターの変更要求をサブミットしてください。キャプチャー・プログラムは稼働を継続しますが、ただし 15 秒ごとにデータをコミットするようになります。

```
asnccmd capture_server=srcdb1 capture_schema=asnprod chgparms  
commit_interval=15
```

重要: 変更するパラメーターは、**chgparms** コマンドの直後に指定する必要があります。

4. 引き続きパフォーマンスのモニターとコミット・インターバルのパラメーターの変更を行う。キャプチャー・プログラムを停止する必要はありません。最終的に、ニーズに合ったコミット・インターバルを見つけたときは、キャプチャー・パラメーター表を (ステップ 1 で説明されている方法で) 更新して、次の始動時にキャプチャー・プログラムがその新しい値をデフォルト・コミット・インターバルとして使用するようになります。

例 (OS/400): キャプチャー・スキーマ ASNPROD のキャプチャー・コミット・インターバルのデフォルト設定を使用しないとします。

1. ASNPORD キャプチャー・スキーマ用のキャプチャー・パラメーター表を更新する。コミット・インターバルを 90 秒に設定してください。これにより、今後のキャプチャー・プログラムの始動時にはデフォルト・コミット・インターバルが 90 秒に設定されます。

CHGDPRCAPA CAPCTLLIB(ASNPORD) FRCFRQ(90)

2. 場合によっては、なんらかのパフォーマンス・チューニングが必要となることがあり、そのために、比較的小さな値を指定したコミット・インターバルを使用してキャプチャーを始動してることがあります。キャプチャー・パラメーター表内の値を変更する代わりに、45 秒に設定したコミット・インターバルのパラメーターを使用してキャプチャー・プログラムを始動してください。キャプチャー・プログラムが 45 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てください。

STRDPRCAPA CAPCTLLIB(ASNPORD) FRCFRQ(45)

3. 値をさらに小さくしたコミット・インターバルを試してみる。キャプチャー・プログラムを停止する代わりに、コミット・インターバルを 30 秒に設定するパラメーターの変更要求をサブミットしてください。キャプチャー・プログラムは稼働を継続しますが、ただし 30 秒ごとにデータをコミットするようになります(注：OS/400 では、30 秒より短いコミット・インターバルは設定できません)。

OVRDPRCAPA CAPCTLLIB(ASNPORD) FRCFRQ(30)

4. 最終的に、ニーズに合ったコミット・インターバルを見つけたときは、キャプチャー・パラメーター表を (ステップ 1 で説明されている方法で) 更新して、次の始動時にキャプチャー・プログラムがその新しい値をデフォルト・コミット・インターバルとして使用するようになります。

キャプチャー・プログラムの始動 (Linux、UNIX、Windows、z/OS)

キャプチャー・プログラムを始動して、ログからのデータのキャプチャーを開始してください。キャプチャー・プログラムがデータをキャプチャーするのは、DB2 データベースからだけです。非 DB2 リレーショナル・ソースから変更をキャプチャーするためにトリガー・ベースのキャプチャーを使用している場合は、登録時にトリガーが作成されるため、キャプチャー・プログラムを始動する必要はありません。

重要: DB2 ユーティリティーは、変更をキャプチャー・プログラムから見える方法ではログに記録しません。そのため、キャプチャー・プログラムは DB2 ユーティリティーによって行われた変更を一切キャプチャーしません。

キャプチャー・プログラムの始動後、キャプチャー・プログラムがデータのキャプチャーをすぐには開始しないことがあります。このプログラムがデータのキャプチャーを開始するのは、ターゲット表を完全にリフレッシュしたことをアプライ・プログラムがキャプチャー・プログラムにシグナルで通知した後だけです。このシグナルを受け取ると、キャプチャー・プログラムは所定のソース表用のログからの変更のキャプチャーを開始します。

ヒント: キャプチャー・ログ・ファイル (Linux、UNIX、および Windows の場合は `db2instance.capture_server.capture_schema.CAP.log`、z/OS の場合は `capture_server.capture_schema.CAP.log`) の中に、変更のキャプチャーが開始されたことを示すメッセージがないかどうか調べます。以下に例を示します。

ASN0104I Change capture has been started for the source table "REGRESS.TABLE1" for changes found in the log beginning with log sequence number "0000:0275:6048".

前提条件:

キャプチャー・プログラムを始動する前に、以下の前提条件が満たされていることを必ず確認してください。

- ソース・サーバーおよびキャプチャー・コントロール・サーバーへの接続が構成済みである。
- 正しい許可を受けている。
- 適切なキャプチャー・スキーマ用のコントロール表が作成されており、登録が定義済みである。
- レプリケーション・プログラムが構成済みである。

手順:

以下の方法のいずれかを使用して、DB2 for Linux、DB2 for UNIX、DB2 for Windows、および DB2 for z/OS でキャプチャー・プログラムを始動してください。

レプリケーション・センター

「キャプチャーの開始」ウィンドウを使用して、レプリケーション・センターのオブジェクト・ツリー内において選択されているキャプチャー・コントロール・サーバーのキャプチャー・スキーマによって識別されるキャプチャー・プログラムを実行します。詳しくは、レプリケーション・センターのヘルプを参照してください。

asncap システム・コマンド

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、342 ページの『asncap: キャプチャーの始動』を参照してください。

MVS コンソールまたは TSO (z/OS)

詳細については、483 ページの『第 20 章 SQL レプリケーション・プログラムの操作 (z/OS)』を参照してください。

Windows サービス (Windows)

詳細については、489 ページの『第 21 章 Windows Service Control Manager を使用して SQL レプリケーションのシステム・コマンドを発行する方法 (Windows)』を参照してください。

キャプチャー・プログラムの始動にどの手順を使用するかにかかわらず、始動パラメーターを選択できます。以下のセクションでは、始動パラメーターについて説明し、各パラメーターで選択するべき値を推奨します。

- 131 ページの『add_partition (Linux、UNIX、Windows)』
- 131 ページの『autoprune (Linux、UNIX、Windows、z/OS)』
- 132 ページの『autostop (Linux、UNIX、Windows、z/OS)』
- 132 ページの『capture_path (Linux、UNIX、Windows、z/OS)』
- 133 ページの『capture_schema (Linux、UNIX、Windows、z/OS)』
- 133 ページの『capture_server (Linux、UNIX、Windows、z/OS)』
- 134 ページの『commit_interval (Linux、UNIX、Windows、z/OS)』

- 134 ページの『lag_limit (Linux、UNIX、Windows、z/OS)』
- 134 ページの『logreuse (Linux、UNIX、Windows、z/OS)』
- 135 ページの『logstdout (Linux、UNIX、Windows、z/OS)』
- 135 ページの『memory_limit (Linux、UNIX、Windows、z/OS)』
- 136 ページの『monitor_interval (Linux、UNIX、Windows、z/OS)』
- 136 ページの『monitor_limit (Linux、UNIX、Windows、z/OS)』
- 136 ページの『prune_interval (Linux、UNIX、Windows、z/OS)』
- 137 ページの『retention_limit (Linux、UNIX、Windows、z/OS)』
- 138 ページの『sleep_interval (Linux、UNIX、Windows、z/OS)』
- 138 ページの『startmode (Linux、UNIX、Windows、z/OS)』
- 139 ページの『term (Linux、UNIX、Windows、z/OS)』
- 139 ページの『trace_limit (Linux、UNIX、Windows、z/OS)』

add_partition (Linux、UNIX、Windows)

デフォルト: **add_partition=n**

add_partition パラメーターは、最後にキャプチャー・プログラムが再始動されてから、新しく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、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。

add_partition=y と設定し、キャプチャー・プログラムがログ・ファイルを読み取るようにします。各新規パーティションにおいて、ウォーム・スタート・モードでキャプチャー・プログラムが開始されると、最初のデータベース CONNECT ステートメントの後に DB2 が使用する、最初のログ・シーケンス番号 (LSN) から始まるログ・ファイルを、キャプチャーが読み取ります。

autoprun (Linux、UNIX、Windows、z/OS)

デフォルト: **autoprun=y**

autoprun パラメーターは、キャプチャー・プログラムがそのコントロール表のいくつかを自動的に整理するかどうかを指定します。デフォルト (**autoprun=y** を使用) では、キャプチャー・プログラムは CD 表と UOW 表の行、ならびにキャプチャー・トレース (IBMSNAP_CAPTRACE) 表、キャプチャー・モニター (IBMSNAP_CAPMON) 表、およびシグナル (IBMSNAP_SIGNAL) 表の行を自動的に整理します。 **autoprun=n** を設定した場合は、整理コマンドを使用してこれらの表を整理する必要があります。

自動整理をオンに設定してキャプチャーを始動する場合は、整理インターバルを設定して、使用しているレプリケーション環境での整理頻度を最適化してください。詳細については、136 ページの『prune_interval (Linux、UNIX、Windows、z/OS)』を参照してください。

キャプチャー・プログラムは以下のパラメーターを使用して、どの行が整理されてもよいほど古いのかを判別します。

- 137 ページの『retention_limit (Linux、UNIX、Windows、z/OS)』 (CD 表、UOW 表、およびシグナル表の場合)

- 136 ページの『monitor_limit (Linux、UNIX、Windows、z/OS)』(モニター表の場合)
- 139 ページの『trace_limit (Linux、UNIX、Windows、z/OS)』(キャプチャー・トレース表の場合)

表の整理についての詳細は、255 ページの『コントロール表の整理』を参照してください。

autostop (Linux、UNIX、Windows、z/OS)

デフォルト: **autostop=n**

autostop パラメーターは、ログの終わりに達した後にキャプチャー・プログラムが稼働し続けるか終了するかをコントロールします。

デフォルト (**autostop=n**) では、キャプチャー・プログラムはトランザクションの検索後に終了しません。

モバイル環境または時々しか接続しない環境でレプリケーションを行っている場合は **autostop=y** オプションを使用してください。 **autostop** を使用すると、キャプチャー・プログラムは確実にすべての適格トランザクションを検索し、ログの終わりに達した時点で必ず停止します。それ以外のトランザクションも検索する場合は、キャプチャーを再始動する必要があります。 **autostop=y** オプションは、テスト環境でも使用されます。

推奨: ほとんどのケースでは、**autostop=y** を使用すべきではありません。なぜなら、これを使用すると、レプリケーションの管理に多大なオーバーヘッドが加えられるからです (たとえば、キャプチャー・プログラムを始終再始動する必要があります)。

capture_path (Linux、UNIX、Windows、z/OS)

キャプチャー・パスは、キャプチャー・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です。デフォルトでは、キャプチャー・パスはキャプチャー・プログラムが始動されるディレクトリーです。キャプチャー・プログラムを Windows サービスとして始動する場合、デフォルトではキャプチャー・プログラムは `¥sqllib¥bin` ディレクトリーで始動されます。z/OS オペレーティング・システムでは、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、デフォルトのキャプチャー・パスは、次に示す方法のどちらでこのプログラムを始動するかによって決まります。

- 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始動する場合。キャプチャー・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。
- キャプチャー・プログラムを開始済みのタスクまたは JCL を使用して始動する場合。デフォルト・キャプチャー・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。

キャプチャー・パスを変更して、キャプチャー・プログラムがファイルを保管する場所を指定できます。パス名 (たとえば、`/home/db2inst/capture_files`) を指定できます。z/OS オペレーティング・システムでは、パス名または `//CAPV8` 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。順次データ・セットは、プログラムを実行中のユーザー ID と関連付けられています。それ以外の場合は、これらのファイル名は、明示的に名前を指定されたディレクトリー・パスに保管されるファイル名 (HLQ がファイル名の最初の部分として連結されているもの) に類似します。たとえば、`sysadm.CAPV8.filename` のようになります。

capture_schema (Linux、UNIX、Windows、z/OS)

デフォルト: `capture_schema=ASN`

capture_schema パラメーターにより、どのキャプチャー・プログラムを始動するのが識別されます。デフォルトでは、キャプチャー・スキーマは ASN です。

別のスキーマをすでにセットアップしてある場合は、**capture_schema** パラメーターを使用してそのスキーマを指定すれば、当該のキャプチャー・プログラムを始動できます。詳細については、28 ページの『キャプチャー・コントロール表のセットを複数作成する』を参照してください。

次のような場合には、複数のキャプチャー・スキーマを使用することがあります。

アプリケーションの独立性を保ってアーカイブする場合

アプリケーション A 用のキャプチャー・プログラムとアプリケーション B 用のキャプチャー・プログラムを別個に持てるように、複数のキャプチャー・スキーマを作成してください。各キャプチャー・プログラムは、それぞれ独自のコントロール表を使用します。キャプチャー・プログラムのいずれかがダウンしても、1 つのアプリケーションだけしか影響を受けません。それ以外のアプリケーションは、別のキャプチャー・プログラムからサービスを受けているために影響されません。

各アプリケーションのそれぞれ異なる要件を満たす場合

同じソース表を使用するが、データ要件がそれぞれ異なる複数のアプリケーションがある場合は、複数のキャプチャー・スキーマを作成してください。たとえば、給与計算アプリケーションは機密従業員データを必要としますが、内部従業員登録にはこのようなデータは必要ありません。その機密情報を 1 つのキャプチャー・スキーマだけに登録して、それ以外のキャプチャー・スキーマには登録しないようにできます。同様に、一部のアプリケーションではキャプチャー・プログラムに異なる動作をさせる必要がある場合は、1 つの表を複数回登録できます。たとえば、一部のアプリケーションではキャプチャー・プログラムが更新の保管を削除と挿入の対として実行する必要があるというような場合です。

登録に関する問題を分離する場合

1 つの登録に問題がある場合は、別のキャプチャー・スキーマを作成して、作業登録をそのスキーマに移動できます。この方法により、その問題のある登録をオリジナルのスキーマ内でデバッグし、影響を受けていない登録を別のスキーマを使用して実行できます。

capture_server (Linux、UNIX、Windows、z/OS)

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、キャプチャー・コントロール・サーバーを指定します。Linux、UNIX、および Windows オペレーティング・システム上でキャプチャーを実行している場合は、キャプチャー・コントロール表 (登録表など) はソース表の登録情報を含み、キャプチャー・コントロール・サーバー上に配置されます。z/OS でキャプチャーを実行している場合、キャプチャー・コントロール表は DB2 サブシステム名に配置されます。キャプチャー・プログラムは DB2 ログを読み取るため、キャプチャー・プログラムがソース・データベースと同じサーバーで実行されている必要があります。

commit_interval (Linux、UNIX、Windows、z/OS)

デフォルト: `commit_interval=30`

`commit_interval` パラメーターは、キャプチャー・プログラムがキャプチャー・コントロール表 (UOW 表および CD 表も含む) に対してデータをコミットする頻度を秒単位で指定します。デフォルトでは、キャプチャー・プログラムは UOW 表および CD 表にデータをコミットする前に 30 秒待ちます。コミット・インターバル内に更新された表に対して、ロックが保留されます。`commit_interval` パラメーターに比較的大きな値を指定すると、キャプチャー・プログラムの CPU 使用量が削減されますが、それと同時に、頻繁に実行されているサブスクリプション・セットの待ち時間が増える可能性があります。なぜなら、アプライ・プログラムが取り出せるのはコミット済みのデータだけだからです。

lag_limit (Linux、UNIX、Windows、z/OS)

デフォルト: `lag_limit=10,080`

`lag_limit` パラメーターは、キャプチャー・プログラムが DB2 ログからのレコードを処理する際に許される遅れの分数を表します。

デフォルトでは、ログ・レコードが 10,080 分 (7 日) より古い場合、キャプチャー・プログラムがコールド・スタートに切り替えるのを許可する値が `startmode` パラメーターに指定されないかぎり、キャプチャー・プログラムは始動しません。

遅れが限度に達したためにキャプチャー・プログラムが始動しない場合は、キャプチャー・プログラムがログの読み取りで遅れを出している理由を判別する必要があります。この遅れの限度に関するパラメーターを実際に使用することのないテスト環境では、遅れの限度をより高く設定して、キャプチャー・プログラムの再始動を試行する場合があります。別の方法として、使用するテスト環境のソース表にはほんの少しのデータしか入っていない場合にはキャプチャー・プログラムをコールド・スタートして、すべてのターゲット表内のデータを完全にリフレッシュすることもあります。

logreuse (Linux、UNIX、Windows、z/OS)

デフォルト: `logreuse=n`

キャプチャー・プログラムは、操作情報をログ・ファイルに保管します。

Linux、UNIX、および Windows オペレーティング・システムでは、ログ・ファイルの名前は `db2instance.capture_server.capture_schema.CAP.log` です。たとえば、`DB2INST.SRCDB1.ASN.CAP.log`。

z/OS オペレーティング・システムの場合のファイル名は DB2 インスタンス名が含まれないことを除いては同じです。たとえば、`SRCDDB1.ASN.CAP.log`。このファイルは、**capture_path** パラメーターで指定されているディレクトリーに保管されます。**capture_path** パラメーターが高位修飾子 (HLQ) として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ます。したがって、ログ・ファイル名を作成するのに使用される **capture_schema** 名は、その名前の最初の 8 文字までの長さに切り捨てられます。

デフォルト (**logreuse=n**) では、キャプチャー・プログラムはメッセージをログ・ファイルに付加します。これは、キャプチャー・プログラムが再始動された後であっても同様に行われます。メッセージの履歴が必要な場合は、デフォルトのままにしておいてください。次のような場合には、キャプチャー・プログラムが再始動時にログを削除して再作成するようにさせる (**logreuse=y**) 必要があります。

- ログが大きくなったため、ログを消去したい場合。
- ログに保管されている履歴が不要な場合。
- スペースを節約したい場合。

logstdout (Linux、UNIX、Windows、z/OS)

デフォルト: **logstdout=n**

logstdout パラメーターが使用可能なのは、**asncap** コマンドを使用する場合だけです。このパラメーターは、レプリケーション・センターでは使用不可です。

デフォルトでは、キャプチャー・プログラムは一部の警告メッセージと通知メッセージをログ・ファイルにのみ送信します。トラブルシューティング中や、テスト環境でキャプチャー・プログラムの動作をモニターしているときには、このようなメッセージは標準出力 (STDOUT) に送信されるようにする (**logstdout=y**) 場合があります。

memory_limit (Linux、UNIX、Windows、z/OS)

デフォルト: **memory_limit=32**

memory_limit パラメーターは、キャプチャー・プログラムが使用できるメモリーの量を MB (M バイト) 単位で指定します。

デフォルトでは、キャプチャー・プログラムは 32 MB のメモリーを使用してトランザクション情報を保管します。その後あふれた分の情報は、**capture_path** ディレクトリーに配置されているファイルに入れられます。このメモリーの限度は、パフォーマンスに関するニーズに基づいて変更できます。メモリーの限度を高めを設定すると、キャプチャーのパフォーマンスは向上しますが、システム上のその他のユーザーが使用可能なメモリーが少なくなります。メモリーの限度を低めに設定すると、その他のユーザーのためにメモリーが解放されます。メモリーの限度をあまりに低く設定して、キャプチャー・プログラムが入りきらない分をファイルに入れる

よくなると、システム上で使用するスペースが増え、さらに入出力のためにシステムがスローダウンすることになります。

メモリーの限度は、レプリケーション・アラート・モニターを使用してモニターできます。また、CAPMON 表内のデータを使用すると、メモリー制限のためにディスクにあふれたソース・システム・トランザクションの数も判別できます。CAPMON 表の TRANS_SPILLED 列の値を合計してください。

monitor_interval (Linux、UNIX、Windows、z/OS)

デフォルト: `monitor_interval=300`

monitor_interval パラメーターは、キャプチャー・プログラムが情報をキャプチャー・モニター (IBMSNAP_CAPMON) 表に書き込む頻度を指定します。

デフォルトでは、キャプチャー・プログラムはキャプチャー・モニター表に 300 秒 (5 分) ごとに行を挿入します。この稼働パラメーターは、コミット・インターバルと関連して機能します。データを細分化されたレベルでモニターしてみたい場合は、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_limit (Linux、UNIX、Windows、z/OS)

デフォルト: `monitor_limit=10080`

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モニター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、キャプチャー・モニター (IBMSNAP_CAPMON) 表内の 10,080 分 (7 日) より古い行が整理されます。IBMSNAP_CAPMON 表には、キャプチャー・プログラムに関する稼働統計が含まれています。1 週間分より少ない統計しか必要でない場合は、デフォルトのモニター限度を使用してください。統計を頻繁に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、キャプチャー・モニター表がより頻繁に整理されるようにして、古い統計が除去されるようにできます。この統計を履歴の分析に使用する場合、および 1 週間より長い期間の統計が必要である場合は、モニター限度の値を大きくしてください。

prune_interval (Linux、UNIX、Windows、z/OS)

デフォルト: `prune_interval=300`

prune_interval パラメーターは、キャプチャー・プログラムが一部のコントロール表から古い行を削除しようとする、その試行頻度を指定します。このパラメーターが有効なのは、`autoprun=y` の場合だけです。

デフォルトでは、キャプチャー・プログラムは CD 表および UOW 表を 300 秒 (5 分) ごとに整理します。表の整理頻度が不十分であると、それらの表が含まれている表スペース用のスペースが使い尽くされる可能性があります。そうなった場合は、キャプチャー・プログラムは強制停止されます。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。オプションとしての整理頻度をレプリケーション環境で設定する場合は、実際のレプリ

ケーション環境に最適な値を使用してください。一般に、パフォーマンスは表のサイズが小さく保たれている場合に最適となります。

整理インターバルの値を低くする前に、整理が発生する可能性があるほどデータが頻繁にアプライされていることを確認してください。アプライ・プログラムがデータを頻繁にアプライしていない場合は、整理インターバルの値を低く設定しても無益です。なぜなら、CD 表と UOW 表を整理するためには、その前にアプライ・プログラムがすべてのターゲットに対してデータを複製する必要があるからです。

整理インターバルにより、キャプチャー・プログラムがどのくらいの頻度で表の整理を試行するのが決定されます。整理間隔は、データが整理の対象となるのに十分なだけ古くなったと見なす時点 を決定するパラメーター (**trace_limit**、**monitor_limit**、**retention_limit**) と関連して機能します。たとえば、**prune_interval** が 300 秒、**trace_limit** が 10080 秒である場合は、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。トレース表で 10080 分 (7 日) より古い行を検出すると、キャプチャー・プログラムはそれらをすべて整理します。

表の整理についての詳細は、255 ページの『コントロール表の整理』を参照してください。

retention_limit (Linux、UNIX、Windows、z/OS)

デフォルト: **retention_limit=10,080**

retention_limit パラメーターは、保持制限整理の対象として適格と見なすまでに、古いデータをどれだけの期間 CD 表、UOW 表、およびシグナル (IBMSNAP_SIGNAL) 表に残しておくかを決定します。

サブスクリプション・セットが非活動にされているか、またはときたましか実行されないために、通常の整理処理が使用禁止になっている場合は、データは長い間 CD 表および UOW 表に残されます。このデータが現行 DB2 タイム・スタンプから保持制限の値を引いたものより古くなると、このデータは保持制限整理処理によって表から削除されます。サブスクリプション・セットを非常にまれにしか実行しないか、またはアプライ・プログラムを停止すると、使用している CD 表と UOW 表が非常に大きくなって、保持制限整理の対象として適格となる可能性があります。

整理される行のいずれか 1 つでも レプリケーションの候補になっており、それにもかかわらずなんらかの理由でそれらがターゲット表にまだアプライされていない場合は、ターゲット表をリフレッシュして、ソースと同期化させる必要があります。比較的高い保持制限を使用すれば、フル・リフレッシュが発生するのを回避できます。ただし、使用している CD 表および UOW 表が大きくなり、システム上のスペースを使用します。

Update-anywhere レプリケーションを実行している場合は、保持制限整理により、リジェクトされたトランザクションが確実に削除されます。レプリカ・ターゲット表に対して競合検出を使用している場合、競合しているトランザクションが検出されると、結果的にトランザクションがリジェクトされます。それらのリジェクトされたトランザクションと関係のある CD 表および UOW 表の行は複製されず、保持制

限に達した時点で整理されます。リジェクトされたトランザクションと関係のあるすべての古い行が削除された場合は、フル・リフレッシュは不要です。

また、保持整理により、もはや不要となったシグナル情報もシグナル (IBMSNAP_SIGNAL) 表から確実に削除されます。

コントロール表の整理についての詳細は、255 ページの『コントロール表の整理』を参照してください。

sleep_interval (Linux、UNIX、Windows、z/OS)

デフォルト: `sleep_interval=5`

スリープ・インターバルは、キャプチャー・プログラムがログの終わりに到達し、バッファが空になった後、キャプチャー・プログラムは何秒待ってから再びログを読み取るかという、その秒数を表します。z/OS オペレーティング・システムでのデータ共有の場合は、スリープ・インターバルは、バッファがその全容量の半分より少ない領域しか使用されていない状態に戻った後、キャプチャー・プログラムがスリープする秒数を表します。

デフォルトでは、キャプチャー・プログラムは 5 秒間スリープします。キャプチャー・プログラムがログを読み取るために生じるオーバーヘッドを削減したい場合は、スリープ・インターバルを変更してください。スリープ・インターバルの値を小さくするということは、その分遅延が生じる機会が少なくなるということを意味します。スリープ・インターバルの値を大きくすると、まばらに更新が行われるシステムでは潜在的な CPU 使用量が節約されます。

startmode (Linux、UNIX、Windows、z/OS)

デフォルト: `startmode=warmsi`

キャプチャーは、以下の始動モードのいずれかを使用して始動できます。

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)

キャプチャー・プログラムはウォーム・スタートします。ただし、今回はじめてキャプチャー・プログラムを始動する場合は、コールド・スタートに切り替えられます。この始動モードは、コールド・スタートになるのは必ずキャプチャー・プログラムの最初の始動時だけにしたい場合に使用してください。

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)

キャプチャー・プログラムはウォーム・スタートする。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 を日常のレプリケーション環境で使用する場合は、ウォーム・スタートの実行を妨げているあらゆる問題 (データベースや表スペースが使用不能など) を修復する機会があります。この始動モードを使用して、予期せぬコールド・スタートが実行されるのを防いでください。ウォーム・スタートした場合、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。キャプチャー・プログラムの始動後にエラーが発生した場合は、キャプチャー・プログラムは終了し、すべての表はそのまま残されます。

ヒント: warmns は、キャプチャー・プログラムをはじめて始動するときには使用できません。なぜなら、キャプチャー・プログラムの最初の始動時に

はウォーム・スタート情報がないからです。キャプチャー・プログラムの最初の始動時に **cold** 始動モードを使用してから、**warmns** 始動モードを使用してください。始動モードを切り替えたくない場合は、代わりに **warmsi** を使用できます。

warmsa (ウォーム・スタート、常に必要に応じてコールド・スタートに切り替える) ウォーム・スタート情報が使用可能な場合、キャプチャー・プログラムは前回の実行で終了したところから処理を再開します。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。通常は、コールド・スタートへの切り替えは望ましくありません。なぜなら、すべてのターゲット表をリフレッシュする必要があるからです。

cold コールド・スタート時には、キャプチャー・プログラムは初期化の間に CD 表および UOW 表のすべての行を削除します。これらのレプリケーション・ソースへのすべてのサブスクリプション・セットは、次のアプライ処理サイクルの間に完全にリフレッシュされます (つまり、すべてのデータがソース表からターゲット表にコピーされます)。キャプチャー・プログラムがコールド・スタートを試行したときにフル・リフレッシュが使用不可にされていた場合、キャプチャー・プログラムは始動しますが、アプライ・プログラムは失敗して、エラー・メッセージを発行します。

キャプチャー・プログラムがコールド・スタートするよう明示的に要求する必要がある場合はめったにありません。コールド・スタートが必要となるのは、キャプチャー・プログラムの最初の始動時、および **warmsi** が推奨始動モードである場合だけです。

重要: 変更データの履歴を正確なものにしておきたい場合は、キャプチャー・プログラムのコールド・スタートは行わないでください。キャプチャー・プログラムをシャットダウンされた後でアプライ・プログラムが変更をレプリケーションできないと、ギャップが生じることがあります。また、コールド・スタートを避けたい以上、キャプチャー・パラメーター (IBMSNAP_CAPPARMS) 表で STARTMODE のデフォルトとしてコールド・スタート (cold) を指定しないでください。

term (Linux、UNIX、Windows、z/OS)

デフォルト: **term=y**

term パラメーターにより、DB2 の状況がキャプチャー・プログラムの動作にどのように影響するかが決定されます。

デフォルトでは、キャプチャー・プログラムは DB2 が終了すると終了します。

term=n は、DB2 がアクティブでない場合に DB2 が始動するのをキャプチャー・プログラムが待つようにする場合に使用してください。DB2 が静止している場合は、キャプチャーは終了しません。これは、キャプチャーはアクティブなままであるが、データベースは使用しないということです。

trace_limit (Linux、UNIX、Windows、z/OS)

デフォルト: **trace_limit=10,080**

trace_limit は、整理の対象にする前に、行がどのくらい古くなるまでキャプチャー・トレース (IBMSNAP_CAPTRACE) 表に残しておく必要があるのかを指定します。

キャプチャーが整理を実行する場合、デフォルトでは、キャプチャー・トレース (IBMSNAP_CAPTRACE) 表内の行は 10,080 分 (7 日) ごとに整理の対象として適格となります。CAPTRACE 表には、キャプチャー・プログラム用の監査証跡情報が含まれています。キャプチャーが実行することはすべてこの表に記録されます。そのため、キャプチャー・プログラムが非常にアクティブであると、この表が急速に大きくなる可能性があります。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。

キャプチャー・プログラムの始動 (OS/400)

キャプチャー・プログラムを始動して、ジャーナルからのデータのキャプチャーを開始してください。

キャプチャー・プログラムの始動後、キャプチャー・プログラムがデータのキャプチャーをすぐには開始しないことがあります。このプログラムがデータのキャプチャーを開始するのは、アプライ・プログラムがキャプチャー・プログラムに対し、所定のソース表用のログからの変更のキャプチャーを開始するよう求めるシグナルを送った後だけです。

前提条件:

キャプチャー・プログラムを始動する前に、17 ページの『第 2 章 SQL レプリケーション用のサーバーの構成』に示されている指示に従って、以下の前提条件が満たされていることを必ず確認してください。

- 正しい許可を受けている。
- 適切なキャプチャー・スキーマ用のコントロール表が作成されており、登録が定義済みである。
- レプリケーション・プログラムが構成済みである (キャプチャー・プログラムがリモート・ジャーナルを読み取っている場合)。

手順:

以下の方法のいずれかを使用して、OS/400 上でキャプチャー・プログラムを始動してください。

レプリケーション・センター

「キャプチャーの開始」ウィンドウを使用して、レプリケーション・センターのオブジェクト・ツリー内において選択されているキャプチャー・コントロール・サーバーのキャプチャー・スキーマによって識別されるキャプチャー・プログラムを実行します。詳しくは、レプリケーション・センターのヘルプを参照してください。

STRDPRCAP システム・コマンド (OS/400)

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、468 ページの『STRDPRCAP: キャプチャー・プログラムの始動 (OS/400)』を参照してください。

実行中のキャプチャー・プログラムの動作の変更

キャプチャー・プログラムの実行中に、1 つまたは複数の稼働パラメーターの値をオーバーライドすると、このプログラムの動作を変更できます。この変更は、キャプチャー・パラメーター (IBMSNAP_CAPPARMS) 表には書き込まれません。キャプチャー・プログラムは、ユーザーがキャプチャー・プログラムを停止するまで、またはさらに新しい値を指定するまで、その新しい値を使用します。

Linux、UNIX、Windows、および z/OS では、キャプチャー・プログラムの実行中に以下のキャプチャー・パラメーターを変更できます。

- Autoprune
- Autostop
- Commit_interval
- Lag_limit
- Logreuse
- Logstdout
- Memory_limit
- Monitor_interval
- Monitor_limit
- Prune_interval
- Retention_limit
- Sleep_interval
- Term
- Trace_limit

OS/400 では、所定のキャプチャー・スキーマについて、以下の稼働パラメーターの値をオーバーライドできます。

- CLNUPITV
- FRCFRQ
- MEMLMT
- MONLMT
- MONITV
- PRUNE
- RETAIN
- TRCLMT

値を変更した場合、その影響がすべてのパラメーターに即時に及ぼされるとはかぎらない場合があります。

前提条件:

特定のキャプチャー・スキーマを指定されたキャプチャー・プログラムが始動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、現行セッションのパラメーターの現行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、キャプチャー・プログラムの実行中に「キャプチャー・プログラムを実行するためのパラメーターの変更」ウィンドウを使用します。この方法を使用すると、実行中のキャプチャー・プログラムによって使用されるパラメーターの現行値を、変更前に参照できます。詳しくは、レプリケーション・センターのヘルプを参照してください。

chgparms システム・パラメーター (Linux、UNIX、Windows、z/OS)

この方法では、パラメーターの現行値は表示されません。新しく指定した値が、実行中のキャプチャー・インスタンスに送信されます。 349 ページの『asncmd: キャプチャーの操作』を参照。

OVRDPRCAPA システム・コマンド (OS/400)

448 ページの『OVRDPRCAPA: DPR キャプチャー属性のオーバーライド (OS/400)』を参照。

キャプチャー・パラメーター表の稼働パラメーターの変更

キャプチャー・パラメーター (IBMSNAP_CAPPARMS) 表には、キャプチャー・プログラム用の稼働パラメーターが含まれています。始動時に、キャプチャー・プログラムはこの表から、デフォルトに設定されている稼働中の動作に関する値を使用します。ただし、始動パラメーターを使用して新しい値が指定された場合を除きます。

キャプチャー・パラメーター表には 1 つの行しか入れられません。デフォルト値のいずれかを変更する場合は、行を挿入する代わりに列を更新することができます。この行を削除しても、出荷時のデフォルトが始動パラメーターによってオーバーライドされないかぎり、キャプチャー・プログラムはそれらのデフォルトを使用してやはり始動します。

キャプチャー・プログラムはこの表を始動時にしか読み取らないため、新しい設定でキャプチャー・プログラムを実行したい場合は、キャプチャー・プログラムの停止と始動を行う必要があります。キャプチャー・プログラムの実行中にキャプチャー・パラメーター表を変更し、さらにキャプチャー・プログラムを再初期化しても、キャプチャー・プログラムの動作は変更されません。この表の列についての記述は、501 ページの『第 24 章 SQL レプリケーションの表構造』を参照してください。

手順:

以下の方法のいずれかを使用して、グローバル稼働パラメーターを変更してください。これらのパラメーターは、キャプチャー・プログラムによって使用されるもので、キャプチャー・パラメーター (IBMSNAP_CAPPARMS) 表に保管されています。

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「キャプチャー・パラメーターの管理」ウィ

ンドウを使用して、キャプチャー・パラメーター表内の任意の値を表示または変更します。詳しくは、レプリケーション・センターのヘルプを参照してください。

CHGDPRCAP システム・コマンド (OS/400)

425 ページの『CHGDPRCAP: DPR キャプチャー属性の変更 (OS/400)』を参照。

キャプチャー・プログラムの停止と始動を行った後でのみ、パラメーターの変更は有効になりません。

キャプチャー・プログラムの停止

特定のキャプチャー・スキーマ用のキャプチャー・プログラムを停止できます。キャプチャー・プログラムを停止すると、それ以降当該のソースからはデータがキャプチャーされなくなります。

OS/400: キャプチャー・プログラムが停止した時点において開かれた状態にあった UOW 表とすべての CD 表を再編成することにした場合、キャプチャー・プログラムがシャットダウンするのに時間が必要です (このプログラムは即時にはシャットダウンしません)。

前提条件:

特定のキャプチャー・スキーマを指定されたキャプチャー・プログラムが始動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、特定のキャプチャー・スキーマ用のキャプチャー・プログラムを停止してください。

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「キャプチャーの停止」ウィンドウを使用して、選択したキャプチャー・スキーマ用の実行中のキャプチャー・プログラムを停止します。詳しくは、レプリケーション・センターのヘルプを参照してください。

asncmd stop システム・コマンド (Linux、UNIX、Windows、z/OS)

349 ページの『asncmd: キャプチャーの操作』を参照。

ENDDPRCAP システム・コマンド (OS/400)

434 ページの『ENDDPRCAP: キャプチャー・プログラムの停止 (OS/400)』を参照。

整理中にキャプチャー・プログラムを停止または中断した場合は、整理も中断されます。キャプチャー・プログラムを再開または再始動すると、整理は **autoprun** パラメーターに基づいて再開されます。

登録をドロップするためにキャプチャー・プログラムを停止する必要はありません。登録をドロップするときには、必ずその前にその登録を非活動化してください。詳細については、214 ページの『登録済みオブジェクトの変更のキャプチャーの停止』を参照してください。

キャプチャーの中断 (Linux、UNIX、Windows、z/OS)

キャプチャー・プログラムを中断すると、キャプチャー・プログラム環境を損なうことなく、ピーク時にオペレーティング・システムのリソースを操作可能なトランザクションのために解放できます。進行中の作業の終了後にキャプチャー・プログラムがシャットダウンしないようにする場合は、キャプチャー・プログラムを停止するのではなく、中断してください。キャプチャーを再開する場合には、キャプチャーを再始動する場合に必然的に生じるオーバーヘッドが生じません。

重要: レプリケーション・ソースを除去する前に、キャプチャー・プログラムを中断しないでください。その代わりに、レプリケーション・ソースを非活動化してから除去してください。

前提条件:

特定のキャプチャー・スキーマを指定されたキャプチャー・プログラムが始動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、キャプチャー・プログラムを実行中に中断してください。

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、「キャプチャーの中断 (Suspend Capture)」ウィンドウを使用してキャプチャー・プログラムを中断します。詳しくは、レプリケーション・センターのヘルプを参照してください。

asncmd suspend システム・コマンド

329 ページの『第 18 章 SQL レプリケーション用のシステム・コマンド (Linux、UNIX、Windows、z/OS)』を参照。

整理中にキャプチャー・プログラムを停止または中断した場合は、整理も中断されます。キャプチャー・プログラムを再開または再始動すると、整理は **autoprun** パラメーターに基づいて再開されます。

キャプチャーの再開 (Linux、UNIX、Windows、z/OS)

中断されたキャプチャー・プログラムに再びデータのキャプチャーを開始させる場合は、このプログラムを再始動する必要があります。

前提条件:

特定のキャプチャー・スキーマを指定されたキャプチャー・プログラムが中断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、キャプチャー・プログラムを再開してください (中断されている場合)。

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、「キャプチャーの再開 (Resume Capture)」

ウィンドウを使用して、中断されていたキャプチャー・プログラムを再開します。詳しくは、レプリケーション・センターのヘルプを参照してください。

asnccmd resume システム・コマンド

329 ページの『第 18 章 SQL レプリケーション用のシステム・コマンド (Linux、UNIX、Windows、z/OS)』を参照。

整理中にキャプチャー・プログラムを停止または中断した場合は、整理も中断されます。キャプチャー・プログラムを再開または再始動すると、整理は **autoprun** パラメーターに基づいて再開されます。

キャプチャーの再初期化

キャプチャー・プログラムの実行中に既存の登録済みオブジェクトの属性を変更する場合は、必ずキャプチャー・プログラムを再初期化してください。たとえば、登録 (IBMSNAP_REGISTER) 表内の値 CONFLICT_LEVEL、CHGONLY、RECAPTURE、および CHG_UPD_TO_DEL_INS などを変更する場合です。

OS/400 でのキャプチャーの場合は、以前キャプチャーされていなかったジャーナルを対象としてデータのキャプチャーを開始する場合にも再初期化が必要です。

前提条件:

特定のキャプチャー・スキーマを指定されたキャプチャー・プログラムが始動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、キャプチャー・プログラムを実行中に再初期化してください。

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、「キャプチャーの再初期化 (Reinitialize Capture)」ウィンドウを使用してキャプチャー・プログラムを再初期化します。詳しくは、レプリケーション・センターのヘルプを参照してください。

asnccmd reinit システム・コマンド

329 ページの『第 18 章 SQL レプリケーション用のシステム・コマンド (Linux、UNIX、Windows、z/OS)』を参照。

INZDPRCAP システム・コマンド

446 ページの『INZDPRCAP: DPR キャプチャー・プログラムの再初期化 (OS/400)』を参照。

関連タスク:

- 483 ページの『第 20 章 SQL レプリケーション・プログラムの操作 (z/OS)』
- 489 ページの『第 21 章 Windows Service Control Manager を使用して SQL レプリケーションのシステム・コマンドを発行する方法 (Windows)』

関連資料:

- 366 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』

- 349 ページの『asnccmd: キャプチャーの操作』
- 342 ページの『asncap: キャプチャーの始動』
- 434 ページの『ENDDPRCAP: キャプチャー・プログラムの停止 (OS/400)』
- 468 ページの『STRDPRCAP: キャプチャー・プログラムの始動 (OS/400)』

第 10 章 SQL レプリケーションに関するアプライ・プログラムの操作

この章では、アプライ・プログラムの始動および停止の方法を説明します。また、ASNDONE および ASNLOAD 出口ルーチンの使用方法も説明します。

この章には、以下のセクションがあります。

- 『アプライ・プログラムのデフォルトの稼働パラメーター』
- 148 ページの『アプライ・プログラムの稼働パラメーターの変更』
- 149 ページの『アプライ・プログラムの始動 (Linux、UNIX、Windows、z/OS)』
- 159 ページの『アプライ・プログラムの始動 (OS/400)』
- 160 ページの『アプライ・パラメーター表での稼働パラメーターの変更 (Linux、UNIX、Windows、z/OS)』
- 161 ページの『アプライ・プログラムの停止』
- 161 ページの『ASNDONE 出口ルーチンの変更 (Linux、UNIX、Windows、z/OS)』
- 162 ページの『ASNDONE 出口ルーチンの変更 (OS/400)』
- 164 ページの『ASNLOAD 出口ルーチンを使ったターゲット表のリフレッシュ』

アプライ・プログラムのデフォルトの稼働パラメーター

アプライには、デフォルト値のあるパラメーターがいくつかあります。製品出荷時のデフォルト値は、Linux、UNIX、Windows、および z/OS の場合は表 8、OS/400 の場合は 160 ページの表 9 に示されています。ほとんどの稼働パラメーターのデフォルト値は、出荷時にはアプライ・パラメーター (IBMSNAP_APPPARMS) 表に保管されています。これらのデフォルトをご使用のレプリケーション環境で使用し、必要に応じて 148 ページの『アプライ・プログラムの稼働パラメーターの変更』で説明されている方法のいずれかを使用して変更してください。

表 8. アプライ稼働パラメーターのデフォルト設定 (Linux、UNIX、Windows、z/OS)

稼働パラメーター	デフォルト値	IBMSNAP_APPPARMS 表の列名
apply_qual	デフォルトなし	APPLY_QUAL
apply_path	アプライが始動されたディレクトリー ¹	APPLY_PATH
control_server	DB2DBDFT ²	該当せず
copyonce	n ³	COPYONCE
db2_subsystem	デフォルトなし ⁴	該当せず
delay	6 秒	DELAY
errwait	300 秒	ERRWAIT
inamsg	y ⁵	INAMSG
loadxit	n ³	LOADXIT
logreuse	n ³	LOGREUSE

表 8. アプライ稼働パラメーターのデフォルト設定 (Linux、UNIX、Windows、z/OS) (続き)

稼働パラメーター	デフォルト値	IBMSNAP_APPPARMS 表の列名
logstdout	n ³	LOGSTDOUT
notify	n ³	NOTIFY
opt4one	n ³	OPT4ONE
pwdfile	asnpwd.aut	該当せず [*]
spillfile	disk ⁶	SPILLFILE
sleep	y ⁵	SLEEP
sqlerrcontinue	n ³	SQLERRCONTINUE
term	y ⁵	TERM
trlreuse	n ³	TRLREUSE

注:

1. アプライが Windows サービスとして始動する場合、そのパスは %sqllib%\bin である。
2. アプライ・コントロール・サーバーは、DB2DBDFT 環境変数の値である (指定されている場合)。Linux、UNIX、および Windows オペレーティング・システムの場合に限る。
3. no
4. DB2 サブシステム名は最大 4 文字。このパラメーターは必須である。DB2 サブシステム名は、z/OS オペレーティング・システムにのみ適用される。
5. yes
6. z/OS オペレーティング・システムでは、デフォルトは MEM である。

これらの稼働パラメーターとそのデフォルトについての詳細は、149 ページの『アプライ・プログラムの始動 (Linux、UNIX、Windows、z/OS)』を参照してください。

アプライ・プログラムの稼働パラメーターの変更

稼働パラメーターのデフォルト値を、ご使用の環境で通常使用している値に変更できます。アプライ・プログラムの開始時に、これらのデフォルト値をオーバーライドすることもできます。

IBMSNAP_APPPARMS 表での新しいデフォルト値の設定

アプライ・パラメーター (IBMSNAP_APPPARMS) 表には、アプライ・プログラムの動作をコントロールするために変更できるパラメーターが含まれています。この表が作成されると、その中にはアプライ・プログラム用の出荷時のデフォルト値が入れられます。APPPARMS 表の列値が設定されていない場合は、Linux、UNIX、Windows、z/OS の場合は 147 ページの表 8、OS/400 の場合は 160 ページの表 9 に示されている、ハードコーディングされたデフォルト値が使用されます。

アプライ・プログラムの始動時に行うパラメーター値の指定

アプライ・プログラムの始動時に、このプログラム用の値を指定できます。始動時に設定された値により、現行セッションの間のアプライの動作がコントロールされ、デフォルトの稼働パラメーター値およびアプライ・パラメーター表内に存在する可能性のあるあらゆる値がオーバーライドされます。アプライ・パラメーター表内の値については、これらの値による更新は行われません。アプライ・プログラムを始動する前にアプライ・パラメーター表を

変更せず、さらにアプライ・プログラムの始動時にどの稼働パラメーターも指定しなかった場合は、稼働パラメーターに対してデフォルト値が使用されます。

例 (Linux、UNIX、Windows): アプライ修飾子 ASNPROD の **errwait** のデフォルト設定を使用しないとします。ASNPROD アプライ修飾子用のアプライ・パラメーター表を更新します。 **errwait** インターバルを 600 秒に設定します。

```
update asn.ibmsnap_appparms set errwait=600 where apply_qual='ASNPROD';
```

アプライ・プログラムの始動 (Linux、UNIX、Windows、z/OS)

アプライ・プログラムのインスタンスを始動して、ターゲットへのデータのアプライを開始できます。

アプライ・プログラムは始動された後、以下のイベントのいずれかが発生するまで、連続的に実行されます (**copyonce** 始動パラメーターが使用されない場合)。

- ユーザーがレプリケーション・センターまたはコマンドを使用してアプライ・プログラムを停止する。
- アプライ・プログラムがアプライ・コントロール・サーバーに接続できない。
- アプライ・プログラムは処理用のメモリーを割り振ることができない。

アプライ・プログラムの状況を照会する方法については、199 ページの『レプリケーション・プログラムの現在の状況のチェック (Linux、UNIX、Windows、z/OS)』を参照してください。

前提条件:

アプライ・プログラムを始動する前に、17 ページの『第 2 章 SQL レプリケーション用のサーバーの構成』の説明に従って、システムが正しくセットアップされていることを確認してください。

- 必要なレプリケーション・サーバーのすべてへの接続が構成されている。
- 正しい許可を受けている。
- 必要なアプライ修飾子に対してソースおよびコントロール・データを含むコントロール表が作成されている。
- レプリケーション・プログラムが構成済みである。
- z/OS の場合、必要なすべてのサーバーにアプライ・プログラムが手動でバインドされている。
- Linux、UNIX、および Windows 上で実行されるリモート・サーバーに対するエンド・ユーザー認証用のパスワード・ファイルが存在している。

また、以下の条件が満たされていることを確認してください。

- アプライ修飾子に対して少なくとも 1 つのアクティブなサブスクリプション・セットが存在し、サブスクリプション・セットが、1 つ、または複数の以下の項目を含んでいる必要があります。
 - サブスクリプション・セット・メンバー
 - SQL ステートメント
 - 手順

- コンデンスされているターゲット表はすべてターゲット・キーを必要とします。ターゲット・キーは、アプライ・プログラムが各アプライ・サイクル中に複製する変更のトラッキングに使用する主キーまたはユニーク索引のいずれかである、ユニーク列のセットです。(非コンデンス CCD 表には主キーまたはユニーク索引はありません。)

手順:

以下の方法のいずれかを使用して、アプライ・プログラムを始動します。

レプリケーション・センター

「アプライの開始」ウィンドウを使用します。詳しくは、レプリケーション・センターのヘルプを参照してください。

asnapply システム・コマンド (Linux、UNIX、Windows、z/OS)

詳細については、334 ページの『asnapply: アプライの始動』を参照してください。

Windows サービス (Windows)

詳細については、489 ページの『第 21 章 Windows Service Control Manager を使用して SQL レプリケーションのシステム・コマンドを発行する方法 (Windows)』を参照してください。

MVS コンソールまたは TSO (z/OS)

詳細については、483 ページの『第 20 章 SQL レプリケーション・プログラムの操作 (z/OS)』を参照してください。

どのプロシージャを使用してアプライ・プログラムを始動した場合にも始動パラメーターを設定する必要があります。以下のセクションでは、始動パラメーターについて説明し、各パラメーターで選択すべき値を推奨します。

- 151 ページの『apply_path (Linux、UNIX、Windows、z/OS)』
- 151 ページの『apply_qual (Linux、UNIX、Windows、z/OS)』
- 152 ページの『control_server (Linux、UNIX、Windows、z/OS)』
- 152 ページの『copyonce (Linux、UNIX、Windows、z/OS)』
- 153 ページの『db2_subsystem (z/OS)』
- 153 ページの『delay (Linux、UNIX、Windows、z/OS)』
- 153 ページの『errwait (Linux、UNIX、Windows、z/OS)』
- 154 ページの『inamsg (Linux、UNIX、Windows、z/OS)』
- 154 ページの『loadxit (Linux、UNIX、Windows、z/OS)』
- 154 ページの『logreuse (Linux、UNIX、Windows、z/OS)』
- 155 ページの『logstdout (Linux、UNIX、Windows、z/OS)』
- 155 ページの『notify (Linux、UNIX、Windows、z/OS)』
- 155 ページの『opt4one (Linux、UNIX、Windows、z/OS)』
- 156 ページの『pwdfile (Linux、UNIX、Windows)』
- 156 ページの『sleep (Linux、UNIX、Windows、z/OS)』
- 157 ページの『spillfile (Linux、UNIX、Windows、z/OS)』
- 157 ページの『sqlerrcontinue (Linux、UNIX、Windows、z/OS)』
- 158 ページの『term (Linux、UNIX、Windows、z/OS)』

apply_path (Linux、UNIX、Windows、z/OS)

デフォルト (Linux、UNIX、Windows、z/OS): `apply_path=current_directory`

デフォルト (Windows 上のサービス): `apply_path=sqllib¥bin`

アプライ・パスは、アプライ・プログラムがログおよび作業ファイルを保管するディレクトリーです。デフォルトのアプライ・パスはプログラムを始動するディレクトリーです。ログおよび作業ファイルを他の場所に保管するには、アプライ・パスを変更します (たとえば、AIX システムの場合は `/home/db2inst1/apply_files`)。アプライ・ログ・ファイルにアクセスするには、このディレクトリーに移動する必要があります。そのため、どのディレクトリーを選択したか覚えておいてください。z/OS オペレーティング・システムの場合、アプライ・パスを変更する方法に関する詳細は、SASNSAMP(ASNSTRA) ジョブを参照してください。

重要: 選択したディレクトリーに、アプライ・プログラムから使用される一時ファイルに十分なスペースがあることを確認してください。詳細については、10 ページの『アプライ・プログラム用予備ファイルのスペース所要量の計画』を参照してください。

Windows システムでのアプライ・プログラムのインスタンスの始動: レプリケーション・センターまたは `asnapply` コマンドを使用してアプライ・プログラムを始動する場合、もし 大文字と小文字の区別以外は同じ複数のアプライ修飾子を使用するのであれば、アプライ・パスを指定する必要があります。Windows システムのファイル名は大文字小文字が区別されません。たとえば、`APPLYQUAL1`、`ApplyQual1`、`applyqual1` という 3 つのアプライ修飾子があるとして、アプライ・プログラムの各インスタンスのログ・ファイルのファイル名が競合しないように、これらのアプライ・インスタンスはそれぞれ異なる `apply_path` を使用して開始する必要があります。

apply_qual (Linux、UNIX、Windows、z/OS)

処理対象のサブスクリプション・セットのアプライ修飾子を指定する必要があります。(アプライ修飾子は、サブスクリプション・セットを作成するときに定義します。) 1 つの始動コマンドで指定できるアプライ修飾子は 1 つだけです。

重要: アプライ修飾子の 大文字小文字は区別されます。入力された値が、サブスクリプション・セット (IBMSNAP_SUBS_SET) 表の `APPLY_QUAL` 列の値と一致する必要があります。

複数のアプライ修飾子を定義してあれば、アプライ・プログラムの別のインスタンスを始動できます。始動されたアプライ・プログラムの各インスタンスは、同じアプライ・コントロール・サーバーで表される異なるサブスクリプション・セットを処理します。たとえば、2 つのサブスクリプション・セットが定義され、それぞれのセットにユニークの修飾子、`APPLY1` および `APPLY2` があるとして、アプライ・プログラムの 2 つのインスタンス (各修飾子に対して 1 つずつ) を始動できます。各インスタンスは、`CNTRLSVR` と呼ばれるアプライ・コントロール・サーバーにあるコントロール表を使用します。アプライ・プログラムの各インスタンス

は、各自のサブスクリプション・セットを別個に処理するため、アプライ・プログラムの 1 つのインスタンスですべてのセットを処理する場合よりもパフォーマンスが向上します。

control_server (Linux、UNIX、Windows、z/OS)

デフォルト (Linux、UNIX、Windows): DB2DBDFT 環境変数の値 (使用可能な場合)

デフォルト (z/OS): なし

アプライ・コントロール・サーバーは、サブスクリプション定義およびアプライ・コントロール表が置かれているサーバーです。1 つのアプライ修飾子に対して、コントロール・サーバーを 1 つだけ指定してください。値が指定されない場合、アプライ・プログラムはデフォルト・サーバー上で始動されます。デフォルトはオペレーティング・システムによって異なります。

アプライ・プログラムがコントロール・サーバーに接続できない場合、アプライ・プログラムは終了します。アプライ・プログラムは他のサーバーに接続できなくても終了しません。この場合には、エラー・メッセージを発行して処理を続けます。

copyonce (Linux、UNIX、Windows、z/OS)

デフォルト: **copyonce=n**

copyonce パラメーターは、アプライ・プログラムのコピー・サイクルを決定します。

copyonce=y を指定して開始されたアプライ・プログラムは、適格なサブスクリプション・セットをそれぞれ一度だけ処理した後で終了します。この場合、以下の条件のいずれかが満たされた場合、サブスクリプション・セットは処理に適格となります。

- サブスクリプション・セットは相対タイミングを使用し、時間が経過し、サブスクリプション・セットはアクティブである。
- サブスクリプション・セットはイベント・ベースのタイミングを使用し、アクティブであり、イベントが発生したが、アプライ・プログラムはまだサブスクリプション・セットを処理していない。

copyonce=n を使用してアプライ・プログラムを始動する典型的な状況は、アプライ・プログラムの実行を継続し、適格なサブスクリプションの処理を続ける必要があるときです。

時々ネットワークに接続するようなダイヤルイン環境でアプライ・プログラムを実行するときには、**copyonce=n** ではなく、**copyonce=y** を使用してください。また、テスト環境でアプライ・プログラムを実行するときにも、**copyonce=y** を使用することが考えられます。

ヒント: サブスクリプション・セットが適格であり、レプリケーションできるデータがあるかぎり、アプライ・プログラムがサブスクリプション・セットの処理を何回

も行うようにしたい場合は、**copyonce=y** ではなく、**sleep=n** を使用してください。**Copyonce=y** は、複製するべきデータが残っている場合も、各セットを一度しか処理しません。

db2_subsystem (z/OS)

db2_subsystem パラメーターは、アプライが z/OS 上で実行している場合に、DB2 サブシステムの名前を指定します。入力する DB2 サブシステム名は最大 4 文字です。このパラメーターにはデフォルトはありません。このパラメーターは必須です。

delay (Linux、UNIX、Windows、z/OS)

デフォルト: **delay=6** 秒

delay パラメーターは、アプライ・プログラムが各アプライ・サイクルの最後に待機する秒数を設定します。

デフォルトでは、連続してレプリケーションを行う場合 (サブスクリプション・セットで **sleep=0** 分を使用する場合)、アプライ・プログラムはサブスクリプション・セットが正常に処理されてから、6 秒間待った後にサブスクリプション・セットを再試行します。複製すべきデータベース・アクティビティーがない場合は、ゼロ以外の値を使用して CPU サイクルを節約してください。待ち時間を少なくするには遅延値を小さくします。

注: **delay** パラメーターは、**copyonce** が指定される場合は無視されます。

errwait (Linux、UNIX、Windows、z/OS)

デフォルト: **errwait=300** 秒 (5 分)

errwait パラメーターは、サブスクリプション・サイクルが失敗した後に、アプライ・プログラムがサブスクリプション・セットを再試行するまでに待つ秒数を指定します。

デフォルトでは、アプライ・プログラムはサブスクリプション・サイクルが失敗した後 300 秒待機してから、サブスクリプション・セットを再試行します。テスト環境ではより小さい値を使用することが考えられます。最小値は 1 秒です。実稼働環境では、このパラメーターのデフォルトを変更する前に、トレードオフを考慮してください。

- 小さい値を使用すると、アプライ・プログラムがハード・エラーを繰り返し再試行することにより、CPU サイクルが無駄になります。たとえば、ターゲット表に問題があるときにアプライ・プログラムがサブスクリプション・セットの処理を繰り返し再試行した場合は、CPU サイクルを不必要に消費することになります。ログ・ファイルの中に、あるいはアプライ・プログラムが z/OS 上で実行されている場合は演算子コンソールに、大量のメッセージが送られることもあります。
- 大きな値を使用すると、アプライ・プログラムが一時的エラー条件を再試行するまで待つ必要がある場合の待ち時間が増えることになります。たとえば、迅速に

解決されるネットワーク・エラーを検出したときにもアプライ・プログラムは不必要に待つことになるため、 **errwait** パラメーターに大きな値を使用すると、待ち時間が増加します。

注: **errwait** パラメーターは、**copyonce** が指定される場合は無視されます。

inamsg (Linux、UNIX、Windows、z/OS)

デフォルト: **inamsg=y**

inamsg パラメーターは、アプライ・プログラムが非アクティブになったときにメッセージを発行するかどうかを指定します。

デフォルトでは、アプライ・プログラムは非アクティブになるとメッセージを発行します。サブスクリプション・セットの処理間のアプライ・プログラムの待ち時間が長くない場合は特にそうですが、メッセージはアプライ・ログ・ファイルのスペースを多量に消費します。このため、アプライ・プログラムが非アクティブになった場合のメッセージを発行させたくないことも考えられます。これらのメッセージをオフにするには、**inamsg=n** を使用します。

loadxit (Linux、UNIX、Windows、z/OS)

デフォルト: **loadxit=n**

loadxit パラメーターは、アプライ・プログラムが ASNLOAD 出口ルーチンを使用してターゲット表をリフレッシュするかどうかを指定します。

デフォルトでは、アプライ・プログラムはターゲット表をリフレッシュするために ASNLOAD 出口ルーチンを使用しません (**loadxit=n**)。アプライ・プログラムが ASNLOAD 出口ルーチン呼び出ししてターゲット表をリフレッシュするようにしたい場合は、**loadxit=y** を使用します。フル・リフレッシュ時にターゲット表に大量のデータをコピーするときには、ASNLOAD 出口を使用することを考えてみてください。アプライ・プログラムで ASNLOAD を使用するべき状況と、ASNLOAD の使用方法については、164 ページの『ASNLOAD 出口ルーチンを使ったターゲット表のリフレッシュ』を参照してください。

logreuse (Linux、UNIX、Windows、z/OS)

デフォルト: **logreuse=n**

アプライ・プログラムは、操作情報をログ・ファイルに保管します。
Linux、UNIX、および Windows の場合、ログ・ファイルの名前は *db2instance.control_server.apply_qualifier.APP.log* になります。z/OS オペレーティング・システムの場合のファイル名は DB2 インスタンス名が含まれないことを除いては同じです。

このパラメーターは、ログ・ファイルに追加するか、上書きするかを指定します。

デフォルトでは、アプライ・プログラムは開始されるたびに、ログ・ファイルにメッセージを追加します (**logreuse=n**)。アプライ・プログラムから発行されたメッセ

ージの履歴を保存する場合は、デフォルトのままにします。以下のような状況では、**logreuse=y** を使用して、アプライ・プログラムが開始時にログを削除し、再作成することができます。

- ログが大きくなったため、ログを消去してスペースを節約したい場合。
- ログに保管されている履歴が不要な場合。

logstdout (Linux、UNIX、Windows、z/OS)

デフォルト: **logstdout=n**

logstdout パラメーターは、**asnapply** コマンドを使用した場合にしか使用できません。**logstdout** はレプリケーション・センターからは使用できません。

logstdout パラメーターは、アプライ・プログラムが、完了メッセージ (ASN10251) をログ・ファイルと標準出力の両方に送信するかどうかを指定します。

デフォルトでは、アプライ・プログラムは完了メッセージを標準出力 (STDOUT) に送信しません。**logstdout=y** と指定すると、アプライ・プログラムは、ログ・ファイルと標準出力 (STDOUT) の両方に送信します。トラブルシューティングの場合や、アプライ・プログラムの稼働状態をモニターする場合は、メッセージを標準出力に送信するように選択できます。

notify (Linux、UNIX、Windows、z/OS)

デフォルト: **notify=n**

notify パラメーターは、アプライ・プログラムがサブスクリプションを処理した後に、ASNDONE 出口ルーチンに通知するかどうかを指定します。

デフォルトでは、アプライ・プログラムはサブスクリプションの処理が完了した後で ASNDONE 出口ルーチンに通知しません。**notify=y** が指定されると、アプライ・プログラムはサブスクリプション・サイクルが完了した後、アプライ・コントロール表の検査や、E メール・メッセージの送信など、追加処理を実行するために、ASNDONE を呼び出します。ASNDONE の詳細については、161 ページの『ASNDONE 出口ルーチンの変更 (Linux、UNIX、Windows、z/OS)』を参照してください。

opt4one (Linux、UNIX、Windows、z/OS)

デフォルト: **opt4one=n**

opt4one パラメーターは、アプライ・プログラムの処理が、1 つのサブスクリプション・セット用に最適化されるかどうかを指定します。

注: **opt4one** パラメーターは、**copyonce** が指定される場合は無視されます。

デフォルトでは、アプライ・プログラムは複数のサブスクリプション・セット用に最適化されています。アプライ・プログラムは各コピー・サイクルの最初にレプリケーション・コントロール表から情報を読み取ります。アプライ修飾子に対してサブスクリプション・セットが 1 つである場合は、アプライ・プログラムがメモリー・キャッシュ内にサブスクリプション・セットのメンバーおよび列の情報を入れてそれを再利用するように、**opt4one=y** を使用してアプライ・プログラムを始動し

てください。アプライ・プログラムを 1 つのサブスクリプション・セット用に最適化すると、アプライ・プログラムから使用される CPU が少なくなるため、スループット率が向上します。

重要: `opt4one=y` を使用した場合に、セットにメンバーを追加するか、その他の方法でセットを変更したときには、アプライ・プログラムがコントロール表の中の変更を入手できるように、アプライ・プログラムを停止してから再度開始する必要があります。

pwdfile (Linux、UNIX、Windows)

デフォルト: `pwdfile=asnpwd.aut`

データを複数のサーバーに分散する場合は、アプライ・プログラムがリモート・サーバー上のデータにアクセスできるように、暗号化されたパスワード・ファイルにユーザー ID とパスワードを保管できます。詳細については、26 ページの『レプリケーション用のユーザー ID およびパスワードの保管 (Linux、UNIX、Windows)』を参照してください。

sleep (Linux、UNIX、Windows、z/OS)

デフォルト: `sleep=y`

sleep パラメーターは、アプライ・プログラムが適格なサブスクリプション・セットを処理した後で、スリープ・モードで実行を続けるか、終了するかを指定します。

デフォルトでは、アプライ・プログラムは `sleep=y` で始動されます。アプライ・プログラムは適格なサブスクリプション・セットがあるかどうかをチェックします。適格なサブスクリプション・セットが検出されると、アプライ・プログラムはセットを処理し、別の適格なセットがないか探します。アプライ・プログラムは適格なセットが検出されると、その処理を続けます。適格なセットが検出されない場合、アプライ・プログラムはスリープ・モードで実行を続け、定期的に「ウェイクアップ」して、適格なサブスクリプション・セットがあるかどうかをチェックします。通常の場合は、長期間にわたって更新を適用するために、アプライ・プログラムをアクティブな状態で実行しておきたいため、この方法でアプライ・プログラムを始動します。

注: `sleep` パラメーターは、`copyonce` が指定される場合は無視されます。

`sleep=n` を指定して始動されたアプライ・プログラムは、適格なサブスクリプション・セットがあるかどうかをチェックし、それを処理します。アプライ・プログラムは適格なサブスクリプション・セットが見つからなくなるまでセットの処理を続け、複製するべきデータがなくなるまで適格なセットの処理を繰り返した後、終了します。`sleep=n` を使用する典型的な状況は、適格なサブスクリプション・セットが検出されたときにのみ アプライ・プログラムが実行され、その後にプログラムが終了される、モバイル環境またはテスト環境です。この場合は、アプライ・プログラムをスリープ・モードで待機させ、適格なセットがないかどうかをチェックするために定期的にウェイクアップさせることはしません。このような環境では、アプライ・プログラムを無期限に実行するのではなく、アプライ・プログラムをいつ実行するかをユーザーがコントロールできるようにします。

ヒント: 各サブスクリプション・セットを一度だけ処理する場合は、**sleep=n** ではなく、**copyonce=y** を使用してください。

spillfile (Linux、UNIX、Windows、z/OS)

デフォルト (Linux、UNIX、Windows): **spillfile=disk**

デフォルト (z/OS): **spillfile=MEM**

アプライ・プログラムはソース表からデータをリトリートし、アプライ・プログラムが実行されているシステム上の予備ファイルに入れます。

Linux、UNIX、および Windows オペレーティング・システムの場合、**spillfile** の唯一有効な設定は **disk** です。予備ファイルは、必ず **apply_path** で指定されたロケーションのディスクに置かれるからです。

USS を含む z/OS オペレーティング・システムの場合は、予備ファイルはデフォルトでメモリー内に保管されます。ディスク上に予備ファイルを保管するように指定する場合、アプライ・プログラムは **ASNASPL DD** ステートメント上の指定を使用して、予備ファイルを割り振ります。ASNASPL DD ステートメントが指定されない場合、VIO が使用されます。

sqlerrcontinue (Linux、UNIX、Windows、z/OS)

デフォルト: **sqlerrcontinue=n**

sqlerrcontinue パラメーターは、アプライ・プログラムが特定の SQL エラーに対処する方法を指定します。

デフォルトでは、アプライ・プログラムは SQL エラーを検出すると、そのサブスクリプション・セットの処理を停止し、エラー・メッセージを生成します。通常の場合、実稼働環境ではデフォルトを使用します。

テスト環境の場合は、ターゲット表にデータを挿入するときに、特定の SQL エラーが発生することが予想されます。これらのエラーはユーザーにとっては許容できる場合もありますが、エラーにより現行のサブスクリプション・サイクルは停止します。このような状況では、アプライ・プログラムがエラーを無視し、そのサイクルで複製したデータをロールバックしないように、**sqlerrcontinue=y** を使用してアプライ・プログラムを始動できます。アプライ・プログラムはターゲット表へのデータ挿入時に SQL エラーを受け取ると、**apply_qualifier.sqls** ファイルの中の値をチェックします。一致するものがあると、エラーに関する詳細をエラー・ファイル、**apply_qualifier.err** に書き込み、処理を続行します。アプライ・プログラムは **apply_qualifier.sqls** ファイルにリストされていない SQL エラーを検出すると、セットの処理を停止し、次のセットに進みます。

sqlerrcontinue=y オプションを使用してアプライ・プログラムを始動する前に、**apply_qualifier.sqls** ファイルを作成し、アプライ・プログラムの呼び出しに使用したディレクトリーにこのファイルを保管する必要があります。このファイルには、最大 20 の 5 バイトの値を続けてリストしてください。アプライ・プログラムの実行中にこのファイルの内容を変更したときには、アプライ・プログラムが新しい値を認識できるように、アプライ・プログラムを停止してから再度開始してください。

例: ターゲット表で次のエラーを受け取ったときにアプライ・プログラムがサブスクリプション・セットの処理を続けるようにするとします (sqlstate/code)。

42704/-803

重複索引違反

以下の SQL 状態を含む SQL 状態ファイルを作成します。

42704

ターゲット表の更新時にこの SQL 状態が戻された場合、アプライ・プログラムはセット内のその他のターゲット表に変更を適用し、エラーおよびリジェクトされた行の両方を示すエラー・ファイルを作成します。

ヒント: アプライ・トレール (IBMSNAP_APPLYTRAIL) 表の STATUS 列をチェックします。16 という値は、アプライ・プログラムがサブスクリプション・セットを正常に処理したが、*apply_qualifier.sqs* ファイルで定義された、許容されるなんらかのエラーが発生したことを意味します。

term (Linux, UNIX, Windows, z/OS)

デフォルト: **term=y**

term パラメーターは、DB2 の状況が、アプライ・プログラムの稼働にどのような影響を与えるかを決定します。

デフォルトでは、アプライ・プログラムは DB2 が終了すると終了します。

DB2 がアクティブでないときに、アプライ・プログラムが DB2 の始動を待つようにするには、**term=n** を使用します。z/OS オペレーティング・システムの場合、DB2 が静止したときにアプライ・プログラムがアクティブであると、アプライ・プログラムはアクティブなまま残り、DB2 が開始されるまで再接続しません。Linux、UNIX、および Windows オペレーティング・システムの場合、DB2 が静止したときにアプライ・プログラムがアクティブであると、アプライ・プログラムはアクティブなまま残り、DB2 が静止モードを終了するまで再接続しません。

注: **term** パラメーターは、**copyonce** が指定される場合は無視されます。

trlreuse (Linux, UNIX, Windows, z/OS)

デフォルト: **trlreuse=n**

trlreuse パラメーターは、アプライ・プログラムの始動時に、アプライ・トレール (IBMSNAP_APPLYTRAIL) 表を再利用する (追加する) か、上書きするかを指定します。

デフォルトでは、アプライ・プログラムは開始されると、アプライ・トレール表に項目を追加します。この表は、アプライ・コントロール・サーバーにあるすべてのアプライ・インスタンスの操作の履歴を保持します。この表は、診断およびパフォーマンス統計のリポジトリです。更新の履歴を保持する場合は、デフォルトを使用します。以下のような状況では、アプライ・プログラムの始動時に、アプライ・トレール表に追加するのではなく、表を空にすることができます (**trlreuse=y**)。

- アプライ・トレール表が大きくなったため、消去してスペースを節約したい場合。
- 表に保管されている履歴が不必要な場合。

ヒント: `trlreuse=y` を使用する代わりに、アプライ・プログラムがサブスクリプション・セットを正常に処理した後で (`status=0`)、SQL 処理を使用してアプライ・トレール表から行を削除できます。

アプライ・プログラムの始動 (OS/400)

アプライ・プログラムのインスタンスを始動して、ターゲットへのデータのアプライを開始できます。

アプライ・プログラムは始動された後、以下の条件のいずれか発生するまで、連続的に実行されます。

- ユーザーが `COPYONCE(*YES)` 始動パラメーターを使用してプログラムを始動する。
- ユーザーが `ALWINACT(*NO)` を指定し、処理するべきデータがない。
- ユーザーがレプリケーション・センターまたはコマンドを使用してアプライ・プログラムを停止する。
- アプライ・プログラムがアプライ・コントロール・サーバーに接続できない。
- アプライ・プログラムは処理用のメモリーを割り振ることができない。

前提条件:

アプライ・プログラムを始動する前に、システムが正しくセットアップされていることを確認してください。

- すべてのレプリケーション・サーバーへの接続が構成されている。
- 正しい許可を受けている。
- コントロール表が作成されている。
- レプリケーション・プログラムが構成済みである。

また、以下の条件が満たされていることを確認してください。

- アプライ修飾子に対して少なくとも 1 つのアクティブなサブスクリプション・セットが存在し、サブスクリプション・セットが、1 つ、または複数の以下の項目を含んでいる必要があります。
 - サブスクリプション・セット・メンバー
 - SQL ステートメント
 - 手順
- コンデンスされているターゲット表はすべてターゲット・キーを必要とします。ターゲット・キーは、アプライ・プログラムが各アプライ・サイクル中に複製する変更のトラッキングに使用する主キーまたはユニーク索引のいずれかである、ユニーク列のセットです。(非コンデンス CCD 表には主キーまたはユニーク索引はありません。)

手順:

以下の方法のいずれかを使用して、アプライ・プログラムを始動します。

レプリケーション・センター

「アプライの開始」ウィンドウを使用します。詳しくは、レプリケーション・センターのヘルプを参照してください。

STRDPRAPY システム・コマンド

詳細については、460 ページの『STRDPRAPY: アプライ・プログラムの始動 (OS/400)』を参照してください。

アプライ・プログラムを始動するときには、稼働パラメーターにこれらのデフォルト設定を使用できます。

表9. アプライ稼働パラメーターのデフォルト設定 (OS/400)

稼働パラメーター	(*value) の記述
USER (*CURRENT)	システムにサインオンしたユーザー。
JOB (*LIBL/QZSNDPR)	製品ライブラリー名/ジョブ記述。
APYQUAL (*USER)	現行ユーザー名 (上記から)。
CTLSVR (*LOCAL)	ローカル RDB サーバー名。
TRACE (*NONE)	トレースを生成しない。
FULLREFPGM (*NONE)	ASNLOAD 出口ルーチンを実行しない。
SUBNFYPGM (*NONE)	ASNDONE 出口ルーチンを実行しない。
INACTMSG (*YES)	アプライ・プログラムは非アクティブ期間を開始すると、プログラムが非アクティブになる期間を示すメッセージ ASN1044 を生成する。
ALWINACT (*YES)	処理するものがない場合はスリープする。
DELAY (6)	再度処理する前に、アプライ・サイクルの後に 6 秒待機する。
RTYWAIT (300)	失敗した操作を再始動する前に 300 秒待機する。
COPYONCE (*NO)	1 つのコピー・サイクルを完了した後で終了せずに処理を続行する。
TRLREUSE (*NO)	アプライ・プログラムの始動時にアプライ・トレール (IBMSNAP_APPLYTRAIL) 表を空にしない。
OPTSNGSET (*NO)	アプライ・プログラムのパフォーマンスを 1 つのサブスクリプション・セットの処理用に最適化しない。

構文図を含めて、これらの稼働パラメーターの詳細は、460 ページの『STRDPRAPY: アプライ・プログラムの始動 (OS/400)』を参照してください。

アプライ・パラメーター表での稼働パラメーターの変更 (Linux、UNIX、Windows、z/OS)

アプライ・パラメーター (IBMSNAP_APPPARMS) 表には、アプライ・プログラム用の稼働パラメーターが含まれています。始動時に、アプライ・プログラムはこの表から、デフォルトに設定されている稼働中の動作に関する値を使用します。ただし、始動パラメーターを使用して新しい値が指定された場合を除きます。

各 *apply_qualifier* には 1 つの行しか入れられません。デフォルト値のいずれかを変更する場合は、行を挿入する代わりに列を更新することができます。この行を削除

しても、出荷時のデフォルトが始動パラメーターによってオーバーライドされないかぎり、アプライ・プログラムはそれらのデフォルトを使用しても始動します。

アプライ・プログラムはこの表を始動時にしか読み取らないため、新しい設定でアプライ・プログラムを実行したい場合は、アプライ・プログラムの停止と始動を行う必要があります。アプライ・プログラムの実行中にアプライ・パラメーター表を変更しても、アプライ・プログラムの動作は変更されません。この表の列についての記述は、501 ページの『第 24 章 SQL レプリケーションの表構造』を参照してください。

アプライ・プログラムの停止

ユーザーはアプライ・プログラムのインスタンスを停止できます。ユーザーがアプライ・プログラムを停止すると、アプライ・プログラムはターゲット表にデータをコピーしなくなり、次にプログラムが開始されたときに正しく開始されるように、コントロール表の中の情報を更新します。

前提条件:

アプライ・プログラムのインスタンスが開始されている必要があります。

手順:

アプライ・プログラムのインスタンスを停止するには、以下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

「アプライの停止」ウィンドウを使用します。詳しくは、レプリケーション・センターのヘルプを参照してください。

asnacmd stop システム・コマンド (Linux、UNIX、Windows、z/OS)

詳細については、330 ページの『asnacmd: アプライの操作』を参照してください。

ENDDPRAPY システム・コマンド (OS/400)

詳細については、431 ページの『ENDDPRAPY: アプライ・プログラムの停止 (OS/400)』を参照してください。

ASNDONE 出口ルーチンの変更 (Linux、UNIX、Windows、z/OS)

このセクションでは、Linux、UNIX、Windows、および z/OS オペレーティング・システムでの ASNDONE 出口ルーチンのカスタマイズ方法を説明します。

notify=y パラメーターを使用してアプライ・プログラムを始動すると、アプライ・プログラムはサブスクリプションの処理が成功したかどうかに関係なく、サブスクリプションの処理が終了した後、ASNDONE 出口ルーチンを呼び出します。以下のリストは、レプリケーション環境で使用するために ASNDONE 出口ルーチンをどのように変更できるかの例を示しています。

- トランザクションがリジェクトされたことが判明した場合は、この出口ルーチンを使用して、リジェクトされたトランザクションを UOW 表で確認し、その後のアクション (たとえば、E メールをレプリケーション演算子に自動的に送信する、メッセージを発行する、またはアラートを生成する) を開始します。

- この出口ルーチンを使用して、失敗したサブスクリプション・セットが訂正されるまでアプライ・プログラムが再試行を続けないように、失敗したサブスクリプション・セットを非活動化します。失敗したサブスクリプション・セットを検出するには、アプライ・トレール (IBMSNAP_APPLYTRAIL) 表で STATUS= -1 を探すように出口ルーチンを変更します。サブスクリプション・セットを非活動化するには、サブスクリプション・セット (IBMSNAP_SUBS_SET) 表に ACTIVATE=0 を設定するように出口ルーチンを構成します。
- データが各 サブスクリプション・セットに適用された後で出口ルーチンを使用してデータを操作します。(この代わりに、SQL ステートメントまたはストアド・プロシージャを使用して、アプライ・プログラムによる特定の サブスクリプション・セットの処理の前または後で実行されるランタイム処理ステートメントを定義することもできます)

手順:

サンプルの ASNDONE 出口ルーチンの変更したバージョンを使用するには、以下のようになります。

1. 要件に合うように、ASNDONE ルーチンを調整します。

Linux、UNIX、Windows の場合: この出口ルーチンの変更方法については、サンプル・プログラム (%sqllib%samples%repl%asndone.smp) の PROLOG セクションを参照してください。

z/OS の場合: サンプル・プログラム SASNSAMP(ASNDONE) の PROLOG セクションを参照してください。

2. プログラムをコンパイル、リンク、バインドし、実行可能ファイルを適切なディレクトリーに入れます。
3. **notify=y** パラメーターを使用してアプライ・プログラムを始動し、ASNDONE 出口ルーチンを呼び出します。

ASNDONE 出口ルーチンの変更 (OS/400)

このセクションでは、OS/400 環境での ASNDONE 出口ルーチンのカスタマイズ方法を説明します。

SUBNFYPGM パラメーターを ASNDONE 出口ルーチンの名前に設定してアプライ・プログラムを始動すると、アプライ・プログラムは、サブスクリプションの処理が成功したかどうかに関係なく、サブスクリプションの処理が終了した後、ASNDONE 出口ルーチンを呼び出します。以下のリストは、レプリケーション環境で使用するために ASNDONE 出口ルーチンをどのように変更できるかの例を示しています。

- トランザクションがリジェクトされたことが判明した場合は、この出口ルーチンを使用して、リジェクトされたトランザクションを UOW 表で確認し、その後のアクション (たとえば、E メールをレプリケーション演算子に自動的に送信する、メッセージを発行する、またはアラートを生成する) を開始します。
- この出口ルーチンを使用して、失敗したサブスクリプション・セットが訂正されるまでアプライ・プログラムが再試行を続けないように、失敗したサブスクリプション・セットを非活動化します。失敗したサブスクリプション・セットを検出するには、アプライ・トレール (IBMSNAP_APPLYTRAIL) 表で STATUS= -1 を

探すように出力ルーチンを変更します。サブスクリプション・セットを非活動化するには、サブスクリプション・セット (IBMSNAP_SUBS_SET) 表に ACTIVATE=0 を設定するように出力ルーチンを構成します。

- データが各 サブスクリプション・セットに適用された後で出力ルーチンを使用してデータを操作します。(この代わりに、SQL ステートメントまたはストアード・プロシージャを使用して、アプライ・プログラムによる特定の サブスクリプション・セットの処理の前または後で実行されるランタイム処理ステートメントを定義することもできます。詳細については、120 ページの『ストアード・プロシージャまたは SQL ステートメントを使用したデータ拡張』を参照してください。)

手順:

サンプルの ASNDONE 出力ルーチンの変更したバージョンを使用するには、以下のようになります。

1. サイトの要件に合うように、ASNDONE 出力ルーチンを調整します。

以下の表は、C、COBOL、および RPG 言語のこのルーチンのソース・コードがどこにあるかを示しています。

コンパイラ言語	ライブラリー名	ソース・ファイル名	メンバー名
C	QDP4	QCSRC	ASNDONE
COBOL	QDP4	QCBLLSRC	ASNDONE
RPG	QDP4	QRPGLESRC	ASNDONE

プログラムの変更時に、以下の活動化グループに関連することを考慮してください。

新しい活動化グループを使用して実行するようにプログラムが作成されている場合: アプライ・プログラムおよび ASNLOAD プログラムは、リレーショナル・データベースの接続およびオープン・カーソルなどの SQL リソースを共有しません。OS/400 オペレーティング・システムにある活動化処理コードは、コントロールがアプライ・プログラムに戻される前に、ASNLOAD プログラムによって割り振られたリソースをすべて解放します。アプライ・プログラムが ASNLOAD プログラムを呼び出すたびに、追加のリソースが使用されます。

このプログラムが呼び出し側の活動化グループで実行するために作成される場合: アプライ・プログラムと SQL リソースを共有します。アプライ・プログラムへの影響が最小限で済むようプログラムを設計します。たとえば、現行のリレーショナル・データベース接続を変更すると、予期しないアプライ・プログラムの処理を引き起こす可能性があります。

このプログラムが名前付き活動化グループで実行するために作成される場合: アプライ・プログラムとの間でリソースを共有しません。名前付き活動化グループを使用すると、ASNLOAD プログラムが呼び出されるたびに、活動化グループのオーバーヘッドは回避されます。ランタイムのデータ構造および SQL リソースは、呼び出し間で共有されます。アプライ・プログラムが終了するまで、アプリケーションの終結処理は実行されません。したがって、サブスクリプション通知プログラムは、コントロールがアプライ・プログラムに戻される時、ソース

表、ターゲット表、またはコントロール表をロックすることにより、アプライ・プログラムでロックの競合が生じることがないように設計します。

2. プログラムをコンパイル、リンク、バインドし、実行可能ファイルを適切なディレクトリーに入れます。
3. アプライ・プログラムを始動し、**STRDPRAPY** コマンドのパラメーター **SUBNFYPGM** を使用して、ASNDONE プログラムの名前を指定します。たとえば、プログラムの名前が ASNDONE_1 でライブラリー APPLIB にある場合には、以下のコマンドを使用します。

```
SUBNFYPGM(APPLIB/ASNDONE_1)
```

ASNLOAD 出口ルーチンを使ったターゲット表のリフレッシュ

デフォルトでは、アプライ・プログラムはサブスクリプション・セット内の各ターゲット表のリフレッシュを実行するときに ASNLOAD 出口ルーチンを使用しません。アプライ・プログラムはソース表に対して全選択を行い、アプライ・プログラムが実行されているサーバー上の予備ファイルにデータを持ち込み、INSERT ステートメントを使用してターゲット表にデータを取り込みます。ソース表が大きい場合には、フル・リフレッシュ時にターゲットにデータを効率的にコピーするために、代わりに ASNLOAD 出口ルーチンを使用できます。

ASNLOAD 出口ルーチンは、Linux、UNIX、Windows、および z/OS オペレーティング・システムでは、ソース形式とコンパイル済みフォーマットの両方で、サンプルの出口ルーチンとして提供されます。OS/400 オペレーティング・システムでは、ソース形式のみで提供されます。サンプルの出口ルーチンは DB2 プラットフォームによってそれぞれ異なり、そのプラットフォームで提供されるユーティリティー・オプションを利用します。

アプライ・プログラムが ASNLOAD 出口ルーチンを呼び出したときにエラーが発生すると、アプライ・プログラムがメッセージを発行し、現行のサブスクリプション・セットの処理を停止し、次のサブスクリプション・セットを処理します。

前提条件:

ASNLOAD 出口ルーチンを使用する前に、以下の前提条件が満たされていることを確認してください。

- ターゲット表の列は、ソース表の順序とデータ・タイプと一致する。
- ターゲット表は、レプリケーション・マッピングの一部である列のみを含む。

制約事項:

- z/OS オペレーティング・システムでは、ASNLOAD 出口ルーチンは、DB2 V7 (またはそれ以降) Utilities Suite で使用可能なクロスローダー関数を呼び出しません。
- Linux、UNIX、および Windows オペレーティング・システムでは、ASNLOAD 出口ルーチンは、カーソル関数からのロードを含む、エクスポート、インポート、およびロード・ユーティリティーを処理します。カーソルからのロードは、サブスクリプション・セット・メンバーのソースが実際にはニックネームである場合、またはターゲット・データベースがソース・データベースと同じである場

合に ASNLOAD 出口が使用するデフォルト・オプションです。また、以下のアクションが実行された場合は、DB2 データ・ソースでカーソルからのロード関数が使用されることもあります。

- ソース表のニックネームがターゲット・データベースで作成された場合。
- サブスクリプション・セット・メンバーの IBMSNAP_SUBS_MEMBR 表の列が、カーソルからのロード関数が使用されることを示すように設定された場合。これらの列の値は、レプリケーション・センターを使って設定できます。
 - LOADX_TYPE が、使用されるカーソルからのロード関数を示すように設定された場合。
 - ソース表を含むサブスクリプション・セット・メンバーの IBMSNAP_SUBS_MEMBR 表にある LOADX_SRC_N_OWNER 列および LOADX_SRC_N_TABLE 列で、ソース・ニックネーム情報が指定された場合。

以下のセクションでは、さまざまなオペレーティング・システム上で ASNLOAD 出口ルーチンを使用する方法を説明します。

- 『ASNLOAD 出口ルーチンを使ったターゲット表のリフレッシュ (Linux、UNIX、Windows)』
- 167 ページの 『ASNLOAD 出口ルーチンを使ったターゲット表のリフレッシュ (z/OS)』
- 170 ページの 『ASNLOAD 出口ルーチンを使ったターゲット表のリフレッシュ (OS/400)』

ASNLOAD 出口ルーチンを使ったターゲット表のリフレッシュ (Linux、UNIX、Windows)

ASNLOAD 出口ルーチンは、DB2 の IMPORT ユーティリティまたは DB2 の LOAD ユーティリティと一緒に DB2 の EXPORT ユーティリティを使用する、または新しい LOAD FROM CURSOR ユーティリティを使用するなど、多数のユーティリティ・オプションを提供します。ユーザーがサンプルの出口ルーチンを呼び出すと、デフォルトでは、ソース・サーバー、ターゲット・サーバー、およびランタイム環境に基づいて、使用されるユーティリティが選択されます。

コンパイル済みの出口ルーチンを使用したり、レプリケーション構成をカスタマイズして動作を構成したり、出口コード自体を変更することができます。レプリケーション構成は、サブスクリプション・メンバー (IBMSNAP_SUBS_MEMBR) 表の列を更新するか、サンプルの構成ファイル (asnload.ini) を更新することによりカスタマイズできます。

手順:

提供されたままの ASNLOAD ルーチンを使用する場合は、**loadxit=y** パラメーターを使用してアプライ・プログラムを始動します。

サンプルの ASNLOAD 出口ルーチンの変更したバージョンを使用するには、以下のようになります。

1. サイトの要件に合うように、ASNLOAD ルーチンを調整します。この出口ルーチンの変更方法については、サンプル・プログラム (`¥sqllib¥samples¥repl¥asnload.smp`) の PROLOG セクションを参照してください。

重要: サンプルのソースは、asnload.ini ファイルからユーザー ID とパスワードの組み合わせを使用します。asnload.ini ファイルに特定のサーバーのユーザー ID とパスワードがない場合、または asnload.ini ファイルを使用できない場合は、user/using 句を使用しないで接続が行われます。

2. プログラムをコンパイル、リンク、バインドし、実行可能ファイルを適切なディレクトリーに入れます。
3. ユーザーが指定したコードを使用して取り込みが行われるメンバーについては、LOADX_TYPE を 2 に設定してください。詳細については、168 ページの『ASNLOAD 出口の動作のカスタマイズ (Linux、UNIX、Windows、z/OS)』を参照してください。
4. `loadxit=y` パラメーターを使用してアプライ・プログラムを始動し、ASNLOAD 出口ルーチンを呼び出します。

ASNLOAD 出口ルーチンへの入力を構成するには、169 ページの『ASNLOAD の構成ファイルの使用 (Linux、UNIX、Windows)』を参照してください。

ASNLOAD 出口ルーチンから生成されるファイル:

これらのファイルは、ASNLOAD 出口ルーチンを呼び出したアプライ・インスタンスの `apply_path` ディレクトリーに保管されます。

- `asnload apply_qualifier.trc`

トレースがオンの場合、このファイルはトレース情報を保持します。ASNLOAD 出口ルーチンがこのファイルを作成します。ファイルが存在する場合は、情報がファイルに追加されます。

- `asnload apply_qualifier.msg`

このファイルは、ロード統計を含めて、一般出口障害、警告メッセージおよび情報メッセージを保持します。ASNLOAD 出口ルーチンがこのファイルを作成します。ファイルが存在する場合は、情報がファイルに追加されます。

- `asnaEXPT apply_qualifier.msg`

このファイルは、DB2 の EXPORT ユーティリティーが発行したエラー・メッセージ、警告メッセージ、または通知メッセージを保持します。ASNLOAD 出口ルーチンがこのファイルを作成します。ファイルが存在する場合は、情報がファイルに追加されます。

- `asnaIMPT apply_qualifier.msg`

このファイルは、DB2 の IMPORT ユーティリティーが発行したエラー・メッセージ、警告メッセージ、または通知メッセージを保持します。ASNLOAD 出口ルーチンがこのファイルを作成します。ファイルが存在する場合は、情報がファイルに追加されます。

- `asnaLOAD apply_qualifier.msg`

このファイルは、DB2 の LOAD ユーティリティーが発行したエラー・メッセージ、警告メッセージ、または通知メッセージを保持します。ASNLOAD 出口ルーチンがこのファイルを作成します。ファイルが存在する場合は、情報がファイルに追加されます。

ASNLOAD 出口ルーチンを使ったターゲット表のリフレッシュ (z/OS)

ASNLOAD 出口ルーチンは、カーソルに基づくフェッチを行い、ソースからデータを入手し、ターゲットにデータをロードする、LOAD ユーティリティーを呼び出します。ASNLOAD 出口ルーチンは LOG NO を指定して LOAD を使用し、表スペースの COPYPEND 状況をリセットします。サンプルの ASNLOAD ソース・コードに変更を加えて、ロード・オプションを変更できます。ソースは、2 つのヘッダー・ファイルと 3 つの C++ プログラムから構成されています。

手順:

提供されたままの ASNLOAD ルーチンを使用する場合は、**loadxit=y** パラメーターを使用してアプライ・プログラムを始動します。

サンプルの ASNLOAD ルーチンの変更したバージョンを使用するには、以下のようになります。

1. サイトの要件に合うように、ASNLOAD ルーチンを調整します。この出口ルーチンの変更方法については、サンプル・プログラム SASNSAMP(ASNLOAD) の PROLOG セクションを参照してください。
2. プログラムをコンパイル、リンク、バインドし、実行可能ファイルを適切なディレクトリーに入れます。アプライ・プランに ASNLOAD パッケージを追加します。
 - a. 以下の条件が満たされていることを確認してください。
 - ユーティリティー・サポートを含む DB2 Universal Database for z/OS および DB2 Universal Database for OS/390 バージョン 7 以上がインストールされている。
 - DSNUTILS ストアード・プロシージャが実行されている。DSNUTILS は WLM 環境で実行する必要があります。DSNUTILS の使用方法の詳細は、『DB2 Universal Database (OS/390 版および z/OS 版)ユーティリティー・ガイドおよび解説書 (SC88-8773)』を参照してください。
 - b. サンプル zmak ファイル (SASNSAMP(ASNCMPLD)) を使用して、USS の ASNLOAD ユーザー出口プログラムをコンパイルし、リンク・エディットします。
 - c. ASNLOAD 出口ルーチンを DSNUTILS およびアプライ・パッケージとバインドします。

サンプルの ASNLOAD は LOG NO を指定してロードを実行した後、表スペースを修理して nocopypend を設定します。表スペースのバックアップは行いません。デフォルトでは、ASNLOAD は、APPLY_PATH=// オプションが指定された **apply_path** パラメーターがそのアプライ・インスタンスに指定されていない限り、アプライ・プログラムのインスタンスを実行中のユーザー ID の下に 2 つの一時ファイルを作成します。この場合、2 つの一時ファ

イルは、APPLY_PATH で指定された上位修飾子の下に作成されます。また、ロードに関するすべての情報を含むファイルも作成します。

3. ユーザーが指定したコードを使用して取り込みが行われるメンバーについては、`loadx_type = 2` を設定します。
4. `loadxit=y` パラメーターを使用してアプライ・プログラムを始動し、ASNLOAD 出口ルーチン呼び出しします。

ASNLOAD 出口ルーチンへの入力を構成するには、169 ページの『ASNLOAD の構成ファイルの使用 (Linux、UNIX、Windows)』を参照してください。

ASNLOAD 出口ルーチンから生成されるファイル: これらのファイルは、ASNLOAD 出口ルーチン呼び出ししたアプライ・インスタンスの `apply_path` ディレクトリーまたは HLQ に保管されます。

- `userid.apply_qual.LOADMSG`

このファイルは、ロード統計を含めて、障害、警告メッセージおよび情報メッセージを保持します。ASNLOAD 出口ルーチンがこのファイルを作成します。ファイルが存在する場合は、情報がファイルに追加されます。

- `userid.apply_qual.LOADTRC`

トレースがオンの場合、このファイルはトレース情報を保持します。ASNLOAD 出口ルーチンがこのファイルを作成します。ファイルが存在する場合は、情報がファイルに追加されます。

ASNLOAD 出口の動作のカスタマイズ (Linux、UNIX、Windows、z/OS)

レプリケーション構成をカスタマイズして ASNLOAD 出口ルーチンの動作を構成したり、出口コード自体をカスタマイズすることができます。レプリケーション構成は、サブスクリプション・メンバー (IBMSNAP_SUBS_MEMBR) 表の列を更新するか、構成ファイルを更新することによりカスタマイズできます。

サブスクリプション・メンバー表の使用

サブスクリプション・メンバー (IBMSNAP_SUBS_MEMBR) 表の列を使用して、ASNLOAD 出口ルーチンの動作をカスタマイズできます。LOADX_TYPE を使用してロード・オプションを選択します。LOADX_TYPE の有効な値は以下のとおりです。

NULL (デフォルト)

z/OS の場合: カーソルからのロード関数を使用します。

Linux、UNIX、Windows の場合: ASNLOAD 出口ルーチンが適切なユーティリティーを決定します (オプション 3、4、または 5)。

- 1 このメンバーについては ASNLOAD 出口ルーチン呼び出ししません。
このメンバーについて ASNLOAD 出口ルーチン呼び出たくない場合は、LOADX_TYPE を 1 に設定します。
- 2 ユーザー自身の出口ロジックを提供します。
ユーザー自身のロジックを ASNLOAD 出口ルーチンで提供する場合は、ASNLOAD 出口ルーチンにより取り込みを行うサブスクリプション・セッ

トのメンバーについて LOADX_TYPE を 2 に設定してください。
LOADX_TYPE を 2 に設定しているのに出口ロジックを提供しない場合、
出口は失敗します。

3 カーソルからのロード関数を使用します。

Linux、UNIX、および Windows オペレーティング・システムの場合、カーソルからのロード関数はターゲット表 (ローカル・データベースになければならない) にロードするデータをフェッチするために SELECT ステートメントを必要とします。このステートメントは、DB2 表またはニックネームのどちらかを参照でき、セットアップは以下のようになります。

IBM 以外のソースから、登録済みソース・ニックネームがターゲット・データベースとは別のデータベース上にある DB2 表に複製している場合、または DB2 表から別の DB2 表へ複製していてソース・データベースがターゲット・データベースと異なる場合、以下のステップを実行する必要があります。

1. ターゲット・サーバー・データベースのソース表にニックネームを作成します。
2. サブスクリプション・メンバー (IBMSNAP_SUBS_MEMBR) 表のニックネーム所有者およびテーブル名列 (LOADX_SRC_N_OWNER および LOADX_SRC_N_TABLE) を更新します。

DB2 表から別の DB2 表へ複製していて、ソース・データベースとターゲット・データベースが同じである場合、または IBM 以外のソースから、登録済みソース・ニックネームがターゲット・データベースと同じデータベース上にある DB2 表に複製している場合、カーソルからのロード関数を使用するための追加のアクションは必要ありません。

4 (Linux、UNIX、および Windows のみ)

EXPORT ユーティリティと LOAD ユーティリティの組み合わせを使用します。

5 (Linux、UNIX、および Windows のみ)

EXPORT ユーティリティと IMPORT ユーティリティの組み合わせを使用します。

ASNLOAD の構成ファイルの使用 (Linux、UNIX、Windows)

ASNLOAD 出口ルーチンへの入力を構成するために、オプションの構成ファイルを使用できます。このファイルは ASNLOAD の実行に必須ではありません。

Linux、UNIX、および Windows オペレーティング・システムの場合、構成ファイルのファイル名は asnload.ini にする必要があります。ASNLOAD 出口ルーチンは、**apply_path** ディレクトリーでこのオプションの構成ファイルを探します。サンプル・ファイル `sqllib/samples/repl/asnload.ini` を編集し、ASNLOAD 出口ルーチンを呼び出したアプライ・プログラムのインスタンスの **apply_path** ディレクトリーに保管してください。

ASNLOAD 出口ルーチンを使ったターゲット表のリフレッシュ (OS/400)

アプライ・プログラムの代わりに出口ルーチンを使用して、もっと効率よくフル・リフレッシュを行います。たとえば、ソース表の各列および各行をターゲット表にコピーしている場合、分散データ管理 (DDM) ファイルおよびファイルのコピー (CPYF) CL コマンドを使用する出口ルーチンのフル・リフレッシュを、ソース表からターゲット表へとファイル全体をコピーするように設計することができます。

手順:

ASNLOAD 出口ルーチンを使用してターゲット表をリフレッシュするには、FULLREFPGM パラメーターを使用してアプライ・プログラムを始動します。

サンプルの ASNLOAD ルーチンの変更したバージョンを使用するには、以下のようになります。

1. サイトの要件に合うように、ASNLOAD 出口ルーチンを調整します。この出口ルーチンの変更方法については、サンプル・プログラムの PROLOG セクションを参照してください。C、COBOL、および RPG 言語のソースを使用することができます。

コンパイラ言語	ライブラリー名	ソース・ファイル名	メンバー名
C	QDP4	QCSRC	ASNLOAD
COBOL	QDP4	QCBLLSRC	ASNLOAD
RPG	QDP4	QRPGLSRC	ASNLOAD

2. プログラムをコンパイル、リンク、バインドし、実行可能ファイルを適切なディレクトリーに入れます。

アプライ・プログラムを妨害しないようにするため、出口ルーチンが (呼び出し側の活動化グループではなく) 新しい活動化グループを使うようにコンパイルします。

名前付き活動化グループまたは新しい活動化グループを使って出口ルーチンをコンパイルすることができます。パフォーマンスを向上させるには、名前付き活動化グループを使用します。名前付き活動化グループを使用すると、出口ルーチンは必要に応じて変更をコミットまたはロールバックする必要があります。アプライ・プログラムは、(終了するまで) 変更をコミットまたはロールバックすることはありません。出口ルーチンは、完了時に変更を明示的にコミットするか、または変更を暗黙的にコミットするためにコンパイルされる必要があります。出口ルーチンの完了時にコミットされていない変更はすべて、以下のいずれかが行われるまでコミットされることはありません。

- アプライ・プログラムが、同じ活動化グループを使って別の出口ルーチンを呼び出す。
- アプライ・プログラムに応じて開始したジョブが終了する。

3. FULLREFPGM パラメーターを ASNLOAD プログラムの名前に設定してアプライ・プログラムを始動します。

始動されたアプライ・プログラムは、ユーザーから指定された ASNLOAD 出口ルーチンを使用します。別の ASNLOAD 出口ルーチンを使用させたい場合は、アプライ・プログラムを終了してから再始動します。

ASNLOAD 出口ルーチンを実行すると、すべてのターゲット表が 1 つずつリフレッシュされます。

関連タスク:

- 483 ページの『第 20 章 SQL レプリケーション・プログラムの操作 (z/OS)』
- 489 ページの『第 21 章 Windows Service Control Manager を使用して SQL レプリケーションのシステム・コマンドを発行する方法 (Windows)』

関連資料:

- 366 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』
- 330 ページの『asnacmd: アプライの操作』
- 334 ページの『asnapply: アプライの始動』
- 431 ページの『ENDDPRAPY: アプライ・プログラムの停止 (OS/400)』
- 460 ページの『STRDPRAPY: アプライ・プログラムの始動 (OS/400)』

第 11 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。以下のトピックには、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されています。

- 『レプリケーション・アラート・モニター』
- 176 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』
- 182 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』
- 185 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活動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・アクティビティーをチェックします。

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[®]、または 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:

- 173 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』
- 180 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』
- 181 ページの『レプリケーション中のアラートを送信する ASNMAIL 出口ルーチン』

関連概念:

- 173 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。

- 177 ページの『Q キャプチャー・プログラムのアラート条件』
- 177 ページの『Q アプライ・プログラムのアラート条件』
- 178 ページの『キャプチャー・プログラムのアラート条件』
- 179 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 10 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 10. 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活動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 11 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 11. 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 11. 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 12 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 12. 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 12. 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 13 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 13. 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活性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:

- 173 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』
- 173 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:

- 187 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 出口ルーチンは、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、この出口ルーチンを変更できます。例えば、ASNMAIL ユーザー出口ルーチンに、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 出口ルーチン』を参照してください。

関連概念:

- 176 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:

- 185 ページの『レプリケーション・アラート・モニターの操作』
- 195 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:

- 181 ページの『レプリケーション中のアラートを送信する ASNMAIL 出口ルーチン』

レプリケーション中のアラートを送信する ASNMAIL 出口ルーチン

ASNMAIL 出口ルーチンは、通知を処理します。この出口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 14 では、ASNMAIL 出口ルーチンの入力について説明しています。

表 14. ASNMAIL 出口ルーチンの入力

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 出口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。

¥sqllib¥samples¥rep¥ ディレクトリーには、ASNMAIL 出口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:

- 180 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』
- 176 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:

- 185 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 つまたは複数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、184 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:

- 173 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 for UNIX、DB2 for Windows、または DB2 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:

- 173 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:

- 「IBM DB2 Information Integrator レプリケーションとイベント・パブリッシングガイドおよびリファレンス」の『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します。
 - 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します。
 - 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します。

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します。

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・サーバーを展開します。「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:

- 180 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:

- 181 ページの『レプリケーション中のアラートを送信する ASNMAIL 出口ルーチン』

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、182 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。以下のトピックには、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されています。

1. モニターの作成。詳細については、186 ページの『レプリケーションまたは公開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、187 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、193 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。
 - 190 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』
 - 190 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、188 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、189 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、196 ページの『モニターの停止』を参照してください。

関連概念:

- 173 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公開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公開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、182 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・プログラムまたは公開プログラムを指定します。

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します。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します。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 次ダイアログを開きます。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・セットを指定します。
 - 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します。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 (Finish)」をクリックします。

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:

- 173 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・プログラムと公開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に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、次のようになります。

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 つを使用します。
 - 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。
2. ユーザーの環境と互換性のあるしきい値を選択してください。

例えば、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てください。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**」フォルダーを展開します。
 - b. モニターを選択します。
 - 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します。
 - d. 「**変更**」を選択します。
2. アラート条件を変更します。
3. モニターを再初期化します。詳細については、189 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:

- 176 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』
- 173 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、186 ページの『レプリケーションまたは公開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、362 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、357 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:

- 173 ページの『レプリケーション・アラート・モニター』
- 「*IBM DB2 Information Integrator* レプリケーションとイベント・パブリッシングガイドおよびリファレンス」の『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:

- 189 ページの『モニターの再初期化』
- 196 ページの『モニターの停止』
- 「*IBM DB2 Information Integrator* レプリケーションとイベント・パブリッシングガイドおよびリファレンス」の『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、354 ページの『asnmcmd: 実行中のレプリケーション・アラート・モニターの処理』を参照してください。

関連タスク:

- 188 ページの『モニターの開始』

- 196 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 15 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 15. レプリケーション・アラート・モニター操作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
autoprune	Y
email_server	デフォルト値なし
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:

- 190 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』
- 173 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:

- 185 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。

- 191 ページの『alert_prune_limit』
- 191 ページの『autoprune』
- 191 ページの『email_server』
- 191 ページの『max_notification_minutes』
- 192 ページの『max_notifications_per_alert』
- 192 ページの『monitor_errors』
- 192 ページの『monitor_interval』

- 192 ページの『monitor_limit』
- 192 ページの『monitor_path』
- 192 ページの『runonce』
- 193 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。

autoprune

デフォルト: **autoprune=y**

autoprune パラメーターは、自動整理を制御します。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 出口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) を使用するように設定されている 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、現行タイム・スタンプに基づいて行がこの制限より古い場合は、キャプチャー・プログラムと Q キャプチャー・プログラムはこれらの表中の行を整理します。

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:

- 190 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』
- 173 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:

- 185 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、『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、『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、195 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、196 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。

180 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:

- 182 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します。
- **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・モニターを開始したときに実行時間を指定することもできます。

関連概念:

- 190 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』
- 190 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』
- 173 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。

- **autoprune** 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します。
- **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します。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します。
- **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します。

関連概念:

- 190 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』
- 190 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』
- 173 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します。 **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します。
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します。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** 出口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ます。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:

- 180 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』
- 190 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』
- 190 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』
- 176 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』
- 173 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:

- 184 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:

- 190 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』
- 190 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・プログラムまたは公開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、354 ページの『asnmcmd: 実行中のレプリケーション・アラート・モニターの処理』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:

- 188 ページの『モニターの開始』
- 189 ページの『モニターの再初期化』

第 12 章 SQL レプリケーションの要求時報告

この章では、レプリケーション環境を報告して分析するために使用できる方式について説明します。この章の情報を使用して、レプリケーション・プログラムの現在の状況をチェックしたり、履歴データを検討して最近のメッセージやスループット、または待ち時間の統計を確認できます。

この章には、以下のセクションがあります。

- 『レプリケーション・プログラムの現在の状況のチェック (Linux、UNIX、Windows、z/OS)』
- 201 ページの『キャプチャー・プログラムおよびアプライ・プログラムのジャーナル・ジョブの状況のチェック (OS/400)』
- 201 ページの『傾向の履歴データの検討』
- 206 ページの『キャプチャー・プログラムの進行のモニター (OS/400)』

レプリケーション・プログラムの現在の状況のチェック (Linux、UNIX、Windows、z/OS)

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・モニターの現在の状況は簡単に評価できます。

レプリケーション・プログラムの現在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 (UNIX、Windows、z/OS)

「状況の照会」ウィンドウを使用して、キャプチャー・プログラムまたはアプライ・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・モニターの状況を照会することはできません。) 詳しくは、レプリケーション・センターのヘルプを参照してください。

コマンド行 (UNIX、Windows、z/OS)

- キャプチャー・プログラムの **asnccmd** システム・コマンド、**status** パラメーター。詳細については、349 ページの『asnccmd: キャプチャーの操作』を参照してください。
- アプライ・プログラムの **asnacmd** システム・コマンド、**status** パラメーター。詳細については、330 ページの『asnacmd: アプライの操作』を参照してください。
- レプリケーション・アラート・モニターの **asnmcmd** システム・コマンド、**status** パラメーター。詳細については、354 ページの『asnmcmd: 実行中のレプリケーション・アラート・モニターの処理』を参照してください。

プログラムの状況を照会すると、そのプログラムに関連する各スレッドの状態を記述するメッセージが戻されます。

- キャプチャー・プログラムには、管理スレッド、整理スレッド、ワーカー・スレッド、シリアライゼーション・スレッドという 4 つのスレッドがあります。
- アプライ・プログラムには、管理スレッドとワーカー・スレッドという 2 つのスレッドがあります。
- レプリケーション・アラート・モニター・プログラムには、管理スレッド、ワーカー・スレッド、およびシリアライゼーション・スレッドという 3 つのスレッドがあります。

ユーザーは受け取ったメッセージから、プログラムが正しく機能しているかどうかを判断できます。一般的にワーカー・スレッド、管理スレッド、および整理スレッドは作動状態にあり、実行が期待されるタスクを実行しています。シリアライゼーション・スレッドは通常は待機状態にあります。このスレッドはグローバル・シグナル・ハンドラーであり、シグナルを待っています。整理スレッドは、CD 表、および次のレプリケーション・コントロール表を整理します。

- 作業単位 (IBMSNAP_UOW) 表
- キャプチャー・トレース (IBMSNAP_CAPTRACE) 表
- キャプチャー・モニター (IBMSNAP_CAPMON) 表
- シグナル (IBMSNAP_SIGNAL) 表

ユーザーが受け取ったメッセージが、プログラムは実行中であることを示しているも、環境がその逆の証拠を示している場合は、さらに調査が必要です。たとえば、アプライ・プログラムの状況を照会した結果、ワーカー・スレッドが作業中であることが判明したが、データが期待どおりにターゲット表にアプライされていない場合は、アプライ・トレール (IBMSNAP_APPLYTRAIL) 表の中で、データがアプライされない理由を EXPLAIN するメッセージがないか探すことができます。プログラムの作業を阻害するシステム・リソースの問題が発生したことも考えられます。

ユーザーが受け取ったメッセージが典型的な状態を示していない場合は、表 16 に示すような追加アクションを実行できます。

表 16. 処理スレッドの状況に関連する問題に対する推奨アクション

処理スレッドの状況	記述および推奨アクション
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	CPU の不足など、システム・リソースの問題の可能性があるので調査してください。
初期化済み	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	この状態は、キャプチャー・プログラムのスレッドにのみ当てはまります。スレッドがこの状態にある場合は、キャプチャー・プログラムはユーザーにより中断され、ユーザーによる操作の再開を待っています。

表 16. 処理スレッドの状況に関連する問題に対する推奨アクション (続き)

処理スレッドの状況	記述および推奨アクション
停止	スレッドは実行されていません。アプライ・トレール (IBMSNAP_APPLYTRAIL) 表またはキャプチャー・トレース (IBMSNAP_CAPTRACE) 表で、スレッドが停止された理由を EXPLAIN するメッセージを調べてください。たとえば、整理スレッドが停止されたことを示すメッセージを受け取った場合は、IBMSNAP_CAPTRACE 表で理由を確認します。表が非常に大きいため、即時に整理する場合は、キャプチャー・プログラムを停止後に再始動すると、整理スレッドを開始できます。

キャプチャー・プログラムおよびアプライ・プログラムのジャーナル・ジョブの状況のチェック (OS/400)

DB2 for iSeries は、サブシステム・ジョブの処理 (WRKSBSJOB) システム・コマンドを使用して、キャプチャー・プログラムおよびアプライ・プログラムのジャーナル・ジョブの状況をチェックします。

1. 次のコマンドを入力します。

```
WRKSBSJOB subsystem
```

この *subsystem* はサブシステム名です。ユーザーが自身のサブシステム記述を作成していないかぎり、多くの場合、サブシステムは QZSNDPR です。

2. 実行中のジョブのリストで、目的のジョブを探します。ジャーナル・ジョブは、割り当てられているジャーナルに従って命名されています。ここでジョブが見つからない場合には、ジョブの処理 (WRKJOB) システム・コマンドまたはサブミットしたジョブの処理 (WRKSBMJOB) システム・コマンドを使用します。ジョブのジョブ・ログを探して、ジョブが正常完了していること、またはジョブが失敗した理由を確認します。

傾向の履歴データの検討

最近のレプリケーション操作からの履歴データを検討して、データの傾向を評価できます。一定期間にわたり認識される傾向から、安定した量のデータが複製されていることを確認したり、パフォーマンスを向上させるために調整の余地があることを認識できます。

履歴データは、アプライ・トレール (IBMSNAP_APPLYTRAIL)、アプライ・トレース (IBMSNAP_APPLYTRACE)、キャプチャー・モニター (IBMSNAP_CAPMON)、およびキャプチャー・トレース (IBMSNAP_CAPTRACE) の各コントロール表から生成できます。これらの表の整理の頻度により、生成できるレポートが異なります。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くことをお勧めします。

表 17 は、ユーザーが表示できる履歴データを示しています。

表 17. 履歴情報の保管場所

質問:	レプリケーション・センターで使用する ウィンドウ:
キャプチャー・プログラム、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？	キャプチャー・メッセージ アプライ・メッセージ モニター・メッセージ
平均で、 <ul style="list-style-type: none"> 一定期間中に CD 表で処理された行数は？ 整理される行数は？ コミットされるトランザクション数は？ キャプチャー・プログラムが使用するメモリー量は？ 	キャプチャー・スループット分析
ソースでデータが更新されてから、キャプチャー・プログラムによってキャプチャーされるまでの平均時間はどれくらいか？	キャプチャー待ち時間
アプライ・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？	アプライ・レポート
平均で、 <ul style="list-style-type: none"> 一定期間中にターゲット表で処理された行数は？ サブスクリプション・セットの処理の経過時間は？ 	アプライ・スループット分析
ソース表が更新されてから、対応するターゲット表が更新されるまでの平均の経過時間はどれくらいか？	エンドツーエンド待ち時間

分析対象とするデータ量を識別するために、時間の範囲を選択できます。時間範囲の開始と終了の両方の日時を指定してから、算出結果の平均値として結果を表示するように指定します。時間インターバル (1 秒、1 分、1 時間、1 日、または 1 週間) を選択して、結果をグループ化します。たとえば、9:00 p.m. から 9:59 p.m. までのアプライ・プログラムのスループットを分析するように選択した場合、データを 1 分インターバルで表示すると、結果は 60 行で表示され、それぞれの行に、60 分の範囲の間の各 1 分間のアクティビティの要約が表示されます。また、1 時間というインターバルを選択した場合は、結果は 1 行に表示され、指定された 1 時間という期間の平均スループットが示されます。ユーザーがインターバルを指定しない場合は、APPLYTRAIL 表のロー・データが表示されます。

レプリケーション・センターのウィンドウは、さまざまなコントロール表およびログ・ファイルに含まれた情報から結果を表示します。次のセクションでは、レプリケーション・センターで、履歴データを使用してレプリケーション操作を評価するため方法をさらに詳しく説明します。

- 『キャプチャー・プログラムのメッセージの検討』
- 『キャプチャー・プログラムのスループットの検査』
- 204 ページの『キャプチャー・プログラムによって処理されるデータの待ち時間の表示』
- 204 ページの『アプライ・プログラムのメッセージの検討』
- 205 ページの『アプライ・プログラムのスループットの検査』
- 205 ページの『トランザクションのレプリケーションに要した平均時間の表示』
- 206 ページの『モニター・プログラムのメッセージの検討』

キャプチャー・プログラムのメッセージの検討

指定された一定期間内にキャプチャー・トレース (IBMSNAP_CAPTRACE) 表に挿入されたメッセージを検討するには、「キャプチャー・メッセージ」ウィンドウを使用します。IBMSNAP_CAPTRACE 表には、キャプチャー・プログラムから発行された、初期化、整理、警告、およびエラーなどの重要なイベントの行が入ります。

たとえば、「キャプチャー・メッセージ」ウィンドウでは、1 週間の間にキャプチャー・プログラムから記録されたすべてのエラー・メッセージと警告メッセージを検討できます。また、「キャプチャー・メッセージ」ウィンドウから、ファイルにデータを印刷または保管することもできます。

キャプチャー・プログラムのスループットの検査

指定された時間範囲のキャプチャー・プログラムのパフォーマンス結果を表示するには、「キャプチャー・スループット分析」ウィンドウを使用します。キャプチャー・プログラムは統計情報を定期的にキャプチャー・モニター (IBMSNAP_CAPMON) 表に記録し、整理時には、整理統計をキャプチャー・トレース (IBMSNAP_CAPTRACE) 表に記録します。「キャプチャー・スループット分析」ウィンドウはこれらの表の情報を使用して、4 つの異なるタスクのパフォーマンス率の計算結果を表示します。

4 つのタイプの情報の結果をすべて使用して、キャプチャー・プログラムのスループット・パフォーマンスを検査できます。結果を絶対値で表示するか平均値で表示するかを指定できます。

- ログで挿入された、またはスキップされた行数
- CD 表で整理された行数
- コミットされたトランザクション数
- メモリー使用

たとえば、「キャプチャー・スループット分析」ウィンドウでは、キャプチャー・プログラムのスループットの 1 週間の平均パフォーマンスを検討できます。これを行うには、時間範囲の開始と終了の両方の日時を指定してから、算出結果の平均値として結果を表示するように指定します。

キャプチャー・プログラムによって処理されるデータの待ち時間の表示

ソースでデータが更新されてから、キャプチャー・プログラムによってキャプチャーされるまでのおおよその時間を表示するには、「キャプチャー待ち時間」ウィンドウを使用します。経過時間は、一定期間にわたり、CD 表の中のデータの新鮮さを示します。この平均待ち時間は、キャプチャー・モニター (CAPMON) 表の情報から生成できます。この情報の派生元は、登録 (IBMSNAP_REGISTER) 表です。

現行のキャプチャー待ち時間は、登録 (IBMSNAP_REGISTER) 表の中のグローバル・レコードからの SYNCHTIME 列の中の CURRENT_TIMESTAMP 値を使用して計算されます。

$(CURRENT_TIMESTAMP) - (SYNCHTIME)$

表 18. 現行キャプチャー待ち時間を計算するための値の例

パラメーター	列値
CURRENT_TIMESTAMP	2001-10-20-10:30:25
SYNCHTIME	2001-10-20-10:30:00

たとえば、表 18 の値を使用すると、現行待ち時間は 25 秒になります。

$10:30:25 - 10:30:00 = 25$

キャプチャー待ち時間は時間とともに変化し、これらの変更の履歴はキャプチャー・モニター (IBMSNAP_CAPMON) 表に保管されます。レプリケーション・センタはキャプチャー・モニター表の中の情報を使用して、平均待ち時間、または履歴待ち時間を計算します。平均待ち時間の場合の公式は、現行待ち時間のもと同じですが、CURRENT_TIMESTAMP 値ではなく、MONITOR_TIME 値が使用されます。MONITOR_TIME 値は、キャプチャー・プログラムによってキャプチャー・モニター表に行が挿入された時間を示すタイム・スタンプです。平均待ち時間は、秒、分、時間、日、または週で表示できます。たとえば、「キャプチャー待ち時間」ウィンドウで、キャプチャー・プログラムの過去 1 週間の平均待ち時間を 1 時間単位で表示できます。

アプライ・プログラムのメッセージの検討

指定された一定期間内にアプライ・トレース (IBMSNAP_APPLYTRACE) 表に挿入されたメッセージを検討するには、「アプライ・メッセージ」ウィンドウを使用します。IBMSNAP_APPLYTRACE 表には、アプライ・プログラムから発行された、初期化、警告、およびエラーなどの重要なイベントの行が入ります。

たとえば、「アプライ・メッセージ」ウィンドウでは、1 週間の間にアプライ・プログラムから記録されたすべてのエラー・メッセージと警告メッセージを検討できます。また、「アプライ・メッセージ」ウィンドウから、ファイルにデータを印刷または保管することもできます。

「アプライ・レポート」ウィンドウを使用して、アプライ・トレール (IBMSNAP_APPLYTRAIL) 表に挿入されたデータを検討して、一定期間について、特定のアプライ・プログラムが成功しているかどうかをチェックします。

IBMSNAP_APPLYTRAIL 表には、サブスクリプション・セットの実行に関するデータとして、サブスクリプション・セットの状況、エラー・メッセージ、および処理された行数が含まれています。

「アプライ・レポート」ウィンドウでは次のデータを表示できます。

- すべてのサブスクリプション・セット
- 失敗したサブスクリプション・セット
- 成功したサブスクリプション・セット
- 失敗したサブスクリプション・セットごとのエラー・サマリー

たとえば、「アプライ・レポート」ウィンドウでは、アプライ・プログラムが先週サブスクリプション・セットを正常に処理したかどうかを確認できます。レプリケーションできなかったサブスクリプション・セットがある場合には、これらのセットに対してアプライ・プログラムから発行されたエラー・メッセージを表示できます。また、「アプライ・レポート」ウィンドウと一緒に「アプライ・スループット分析」ウィンドウを使用できます。「アプライ・レポート」ウィンドウを使用してどのセットが正常に複製されたかを確認した後、「アプライ・スループット分析」ウィンドウを使用して複製された行数と、レプリケーションに要した時間を確認できます。

「アプライ・レポート」ウィンドウを使用して、IBMSNAP_APPLYTRAIL 表で、特定の行からのすべてのデータを表示することもできます。

アプライ・プログラムのスループットの検査

特定のアプライ修飾子のパフォーマンス統計を検査するには「アプライ・スループット分析」ウィンドウを使用します。SQL ステートメントを作成しなくても、データをフィルターに掛け、グループ化できます。たとえば、特定のアプライ修飾子により処理されたサブスクリプション・セット内のターゲット表で挿入、更新、削除、および再処理された行数を表示できます。また、特定のアプライ修飾子について、アプライ・プログラムがサブスクリプション・セットの処理に費やした時間も表示できます。

トランザクションのレプリケーションに要した平均時間の表示

特定のサブスクリプション・セット内のトランザクションのレプリケーションに使用された平均時間のおおよその値を表示するには、「エンドツーエンド待ち時間」ウィンドウを使用します。変更キャプチャー・レプリケーションで実行されるイベントの順序の記述は、レプリケーション・センターのヘルプを参照してください。

「エンドツーエンド待ち時間」ウィンドウでは、たとえば、一定期間内のアプライ・サイクルごとに、サブスクリプション・セットのおおよその待ち時間を表示できます。また、時間をインターバルで分けて、インターバルごとの平均待ち時間を表示することもできます。

レプリケーション・センターでは次の公式を使用してエンドツーエンド待ち時間を計算します。

$$(ENDTIME - LASTRUN) + (SOURCE_CONN_TIME - SYNCHTIME)$$

ここで、

- ENDTIME は、アプライ・プログラムがサブスクリプション・セットの処理を終了した時間です。
- LASTRUN は、アプライ・プログラムがサブスクリプション・セットの処理を開始した時間です。
- SOURCE_CONN_TIME は、アプライ・プログラムがデータをフェッチするためにキャプチャー・コントロール・サーバーに接続した時間です。
- SYNCHTIME は、キャプチャー・プログラムによる、CD 表へのデータのコミットの最新の時刻です。

表 19. エンドツーエンド待ち時間の値の計算の例

パラメーター	列値
ENDTIME	2001-10-20-10:01:00
LASTRUN	2001-10-20-10:00:30
SOURCE_CONN_TIME	2001-10-20-10:00:32
SYNCHTIME	2001-10-20-10:00:00

たとえば、あるサブスクリプション・セットに、表 19 に示すような値があるとし、前述の式を使用すると、このサブスクリプション・セットのエンドツーエンドの平均待ち時間は 62 秒になります。

$$(10:01:00 - 10:00:30) + (10:00:32 - 10:00:00) = 62$$

モニター・プログラムのメッセージの検討

指定された一定期間内にモニター・トレース (IBMSNAP_MONTRACE) 表に挿入されたメッセージを検討するには、「モニター・メッセージ」ウィンドウを使用します。IBMSNAP_MONTRACE 表には、モニター・プログラムから発行された、アクション、警告、およびエラーなどの重要なイベントの行が入ります。

たとえば、「モニター・メッセージ」ウィンドウでは、1 週間の間にモニター・プログラムから記録されたすべてのエラー・メッセージと警告メッセージを検討できます。また、「モニター・メッセージ」ウィンドウから、ファイルにデータを印刷または保管することもできます。

キャプチャー・プログラムの進行のモニター (OS/400)

キャプチャー・プログラムが終了した場合は、再始動 (IBMSNAP_RESTART) 表を調べて、キャプチャー・プログラムがどこまで進行したかを確認できます。ソース表によって使用される各ジャーナルごとに 1 つの行があります。LOGMARKER 列には、正常に処理された最後のジャーナル項目のタイム・スタンプが記されています。SEQNBR 列には、その項目のジャーナルを入力した順番を表す数値が記されています。

キャプチャー・プログラムがまだ実行中の場合には、次のステップを使用して、進行状況を確認できます。

1. キャプチャーされる各ソース表ごとに、その CD 表を開きます。
2. CD 表の最後の行の COMMITSEQ 列にある 16 進値をメモします。

3. 同じ COMMITSEQ 値を使用した行の作業単位 (IBMSNAP_UOW) 表を見ます。一致する COMMITSEQ が IBMSNAP_UOW 表に存在しない場合には、CD 表の 2 番目の行から最後の行で同じ処理を繰り返してください。一致するものが見つかるまで、CD 表全体を逆方向にも検索してみます。
4. 一致する COMMITSEQ が見つかったら、UOW 行の LOGMARKER 列にある値をメモします。これが、処理されたジャーナル項目のタイム・スタンプです。その時間までソース表に加えられているすべての変更は、適用される準備ができています。
5. ジャーナルの表示 (**DSPJRN**) システム・コマンドを使用して、キャプチャー・プログラムによる処理待ちのジャーナル項目の数を確認します。以下の例で示すようにして、出力ファイル (または、印刷レポート用のプリンター) に直接出力します。

```
DSPJRN FILE(JRNLIB/DJRN1)
        RCVRNG(*CURCHAIN)
        FROMTIME(timestamp)
        TOTIME(*LAST)
        JRNCDE(J F R C)
        OUTPUT(*OUTFILE)
        ENTDTALEN(1) OUTFILE(library/outfile)
```

この *timestamp* は、4 で識別されたタイム・スタンプです。

出力ファイルにあるレコード数が、キャプチャー・プログラムが処理しなければならない残っているジャーナル項目の大体の数です。

第 13 章 SQL レプリケーション環境の変更

この章では、レプリケーション環境の日常の変更を行うときに考慮する必要のある問題について説明します。

この章には、以下のセクションがあります。

- 『新規オブジェクトの登録』
- 210 ページの『登録済みオブジェクトの登録属性の変更』
- 211 ページの『ソース表への列の追加』
- 214 ページの『登録済みオブジェクトの変更のキャプチャーの停止』
- 215 ページの『登録の再活動化』
- 216 ページの『登録の除去』
- 217 ページの『キャプチャー・スキーマの変更』
- 220 ページの『新規サブスクリプション・セットの作成』
- 220 ページの『既存のサブスクリプション・セットに新しいサブスクリプション・セット・メンバーを追加する』
- 221 ページの『既存のサブスクリプション・セットでサブスクリプション・セット・メンバーを使用不可にする』
- 221 ページの『既存のサブスクリプション・セットに対してサブスクリプション・セット・メンバーを使用可能にする』
- 222 ページの『サブスクリプション・セットの属性の変更』
- 222 ページの『サブスクリプション・セット名の変更』
- 224 ページの『サブスクリプション・セットの分割』
- 229 ページの『サブスクリプション・セットのマージ』
- 232 ページの『サブスクリプション・セットのアプライ修飾子の変更』
- 235 ページの『サブスクリプション・セットの非活動化』
- 236 ページの『サブスクリプション・セットの除去』
- 237 ページの『データベース・アプリケーション・イベントとレプリケーション・イベントの調整』
- 245 ページの『別のシステムへのレプリケーション構成のプロモート』

新規オブジェクトの登録

レプリケーション環境では、いつでも新規の表、ビュー、またはニックネームを登録できます。キャプチャー・プログラムを再初期化する必要はありません。

手順:

オブジェクトを登録するには、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

「表の登録」、「ビューの登録」、または「ニックネームの登録」ウィンドウを使用します。詳しくは、レプリケーション・センターのヘルプを参照してください。

ADDDPRREG システム・コマンド (OS/400)

パラメーター記述およびコマンド構文の詳細については、386 ページの『ADDDPRREG: DPR 登録の追加 (OS/400)』を参照してください。

新しく登録されたオブジェクトは、そのオブジェクトを参照するサブスクリプション・セットがアプライ・プログラムによって処理されたときに、キャプチャー・プログラムによって自動的に初期化されます。アプライ・プログラムは、この新規オブジェクトの変更のキャプチャーを開始するようにキャプチャー・プログラムにシグナルを送ります。オブジェクトの登録の詳細は、43 ページの『第 3 章 表およびビューを SQL レプリケーション・ソースとして登録する』を参照してください。

登録済みオブジェクトの登録属性の変更

既存の登録済みオブジェクトの登録属性はいつでも変更できます。登録属性には次の属性が含まれます。

- CHGONLY
- CONFLICT_LEVEL
- RECAPTURE
- DISABLE_REFRESH
- CHG_UPD_TO_DEL_INS
- STOP_ON_ERROR
- BEFORE_IMG_PREFIX

注: 変更前イメージ接頭部の値は、この値が NULL のときにしか更新できません。

手順:

1. 次の方法を使用して属性を変更します。

レプリケーション・センター

「登録済み表」フォルダーで内容ペインの中の登録済み表を右マウス・ボタンでクリックし、「プロパティ」を選択します。詳しくは、レプリケーション・センターのヘルプを参照してください。

2. 属性を変更したら、変更が認識されるように、キャプチャー・プログラムを再初期化する必要があります。キャプチャー・プログラム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

「キャプチャー・コントロール・サーバー」フォルダーで、内容ペインの中のキャプチャー・コントロール・サーバーを右マウス・ボタンでクリックし、「キャプチャーの再初期化」を選択します。詳しくは、レプリケーション・センターのヘルプを参照してください。

asnccmd システム・コマンド (Windows、UNIX、z/OS)

reinit パラメーターを使用します。パラメーター記述およびコマンド構文の詳細については、349 ページの『asnccmd: キャプチャーの操作』を参照してください。

INZDPRCAP システム・コマンド (OS/400)

パラメーター記述およびコマンド構文の詳細については、446 ページの『INZDPRCAP: DPR キャプチャー・プログラムの再初期化 (OS/400)』を参照してください。

ソース表への列の追加

登録済みソース表に列を追加する必要がある場合は、最初に DB2 レプリケーションがこの表をどのように使用しているかを考えてください。このソース表の中の新しい列を複製する必要がある場合には、既存のキャプチャー・プログラムおよびアプライ・プログラムが新規列を認識し、中断なしに処理を続行できるようにしてください。新規列の中のデータを複製するかどうかによって、特別な処理ステップを実行する必要があることがあります。

複製しない

新規列の中のデータを複製する必要がある場合には、特別な処理ステップを実行する必要はありません。キャプチャー・プログラムは即時に変更を認識し、実行を続けます。

複製する

これらの新規列の中のデータを複製する必要がある場合には、新規列のデータがキャプチャーされ、キャプチャー・プログラムおよびアプライ・プログラムがエラーなしに実行を継続できるように、以下のステップを実行します。

前提条件:

この手順を使用する前に、ソース表、変更データ (CD) 表、およびターゲット表の構造、そしてシステムで定義されている登録およびサブスクリプション・セットについてよく調べておいてください。

制約事項:

主キーとして相対レコード番号 (RRN) を使用する iSeries の表に列を追加する場合は、これらのステップを使用しないでください。RRN は CD 表の最後の列である必要があります。RRN を含む iSeries の表に列を追加するときには、登録を除去し、ソース表に列を追加してから、RRN がキャプチャーされることを指定して、新しい登録として再度この表を追加してください。iSeries の登録の除去および追加の詳細は、453 ページの『RMVDPREG: DPR 登録の除去 (OS/400)』および 386 ページの『ADDDPRREG: DPR 登録の追加 (OS/400)』を参照してください。

DB2 以外のリレーショナル・データベース上の登録済みソースに列を追加するためにこれらのステップを使用することはできません。DB2 以外のリレーショナル・ソースの登録には、変更のキャプチャーに使用されるトリガーのセットが含まれます。これらのトリガーを変更することはできません。

このため、このソース表に新規列を追加し、これらの列の中のデータを複製する必要があるときには、既存の登録済みソースをドロップしてから、再作成する必要があります。

手順:

1. 変更するソース表に対するすべてのアクティビティーを静止します。
2. 次の方法のいずれかを使用して、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

「キャプチャーの停止」ウィンドウを使用します。詳しくは、レプリケーション・センターのヘルプを参照してください。

asnccmd システム・コマンド (Windows、UNIX、z/OS)

stop パラメーターを使用します。パラメーター記述およびコマンド構文の詳細については、349 ページの『asnccmd: キャプチャーの操作』を参照してください。

ENDDPRCAP システム・コマンド (OS/400)

パラメーター記述およびコマンド構文の詳細については、434 ページの『ENDDPRCAP: キャプチャー・プログラムの停止 (OS/400)』を参照してください。

ヒント: この手順の間キャプチャー・プログラムをアクティブにしておく必要がある場合には、ソース表に対するアクティビティーを停止した後、シグナル (IBMSNAP_SIGNAL) 表に USER シグナルを挿入します。キャプチャー・プログラムが USER シグナルを処理するまで待ちます。

キャプチャー・プログラムは USER シグナルを処理すると、関連する CD 表に対して処理を必要とするアクティビティーがなくなるため、この CD 表へのアクセスを必要としなくなります。

3. 次の方法を使用して、このソース表をサブスクライブするすべてのサブスクリプション・セットを非活動化します。

レプリケーション・センター

「サブスクリプション・セット」フォルダーで内容ペインの中のアクティブなサブスクリプション・セットを右マウス・ボタンでクリックし、「非アクティブ化」を選択します。詳しくは、レプリケーション・センターのヘルプを参照してください。

注: この処理中にサブスクリプション・セットを非活動化したくない場合は、新規列を追加しているときに、このソース表に対して、これらのサブスクリプション・セットに関連付けられたアプライ・プログラムが実行されていないことを確認してください。この代わりに、これらのアプライ・プログラムが、前の USER シグナルに関連付けられたシグナルのログ・シーケンス番号 (LSN) までデータの処理を終了していることを確認することもできます。

このステップの方法では、表を変更できるように、CD 表に対して排他的アクセスを確保しています。

4. SQL を使用して ALTER TABLE ADD ステートメントをサブミットし、ソース表に新規列を追加します。
5. CD 表に新規列を追加するには、次の方法を使用します。

レプリケーション・センター

「登録済み表」フォルダーで内容ペインの中の登録済み表を右マウス・ボタンでクリックし、「プロパティ」を選択します。詳しくは、レプリケーション・センターのヘルプを参照してください。

キャプチャー・プログラムは自動的に登録を再初期化し、新規列で初めてログ・データを読み取ったときに、これらの新規列の変更をキャプチャーします。

6. SQL を使用して ALTER TABLE ADD ステートメントをサブミットし、ターゲット表に新規列を追加します。
7. 次の方法を使用して、ステップ 3 でまだ非活動化していなかった、関連するサブスクリプション・セットを非活動化します。

レプリケーション・センター

「サブスクリプション・セット」フォルダーで内容ペインの中のアクティブなサブスクリプション・セットを右マウス・ボタンでクリックし、「非アクティブ化」を選択します。詳しくは、レプリケーション・センターのヘルプを参照してください。

絶対に必要である場合は、ここでこのソース表に対するアクティビティを再開できます。しかし、関連するサブスクリプション・セットがまだ変更されていないため、これらの新規列に対して行われた変更が失われるないように、これらのサブスクリプション・セットは非活動化しておく必要があります。

8. 次の方法を使用して、関連するサブスクリプション・セット・メンバーに新規列を追加します。

レプリケーション・センター

「ターゲット表への列の追加」ウィンドウを使用します。ターゲット表への列の追加に関する詳細は、レプリケーション・センターのヘルプを参照してください。

9. **UNIX、Windows、z/OS の場合: opt4one** を y に設定してアプライ・プログラムを実行している場合には、アプライ・プログラムを停止してから再始動します。
10. 次の方法を使用してサブスクリプション・セットを再活動化します。

レプリケーション・センター

「サブスクリプション・セット」フォルダーで内容ペインの中の非活動化されたサブスクリプション・セットを右マウス・ボタンでクリックし、「活動化」を選択します。詳しくは、レプリケーション・センターのヘルプを参照してください。

登録済みオブジェクトの変更のキャプチャーの停止

キャプチャー・プログラムがオブジェクトに必要なすべての処理を終了できるように、登録済みオブジェクトを削除する前に、オブジェクトを非活動化する必要があります。また、このオブジェクトで一時的に変更のキャプチャーを停止しても、他の登録済みオブジェクトに対してはキャプチャー・プログラムを実行し続けておきたい場合も、登録済みオブジェクトを非活動化できます。

キャプチャー・プログラムは、非活動化されたソース・オブジェクトについては変更のキャプチャーを停止します。しかし、これらのソース・オブジェクトに関連する変更データ (CD) 表、登録属性、およびサブスクリプション・セットは、システム上に残ります。

登録済みオブジェクトを非活動化する前に、この登録済みオブジェクトに関連付けられたすべてのサブスクリプション・セットを非活動化する必要があります。これにより、ユーザーがオブジェクトを削除する、または再度活動化する準備が整う前に、アプライ・プログラムがオブジェクトを自動的に再活動化し、非活動化処理に介入してくることを防止できます。

オブジェクトが非活動化され、DB2 レプリケーションがそのオブジェクトに対する変更のキャプチャーを停止すると、登録済みオブジェクトに関連付けられたすべてのサブスクリプション・セットが影響を受けます。これらのサブスクリプション・セットの実行を続けたい場合は、この登録済みオブジェクトをソースとして使用するサブスクリプション・セット・メンバーを、非活動化されたサブスクリプション・セットから除去する必要があります。

制約事項:

非活動化できる DB2 登録済みオブジェクトは、キャプチャー・プログラム・ソースとして定義されているものだけです。

キャプチャー・トリガーから使用される DB2 以外のリレーショナル・データベース・オブジェクトを非活動化することはできません。

手順:

登録済みオブジェクトを非活動化するには、次のようにします。

1. 次の方法を使用して、関連するすべてのサブスクリプション・セットを非活動化します。

レプリケーション・センター

「サブスクリプション・セット」フォルダーで内容ペインの中のアクティブなサブスクリプション・セットを右マウス・ボタンでクリックし、「非アクティブ化」を選択します。詳しくは、レプリケーション・センターのヘルプを参照してください。

詳細については、235 ページの『サブスクリプション・セットの非活動化』を参照してください。

2. 次の方法のいずれかを使用して、登録済みオブジェクトを非活動化します。

レプリケーション・センター

「登録済み表」フォルダーで内容ペインの中の登録済み表を右マウス・

ボタンをクリックし、「変更の取り込みを停止する」を選択します。詳しくは、レプリケーション・センターのヘルプを参照してください。

CAPSTOP シグナル

シグナル (IBMSNAP_SIGNAL) 表に手動で CAPSTOP シグナルを挿入します。詳細については、543 ページの『*schema.IBMSNAP_SIGNAL*』を参照してください。

登録の再活動化

登録および関連するサブスクリプション・セットを一時的に非活動化し、その後に登録を再活動化してデータのキャプチャーを再開するには、レプリケーション・センターを使用して単純にこれらのサブスクリプション・セットを再活動化します。キャプチャー・プログラムは、アプライ・プログラムが CAPSTART シグナルを送信した後に登録を再活動化します。

しかし、予期しないエラーのためにキャプチャー・プログラムが登録を非活動化した場合は、登録を再活動化するために特別な処理を行う必要があります。予期しないエラーが発生すると、この登録の STOP_ON_ERROR 列の値が N に設定されている場合、キャプチャー・プログラムは、登録 (IBMSNAP_REGISTER) 表の中の STATE 列の値を S (Stopped) に設定します。この STATE 列の値は、キャプチャー・プログラムがこの登録の処理を停止したこと、そして登録の修復が必要であることを意味します。アプライ・プログラムは停止状態の登録に対して CAPSTART シグナルを発行することはありません。

これらの予期しないエラーを訂正し、登録を再活動化に適格なものにするには、以下の手順を使用します。

前提条件:

この非活動化された登録に関してキャプチャー・プログラムから生成されたエラー・メッセージを読みます。

DB2 レプリケーションのキャプチャー・コントロール表の構成、およびシステム上で実行中のキャプチャー・プログラムについて調べておいてください。

手順:

1. エラー・メッセージに含まれた情報を使用して、登録を変更します。
2. キャプチャー・コントロール・サーバーから、次の SQL スクリプトを実行して、IBMSNAP_REGISTER 表の中の STATE 列をリセットします。

```
UPDATE Schema.IBMSNAP_REGISTER
   SET STATE = 'I'
  WHERE
    SOURCE_OWNER = 'SrcSchema'    AND
    SOURCE_TABLE = 'SrcTbl'       AND
    SOURCE_VIEW_QUAL = SrcVwQual  AND
    STATE = 'S';
```

この *Schema* はキャプチャー・スキーマの名前であり、*SrcSchema* は登録済みソース表スキーマであり、*SrcTbl* は登録済みソース表の名前であり、*SrcVwQual* はこのソース表のソース・ビュー修飾子です。

STATE 列が I (Inactive) に設定されると、キャプチャー・プログラムは、通常はアプライ・プログラムから出される CAPSTART シグナルを受け取りしだい、データのキャプチャーを開始することができます。

例: アクティブな登録のソース表が誤って DATA CAPTURE NONE に変更されたとします (本来は DATA CAPTURE CHANGES である必要があります)。また、エラーを検出してもキャプチャー・プログラムが停止しないことを指定する STOP_ON_ERROR = 'N' を使用してこの登録が定義されていたとします。キャプチャー・プログラムは次回の再始動、または再初期化時に、ソース表のこの誤った条件を認識し、この登録について、登録 (IBMSNAP_REGISTER) 表の中の STATE 列を S (Stopped) に設定します。登録は停止状態になるため、アプライ・プログラムが対応するサブスクリプション・セットを処理しようとする、エラー・メッセージが発行されます。以下のことを行う必要があります。

- 表のオプションを DATA CAPTURE CHANGES にリセットする ALTER TABLE ステートメントをサブミットして、SQL によりソース表の設定を訂正します。
- 上記の SQL スクリプトを使用して、手動で登録を停止状態から非活動状態にリセットします。

アプライ・プログラムはサブスクリプション・セット全体に対してフル・リフレッシュを実行します。

登録の除去

ユーザーが登録をドロップすると、DB2 レプリケーションはオブジェクトの登録をドロップし、関連する変更データ (CD) 表または整合変更データ (CCD) 表をドロップし、DB2 以外のリレーショナル・データベース・ソースの CCD オブジェクト・ニックネームおよびキャプチャー・トリガーをドロップします。実際のソース表またはビューはデータベース内に残ります。

前提条件:

キャプチャー・プログラムがこのオブジェクトの現行の処理を終了できるように、最初にソース・オブジェクトを非活動化します。

重要: 非活動化は非同期処理です。オブジェクトの除去前に必ず非活動化処理が終了するようにしてください。

手順:

ソース表またはビューの登録を除去するには、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

「登録済み表の削除」または「登録済みビューの削除」ウィンドウを使用します。詳しくは、レプリケーション・センターのヘルプを参照してください。

RMVDPREG システム・コマンド (OS/400)

パラメーター記述およびコマンド構文の詳細については、453 ページの『RMVDPREG: DPR 登録の除去 (OS/400)』を参照してください。

キャプチャー・スキーマの変更

既存のキャプチャー・スキーマを変更するには、以下の手順を使用します。

前提条件:

次の SQL ステートメントを実行する前に、DB2 レプリケーション・コントロール表、およびシステムに定義されているサブスクリプション・セットについてよく調べておいてください。

UNIX、Windows、z/OS の場合: モニター定義が設定されているか、あるいは、変更しようとしているキャプチャー・スキーマの下でレプリケーション・アラート・モニター・プログラムを始動している場合には、これらのモニター定義をドロップしてください。キャプチャー・スキーマを変更した後、レプリケーション・センターを使用して、新しいキャプチャー・スキーマ名を使用してモニター定義を再作成します。次に、**asnmcmd** システム・コマンドで **reinit** パラメーターを使用して、関連するレプリケーション・アラート・モニター・プログラムを再初期化できます。この代わりに、**asnmcmd** システム・コマンドで **stop** 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・プログラムを停止してから、**asnmon** システム・コマンドを使用してプログラムを再始動することもできます。

新しいキャプチャー・スキーマ名を決定します。詳細については、327 ページの『第 17 章 SQL レプリケーション・オブジェクトの命名規則』を参照してください。

この手順を使用する前に、キャプチャー・コントロール・サーバー、およびこのキャプチャー・コントロール・サーバーに関連するすべてのアプライ・コントロール・サーバーが、バージョン 8 に移行済みであることを確認してください。

制約事項:

ソース・サーバーが DB2 以外のリレーショナル・データベースである場合は、この手順は使用しないでください。

手順:

1. 次の方法のいずれかを使用して、新しいキャプチャー・スキーマ用のコントロール表を作成します。

レプリケーション・センター (UNIX、Windows、z/OS)

「レプリケーション・コントロール表の作成」ノートブックを使用します。詳しくは、レプリケーション・センターのヘルプを参照してください。

CRTDPRTBL システム・コマンド (OS/400)

パラメーター記述およびコマンド構文の詳細については、430 ページの『CRTDPRTBL: レプリケーション・コントロール表の作成 (OS/400)』を参照してください。

2. 次の方法のいずれかを使用して、既存のキャプチャー・スキーマを使用しているキャプチャー・プログラムを停止します。(キャプチャー・プログラムが実行されていない場合は、このステップはスキップしてください):

レプリケーション・センター

「キャプチャーの停止」ウィンドウを使用します。詳しくは、レプリケーション・センターのヘルプを参照してください。

asnccmd システム・コマンド (UNIX、Windows、z/OS)

stop パラメーターを使用します。パラメーター記述およびコマンド構文の詳細については、349 ページの『asnccmd: キャプチャーの操作』を参照してください。

ENDDPRCAP システム・コマンド (OS/400)

パラメーター記述およびコマンド構文の詳細については、434 ページの『ENDDPRCAP: キャプチャー・プログラムの停止 (OS/400)』を参照してください。

3. 次の方法を使用して、関連するすべてのサブスクリプション・セットを非活動化します。

レプリケーション・センター

「サブスクリプション・セット」フォルダーで内容ペインの中のアクティブなサブスクリプション・セットを右マウス・ボタンでクリックし、「非アクティブ化」を選択します。詳しくは、レプリケーション・センターのヘルプを参照してください。

4. アプライ・コントロール・サーバーから次の SQL ステートメントを実行して、このキャプチャー・スキーマに属するソース表を持つ関連するサブスクリプション・セットのキャプチャー・スキーマ名を変更します。

```
UPDATE ASN.IBMSNAP_SUBS_SET
  SET CAPTURE_SCHEMA = 'NewSchema'
WHERE
  CAPTURE_SCHEMA = 'ExistingSchema';
```

この *NewSchema* は新しいキャプチャー・スキーマ名であり、*ExistingSchema* は、変更しようとしているキャプチャー・スキーマの名前です。

5. このキャプチャー・スキーマに登録されたターゲット表 (たとえば CCD 表またはレプリカ・タイプ表) を持つサブスクリプション・セットを作成してある場合には、アプライ・コントロール・サーバーから次の SQL ステートメントを実行して、これらのサブスクリプション・セットのターゲット・スキーマ名を変更します。

```
UPDATE ASN.IBMSNAP_SUBS_SET
  SET TGT_CAPTURE_SCHEMA = 'NewSchema'
WHERE
  TGT_CAPTURE_SCHEMA = 'ExistingSchema';
```

この *NewSchema* は新しいキャプチャー・スキーマ名であり、*ExistingSchema* は、変更しようとしているキャプチャー・スキーマの名前です。

6. キャプチャー・コントロール・サーバーから SQL ステートメントを実行して、既存のキャプチャー・コントロール表のそれぞれから、ステップ 1 で作成した対応する新しいキャプチャー・コントロール表のそれぞれにアクティブ情報をコピーします。たとえば、新しい登録 (IBMSNAP_REGISTER) 表にアクティブ情報をコピーするには、次のようにします。

```
INSERT INTO NewSchema.IBMSNAP_REGISTER
  SELECT * FROM
    ExistingSchema.IBMSNAP_REGISTER;
```

この *NewSchema* は新しいキャプチャー・スキーマ名であり、*ExistingSchema* は、変更しようとしているキャプチャー・スキーマの名前です。

次の表を含めて、既存のキャプチャー・コントロール表のそれぞれに対してこのステップを繰り返します。

- IBMSNAP_CAPMON
- IBMSNAP_CAPPARMS
- IBMSNAP_CAPTRACE
- IBMSNAP_PRUNCNTL
- IBMSNAP_PRUNE_SET
- IBMSNAP_REG_EXT (OS/400 のみ)
- IBMSNAP_REGISTER
- IBMSNAP_RESTART
- IBMSNAP_SIGNAL
- IBMSNAP_UOW

(IBMSNAP_CAPENQ [UNIX、Windows、z/OS の場合] または IBMSNAP_PRUNE_LOCK コントロール表には行が含まれていないため、これらの表についてはこのステップを繰り返す必要はありません。)

CD 表は変更しないでください。。

7. 既存のスキーマ、および関連するキャプチャー・コントロール表をドロップするには、次の方法を使用します。

レプリケーション・センター

「キャプチャー・コントロール・サーバー」フォルダーで、キャプチャー・コントロール表をドロップするデータベースを右マウス・ボタンでクリックし、「キャプチャー・コントロール表のドロップ」を選択します。詳しくは、レプリケーション・センターのヘルプを参照してください。

8. 新しいスキーマ名を使用してキャプチャー・プログラムを再始動するには、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

「キャプチャーの開始」ウィンドウを使用します。詳しくは、レプリケーション・センターのヘルプを参照してください。

asncap システム・コマンド (UNIX、Windows、z/OS)

capture_schema=*NewSchema* および **startmode**=*warmsi* または *warmns* パラメーター・オプションを使用してください。パラメーター記述およびコマンド構文の詳細については、342 ページの『asncap: キャプチャーの始動』を参照してください。

STRDPRCAP システム・コマンド (OS/400)

RESTART(*YES) パラメーターを使用します。パラメーター記述およびコマンド構文の詳細については、468 ページの『STRDPRCAP: キャプチャー・プログラムの始動 (OS/400)』を参照してください。

9. 次の方法を使用して、関連するサブスクリプション・セットを再活動化します。

レプリケーション・センター

「サブスクリプション・セット」フォルダーで内容ペインの中の非活動化されたサブスクリプション・セットを右マウス・ボタンでクリックし、「活動化」を選択します。詳しくは、レプリケーション・センターのヘルプを参照してください。

新規サブスクリプション・セットの作成

既存の登録済みオブジェクトについては、いつでも新しいサブスクリプション・セットを作成し、新しいサブスクリプション・セット・メンバーをセットに追加できます。

この手順は、サブスクリプション・セット・メンバーを含む、または含まない、新しいサブスクリプション・セットの追加に使用されます。

前提条件:

新しいサブスクリプション・セットを作成する前に、ソースとして使用する表またはビューを登録します。

手順:

次の方法のいずれかを使用して、新しいサブスクリプション・セットを作成します。

レプリケーション・センター

「サブスクリプション・セットの作成」ノートブックを使用します。詳しくは、レプリケーション・センターのヘルプを参照してください。

ADDDPRSUB システム・コマンド (OS/400)

パラメーター記述およびコマンド構文の詳細については、395 ページの『ADDDPRSUB: DPR サブスクリプション・セットの追加 (OS/400)』を参照してください。

追加情報については、69 ページの『第 4 章 SQL レプリケーションのソースのサブスクライブ』を参照してください。

重要: 対応するアプライ・プログラムがアクティブである場合は、サブスクリプション・セットが完全に定義されるまで、新しいサブスクリプション・セットをアクティブ化しないでください。

既存のサブスクリプション・セットに新しいサブスクリプション・セット・メンバーを追加する

手順:

既存のサブスクリプション・セットに新しいサブスクリプション・セット・メンバーを追加するには、レプリケーション・センターを使用します。

レプリケーション・センター

「サブスクリプション・セットにメンバーを追加する」ノートブックを使用します。詳しくは、レプリケーション・センターのヘルプを参照してください。

アプライ・プログラムは、次のアプライ・サイクルのセットに追加される新しいメンバーで、フル・リフレッシュを実行します。すべてのターゲット表に対するレプリケーション変更は、後続のサイクルで続行します。 **opt4one** 変数を *y* に設定してアプライ・プログラムを実行している場合には、新しいメンバーがアプライによって処理される前に、アプライ・プログラムを停止してから再始動する必要があります。

既存のサブスクリプション・セットでサブスクリプション・セット・メンバーを使用不可にする

サブスクリプション・セット中の表に対するレプリケーションの問題がある場合、アプライ・プログラムは `IBMSNAP_APPLYTRAIL` 表にエラー・メッセージを入れ、そのサブスクリプション・セットのメンバーはこのアプライ・サイクルでは処理されなくなります。アプライ・プログラムが障害のあるサブスクリプション・セット・メンバーを無視して、残りのサブスクリプション・セットの処理を続行できるようにするには、障害のあるサブスクリプション・セット・メンバーを使用不可にする必要があります。サブスクリプション・セット・メンバーを使用不可にするには、次の SQL UPDATE ステートメントを使用します。

```
UPDATE ASN.IBMSNAP_SUBS_MEMBR
  SET MEMBER_STATE = 'D'
  WHERE APPLY_QUAL = apply_qualifier
        SET_NAME = set_name
        WHOS_ON_FIRST = whos_on_first
        SOURCE_OWNER = source_owner
        SOURCE_TABLE = source_table
        SOURCE_VIEW_QUAL = source_view_qualifier
        TARGET_OWNER = target_owner
        TARGET_TABLE = target_table
```

アプライ・プログラムは、メンバーが再度使用可能になるまでこのメンバーを処理しません。

既存のサブスクリプション・セットに対してサブスクリプション・セット・メンバーを使用可能にする

`MEMBER_STATE` を 'N' (新規) に変更することによって、サブスクリプション・セットの使用不可のメンバーを追加したり再度使用可能にしたりすることができます。サブスクリプション・セット・メンバーを再度使用可能にするには、次の SQL UPDATE ステートメントを使用します。

```
UPDATE ASN.IBMSNAP_SUBS_MEMBR
  SET MEMBER_STATE = 'N'
  WHERE APPLY_QUAL = apply_qualifier
        SET_NAME = set_name
        WHOS_ON_FIRST = whos_on_first
        SOURCE_OWNER = source_owner
```

```
SOURCE_TABLE = source_table
SOURCE_VIEW_QUAL = source_view_qualifier
TARGET_OWNER = target_owner
TARGET_TABLE = target_table
```

サブスクリプション・セットの属性の変更

既存のサブスクリプション・セットの属性を変更する必要があることもあります。ユーザーによる変更が考えられる属性には次のものがあります。

- 更新を適用するスケジュール (時間ベースのレプリケーションまたはイベントベースのレプリケーション)
- サブスクリプション・ステートメント
- サブスクリプション・セット・メンバーの WHERE 文節述部
- コミット・カウント
- データ・ブロック値 (MAX_SYNCH_MINUTES)

手順:

サブスクリプション・セットの属性を変更するには、レプリケーション・センターを使用して、以下のステップを実行します。

1. サブスクリプション・セットを非活動化します。
2. サブスクリプション・セットおよびサブスクリプション・セット・メンバーを変更します。
3. サブスクリプション・セットを再活動化します。

最初にサブスクリプション・セットを非活動化することにより、アプライ・プログラムをアクティブ状態で実行したまま、ユーザーが変更を入力している間は、アプライ・プログラムがこのサブスクリプション・セットを処理することを防止できます。アプライ・プログラムは、サブスクリプション・セットが再活動化された後、次のアプライ・サイクル中に、サブスクリプション・セットの変更を認識します。

注: アプライ・プログラムの **opt4one** パラメーターを **y** に設定すると、アプライ・プログラムを停止して再始動しないかぎり、変更は認識されません (UNIX、Windows、z/OS)。

サブスクリプション・セット名の変更

サブスクリプション・セットおよびそのすべてのメンバーをドロップしてから再作成する必要なしに、サブスクリプション・セットの名前を変更するには、以下の手順を使用します。

前提条件:

これらの SQL ステートメントを実行する前に、DB2 レプリケーション・コントロール表の構造、およびシステムで定義されているサブスクリプション・セットについてよく調べておいてください。

UNIX、Windows、z/OS の場合: モニター定義が設定されているか、変更しようとしているサブスクリプション・セットでアラート条件を検出するようにレプリケーション・アラート・モニター・プログラムを始動してある場合には、これらのモニター定義をドロップしてください。サブスクリプション・セット名を変更した後、

レプリケーション・センターを使用してモニター定義を再作成します。次に、**asnmcmd** システム・コマンドで **reinit** パラメーターを使用して、関連するレプリケーション・アラート・モニター・プログラムを再初期化できます。この代わりに、**asnmcmd** システム・コマンドで **stop** 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・プログラムを停止してから、**asnmon** システム・コマンドを使用してプログラムを再始動することもできます。

使用する新しいサブスクリプション・セット名を決定します。

手順:

1. 次の方法を使用して、変更するサブスクリプション・セットを非活動化します。

レプリケーション・センター

「サブスクリプション・セット」フォルダーで内容ペインの中のアクティブなサブスクリプション・セットを右マウス・ボタンでクリックし、「非アクティブ化」を選択します。詳しくは、レプリケーション・センターのヘルプを参照してください。

2. アプライ・コントロール・サーバーから次の SQL ステートメントを実行して、サブスクリプション・セット (IBMSNAP_SUBS_SET) 表、サブスクリプション・メンバー (IBMSNAP_SUBS_MEMBR) 表、およびサブスクリプション列 (IBMSNAP_SUBS_COLS) 表の中のサブスクリプション・セットの名前を変更します。

```
UPDATE ASN.IBMSNAP_SUBS_SET
  SET SET_NAME      = 'NewSetName'
  WHERE
    APPLY_QUAL      = 'ApplyQual' AND
    SET_NAME        = 'ExistSetName' AND
    WHOS_ON_FIRST  = 'Val';
UPDATE ASN.IBMSNAP_SUBS_MEMBR
  SET SET_NAME      = 'NewSetName'
  WHERE
    APPLY_QUAL      = 'ApplyQual' AND
    SET_NAME        = 'ExistSetName' AND
    WHOS_ON_FIRST  = 'Val';
UPDATE ASN.IBMSNAP_SUBS_COLS
  SET SET_NAME      = 'NewSetName'
  WHERE
    APPLY_QUAL      = 'ApplyQual' AND
    SET_NAME        = 'ExistSetName' AND
    WHOS_ON_FIRST  = 'Val';
```

この *NewSetName* は新しいサブスクリプション・セット名であり、*ApplyQual* はアプライ修飾子であり、*ExistSetName* はサブスクリプション・セットの既存の名前であり、*Val* は F または S のいずれかです。

3. このサブスクリプション・セットが事前または事後に実行される SQL ステートメントまたはプロシージャ呼び出しを使用する場合は、アプライ・コントロール・サーバーから次の SQL スクリプトを実行して、サブスクリプション・ステートメント (IBMSNAP_SUBS_STMTS) 表の中のサブスクリプション・セット名を変更します。

```
UPDATE ASN.IBMSNAP_SUBS_STMTS
  SET SET_NAME      = 'NewSetName'
  WHERE
    APPLY_QUAL      = 'ApplyQual' AND
    SET_NAME        = 'ExistSetName' AND
    WHOS_ON_FIRST  = 'Val';
```

この *NewSetName* は新しいサブスクリプション・セット名であり、*ApplyQual* はアプライ修飾子であり、*ExistSetName* はサブスクリプション・セットの既存の名前であり、*Val* は F または S のいずれかです。

4. キャプチャー・コントロール・サーバーから次の SQL ステートメントを実行して、整理セット (IBMSNAP_PRUNE_SET) 表および整理コントロール (IBMSNAP_PRUNCNTL) 表の中のサブスクリプション・セット名を変更します。

```
UPDATE Schema.IBMSNAP_PRUNE_SET
  SET SET_NAME      = 'NewSetName'
  WHERE
    APPLY_QUAL      = 'ApplyQual'      AND
    SET_NAME        = 'ExistSetName'   AND
    TARGET_SERVER   = 'Target_Server';
```

```
UPDATE Schema.IBMSNAP_PRUNCNTL
  SET SET_NAME      = 'NewSetName'
  WHERE
    APPLY_QUAL      = 'ApplyQual'      AND
    SET_NAME        = 'ExistSetName'   AND
    TARGET_SERVER   = 'Target_Server';
```

この *Schema* はキャプチャー・スキーマの新しい名前であり、*NewSetName* は新しいサブスクリプション・セットの名前であり、*ApplyQual* はアプライ修飾子であり、*ExistSetName* はサブスクリプション・セットの既存の名前であり、*Target_Server* はターゲット表のデータベース・ロケーションです。

5. **UNIX、Windows、zOS の場合: opt4one** を y に設定してアプライ・プログラムを実行している場合には、アプライ・プログラムを停止してから再始動します。
6. 次の方法を使用して、サブスクリプション・セットを再活動化します。

レプリケーション・センター

「サブスクリプション・セット」フォルダーで内容ペインの中の非活動化されたサブスクリプション・セットを右マウス・ボタンでクリックし、「活動化」を選択します。詳しくは、レプリケーション・センターのヘルプを参照してください。

サブスクリプション・セットの分割

以下の手順を使用すると、サブスクリプション・セット情報を除去して再作成する必要なしに、サブスクリプション・セットを複数のサブスクリプション・セットに分割できます。

前提条件:

これらの SQL ステートメントを実行する前に、DB2 レプリケーション・コントロール表の構造、およびシステムで定義されているサブスクリプション・セットについてよく調べておいてください。

分割するサブスクリプション・セットのサブスクリプション・セット・メンバーを識別し、これらのサブスクリプション・セット・メンバーに関連するソース表およびターゲット表を判別します。

分割するサブスクリプション・セットのキャプチャー・コントロール・サーバー、ターゲット・サーバー、およびアプライ・コントロール・サーバーを識別します。この手順を使用して作成する新しいサブスクリプション・セットでは、これらのキャプチャー・コントロール・サーバー、ターゲット・サーバー、およびアプライ・コントロール・サーバーのロケーションを使用する必要があります。

UNIX、Windows、z/OS の場合: モニター定義が設定されているか、分割しようとしているサブスクリプション・セットでアラート条件を検出するようにレプリケーション・アラート・モニター・プログラムを始動してある場合には、これらのモニター定義をドロップしてください。サブスクリプション・セットを分割した後、レプリケーション・センターを使用して、モニター定義を再作成します。次に、**asnmcmd** システム・コマンドで **reinit** パラメーターを使用して、関連するレプリケーション・アラート・モニター・プログラムを再初期化できます。この代わりに、**asnmcmd** システム・コマンドで **stop** 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・プログラムを停止してから、**asnmon** システム・コマンドを使用してプログラムを再始動することもできます。

手順:

1. 次の方法を使用して、分割するサブスクリプション・セットを非活動化します。

レプリケーション・センター

「サブスクリプション・セット」フォルダーで内容ペインの中のアクティブなサブスクリプション・セットを右マウス・ボタンでクリックし、「非アクティブ化」を選択します。詳しくは、レプリケーション・センターのヘルプを参照してください。

2. 次の方法のいずれかを使用して、新しいサブスクリプション・セットを作成します。

レプリケーション・センター

「サブスクリプション・セットの作成」ノートブックを使用します。詳しくは、レプリケーション・センターのヘルプを参照してください。

ADDDPRSUB システム・コマンド (OS/400)

SRCTBL(*NONE)、TGTTBL(*NONE)、および ACTIVATE(*NO) パラメーター・オプションを使用します。パラメーター記述およびコマンド構文の詳細については、395 ページの『ADDDPRSUB: DPR サブスクリプション・セットの追加 (OS/400)』を参照してください。

これらの方法は両方とも、サブスクリプション・セット (IBMSNAP_SUBS_SET) 表内に新しい行を作成します。

この新しいサブスクリプション・セットは非アクティブなままおいておきます。

3. アプライ・コントロール・サーバーから次の SQL ステートメントを実行して、既存のサブスクリプション・セットから IBMSNAP_SUBS_SET 表の中の新しいサブスクリプション・セットの行に情報をコピーします。

```
UPDATE ASN.IBMSNAP_SUBS_SET
SET STATUS =
(SELECT STATUS FROM ASN.IBMSNAP_SUBS_SET B
WHERE APPLY_QUAL = 'ApplyQual' AND
```

```

                SET_NAME      = 'ExistName' AND
                WHOS_ON_FIRST = 'Val'),
    LASTRUN      =
    (SELECT LASTRUN FROM ASN.IBMSNAP_SUBS_SET B
     WHERE APPLY_QUAL = 'ApplyQual' AND
           SET_NAME    = 'ExistName' AND
           WHOS_ON_FIRST = 'Val'),
    SYNCHPOINT   =
    (SELECT SYNCHPOINT FROM ASN.IBMSNAP_SUBS_SET B
     WHERE APPLY_QUAL = 'ApplyQual' AND
           SET_NAME    = 'ExistName' AND
           WHOS_ON_FIRST = 'Val'),
    SYNCHTIME    =
    (SELECT SYNCHTIME FROM ASN.IBMSNAP_SUBS_SET B
     WHERE APPLY_QUAL = 'ApplyQual' AND
           SET_NAME    = 'ExistName' AND
           WHOS_ON_FIRST = 'Val'),
    LASTSUCCESS  =
    (SELECT LASTSUCCESS FROM ASN.IBMSNAP_SUBS_SET B
     WHERE APPLY_QUAL = 'ApplyQual' AND
           SET_NAME    = 'ExistName' AND
           WHOS_ON_FIRST = 'Val')
WHERE
    APPLY_QUAL = 'ApplyQual' AND
    SET_NAME   = 'NewName'   AND
    WHOS_ON_FIRST = 'Val';

```

この *ApplyQual* はアプライ修飾子であり、*ExistName* は分割する既存のサブスクリプション・セットの名前であり、*Val* は F または S のいずれかであり、*NewName* は、ユーザーが作成している新しいサブスクリプション・セットの名前です。

4. キャプチャー・コントロール・サーバーから次の SQL ステートメントを実行して、整理セット (IBMSNAP_PRUNE_SET) 表に新しいサブスクリプション・セット用の新しい行を挿入します。

```

INSERT INTO Schema.IBMSNAP_PRUNE_SET
    (APPLY_QUALIFIER,
     SET_NAME,
     TARGET_SERVER,
     SYNCHTIME,
     SYNCHPOINT
    VALUES ('ApplyQual',
            'NewName',
            'Target_Server',
            NULL,
            x'00000000000000000000');

```

この *Schema* はキャプチャー・スキーマの名前であり、*ApplyQual* はアプライ修飾子であり、*NewName* は作成している新しいサブスクリプション・セットの名前であり、*Target_Server* はターゲット表のデータベース・ロケーションです。

5. キャプチャー・コントロール・サーバーから次の SQL ステートメントを実行して、IBMSNAP_PRUNE_SET 表の中の既存のサブスクリプション・セットの行から新しいサブスクリプション・セットの行に情報をコピーします。

```

UPDATE Schema.IBMSNAP_PRUNE_SET
    SET SYNCHPOINT =
        (SELECT SYNCHPOINT FROM Schema.IBMSNAP_PRUNE_SET B
         WHERE APPLY_QUAL = 'ApplyQual' AND
               SET_NAME   = 'ExistName' AND
               TARGET_SERVER = 'Target_Server'),
    SYNCHTIME =

```

```

        (SELECT SYNCHTIME FROM Schema.IBMSNAP_PRUNE_SET B
         WHERE APPLY_QUAL = 'ApplyQual' AND
              SET_NAME    = 'ExistName' AND
              TARGET_SERVER = 'Target_Server')
WHERE
    APPLY_QUAL = 'ApplyQual' AND
    SET_NAME   = 'NewName'   AND
    TARGET_SERVER = 'Target_Server';

```

この *Schema* はキャプチャー・スキーマの名前であり、*ApplyQual* はアプライ修飾子であり、*ExistName* は分割する既存のサブスクリプション・セットの名前であり、*Target_Server* はターゲット表のデータベース・ロケーションであり、*NewName* はユーザーが作成しようとしている新しいサブスクリプション・セットの名前です。

6. アプライ・コントロール・サーバーから次の SQL ステートメントを実行して、サブスクリプション・メンバー (IBMSNAP_SUBS_MEMBR) 表およびサブスクリプション列 (IBMSNAP_SUBS_COLS) 表で、新しいサブスクリプション・セットに移動する各 サブスクリプション・セット・メンバーのサブスクリプション・セット名を変更します。

```

UPDATE ASN.IBMSNAP_SUBS_MEMBR
  SET SET_NAME = 'NewName'
WHERE
  APPLY_QUAL = 'ApplyQual' AND
  SET_NAME   = 'ExistName'   AND
  WHOS_ON_FIRST = 'Val'     AND
  SOURCE_OWNER = 'SrcSchema' AND
  SOURCE_TABLE = 'SrcTbl'   AND
  SOURCE_VIEW_QUAL = SrcVwQual AND
  TARGET_OWNER = 'TgtSchema' AND
  TARGET_TABLE = 'TgtTbl';

UPDATE ASN.IBMSNAP_SUBS_COLS
  SET SET_NAME = 'NewName'
WHERE
  APPLY_QUAL = 'ApplyQual' AND
  SET_NAME   = 'ExistName'   AND
  WHOS_ON_FIRST = 'Val'     AND
  TARGET_OWNER = 'TgtSchema' AND
  TARGET_TABLE = 'TgtTbl';

```

この *NewName* はユーザーが作成している新しいサブスクリプション・セットであり、*ApplyQual* はアプライ修飾子であり、*ExistName* は分割する既存のサブスクリプション・セットであり、*Val* は F または S のいずれかであり、*SrcSchema* はソース表スキーマであり、*SrcTbl* はソース表名であり、*SrcVwQual* はこのソース表のソース・ビュー修飾子であり、*TgtSchema* はターゲット表のスキーマであり、*TgtTbl* はターゲット表名です。

新しいサブスクリプション・セットに移動するサブスクリプション・セット・メンバーごとにこのステップを繰り返します。

7. 分割するサブスクリプション・セットが事前または事後に実行される SQL ステートメントまたはプロシージャ呼び出しを使用する場合は、サブスクリプション・ステートメント (IBMSNAP_SUBS_STMTS) 表の中で適切なステートメントを新しいサブスクリプション・セットに移動します。
 - a. アプライ・コントロール・サーバーから次の SQL スクリプトを実行して、ステートメントを移動します。

```

UPDATE ASN.IBMSNAP_SUBS_STMTS
  SET SET_NAME      = 'NewName'
  WHERE
    APPLY_QUAL      = 'ApplyQual' AND
    SET_NAME        = 'ExistName'   AND
    WHOS_ON_FIRST   = 'Val'         AND
    STMT_NUMBER     in (Stmt1,Stmt2,..Stmtn);

```

この *NewName* はユーザーが作成している新しいサブスクリプション・セットの名前であり、*ApplyQual* はアプライ修飾子であり、*ExistName* は分割する既存のサブスクリプション・セットの名前であり、*Val* は F または S のいずれかであり、*Stmt1*、*Stmt2*、および *Stmtn* は、新しいサブスクリプション・セットに移動するステートメントの番号に対応します。

- b. 両方のサブスクリプション・セットについて、ステートメントの新しいカウントを反映するように、IBMSNAP_SUBS_SET 表の中の AUX_STMTS 列の値を調整します。必要であれば、重複しないようにステートメントの番号を変更します。
8. キャプチャー・コントロール・サーバーから次の SQL ステートメントを実行して、移動した各 サブスクリプション・セット・メンバーについて、整理コントロール (IBMSNAP_PRUNCNTL) 表の中のサブスクリプション・セットの名前を変更します。

```

UPDATE Schema.IBMSNAP_PRUNCNTL
  SET SET_NAME      = 'NewName'
  WHERE
    APPLY_QUAL      = 'ApplyQual' AND
    SET_NAME        = 'ExistName'   AND
    TARGET_SERVER   = 'Target_Server' AND
    SOURCE_OWNER    = 'SrcSchema'   AND
    SOURCE_TABLE    = 'SrcTbl'      AND
    SOURCE_VIEW_QUAL = SrcVwQual    AND
    TARGET_OWNER    = 'TgtSchema'   AND
    TARGET_TABLE    = 'TgtTbl';

```

この *Schema* はキャプチャー・スキーマの名前であり、*NewName* はステップ 2 で作成した新しいサブスクリプション・セットの名前であり、*ApplyQual* はアプライ修飾子であり、*ExistName* は分割された既存のサブスクリプション・セットの名前であり、*Target_Server* はターゲット表のデータベース・ローケーションであり、*SrcSchema* はソース表スキーマであり、*SrcTbl* はソース表名であり、*SrcVwQual* はこのレプリケーション・ソース表のソース・ビュー修飾子であり、*TgtSchema* はターゲット表スキーマであり、*TgtTbl* はターゲット表名です。

新しいサブスクリプション・セットに移動したサブスクリプション・セット・メンバーごとにこのステップを繰り返します。

9. **UNIX、Windows、z/OS の場合: opt4one** を *y* に設定してアプライ・プログラムを実行している場合には、アプライ・プログラムを停止してから再始動します。
10. 次の方法を使用して、両方のサブスクリプション・セットを再活動化します。

レプリケーション・センター

「サブスクリプション・セット」フォルダーで内容ペインの中の非活動

化された両方のサブスクリプション・セットを右マウス・ボタンでクリックし、「活動化」を選択します。詳しくは、レプリケーション・センターのヘルプを参照してください。

サブスクリプション・セットのマージ

2 つのサブスクリプション・セットを 1 つにマージする場合は以下の手順を使用します。2 つのサブスクリプション・セットの中のターゲット表が同じトランザクション整合性を持つようにしたいが、サブスクリプション・セット情報を削除して再作成したくない場合は、サブスクリプション・セットをマージできます。

前提条件:

これらの SQL ステートメントを実行する前に、DB2 レプリケーション・コントロール表の構造、およびシステムで定義されているサブスクリプション・セットについてよく調べておいてください。

マージする各サブスクリプション・セットのキャプチャー・コントロール・サーバー、ターゲット・サーバー、およびアプライ・コントロール・サーバーを識別します。マージするすべてのサブスクリプション・セットが、同じキャプチャー・コントロール・サーバー、ターゲット・サーバー、およびアプライ・コントロール・サーバーを使用して作成されていることを確認します。

UNIX、Windows、z/OS の場合: モニター定義が設定されているか、マージしようとしているサブスクリプション・セットでアラート条件を検出するようにレプリケーション・アラート・モニター・プログラムを始動してある場合には、これらのモニター定義をドロップしてください。サブスクリプション・セットをマージした後、レプリケーション・センターを使用して、モニター定義を再作成します。次に、**asnmcmd** システム・コマンドで **reinit** パラメーターを使用して、関連するレプリケーション・アラート・モニター・プログラムを再初期化できます。この代わりに、**asnmcmd** システム・コマンドで **stop** 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・プログラムを停止してから、**asnmon** システム・コマンドを使用してプログラムを再始動することもできます。

制約事項:

マージされる 2 つのサブスクリプション・セットは、同じキャプチャー・サーバーから、そして同じキャプチャー・スキーマを通じてソース・データを得ている必要があります。

手順:

1. 次の方法のいずれかを使用して、関連するキャプチャー・プログラムを停止します。

レプリケーション・センター

「キャプチャーの停止」ウィンドウを使用します。詳しくは、レプリケーション・センターのヘルプを参照してください。

asncmd システム・コマンド (Windows、UNIX、z/OS)

stop パラメーターを使用します。パラメーター記述およびコマンド構文の詳細については、349 ページの『asncmd: キャプチャーの操作』を参照してください。

ENDDPRCAP システム・コマンド (OS/400)

パラメーター記述およびコマンド構文の詳細については、434 ページの『ENDDPRCAP: キャプチャー・プログラムの停止 (OS/400)』を参照してください。

両方のサブスクリプション・セットが、サブスクリプション・セット (IBMSNAP_SUBS_SET) 表に示されたものと同じ同期点および同期時刻に達するまで待ちます。

重要: サブスクリプション・セットがマージされたときにデータが失われないように、2 つのサブスクリプション・セットでは、同じ同期点値までのソース・データが処理済みである必要があります。

ヒント: キャプチャー・プログラムを停止したくない場合は、シグナル (IBMSNAP_SIGNAL) 表の中に USER シグナルを挿入し、END_SYNCHPOINT (サブスクリプション・イベント [IBMSNAP_SUBS_EVENT] 表の中にある) を IBMSNAP_SIGNAL 表の中の SIGNAL_LSN 列の値に設定してイベントを生成し、このエンドポイントまでのデータのみが適用されるようにします。

2. 次の方法を使用して、2 つのサブスクリプション・セットを非活動化します。

レプリケーション・センター

「サブスクリプション・セット」フォルダーで内容ペインの中の 2 つのアクティブなサブスクリプション・セットを右マウス・ボタンでクリックし、「非アクティブ化」を選択します。詳しくは、レプリケーション・センターのヘルプを参照してください。

3. アプライ・コントロール・サーバーから次の SQL ステートメントを実行して、別のサブスクリプション・セット内に移動するサブスクリプション・セットに対応する行を、IBMSNAP_SUBS_SET 表から削除します。

```
DELETE FROM ASN.IBMSNAP_SUBS_SET
WHERE
  APPLY_QUAL = 'ApplyQual' AND
  SET_NAME = 'Subset_To_Move' AND
  WHOS_ON_FIRST = 'Val';
```

この *ApplyQual* はアプライ修飾子であり、*Subset_To_Move* は別の既存のサブスクリプション・セット内に移動するサブスクリプション・セットの名前であり、*Val* は F または S のいずれかです。

4. キャプチャー・コントロール・サーバーから次の SQL ステートメントを実行して、別のサブスクリプション・セット内に移動するサブスクリプション・セットに対応する行を、整理セット (IBMSNAP_PRUNE_SET) 表から削除します。

```
DELETE FROM Schema.IBMSNAP_PRUNE_SET
WHERE
  APPLY_QUAL = 'ApplyQual' AND
  SET_NAME = 'Subset_To_Move' AND
  TARGET_SERVER = 'Target_Server' ;
```

この *Schema* はキャプチャー・スキーマの名前であり、*ApplyQual* はアプライ修飾子であり、*Subset_To_Move* は別の既存のサブスクリプション・セット内に

移動するサブスクリプション・セットの名前であり、 *Target_Server* はターゲット表のデータベース・ロケーションです。

5. アプライ・コントロール・サーバーから次の SQL ステートメントを実行して、サブスクリプション・メンバー (IBMSNAP_SUBS_MEMBR) 表およびサブスクリプション列 (IBMSNAP_SUBS_COLS) 表の中で、移動するサブスクリプション・セットの名前を、別のサブスクリプション・セットの名前に変更します。

```
UPDATE ASN.IBMSNAP_SUBS_MEMBR
  SET SET_NAME      = 'Existing_Merged_Subset'
  WHERE
    APPLY_QUAL      = 'ApplyQual' AND
    SET_NAME         = 'Subset_To_Move' AND
    WHOS_ON_FIRST   = 'Val';

UPDATE ASN.IBMSNAP_SUBS_COLS
  SET SET_NAME      = 'Existing_Merged_Subset'
  WHERE
    APPLY_QUAL      = 'ApplyQual' AND
    SET_NAME         = 'Subset_To_Move' AND
    WHOS_ON_FIRST   = 'Val';
```

この *Existing_Merged_Subset* は、移動するサブスクリプション・セットとマージされる既存のサブスクリプション・セットの名前であり、 *ApplyQual* はアプライ修飾子であり、 *Subset_To_Move* は既存のサブスクリプション・セット内に移動するサブスクリプション・セットの名前であり、 *Val* は F または S のいずれかです。

6. 移動するサブスクリプション・セットが事前または事後に実行される SQL ステートメントまたはプロシージャ呼び出しを使用する場合は、サブスクリプション・ステートメント (IBMSNAP_SUBS_STMTS) 表の中でサブスクリプション・セットの名前を変更します。
 - a. アプライ・コントロール・サーバーから次の SQL スクリプトを実行して、サブスクリプション・セットの名前を変更します。

```
UPDATE ASN.IBMSNAP_SUBS_STMTS
  SET SET_NAME      = 'Existing_Merged_Subset'
  WHERE
    APPLY_QUAL      = 'ApplyQual' AND
    SET_NAME         = 'Subset_To_Move' AND
    WHOS_ON_FIRST   = 'Val';
```

この *Existing_Merged_Subset* は、移動するサブスクリプション・セットとマージされる既存のサブスクリプション・セットの名前であり、 *ApplyQual* はアプライ修飾子であり、 *Subset_To_Move* は既存のサブスクリプション・セット内に移動するサブスクリプション・セットの名前であり、 *Val* は F または S のいずれかです。

- b. 既存のマージされたサブスクリプション・セット内のステートメントの新しいカウントを反映するように、IBMSNAP_SUBS_SET 表の中の AUX_STMTS 列の値を調整します。必要であれば、重複しないようにステートメントの番号を変更します。
7. キャプチャー・コントロール・サーバーから次の SQL ステートメントを実行して、整理コントロール (IBMSNAP_PRUNCNTL) 表の中で、移動したサブスクリプション・セットの名前を、マージされたサブスクリプション・セットの名前に変更します。

```

UPDATE Schema.IBMSNAP_PRUNCNTL
  SET SET_NAME      = 'Existing_Merged_Subset'
  WHERE
    APPLY_QUAL      = 'ApplyQual' AND
    SET_NAME        = 'Subset_To_Move' AND
    TARGET_SERVER   = 'Target_Server' ;

```

この *Schema* はキャプチャー・スキーマの名前であり、*Existing_Merged_Subset* は移動するサブスクリプション・セットとマージされる既存のサブスクリプション・セットの名前であり、*ApplyQual* はアプライ修飾子であり、*Subset_To_Move* は別の既存のサブスクリプション・セット内に移動するサブスクリプション・セットの名前であり、*Target_Server* はターゲット表のデータベース・ロケーションです。

8. **UNIX、Windows、z/OS の場合: opt4one** を *y* に設定してアプライ・プログラムを実行している場合には、アプライ・プログラムを停止してから再始動します。
9. 次の方法を使用して、マージしたサブスクリプション・セットを再活動化します。

レプリケーション・センター

「サブスクリプション・セット」フォルダーで内容ペインの中の非活動化されたサブスクリプション・セットを右マウス・ボタンでクリックし、「活動化」を選択します。詳しくは、レプリケーション・センターのヘルプを参照してください。

サブスクリプション・セットのアプライ修飾子の変更

サブスクリプション・セットのアプライ修飾子を変更する必要がある場合には、サブスクリプション・セットを削除および再作成することなく、SQL を使用して変更を行うことができます。

同じアプライ修飾子を使用する複数のサブスクリプション・セットがある場合には、アプライ・プログラムのワークロードのバランスを取るために、いくつかのサブスクリプション・セットを新しいアプライ修飾子に移動することも考えられます。

移動するサブスクリプション・セットごとに この手順の SQL ステートメントを実行する必要があります。

前提条件:

これらの SQL ステートメントを実行する前に、DB2 レプリケーション・コントロール表の構造、およびシステムで定義されているサブスクリプション・セットについてよく調べておいてください。

また、次の情報も確認してください。

- 新しいアプライ修飾子の名前。(詳細については、327 ページの『第 17 章 SQL レプリケーション・オブジェクトの命名規則』を参照してください。)
- 既存のアプライ修飾子から新しいアプライ修飾子に移動するサブスクリプション・セット。

- これらのサブスクリプション・セットに定義されている、事前または事後に実行される SQL ステートメントまたはプロシージャ呼び出し。

UNIX、Windows、z/OS の場合: モニター定義が設定されているか、変更しようとしているアプライ修飾子の下でレプリケーション・アラート・モニター・プログラムを始動してある場合には、これらのモニター定義をドロップしてください。アプライ修飾子を変更した後、レプリケーション・センターを使用して、新しいアプライ修飾子名を使用してモニター定義を再作成します。次に、**asnmcmd** システム・コマンドで **reinit** パラメーターを使用して、関連するレプリケーション・アラート・モニター・プログラムを再初期化できます。この代わりに、**asnmcmd** システム・コマンドで **stop** 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・プログラムを停止してから、**asnmcmd** システム・コマンドを使用してプログラムを再始動することもできます。

手順:

1. 次の方法を使用して、変更するサブスクリプション・セットを非活動化します。

レプリケーション・センター

「サブスクリプション・セット」フォルダーで内容ペインの中のアクティブなサブスクリプション・セットを右マウス・ボタンでクリックし、「非アクティブ化」を選択します。詳しくは、レプリケーション・センターのヘルプを参照してください。

2. アプライ・コントロール・サーバーから次の SQL ステートメントを実行して、サブスクリプション・セット (IBMSNAP_SUBS_SET) 表、サブスクリプション・メンバー (IBMSNAP_SUBS_MEMBR) 表、およびサブスクリプション列 (IBMSNAP_SUBS_COLS) 表の中のサブスクリプション・セットのアプライ修飾子を変更します。

```
UPDATE ASN.IBMSNAP_SUBS_SET
  SET APPLY_QUAL = 'NewApplyQual'
WHERE
  APPLY_QUAL = 'ExistApplyQual' AND
  SET_NAME = 'Name' AND
  WHOS_ON_FIRST = 'Val';

UPDATE ASN.IBMSNAP_SUBS_MEMBR
  SET APPLY_QUAL = 'NewApplyQual'
WHERE
  APPLY_QUAL = 'ExistApplyQual' AND
  SET_NAME = 'Name' AND
  WHOS_ON_FIRST = 'Val';

UPDATE ASN.IBMSNAP_SUBS_COLS
  SET APPLY_QUAL = 'NewApplyQual'
WHERE
  APPLY_QUAL = 'ExistApplyQual' AND
  SET_NAME = 'Name' AND
  WHOS_ON_FIRST = 'Val';
```

この *NewApplyQual* は新しいアプライ修飾子であり、*ExistApplyQual* は既存のアプライ修飾子であり、*Name* はサブスクリプション・セットの名前であり、*Val* は F または S のいずれかです。

3. このサブスクリプション・セットが事前または事後に実行される SQL ステートメントまたはプロシージャ呼び出しを使用する場合は、アプライ・コントロー

ル・サーバーから次の SQL スクリプトを実行して、サブスクリプション・ステートメント (IBMSNAP_SUBS_STMTS) 表の中のサブスクリプション・セットのアプライ修飾子を変更します。

```
UPDATE ASN.IBMSNAP_SUBS_STMTS
  SET APPLY_QUAL = 'NewApplyQual'
 WHERE
  APPLY_QUAL = 'ExistApplyQual' AND
  SET_NAME = 'Name' AND
  WHOS_ON_FIRST = 'Val';
```

この *NewApplyQual* は新しいアプライ修飾子であり、*ExistApplyQual* は既存のアプライ修飾子であり、*Name* はサブスクリプション・セットの名前であり、*Val* は F または S のいずれかです。

4. キャプチャー・コントロール・サーバーから次の SQL ステートメントを実行して、整理セット (IBMSNAP_PRUNE_SET) 表および整理コントロール (IBMSNAP_PRUNCNTL) 表の中のサブスクリプション・セットのアプライ修飾子を変更します。

```
UPDATE Schema.IBMSNAP_PRUNE_SET
  SET APPLY_QUAL = 'NewApplyQual'
 WHERE
  APPLY_QUAL = 'ExistApplyQual' AND
  SET_NAME = 'Name' AND
  TARGET_SERVER = 'Target_Server';

UPDATE Schema.IBMSNAP_PRUNCNTL
  SET APPLY_QUAL = 'NewApplyQual'
 WHERE
  APPLY_QUAL = 'ExistApplyQual' AND
  SET_NAME = 'Name' AND
  TARGET_SERVER = 'Target_Server';
```

この *Schema* はキャプチャー・スキーマの名前であり、*NewApplyQual* は新しいアプライ修飾子であり、*ExistApplyQual* は既存のアプライ修飾子であり、*Name* はサブスクリプション・セットの名前であり、*Target_Server* はターゲット表のデータベース・ロケーションです。

5. 移動する残りのサブスクリプション・セットのそれぞれについて、ステップ 2 から 4 を繰り返します。
6. **UNIX、Windows、z/OS の場合: opt4one** を y に設定してアプライ・プログラムを実行している場合には、アプライ・プログラムを停止してから再始動します。
7. 次の方法を使用してサブスクリプション・セットを再活動化します。

レプリケーション・センター

「サブスクリプション・セット」フォルダーで内容ペインの中の非活動化されたサブスクリプション・セットを右マウス・ボタンでクリックし、「活動化」を選択します。詳しくは、レプリケーション・センターのヘルプを参照してください。

サブスクリプション・セットの非活動化

サブスクリプション・セットは、除去することなく、非活動化できます。サブスクリプション・セットを非活動化すると、アプライ・プログラムは、現在の処理サイクルを完了させてから、サブスクリプション・セットの処理を停止します。サブスクリプション・セットを非活動化させる時間の長さによっては、これらの非活動化したサブスクリプション・セットに関して特別の保守が必要になります。

短期間 一時的に非活動化したサブスクリプション・セットについては、特別な処理要件はありません。サブスクリプション・セットの属性を変更するとき、またはターゲット表の障害を修復するときには、サブスクリプション・セットを一時的に非活動化する必要があります。

サブスクリプション・セットを非活動化、変更、および再活動化するには、レプリケーション・センターを使用します。

より長い期間

現在は必要ないが、将来使用する可能性のあるサブスクリプション・セットは非活動化しておくことができます。しかし、このサブスクリプション・セットを長期間にわたって非活動化しておく必要がある場合には、累積した変更データによってキャプチャー・プログラムおよびアプライ・プログラムのパフォーマンスが影響を受ける可能性があるため、追加の処理が必要になります。

キャプチャー・プログラムは、整理プロセス時にはアプライ・プログラムからの情報を使用します。長い時間にわたりアプライ・プログラムが非アクティブになるか、サブスクリプション・セットが非アクティブ化されると、整理情報が不整合になり、非アクティブ化されたサブスクリプション・セットに関連してアクティブな登録が残っている場合には、作業単位 (UOW) 表および変更データ (CD) 表で迅速かつ効果的な整理を行うことができなくなります。この不整合な情報により、残りのアクティブなアプライ・プログラムのパフォーマンスが大幅に低下したり、整理処理が高価な CPU を不必要に消費してしまう可能性があります。UOW 表および CD 表は、最終的にはキャプチャー・プログラムの保持制限 (デフォルト値は 7 日) に基づいて整理されます。しかし、レプリケーション環境の規模によっては、この期間に大量のデータが累積される可能性があります。

このような整理の問題を防止するために、長期間にわたり非活動化しておく必要のあるサブスクリプション・セットについては、SQL を使用して整理情報をリセットすることができます。

前提条件:

これらの SQL ステートメントを実行する前に、DB2 レプリケーション・コントロール表の構造、およびシステムで定義されているサブスクリプション・セットについてよく調べておいてください。

手順:

1. レプリケーション・センターから、サブスクリプション・セットがアクティブでないことを確認します。
2. キャプチャー・コントロール・サーバーから次の SQL ステートメントを実行して、整理セット (IBMSNAP_PRUNE_SET) 表および整理コント

ロール (IBMSNAP_PRUNCNTL) 表で、非活動化したサブスクリプション・セットの整理情報をリセットします。

```
UPDATE Schema.IBMSNAP_PRUNE_SET
SET SYNCHPOINT = x'00000000000000000000' AND
    SYNCHTIME = NULL
WHERE
    APPLY_QUAL = 'ApplyQual' AND
    SET_NAME = 'Name' AND
    TARGET_SERVER = 'Target_Server';

UPDATE Schema.IBMSNAP_PRUNCNTL
SET SYNCHPOINT = NULL AND
    SYNCHTIME = NULL
WHERE
    APPLY_QUAL = 'ApplyQual' AND
    SET_NAME = 'Name' AND
    TARGET_SERVER = 'Target_Server';
```

この *Schema* はキャプチャー・スキーマの名前であり、*ApplyQual* はアプライ修飾子であり、*Name* はサブスクリプション・セットの名前であり、*Target_Server* はターゲット表のデータベース・ロケーションです。

登録済みオブジェクトに関連付けられたすべてのサブスクリプション・セットを非活動化したときには、登録済みオブジェクトも非活動化して、キャプチャー・プログラムが不必要にデータをキャプチャーしないようにする必要があります。

サブスクリプション・セットの除去

特定のサブスクリプション・セットでデータのレプリケーションが必要なくなったときには、サブスクリプション・セットを除去できます。しかし、除去するサブスクリプション・セットに対してアプライ・プログラムの処理が行われている場合には、アプライ・プログラムのジョブはアベンドし、このジョブの中の他のサブスクリプション・セットは、ユーザーがジョブを再始動するまで処理されません。

手順:

1. サブスクリプション・セットに対するアプライ・プログラムの現在の処理がすべて完了していることを確認するために、次の方法を使用して、サブスクリプション・セットを除去する前にサブスクリプション・セットを非活動化してください。

レプリケーション・センター

「サブスクリプション・セット」フォルダーで内容ペインの中のアクティブなサブスクリプション・セットを右マウス・ボタンでクリックし、「非アクティブ化」を選択します。詳しくは、レプリケーション・センターのヘルプを参照してください。

2. 次の方法のいずれかを使用して、非活動化したサブスクリプション・セットを除去します。

レプリケーション・センター

「サブスクリプション・セットの削除」ウィンドウを使用します。詳しくは、レプリケーション・センターのヘルプを参照してください。

RMVDPRSUB システム・コマンド (OS/400)

パラメーター記述およびコマンド構文の詳細については、454 ページの『RMVDPRSUB: DPR サブスクリプション・セットの除去 (OS/400)』を参照してください。

重要: キャプチャー・プログラムは、登録済みオブジェクトのすべてのサブスクリプション・セットが除去されても、変更データ (CD) 表でのデータのキャプチャーおよび行の書き込みを続けます。このキャプチャー・プログラムによる処理の続行を防止するために、サブスクリプション・セットを除去した後で、登録済みオブジェクトを非活動化または除去してください。

データベース・アプリケーション・イベントとレプリケーション・イベントの調整

シグナル (IBMSNAP_SIGNAL) 表に手動で行を挿入すれば、データベース・イベントとレプリケーション・イベントを調整できます。シグナルと呼ばれる、手動で挿入された IBMSNAP_SIGNAL 行は、実行中のキャプチャー・プログラムに特定のアクションを実行するように指示します。

USER タイプ・シグナルを使用したイベント END_SYNCHPOINT の設定

SIGNAL_TYPE 列の値を USER に設定することにより、DB2 リカバリー・ログ内の正確な時点を確認し、データベース・アプリケーション・イベントとレプリケーション・イベントを調整できます。

たとえば、オンライン・トランザクション処理 (OLTP) データを、別個に保守されるデータウェアハウスに複製する場合、ウェアハウスのデータは、随時の照会のために比較的安定したものにしておく必要があります。このため、ウェアハウスのデータでは、OLTP アプリケーションの労働日の特定の時点までに発生した変更のみを更新することにします。この場合のデータベース・アプリケーション・イベントは、労働日の論理的な終了時です。レプリケーション・イベントは、特定の日のビジネスがクローズしてから、次の日のビジネスがクローズするまでの変更の適用ということになります。サブスクリプション・セットはイベント処理に対してのみ構成されていると仮定します。

手順:

USER タイプのシグナルを作成するには、次のようにします。

1. IBMSNAP_SIGNAL 表に次の行を挿入して、キャプチャー USER タイプのシグナルを作成します。

```
INSERT INTO Schema.IBMSNAP_SIGNAL
    (signal_type,
     signal_subtype,
     signal_state)
VALUES ('USER',
       'USER APPLY EVENT SIGNAL',
       'P');
```

データベース・アプリケーション・イベントが発生したとき (この場合はアプリケーションの労働日の終わりに)、この SQL INSERT ステートメントを実行します。

キャプチャー・プログラムは、データベース・リカバリー・ログでこのレコードを検出した後で、このシグナル表のログ・レコードを処理します。キャプチャー・プログラムが反応するのは、この挿入に対応するコミット・レコードが検出された場合、つまりこのイベントがコミットされたことが検証された場合だけです。

USER タイプのシグナルがコミットされると、キャプチャー・プログラムは、処理される挿入ログ・レコードに対応する、次の IBMSNAP_SIGNAL 列値を更新します。

- SIGNAL_STATE = 'R' (キャプチャー・プログラムが受け取り済み)
 - SIGNAL_LSN = このシグナル行の挿入を含む DB2 の作業単位のコミット・ログ・レコードからのログ・シーケンス番号
2. 挿入されたシグナル行の SIGNAL_LSN 列にある値を、サブスクリプション・イベント (IBMSNAP_SUBS_EVENT) コントロール表の END_SYNCHPOINT 値として使用します。この新しい値は、新しい労働日のすべてのデータがキャプチャー・プログラムによってキャプチャーされたこと、そしてアプライ・プログラムが SIGNAL_LSN 列の値までのデータのみをフェッチしてアプライする必要があることをアプライ・プログラムに知らせます。

IBMSNAP_SIGNAL 表に対する更新トリガーを作成すれば、IBMSNAP_SUBS_EVENT への挿入を自動化できます。

```
CREATE TRIGGER EVENT_TRIG
NO CASCADE AFTER UPDATE ON Schema.IBMSNAP_SIGNAL
REFERENCING NEW AS N
FOR EACH ROW MODE DB2SQL
WHEN (N.SIGNAL_SUBTYPE = 'USER APPLY EVENT SIGNAL')
INSERT INTO ASN.IBMSNAP_SUBS_EVENT VALUES
('WH APPLY_EVENT',
(CURRENT_TIMESTAMP + 2 MINUTES),
N.SIGNAL_LSN,
null);
```

このトリガーは、キャプチャー・プログラムによって IBMSNAP_SIGNAL 表が更新されるたびに起動されます。SIGNAL_SUBTYPE 列が 'USER APPLY EVENT SIGNAL' に更新されると、トリガーは IBMSNAP_SUBS_EVENT 表に行を挿入します。この行は、2 分間が経過した後、最後の労働日 (キャプチャー・プログラムにより算出された SIGNAL_LSN 値の以前にコミットされている) 以降のフェッチおよび適用を処理する必要があることをアプライ・プログラムに指示します。

リモート・ジャーナリング用にジャーナル・シグナル表を作成する

iSeries オペレーティング・システムでは、シグナル表は、ソース表で使用される各ジャーナルに関連しています。これらの表はジャーナル・シグナル表と呼ばれ、グローバル・シグナル表 schema.IBMSNAP_SIGNAL と同じ構造です。

リモート・ジャーナリング環境でシグナルを使用するためには、ソース・システム上でジャーナル・シグナル表を作成する必要があります。また、場合によってはキ

キャプチャー・コントロール・サーバーで使用中のキャプチャー・スキーマと同じ名前をもつソース・システムでコレクションを作成する必要があります。

ジャーナル・シグナル表の名前は、`schema.IBMSNAP_SIGNAL_XXXX1_YYYY1` で、`XXXX1` はジャーナル・ライブラリー、`YYYY1` はキャプチャー・コントロール・サーバー上のリモート・ジャーナルのジャーナル名です。この表は、ソース・サーバー上のソース・ジャーナルに対してジャーナリングされる必要があります。

手順:

ソース・サーバー上のソース・ジャーナルに対してシグナル表をジャーナリングする方法は次のとおりです。

1. キャプチャー・コントロール・サーバー上に、キャプチャー・スキーマから取られた名前を持つコレクションが存在することを確認します。
2. 次のようにジャーナル・シグナル表 `schema.IBMSNAP_SIGNAL_XXXX1_YYYY1` を作成します。

```
CREATE TABLE schema/IBMSNAP_SIGNAL_XXXX1_YYYY1
(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,
 SIGNAL_LSN CHAR(10) FOR BIT DATA)
```

3. ジャーナル物理ファイルの終了 (**ENDJRNPf**) コマンドを使用して、ジャーナリングを終了します。

```
ENDJRNPf FILE(schema/IBMSNnnnnn)
```

ここで、`IBMSNnnnnn` は `IBMSNAP_SIGNAL_XXXX1_YYYY1` の短縮名です。

4. 物理ファイルのジャーナル開始 (**STRJRNPf**) コマンドを使って、ソース・ジャーナルのジャーナリングを開始します。

```
STRJRNPf FILE(schema/IBMSNnnnnn) JRN(XXXX2/YYYY2)
```

ここで、`XXXX2` はジャーナル・ライブラリー、`YYYY2` はソース・サーバー上のソース・ジャーナルのジャーナル名です。

キャプチャー CMD STOP シグナルの使用方法

`SIGNAL_TYPE` 列の値を `CMD` に設定し、`SIGNAL_SUBTYPE` 列の値を `STOP` に設定することにより、キャプチャー・プログラムの処理を、DB2 リカバリー・ログの正確な時点で停止できます。この機能は主に 2 つの目的で使用されます。

- 以前のログ・レコードを読み取れない状態にした、ソース表の変更に対してキャプチャー・プログラムを調整するため。このような状態は、ユーザーが表をドロップしてから再作成した場合、またはユーザーが `KEEPDICTIONARY` オプションを `YES` に設定しないで表を再編成したときに発生する可能性があります。
- 複製された分散データベース・システム間で共通のリカバリー点を調整するため。

ソース表の変更とキャプチャー・プログラムの調整

キャプチャー `CMD` タイプ `STOP` サブタイプのシグナルを使用して、キャプチャー・プログラムをシャットダウンして、ソース表の変更を調整できます。

手順:

ソース表の変更を調整するには次のようにします。

1. 次の SQL ステートメントを使用して、シグナル (IBMSNAP_SIGNAL) 表に行を挿入してキャプチャー CMD タイプ STOP サブタイプのシグナルを作成します。

```
INSERT INTO Schema.IBMSNAP_SIGNAL
    (signal_type,
     signal_subtype,
     signal_state)
VALUES ('CMD',
        'STOP',
        'P');
```

この行は、データベース・アプリケーション・イベントが発生したときに、ソース表のアクティビティーが静止した後、そしてログ・レコードに問題を発生させたアクティビティーが変更される前に、挿入する必要があります。

キャプチャー・プログラムは、データベース・リカバリー・ログでこのレコードを検出した後で、このシグナル表のログ・レコードを処理します。キャプチャー・プログラムが反応するのは、この挿入に対応するコミット・レコードが検出された場合、つまりこのイベントがコミットされたことが検証された場合だけです。

キャプチャー・プログラムは、この挿入された IBMSNAP_SIGNAL 行を含む DB2 作業単位のコミット・ログ・レコードより以前の、ログ上のトランザクションからのすべてのキャプチャーされたデータをコミットした後で、すべてのキャプチャー・スレッドを順番にシャットダウンします。キャプチャー・プログラムは終了する前に、処理される挿入ログ・レコードに対応する IBMSNAP_SIGNAL 表の行の中の次の値も更新します。

- SIGNAL_STATE = 'R' (キャプチャー・プログラムが受け取り済み)
- SIGNAL_LSN = このシグナル行の挿入を含む DB2 の作業単位のコミット・ログ・レコードからのログ・シーケンス番号

変更のあるソース表のすべてのログ・レコードは、終了時にキャプチャー・プログラムにより処理されます。

2. ユーザーのシナリオに従って、ソース表をドロップして再作成するか、KEEPDICTIONARY オプションを YES に設定せずにソース表を再編成および圧縮します。
3. 複製された列をドロップまたは変更した場合は、このソース表に対して作成された、対応する登録およびサブスクリプション・セットをここで変更する必要があります。必要であれば、影響を受けたサブスクリプション・セットが、現在停止中のキャプチャー・プログラムに追いつくのを待つことにより、アプライ・プログラムとの間でこのような変更をさらに調整することができます。サブスクリプション・セットは、サブスクリプション・セット (IBMSNAP_SUBS_SET) 表の中の SYNCHPOINT 列の値が Schema.IBMSNAP_RESTART 表の中の MAX_COMMITSEQ 列の値と等しくなると、キャプチャー・プログラムと同期します。

分散リカバリ一点の設定

キャプチャー CMD タイプ STOP サブタイプのシグナルを使用して、ソース・データベースおよびターゲット・データベースを同じリカバリ一点に設定し、共通整合点でデータベースをリカバリできます。

前提条件:

この手順を使用する前に、ターゲット・データベースにアプライ・コントロール表が作成されていることを確認してください。

また、IBMSNAP_SIGNAL 表に行を挿入する前に、ソース・データベースに対するすべてのアクティビティが静止していることを確認します。しかし、IBMSNAP_SIGNAL 表に行を挿入するまでは、データベース表のバックアップまたはイメージ・コピーを作成しないでください。

サブスクリプション・セットがイベント処理用の典型的な構成になっていない場合には、サブスクリプション・セットを一時的にイベント・ベースのタイミング用に設定する必要があります。次の SQL ステートメントを使用して、サブスクリプション・イベント (IBMSNAP_SUBS_EVENT) 表に行を挿入します。

```
INSERT INTO ASN.IBMSNAP_SUBS_EVENT
VALUES('RECOVERY_EVENT',
      CURRENT_TIMESTAMP + 2 MINUTES,
      SIGNAL_LSN_value,
      NULL);
```

この *SIGNAL_LSN_value* は、キャプチャー・プログラムによって設定され、IBMSNAP_SIGNAL 表の中に保管されるログ・シーケンス番号です。

手順:

分散リカバリ一点を設定するには、次のようにします。

1. 次の SQL ステートメントを使用して、IBMSNAP_SIGNAL 表に行を挿入してキャプチャー CMD タイプ STOP サブタイプのシグナルを作成します。

```
INSERT INTO Schema.IBMSNAP_SIGNAL
(signal_type,
 signal_subtype,
 signal_state)
VALUES('CMD',
      'STOP',
      'P');
```

キャプチャー・プログラムは、データベース・リカバリ・ログでこのレコードを検出した後で、このシグナル表のログ・レコードを処理します。キャプチャー・プログラムが反応するのは、この挿入に対応するコミット・レコードが検出された場合、つまりこのイベントがコミットされたことが検証された場合だけです。

キャプチャー・プログラムは、この挿入された IBMSNAP_SIGNAL 行を含む DB2 作業単位のコミット・ログ・レコードより以前の、ログ上のトランザクションからのすべてのキャプチャーされたデータをコミットした後で、すべてのキャプチャー・スレッドを順番にシャットダウンします。キャプチャー・プログラムは終了する前に、処理される挿入ログ・レコードに対応する IBMSNAP_SIGNAL 表の行の中の次の値も更新します。

- SIGNAL_STATE = 'R' (キャプチャー・プログラムが受け取り済み)
- SIGNAL_LSN = このシグナル行の挿入を含む DB2 の作業単位のコミット・ログ・レコードからのログ・シーケンス番号

ソース・データベースのすべてのログ・レコードは、終了時にキャプチャー・プログラムにより処理されます。

2. ソース・データベースのバックアップまたはイメージ・コピー・ユーティリティーを実行します。
3. 挿入した IBMSNAP_SIGNAL 表の行の SIGNAL_LSN 列の値を、IBMSNAP_SUBS_EVENT 表の中の END_SYNCHPOINT 値として使用します。この値は、バックアップ時点以前にコミットされたすべてのデータがキャプチャー・プログラムによってキャプチャーされたこと、そしてアプライ・プログラムが SIGNAL_LSN 列の値までのデータのみをフェッチしてアプライする必要があることをアプライ・プログラムに知らせます。

サブスクリプション・セットは、SIGNAL_LSN 値までのすべてのデータを処理します。

4. ターゲット・データベースのバックアップまたはイメージ・コピー・ユーティリティーを実行します。これでソース・データベースとターゲット・データベースは同じリカバリー点を持つことになるため、共通整合点で両方のデータベースをリカバリーできます。

アプライ・イベントが設定され、ソース・データベースのバックアップまたはイメージ・コピー・ユーティリティーのアクティビティーが完了したらすぐに、ソース・データベースのすべてのアクティビティーを再開できます。キャプチャー・プログラムも開始することができます。ターゲット・データベースのバックアップまたはイメージ・コピー・ユーティリティーのアクティビティーが完了したら、サブスクリプション・セットのスケジューリング・オプションを元の設定 (時間ベース、イベント・ベースまたはその両方) に戻すことができます。

iSeries オペレーティング・システムでは、STOP シグナルを送信して、1 つのジャーナル・ジョブまたはすべてのジャーナル・ジョブを停止できます。単一のジャーナル・ジョブを停止するには、そのジャーナルに指定されたシグナル表 (IBMSNAP_SIGNAL_xxxx_yyyy 表。xxxx はジャーナル・ライブラリー、yyyy はジャーナル名) にシグナルを挿入します。すべてのジャーナル・ジョブを停止するには、シグナルを表 schema.IBMSNAP_SIGNAL に挿入します。リモート・ジャーナル構成中の単一のジャーナル・ジョブを停止するには、ソース・サーバー上のジャーナル・シグナル表にシグナルを挿入します。リモート・ジャーナル構成でジャーナル・シグナル表を作成する方法に関する詳細は、238 ページの『リモート・ジャーナリング用にジャーナル・シグナル表を作成する』を参照してください。

アプライ・プログラム外部の CAPSTART ハンドシェーク・シグナルの実行

アプライ・プログラムがサブスクリプション・セットを使用して CD 表の変更をフェッチおよびアプライできるようにするためには、そのサブスクリプション・セット内の各サブスクリプション・セット・メンバーのキャプチャー・プログラムとアプライ・プログラムの間でハンドシェーク (同期化された通信) が必要になります。

アプライ・プログラムは、CMD タイプ CAPSTART サブタイプのシグナルをシグナル (IBMSNAP_SIGNAL) 表に挿入することにより、ハンドシェークを開始します。アプライ・プログラムは、コンプリートとして定義されているターゲット表でサブスクリプション・メンバーのフル・リフレッシュを実行する前に、このシグナルを挿入します。

手順:

CAPSTART ハンドシェーク・シグナルを実行するには、次のようにします。

- 次の SQL ステートメントを使用して、IBMSNAP_SIGNAL 表に行を挿入してキャプチャー CMD タイプ CAPSTART サブタイプのシグナルを作成します。

```
INSERT INTO Schema.IBMSNAP_SIGNAL
  (signal_type,
   signal_subtype,
   signal_input_in,
   signal_state)
VALUES ('CMD',
       'CAPSTART',
       mapid,
       'P');
```

この *mapid* は *Schema.IBMSNAP_PRUNCNTL* 表の *MAP_ID* 列の値であり、ハンドシェークを必要とするサブスクリプション・セット・メンバーの行に対応します。

注: 必要であれば、サブスクリプション・セット・メンバーのフル・リフレッシュを実行する前に、この SQL INSERT ステートメントを実行してください。

キャプチャー・プログラムは、データベース・リカバリー・ログでこのレコードを検出した後で、このシグナル表のログ・レコードを処理します。キャプチャー・プログラムが反応するのは、この挿入に対応するコミット・レコードが検出された場合、つまりこのイベントがコミットされたことが検証された場合だけです。

キャプチャー・プログラムは、登録済み表の以前の使用に基づいて、関連する登録がすでにメモリー内に入れているかどうかをチェックします。登録済み表が使用されていない場合、キャプチャー・プログラムは関連する登録情報をメモリー内に読み込み、この登録済み表がアクティブであり、使用中であることを示す値を登録 (IBMSNAP_REGISTER) 表の中に設定します。

キャプチャー・プログラムは登録済み表が使用中であるかどうかに関係なく、*Schema.IBMSNAP_PRUNCNTL* 表の中の関連する行の *SYNCHPOINT* 列および *SYNCHTIME* 列の値を、この挿入されたシグナル行を含む DB2 作業単位のコミット・ログ・レコードからのログ・シーケンス番号、およびこの同じコミット・ログ・レコードからのタイム・スタンプにそれぞれ設定します。

キャプチャー・プログラムは、処理される挿入ログ・レコードに対応する IBMSNAP_SIGNAL 表の行の中の次の値を更新します。

- SIGNAL_STATE = 'C' (キャプチャー・プログラムにより受け取り済みで完全)
- SIGNAL_LSN = このシグナル行の挿入を含む DB2 の作業単位のコミット・ログ・レコードからのログ・シーケンス番号

CAPSTOP シグナルの実行

登録の変更のキャプチャーを手動で停止したい場合は、CAPSTOP シグナルを開始できます。このシグナルは、登録を非活動化する場合、または登録を除去する前に使用できます。

手順:

CAPSTOP シグナルを実行するには、次のようにします。

1. 次の SQL ステートメントを使用して、IBMSNAP_SIGNAL 表に行を挿入してキャプチャー CMD タイプ CAPSTOP サブタイプのシグナルを作成します。

```
INSERT INTO Schema.IBMSNAP_SIGNAL
    (signal_type,
     signal_subtype,
     signal_input_in,
     signal_state)
VALUES('CMD',
       'CAPSTOP',
       source_owner.source_table,
       'P');
```

この *Schema* はキャプチャー・スキーマの名前であり、*source_owner.source_table* はキャプチャーされた変更を必要としなくなった表の完全修飾名です。

キャプチャー・プログラムは、データベース・リカバリー・ログでこのレコードを検出した後で、このシグナル表のログ・レコードを処理します。キャプチャー・プログラムが反応するのは、この挿入に対応するコミット・レコードが検出された場合、つまりこのイベントがコミットされたことが検証された場合だけです。

キャプチャー・プログラムは、登録済み表の以前の使用に基づいて、関連する登録がすでにメモリー内に入られているかどうかをチェックします。登録済み表が現在使用中でない場合には、キャプチャー・プログラムは CAPSTOP シグナルを無視します。

登録済み表が使用されている場合は、キャプチャー・プログラムはこの登録に関連するメモリーをクリアし、登録を非活動化します (IBMSNAP_REGISTER 表の STATE 列を 'I' に設定することにより)。その後キャプチャー・プログラムはこの登録済み表について変更のキャプチャーを停止します。

キャプチャー・プログラムは、処理される挿入ログ・レコードに対応する IBMSNAP_SIGNAL 表の行の中の次の列値を更新します。

- SIGNAL_STATE = 'C' (キャプチャー・プログラムにより受け取り済みで完全)
- SIGNAL_LSN = このシグナル行の挿入を含む DB2 の作業単位のコミット・ログ・レコードからのログ・シーケンス番号

2. オプション: 登録を除去します。

iSeries オペレーティング・システムでは、CAPSTOP シグナルを送信して、IBMSNAP_SIGNAL_XXXX_YYYY 表 (XXXX はジャーナル・ライブラリー、YYYY はジャーナル名) にシグナルを挿入することによって、登録のキャプチャー変更を停止す

することもできます。リモート・ジャーナル構成中の登録のキャプチャー変更を停止するには、ソース・サーバーで CAPSTOP シグナルを挿入します。リモート・ジャーナル構成でジャーナル・シグナル表を作成する方法に関する詳細は、238 ページの『リモート・ジャーナリング用にジャーナル・シグナル表を作成する』を参照してください。

別のシステムへのレプリケーション構成のプロモート

あるシステム (たとえばテスト・システム) で登録済みオブジェクトまたはサブスクリプション・セットを定義し、レプリケーション環境を別のシステム (たとえば実動システム) にコピーする必要がある場合には、レプリケーション・センターのプロモート関数を使用できます。これらの関数は、登録済みオブジェクトまたはサブスクリプション・セットをリバース・エンジニアリングして、適切なデータ定義言語 (DDL) およびデータ操作言語 (DML) 付きのスクリプト・ファイルを作成します。ソースの再登録またはサブスクリプション・セットの再作成を行う必要なしに、レプリケーション定義を別のデータベースにコピーできます。

たとえば、プロモート関数を使用して、リモートのターゲット・データベース用のサブスクリプション・セットを定義します。テスト環境でモデルになるターゲット・システムを定義した後、リモート・ターゲット・システム用のサブスクリプション・セットのスクリプトを作成できます (そして使用するアプライ修飾子の変更などを行うことができます)。こうしない場合、中央のコントロール・ポイントからはサポートされません。

重要: プロモート関数は宛先ターゲット・システムとは接続せず、そのシステムのレプリケーション構成パラメーターの検証を行いません。

3 つのプロモート関数があります。

登録済み表のプロモート

この関数は、指定された表の登録情報をプロモートします。この関数は、オプションで、基本表、索引および表スペース定義のプロモートも行います。プロモートする複数の表に対して異なるキャプチャー・スキーマおよび異なるサーバー名を指定できます。また、プロモートされたソース表に関連する変更データ (CD) 表のスキーマ名も変更できます。

複数の登録済み表を同時にプロモートできます。ユーザーが指定した新しいスキーマ名は、プロモートされるすべての表に適用されます。

この関数は、DB2 Universal Database バージョン 8 で登録された表しかプロモートしません。

登録済みビューのプロモート

この関数は、指定されたビューの登録情報をプロモートします。この関数は、オプションで、基本ビュー、登録抹消された基本表 (ビューのベースである)、索引および表スペース定義のプロモートも行います。プロモートする複数のビューに対して異なるキャプチャー・スキーマおよび異なるサーバー名を指定できます。また、プロモートされたソース・ビューに関連する CD ビュー、およびこれらの CD ビューのベースである CD 表のスキーマ名も変更できます。

複数の登録済みビューを同時にプロモートできます。ユーザーが指定した新しいスキーマ名は、プロモートされるすべてのビューに適用されます。

重要: プロモートするビューが登録済みソース表に基づくものである場合は、登録済み表のプロモート関数を使用して、登録済みのソース表を別個にプロモートする必要があります。これらの登録済みソース表が、登録済みビューのプロモート関数によって自動的にプロモートされることはありません。しかし、このビューのベースである、登録抹消された基本表は、必要であればこの関数によってプロモートされます。

サブスクリプション・セットのプロモート

この関数はサブスクリプション・セットをプロモートします。この関数により、データベース間でサブスクリプション・セット (すべてのサブスクリプション・セット・メンバーを含めて) をコピーできます。

サブスクリプション・セットのプロモート関数は、登録済み表のプロモート関数と一緒に使用する必要があります。

重要: プロモート関数を使用して、OS/400、UNIX、Windows、および z/OS オペレーティング・システム上の登録済みオブジェクトおよびサブスクリプション・セットをプロモートできます。プロモート関数は、同種のシステム間でのみレプリケーション定義をコピーします。たとえば、1 つの DB2 Universal Database for z/OS システムから別の DB2 Universal Database for z/OS システムへコピーします。

プロモート関数を使用して、DB2 以外のリレーショナル・データベースとの間でレプリケーション定義をコピーすることはできません。また、プロモート関数を使用して、OS/400 リモート・ジャーナルを含むレプリケーション定義をコピーすることもできません。

関連概念:

- 265 ページの『第 15 章 SQL レプリケーションの場合のレプリケーション・センターの使用法』

関連タスク:

- 43 ページの『第 3 章 表およびビューを SQL レプリケーション・ソースとして登録する』
- 69 ページの『第 4 章 SQL レプリケーションのソースのサブスクライブ』

関連資料:

- 543 ページの『*schema.IBMSNAP_SIGNAL*』

第 14 章 SQL レプリケーション環境の保守

この章では、データベース内にあり、DB2 レプリケーションに使用される、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。

DB2 レプリケーションはデータベース・システムと共同で処理を行うため、既存のデータベース・アクティビティーの変更は最小限ですみます。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。この章では、DB2 レプリケーションの 3 つの関数コンポーネントの保守の要件について説明します。

- 『ソース・システムの保守』
- 252 ページの『コントロール表の保守』
- 261 ページの『ターゲット表の保守』

ソース・システムの保守

レプリケーション・ソース・システムは、変更キャプチャー・メカニズム、複製するソース表 (OS/400 システムで使用されるリモート・ジャーナルを含む)、キャプチャー・プログラムから使用されるログ・データ、および DB2 以外のリレーショナル・データベース・ソースで使用されるキャプチャー・トリガーで構成されます。このセクションでは、ソース表とログ・ファイルを正しく保守する方法、そしてこれらの表およびファイルが常に DB2 レプリケーションからアクセス可能であるようにする方法を説明します。

ソース・オブジェクトの保守

レプリケーション・ソース・オブジェクトは、システム上の他のデータベース表およびビューと同じ保守を必要とする、データベース表およびビューです。これらのオブジェクトに関して、既存のユーティリティーおよび保守ルーチンを引き続き実行してください。

キャプチャー・プログラムおよびアプライ・プログラムが常に処理を進められるように、DB2 レプリケーションでのこれらのソース表の可用性を考慮する必要があります。DB2 レプリケーションは、ほとんどのレプリケーション処理においては、ソース表に直接アクセスする必要はありません。しかし、DB2 レプリケーションは、次の 2 つのアクションのいずれかが発生したときには、ソース表または表スペースにアクセスする必要があります。

- アプライ・プログラムがフル・リフレッシュを実行したとき。
- 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 のみ)。

フル・リフレッシュ時にレプリケーション・アプライ・プログラムの処理が阻害されないように、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してください。また、z/OS の場合は、ソース表が圧縮されている場合に、DB2 が圧縮されたログ・レコードの表スペースに対してラッチを入手できるように、ユーティリ

ティーがオンライン・モードで実行されることを確認してください。ユーティリティーおよび保守ルーチンが、データベース (または z/OS の場合は圧縮された表スペース) をオフラインにする必要のある排他モードで実行される場合は、レプリケーションでソース・オブジェクトを使用することはできません。

ソース・ログおよびジャーナル・レシーバーの保守および保存

DB2 リカバリー・ログには、DB2 リカバリー機能の提供と、実行中のキャプチャー・プログラムへの情報の提供という 2 つの目的があります。DB2 リカバリー、および DB2 レプリケーション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、キャプチャー・プログラムおよび DB2 が、ログまたはジャーナル・レシーバー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。

注: DB2 レプリケーションは、DB2 以外のリレーショナル・データベースからのログ・データは使用しません。

ログ・データの保存 (Linux、UNIX、Windows、z/OS)

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。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。

Linux、UNIX および Windows の場合: キャプチャー・プログラムがアーカイブ・ログからデータをリトリブできるように、ユーザー出力アーカイブを使用するようにデータベースを構成する必要があります。

DB2 の実行時は常にキャプチャー・プログラムを実行するようにしておけば、キャプチャー・プログラムは一般的に DB2 のリカバリー・ログに合わせた最新の状態になります。DB2 がアクティブになっている間は常にキャプチャー・プログラムを実行するか、ログ・レコードを 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、DB2 レプリケーションに合わせて、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。

手順:

キャプチャー・プログラムで使用するために保存する必要のあるログ・レコードと、削除できるログ・レコードを判別するには、次のようにします。

Linux、UNIX および Windows の場合:

1. 次の SQL ステートメントを実行し、再始動 (IBMSNAP_RESTART) 表から MIN_INFLIGHTSEQ 値を入手します。

```
SELECT MIN_INFLIGHTSEQ
FROM ASN.IBMSNAP_RESTART
WITH UR;
```


MIN_INFLIGHTSEQ 値が表示されます。(IBMSNAP_RESTART 表には行が 1 つしかありません。マルチパーティション環境では、各パーティションはログ・ファイルの独自のセットを保守するため、この手順は各パーティションに拡張される必要があります。IBMSNAP_PARTITIONINFO 表の SEQUENCE 列を使用して、各パーティションごとにこの情報を判別してください。)

MIN_INFLIGHTSEQ 値は、ビット・データ列の場合は char(10) 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
```

MIN_INFLIGHTSEQ 値の最後の 12 文字に注目してください。この例では次のようになっています。

```
123456123456
```

2. コマンド行から **db2 get db cfg** コマンドを入力し、アクティブ・ログ・ファイルのパスを入手します。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. DB2 コマンド行から **db2flsn** コマンドを入力し、MIN_INFLIGHTSEQ 値の最後の 12 文字を入力します。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 ファイルにアクセスする必要があります。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リトリブして表示します。以下に例を示します。

```
Given LSN is contained in the log file S000123.LOG
```

4. リトリブされたログ・ファイルの経過日数を確認してください。

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。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が、これより古いログは削除できます。

z/OS の場合、

1. 次の SQL ステートメントを実行し、再始動 (IBMSNAP_RESTART) 表から MIN_INFLIGHTSEQ 値を入手します。

```
SELECT MIN_INFLIGHTSEQ
FROM ASN.IBMSNAP_RESTART
WITH UR;
```

MIN_INFLIGHTSEQ 値が表示されます。(IBMSNAP_RESTART 表には行が 1 つしかありません。) 以下に例を示します。

```
000055551F031230000
```

常に 0000 である、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します。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. MIN_INFLIGHTSEQ 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 の範囲と比較します。
4. 自分の 12 文字のアクティブ・ログ番号の値が存在する行を探します。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・セット情報を確認します。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します。

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。

このログ・ファイルまたはデータ・セットの経過日数はベンチマークと考えてください。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が、これより古いログは削除できます。

推奨: キャプチャー・プログラムはログ・バッファーから直接ログ・レコードを読み取るため、最適なパフォーマンスを実現するために、DB2 がアクティブになっている間は常にキャプチャー・プログラムを実行してください。

ジャーナル・レシーバーの保存 (OS/400)

キャプチャー・プログラムから必要とされるすべてのジャーナル・レシーバーを保存することが重要です。RESTART(*YES) パラメーターを指定してキャプチャー・プログラムを再始動すると、キャプチャー・プログラムは以前に終了した場所から処理を続行し、1 つまたは複数のソース表により使用されるすべてのジャーナル・レシーバーを必要とします。

キャプチャー・プログラムが必要なジャーナル・レシーバーのすべてにアクセスできるように、DB2 DataPropagator for iSeries のインストール時に自動的に登録された、ジャーナル・レシーバー削除出口プログラムを使用してください。この出口プログラムは、ユーザー、またはユーザーのアプリケーション・プログラムの 1 つがジャーナル・レシーバーの削除を試みるたびに呼び出されます。この出口プログラムは、ジャーナル・レシーバーを削除できるかどうかを決定します。

推奨: CHGJRN または CRTJRN コマンドで DLTRCV(*YES) および MNGRCV(*SYSTEM) を指定し、ジャーナル・レシーバー削除出口ルーチンを使用し、ジャーナル管理をシステムに任せるようにします。

ジャーナル・レシーバーが 1 つまたは複数のソース表で使用されている場合、ジャーナル・レシーバー削除出口ルーチンは、キャプチャー・プログラムによってまだ処理されていない項目が、削除対象のレシーバーに含まれていないことを確認します。キャプチャー・プログラムがそのレシーバーの項目をさらに処理する必要がある場合には、出口プログラムはレシーバーの削除を承認しません。詳細については、38 ページの『ジャーナルおよびジャーナル・レシーバーの管理 (OS/400)』を参照してください。

コンプレッション・ディクショナリーの使用方法 (z/OS)

DB2 のコンプレッション・ディクショナリー・ユーティリティを使用する場合は、キャプチャー・プログラムとの間でこれらのユーティリティ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 (z/OS)

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 はキャプチャー・プログラムにエラーを戻します。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、キャプチャー・プログラムは ASN0011E メッセージを発行して終了します。また、コンプレッション・ディクショナリーがもはや使用できない場合には、キャプチャー・プログラムは登録を非活動化します。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を実行

する前に、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

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。

しかし、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. 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。

DB2 コンプレッション・ディクショナリーのラッチ (z/OS)

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。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、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、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、キャプチャー・プログラムを中断してください。

コントロール表の保守

DB2 レプリケーションは、ソース定義、サブスクリプション・セット定義、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動的に拡大 (または縮小) するものもあります。

以下のコントロール表のサイズは、通常処理時に頻繁に変更されます。

- アプライ・ジョブ (IBMSNAP_APPLY_JOB) (OS/400 のみ)
- アプライ・トレース (IBMSNAP_APPLYTRACE)
- アプライ・トレール (IBMSNAP_APPLYTRAIL)
- キャプチャー・モニター (IBMSNAP_CAPMON)
- キャプチャー・トレース (IBMSNAP_CAPTRACE)
- 変更データ (*schema.CD_table*)
- 整合変更データ (*schema.target_table*)
- レプリケーション・アラート・モニター・アラート (IBMSNAP_ALERTS)
- レプリケーション・アラート・モニター・トレース (IBMSNAP_MONTRACE)
- レプリケーション・アラート・モニター・トレール (IBMSNAP_MONTRAIL)
- シグナル (IBMSNAP_SIGNAL)
- サブスクリプション・イベント (IBMSNAP_SUBS_EVENT)
- 作業単位 (IBMSNAP_UOW)

これらの動的なコントロール表のサイズおよび拡大により、システムのパフォーマンスが影響を受ける可能性があります。

このセクションでは、コントロール表に対して実行する必要がある保守のアクティビティについて説明します。

RUNSTATS ユーティリティの使用方法 (Linux、UNIX、Windows、z/OS)

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 DB2 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する必要があります。しかし、表に含まれるデータの量が大量である場合には、変更データ (CD)、作業単位 (IBMSNAP_UOW)、およびその他の動的なコントロール表に対して 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。オプ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。

パッケージおよびプランの再バインド (Linux、UNIX、Windows、z/OS)

DB2 レプリケーションのパッケージおよびプランの多くは、分離 UR (非コミット読み取り) 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、キャプチャー・プログラムおよびアプライ・プログラムがカーソル固定などの標準オプションを使用してレプリケーション・パッケージを再バインドした場合に、プログラム間で競合の問題を発生させる可能性がありますので、パッケージおよびプランの再バインドが必要な場合は注意してください。DB2 レプリケーション・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。

詳細については、29 ページの『レプリケーション・プログラムのセットアップ』を参照してください。

コントロール表の再編成

頻繁に更新される動的なコントロール表は定期的に再編成する必要があります。変更データ (CD) 表および作業単位 (IBMSNAP_UOW) 表は、変更キャプチャー時には多数の INSERTS を受け取り、整理時には多数の DELETES を受け取ります。キャプチャー・モニター (IBMSNAP_CAPMON) 表、キャプチャー・トレース (IBMSNAP_CAPTRACE) 表、およびアプライ・トレール (IBMSNAP_APPLYTRAIL) 表のサイズは、レプリケーションのソース表の更新率によって、大きく変化する可能性があります。

手順:

データのフラグメント化を防止し、スペースをレクラメーション処理するために、以下の表の再編成方式のいずれかを使用してください。

REORG コマンド (Linux、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、キャプチャー・プログラムの挿入処理を迅速化します。

RGZPFM (物理ファイル・メンバーの再編成) コマンド (OS/400)

ENDDPRCAP コマンドで **RGZCTLBL(*YES)** パラメーターを指定すると、キャプチャー・プログラムが終了したときに UOW 表およびアクティブな CD 表を再編成できます。(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、434 ページの『ENDDPRCAP: キャプチャー・プログラムの停止 (OS/400)』を参照してください。)

推奨: 以下の動的なコントロール表は週に一度は再編成してください。

- CD 表
- IBMSNAP_ALERTS
- IBMSNAP_APPLYTRACE
- IBMSNAP_APPLYTRAIL
- IBMSNAP_CAPMON
- IBMSNAP_CAPTRACE
- IBMSNAP_MONTRAIL
- IBMSNAP_MONTRACE
- IBMSNAP_UOW

静的なコントロール表に対しては、未使用のスペースをレクラメーション処理するためのユーティリティを実行したり、頻繁に更新されるオプティマイザーの統計を生成する必要はありません。

- アプライ・エンキュー (IBMSNAP_APPENQ)
- アプライ・パラメーター (IBMSNAP_APPPARMS)
- キャプチャー・エンキュー (IBMSNAP_CAPENQ)
(Linux、UNIX、Windows、z/OS)

- キャプチャー・パラメーター (IBMSNAP_CAPPARMS)
- キャプチャー・パーティション情報 (IBMSNAP_PARTITIONINFO)
- キャプチャー・スキーマ (IBMSNAP_CAPSCHEMAS)
- 整理ロック (IBMSNAP_PRUNE_LOCK)
- 整理セット (IBMSNAP_PRUNE_SET)
- 整理コントロール (IBMSNAP_PRUNCNTL)
- レジスター (IBMSNAP_REGISTER)
- 登録拡張 (IBMSNAP_REG_EXT) (OS/400 のみ)
- 登録同期 (IBMSNAP_REG_SYNCH)
- レプリケーション・アラート・モニター条件 (IBMSNAP_CONDITIONS)
- レプリケーション・アラート・モニター連絡先 (IBMSNAP_CONTACTS)
- レプリケーション・アラート・モニター連絡先グループ (IBMSNAP_CONTACTGRP)
- レプリケーション・アラート・モニター・エンキュー (IBMSNAP_MONENQ)
- レプリケーション・アラート・モニター・グループ (IBMSNAP_GROUPS)
- レプリケーション・アラート・モニター・パラメーター (IBMSNAP_MONPARMS)
- レプリケーション・アラート・モニター・サーバー (IBMSNAP_MONSERVERS)
- 再始動 (IBMSNAP_RESTART)
- 順序付け (IBMSNAP_SEQTABLE)
- サブスクリプション列 (IBMSNAP_SUBS_COLS)
- サブスクリプション・メンバー (IBMSNAP_SUBS_MEMBR)
- サブスクリプション・セット (IBMSNAP_SUBS_SET)
- サブスクリプション・ステートメント (IBMSNAP_SUBS_STMTS)

コントロール表の整理

古いデータを除去し、システム・パフォーマンスを向上させるために、レプリケーション・コントロール表の定期的な整理が必要です。このセクションでは、コントロール表の整理のさまざまな方式と、これらの方式がシステムのパフォーマンスにどのような影響を与えるかについて説明します。

キャプチャー・プログラムによって保守される動的なコントロール表の整理

以下の動的なコントロール表の拡張をモニターし、使用可能な各種の整理方式を考慮する必要があります。

- CD 表
- IBMSNAP_UOW
- IBMSNAP_CAPMON
- IBMSNAP_CAPTRACE
- IBMSNAP_SIGNAL
- IBMSNAP_AUTHTKN (OS/400 のみ)

これらの表で整理が一定インターバルで自動的に行われるように、キャプチャー・プログラムを設定できます。また、整理プロセスを立ち上げることにより、要求時に一度だけ整理を行うこともできます。ユーザーが次に整理コマンドを入力するまで、キャプチャー・プログラムは整理を行いません。

推奨: これらのコントロール表の拡張を管理するために、自動整理を使用することを考えてみてください。自動整理により、ストレージ・コストは最小化され、アプライ・プログラムの効率は向上し、また、これらの表の古いデータが定期的に除去されるため、ストレージのオーバーフローによるシステム障害のリスクが一般的に減少します。自動整理を呼び出すには以下のようにします。

- キャプチャー・プログラムの **autoprune** パラメーターを *y* に設定します (Linux、UNIX、Windows、z/OS)。
- キャプチャー・プログラムのパラメーター設定、**CLNUPITV(*IMMED)** または **CLNUPITV(*DELAYED)** を使用します (OS/400)。

自動整理では、**prune_interval** 稼働パラメーター (Linux、UNIX、Windows、および z/OS の場合)、または **RETAIN** パラメーター (OS/400 の場合) を設定して、自動整理処理を発生させる頻度を指定します。

手順:

以下の方法のいずれかを使用して整理を開始します。

レプリケーション・センター

「キャプチャー・コントロール表の整理」ウィンドウを使用して、表の整理を一度行います。詳しくは、レプリケーション・センターのヘルプを参照してください。

asncap システム・コマンドで **autoprune=y** を指定 (Linux、UNIX、Windows、z/OS)

自動整理を指定してキャプチャー・プログラムを始動するには、この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、342 ページの『asncap: キャプチャーの始動』を参照してください。

asnccmd システム・コマンドで **chgparms autoprune=y** を指定 (Linux、UNIX、Windows、z/OS)

実行中のキャプチャー・プログラムで自動整理を使用可能にするには、この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、349 ページの『asnccmd: キャプチャーの操作』を参照してください。

asnccmd システム・コマンドで整理パラメーターを使用 (Linux、UNIX、Windows、z/OS)

実行中のキャプチャー・プログラムで整理を一度開始するには、この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、349 ページの『asnccmd: キャプチャーの操作』を参照してください。

STRDPRCAP CLNUPITV(*IMMED) または **STRDPRCAP CLNUPITV(*DELAYED)** システム・コマンド (OS/400)

キャプチャー・プログラムを始動した後で、指定したインターバルで古い行を削除するには、これらのコマンドを使用します。パラメーター記述およびコマンド構文の詳細については、468 ページの『STRDPRCAP: キャプチャー・プログラムの始動 (OS/400)』を参照してください。

OVRDPRCAPA PRUNE(*IMMED) または OVRDPRCAPA PRUNE(*DELAYED) システム・コマンド (OS/400)

実行中のキャプチャー・プログラムによるコントロール表の整理の方式を変更するには、この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、448 ページの『OVRDPRCAPA: DPR キャプチャー属性のオーバーライド (OS/400)』を参照してください。

CD 表および UOW 表の整理: キャプチャー・プログラムは、自動的に呼び出された場合も、要求時に実行する場合も、アプライ・プログラムから報告された進行に基づいて、各整理サイクルで CD 表および UOW 表の整理を行います。進行は、整理セット (IBMSNAP_PRUNE_SET) 表の SYNCHPOINT 列で示されます。この通常の整理は、各 CD 表をサブスクライプするすべてのアプライ・プログラムを通じて最小の同期点値に基づいて、また UOW 表の場合は全体を通じて最小の同期点値に基づいて行われます。

しかし通常の整理では、関連するサブスクリプション・セットが非常にまれにしか実行されない場合は、CD 表および UOW 表の効果的な整理を行うことはできません。関連するアプライ・プログラムの実行頻度を決定するとき、これらのアプライ・プログラムを停止するとき、そしてサブスクリプション・セットを比較的長い間非活動化するときには、整理の効率性を考慮してください。

サブスクリプション・セットを非常にまれにしか実行しないか、またはアプライ・プログラムを停止すると、使用している CD 表と UOW 表が非常に大きくなって、保持制限整理の対象として適格となる可能性があります。保持制限は、キャプチャー・プログラムの稼働パラメーターであり、このデフォルト値は 1 週間です。この値は、表の中の古いデータがどれくらいたつと保持制限整理に適格になるかを決定します。

サブスクリプション・セットが非アクティブになるか、まれにしか実行されないために、通常の整理処理が使用禁止になると、データが非常に長い時間表の中に留まる可能性があります。このデータが DB2 の現行タイム・スタンプから保持制限値を引いた値よりも古くなると、保持制限整理処理は、このデータを表から整理します。

保持制限整理を必要とするような条件の発生は回避してください。古いデータが累積されることにより、ストレージのオーバーフローが発生し、性能低下を招く可能性があります。詳細については、235 ページの『サブスクリプション・セットの非活動化』を参照してください。

推奨: すべてのサブスクリプション・セットに対して、アプライ・プログラムを少なくとも 1 日に 1 回実行してください。

ソース・サーバーが多様なターゲット・システムに変更データを提供している場合、そして、各ターゲットの要件が大きく異なるものであり、また、ターゲットによっては、少数の登録済みソースに対してアプライ・プログラムがまれにしか実行されない場合は、複数のキャプチャー・プログラムを使用することを考えてみてください。複数のキャプチャー・プログラムを実行することにより、異なるキャプチャー・スキーマを使用して、多様な処理要件を管理できます。1 つのキャプチャ

ー・スキーマを使用して、サブスクリプション・セットの固有のタイミング要件によりまれにしか整理されない表を分離し、残りのソース表に対しては、別のキャプチャー・スキーマを使用できます。

キャプチャー・モニター表とキャプチャー・トレース表の整理: キャプチャー・プログラムは、キャプチャー・プログラムの以下の稼働パラメーターの値に基づいて、各整理サイクルでキャプチャー・モニター (IBMSNAP_CAPMON) 表およびキャプチャー・トレース (IBMSNAP_CAPTRACE) 表の整理を行います。

- 行が IBMSNAP_CAPMON 表の中に維持される時間の長さを示す **monitor_limit** パラメーター (Linux、UNIX、Windows、z/OS の場合) および **MONLMT** パラメーター (OS/400 の場合)
- 行が IBMSNAP_CAPTRACE 表の中に維持される時間の長さを示す **trace_limit** パラメーター (Linux、UNIX、Windows、z/OS の場合) および **TRCLMT** パラメーター (OS/400 の場合)

モニター限度パラメーターおよびトレース限度パラメーターのデフォルト値は両方ともに 1 週間です。IBMSNAP_CAPMON 表の中にキャプチャー・プログラムの待ち時間およびスループットの履歴情報をどのくらい長く保持したいか、そして IBMSNAP_CAPTRACE 表の中に監査およびトラブルシューティングの情報をどのくらい長く保持したいかによって、これらの値は変更できます。

シグナル表の整理: シグナル (IBMSNAP_SIGNAL) 表もまた、各整理サイクル中に整理されます。シグナル行は、SIGNAL_STATE 列の値が C になると整理に適格になります。C という値は、シグナル情報が完成しており、キャプチャー・プログラム、またはその他のユーザー処理から必要とされることはないため、整理に適格であることを意味します。SIGNAL_TIME 列の値が、DB2 の現行タイム・スタンプから保持制限パラメーターの値を引いたものよりも古いシグナル行は、保持制限整理に適格です。

その他の動的コントロール表の整理

キャプチャー・プログラムは、自分が保守している表に対してのみ整理操作を実行します。整合変更データ (CCD) 表はアプライ・プログラムによって保守されているため、キャプチャー・プログラムはこれらの表を自動的に整理しません。CCD 表のタイプによっては、整理の必要のないものもあります。完全なコンデンス CCD 表は、同じ場所で更新されます。

完全なコンデンス CCD 表からユーザーが除去できる唯一のレコードは、従属するターゲット表にすでに複製済みで、IBMSNAP_OPERATION 列の値が D (削除) のものです。非コンデンス CCD 表は、履歴データを含むものであり、非常に大きくなる可能性があります。このデータは監査のために保存しておく必要があるため、非コンデンス CCD 表に対しては整理操作を実行しないでください。

しかし、内部 CCD 表の整理は考慮に値します。これらの表は、システムの更新アクティビティが多い場合は、すぐに大きくなる可能性があります。内部 CCD 表からは最新の変更だけがフェッチされるため、古い行を保存しておく必要はありません。

内部 CCD 表の整理を使用可能にするために、従属するすべてのターゲットへのアプライがすでに完了している変更データを整理する、事後に実行される SQL ステ

ートメントを、関連するサブスクリプション・セットに追加することを考えてみてください。また、自動スケジューリング機能に必要な SQL DELETE ステートメントを追加して、これらの表から行を削除することもできます。

アプライ・トレール (IBMSNAP_APPLYTRAIL) 表およびアプライ・トレース (IBMSNAP_APPLYTRACE) 表も手動で整理する必要があります。頻繁に実行されるアプライ・プログラムで複数のサブスクリプション・セットを定義および使用する場合、IBMSNAP_APPLYTRAIL 表は急速に拡張するため、頻繁な整理が必要です。これらの表の拡張を管理する最良の方法は、サブスクリプション・セットの 1 つに、事後に実行される SQL ステートメントまたはプロシージャー呼び出しを追加することです。また、自動スケジューリング機能に SQL DELETE ステートメントを追加することもできます。

レプリケーションの失敗の防止およびエラーからのリカバリー

このセクションでは、コントロール表およびレプリケーション・データに影響を与える可能性のある、レプリケーションの失敗を防止し、失敗から回復する方法について説明します。

- キャプチャー・プログラムのコールド・スタートの防止
- コントロール表の入出力エラーおよび接続障害からのリカバリー
- 脱落したソース・データのリトリート

キャプチャー・プログラムのコールド・スタートの防止

キャプチャー・プログラムのコールド・スタートは、プログラムを初めて開始するとき、またはコントロール表およびターゲット表のリフレッシュが必要な場合にだけ実行してください。キャプチャー・プログラムをコールド・スタートすると、レプリケーション環境内のすべてのターゲット表がリフレッシュされます。

Linux、UNIX、Windows、または z/OS 上で、warmns、warmsa、または warmsi オプションを指定してキャプチャー・プログラムを始動した場合、プログラムは再始動 (IBMSNAP_RESTART) 表の中の再始動点に基づいてログ・レコードのリトリートを試みます。キャプチャー・プログラムがログを検出できないと、キャプチャー・プログラムのウォーム・スタートは失敗します。ユーザーが warmns または warmsi オプションを指定してキャプチャー・プログラムを始動すると、再始動プロセスは終了し、エラー・メッセージが発行されます。ユーザーが warmsa オプションを指定してキャプチャー・プログラムを始動すると、再始動プロセスは停止し、キャプチャー・プログラムはコールド・スタートを実行し、CD 表および UOW 表の中のすべてのレコードを削除します。

キャプチャー・プログラムのコールド・スタートを防止するために、以下の推奨事項を考慮してください。

- Linux、UNIX、Windows および z/OS オペレーティング・システムの場合は warmsa を使用せずに、できるだけ warmns または warmsi 始動モードを使用してキャプチャー・プログラムを再始動してください。warmns および warmsi オプションを使用すると、再始動プロセスが失敗した場合の、キャプチャー・プログラムの自動コールド・スタートを防止できます。詳細については、342 ページの『asnacp: キャプチャーの始動』を参照してください。

- OS/400 オペレーティング・システムの場合は、**RESTART(*YES)** パラメーターを指定してキャプチャー・プログラムを始動してください。キャプチャー・プログラムは、以前に終了したときに停止したポイントから処理を継続します。詳細については、468 ページの『STRDPRCAP: キャプチャー・プログラムの始動(OS/400)』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、またはその他のメカニズムを使用して、キャプチャー・プログラムからの履歴データの状況を確認してください。次にこの情報を使用して、DB2 がアクティブのときには必ずキャプチャー・プログラムが実行されていることを検証できます。詳細については、199 ページの『第 12 章 SQL レプリケーションの要求時報告』を参照してください。
- システム上に十分な DB2 ログ・データおよびジャーナル・レシーバーが保存され、このデータが DB2 レプリケーションから使用可能であることを確認してください。ログ保存の詳細は、248 ページの『ソース・ログおよびジャーナル・レシーバーの保守および保存』を参照してください。

コントロール表の入出力エラーおよび接続障害からのリカバリー

コントロール表で入出力エラーまたは接続障害が発生した場合には、DB2 の標準リカバリー手順を使用して、表の順方向リカバリーを行います。表のデータが失われることはありません。

キャプチャー・プログラムは入出力エラーまたは接続障害を検出すると、適切なエラー・メッセージを発行してシャットダウンします。ユーザーはエラーを訂正してから、障害発生時点からキャプチャー・プログラムを再開できます。

アプライ・プログラムは、コントロール表に重大なエラーを検出するとシャットダウンします。アプライ・プログラムはターゲット表のエラー、またはネットワーク接続のエラーを検出すると、アプライ・トレール (IBMSNAP_APPLYTRAIL) 表にエラーを書き込み、処理を続行します。

脱落したソース・データのリトリート

ソース表で障害時点まで順方向リカバリーが行われると、DB2 レプリケーションは正常に進行します。表がリカバリーされると、キャプチャー・プログラムは表のデータ変更の収集を続行します。

しかし、キャプチャー・プログラムおよびアプライ・プログラムは、読み取り専用ターゲット表のポイント・イン・タイム・リカバリーは検出しません。ユーザーがソース表を回復した場合、アプライ・プログラムが、ソース表にもはや存在していない変更をターゲット表に複製した可能性があるため、ユーザーがターゲット表を同じ論理ポイント・イン・タイムまで戻せない場合は、ソース表とターゲット表の間で不整合が生じる可能性があります。

複数レベルでレプリケーションが行われる場合は、この状況がより複雑なものになります。各レベル間のリカバリー点を照合するためのメカニズムを提供するか、リカバリー方式としてフル・リフレッシュを使用する必要があります。

分散リカバリー点の詳細は、237 ページの『データベース・アプリケーション・イベントとレプリケーション・イベントの調整』を参照してください。

ターゲット表の保守

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、DB2 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。

重要: 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、アプライ・プログラムを非活動化してください。

関連概念:

- 495 ページの『第 23 章 SQL レプリケーション・コンポーネントの通信方法』

関連タスク:

- 3 ページの『第 1 章 SQL レプリケーションの計画』
- 17 ページの『第 2 章 SQL レプリケーション用のサーバーの構成』

第 2 部 レプリケーション・センター

本書の第 2 部は、以下の章で構成されています。

265 ページの『第 15 章 SQL レプリケーションの場合のレプリケーション・センターの使用法』では、レプリケーション・センターについて説明します。

295 ページの『第 16 章 基本的な SQL レプリケーションのシナリオ: DB2 for Windows』では、レプリケーション・センターを使用して、サンプル・データを使って簡単なレプリケーション・シナリオを実行する方法について説明します。

第 15 章 SQL レプリケーションの場合のレプリケーション・センターの使用法

レプリケーション・センターは、レプリケーション環境をセットアップおよび管理するため、また、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・プログラムを実行するために使用できるユーザー・インターフェース・ツールです。レプリケーション・センターを使用して、次のような管理タスクを実行することができます。

- レプリケーション・コントロール表の作成
- レプリケーション・ソースの登録
- サブスクリプション・セットを作成し、そのセットにサブスクリプション・セット・メンバーを追加
- キャプチャー・プログラムの操作
- アプライ・プログラムの操作
- レプリケーション処理のモニター

レプリケーション・センターにはランチパッドもあり、これを使用して DB2® レプリケーション環境のセットアップに必要な基本関数を実行することができます。ランチパッドは、さまざまなステップがお互いにどのように関係しているかをグラフィカルに表示します。

レプリケーション・センターを使用して、DB2 から DB2 へのレプリケーション環境や、DB2 と DB2 以外のリレーショナル・データベース間のレプリケーションをセットアップすることができます。レプリケーション・センターは、DB2 コントロール・センターのツールのセットの一部です。レプリケーション・センターの詳細なタスク情報については、オンライン・ヘルプを参照してください。

この章では、以下のタスクについて説明します。

レプリケーション・センターの使用

- 267 ページの『レプリケーション・センターの前提条件』を参照。レプリケーション・センターの前提条件は、DB2 コントロール・センターの前提条件と非常に似ています。
- 268 ページの『レプリケーション・センターの始動』を参照。レプリケーション・センターはいくつかの方法で始動することができます。
- 269 ページの『SQL レプリケーションの場合のレプリケーション・センター・ランチパッドの使用』を参照。ランチパッドの使用はオプションですが、初めてのユーザーには非常に役に立ちます。

レプリケーション・センターのセットアップ

- 271 ページの『レプリケーション・センターのユーザー ID およびパスワードの管理』を参照。レプリケーション・センターがデータベースに接続し、システムにログオンするために使用するパスワードを保守することができます。

- 272 ページの『レプリケーション・プロファイルの作成』を参照。プロファイルの作成はオプションですが、大規模なレプリケーション環境を管理する場合はとても便利です。
- 275 ページの『レプリケーション・コントロール表の作成』を参照。レプリケーション・コントロール・サーバーとして働く各データベース内に、コントロール表を作成する必要があります。
- 278 ページの『レプリケーション・センターへのサーバーの追加』を参照。ユーザーがレプリケーション・コントロール表を作成すると、サーバーは自動的にレプリケーション・センターに追加されます。管理したいサーバーだけを追加することにより、レプリケーション環境のビューをカスタマイズすることができます。
- 280 ページの『データベースの変更キャプチャーの使用可能化 (UNIX および Windows)』を参照。Linux、UNIX[®] および Windows[®] システム上のキャプチャー・コントロール・サーバーが、変更をキャプチャーし、データベース・バックアップを開始できるようにする必要があります。

レプリケーション環境の定義

- 281 ページの『ソースの登録』を参照。表またはビューをレプリケーション・ソースとして登録できます。
- 282 ページの『サブスクリプション・セットの作成』を参照。空のセットを作成し、いつでもここにサブスクリプション・セット・メンバーを追加するか、またはサブスクリプション・セットの作成時にサブスクリプション・セット・メンバーを作成することができます。

レプリケーション環境の保守

- 286 ページの『サブスクリプション・セットの活動化または非活動化』を参照。どのサブスクリプション・セットでも、一時的に、または永続的に、非活動化または活動化することができます。
- 287 ページの『レプリケーション・オブジェクトのプロモート』を参照。テスト環境から実稼働環境へ、表の登録およびサブスクリプション・セットをプロモートすることができます。
- 289 ページの『ターゲット表の強制フル・リフレッシュ』を参照。アプライ・プログラムがサブスクリプション・セットのフル・リフレッシュを実行する時期をコントロールすることができます。
- 290 ページの『レプリケーション定義の除去または削除』を参照。レプリケーション・センターからレプリケーション・オブジェクトを除去し、レプリケーション・コントロール・サーバーからレプリケーション定義を削除することができます。

レプリケーション環境の操作

- 290 ページの『キャプチャー・プログラムの操作』を参照。ネットワーク内のどのサーバーからでも、キャプチャー・プログラムを始動および停止することができます。また、キャプチャー・プログラム用のその他の多くの操作も実行することができます。
- 291 ページの『アプライ・プログラムの操作』を参照。ネットワーク内のどのサーバーからでも、アプライ・プログラムを始動および停止することができます。また、アプライ・プログラム用のその他の多くの操作も実行することができます。

- 292 ページの『レプリケーション・アラート・モニターの操作』を参照。レプリケーション・アクティビティをモニターするためのアラート条件を定義することができます。

レプリケーション・センターの前提条件

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 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。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 Java 2 Runtime Environment または Java 2 Software Development Kit のどちらかのバージョン 1.3 があることを確認してください。

z/OS™ バッファ・プールを表示し、レプリケーション・センターからキャプチャー・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・プログラムを操作するには、DB2 Administration Server for z/OS および 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ないストアード・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・センターは、DB2 に付属のストアード・プロシージャ DSNWZP を呼び出して、SQL 拡張文字などの DB2 サブシステム・パラメーター情報を検索します。DSNWZP ストアード・プロシージャは、390 Enablement パッケージの一部ではありません。DB2 for z/OS のインストール内容に含まれる DSNTIJSJ ジョブを実行して、DSNWZP ストアード・プロシージャを定義してください。

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、キャプチャー・プログラムまたはアプライ・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 (DAS) が実行されていることを確認してください。DB2 Administration Server for z/OS は、DB2 for OS/390 and z/OS V7 またはそれ以降でのみ使用可能です。インストール後は、DB2 for OS/390 V6 アプリケーションで使用することも可能です。

ホスト RDBMS 用のレプリケーション・センターの構成

レプリケーション・センターまたはコマンド行ツールを使用してホスト・データベースまたはサブシステム上でレプリケーションを管理している場合、分散データベース接続サービス (DDCS) パッケージを、z/OS (MVS/ESA)、VSE、VM、または AS/400® 上のホスト DB2 UDB サーバーにバインドする必要があります。

前提条件:

DDCS パッケージをバインドする前に、以下のことを実行してください。

- ホスト・データベースまたはサブシステムへの接続を構成する。
- ホスト・データベースまたはサブシステムをカタログする。

手順:

DDCS パッケージをホスト・データベースまたはサブシステムにバインドするには、以下のようにします。

1. キャプチャー・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進みます。このディレクトリーは通常、DB2 UDB またはクライアント・アプリケーション・イネーブラー (CAE) をインストールしたドライブ上の ¥SQLLIB¥BND ディレクトリーです。たとえば、Windows 上のデフォルト・ディレクトリーにナビゲートするには、次のコマンドを発行します。

```
cd C:¥Program Files¥IBM¥SQLLIB¥BND
```

2. 以下のコマンドを発行して、パッケージをホスト DB2 UDB for z/OS (MVS/ESA)、VSE、VM、AS/400 データベースまたはサブシステムにバインドします。

```
DB2 CONNECT TO dbname USER userid USING password  
DB2 BIND @DDCSxxx.LST ISOLATION CS BLOCKING ALL SQLERROR CONTINUE
```

ここで、*dbname* はホスト・データベースまたはサブシステムの名前であり、*xxx* はホスト・システムのプラットフォームを指定し (レプリケーションの選択項目は、MVS™、VSE、VM、または AS/400)、さらに CS はカーソル固定分離レベルを指定します。

レプリケーション・センターの始動

レプリケーション・センターは、UNIX または Windows 操作環境用の通常の DB2 インストールの一部としてインストールされます。カスタム・インストールを実行する場合は、汎用管理ツール (General Administration Tools) コンポーネントを選択してレプリケーション・センターをインストールする必要があります。

レプリケーション・センターを始動するには、コマンド・ウィンドウに **db2rc** コマンドを入力します。

Windows システムでは、以下に示すように、Windows の「スタート」メニューを使用してレプリケーション・センターを始動することもできます。

1. 「スタート」をクリックします。
2. 「プログラム」を選択します。
3. 「IBM DB2」を選択します。
4. 「汎用管理ツール」を選択します。
5. 「レプリケーション・センター」をクリックします。

すでに DB2 コントロール・センターを実行中の場合は、「ツール」メニューから「レプリケーション・センター」を選択するか、またはレプリケーション・センターのアイコンをクリックして、レプリケーション・センターを始動することができます。

図7 はレプリケーション・センターを示しています。

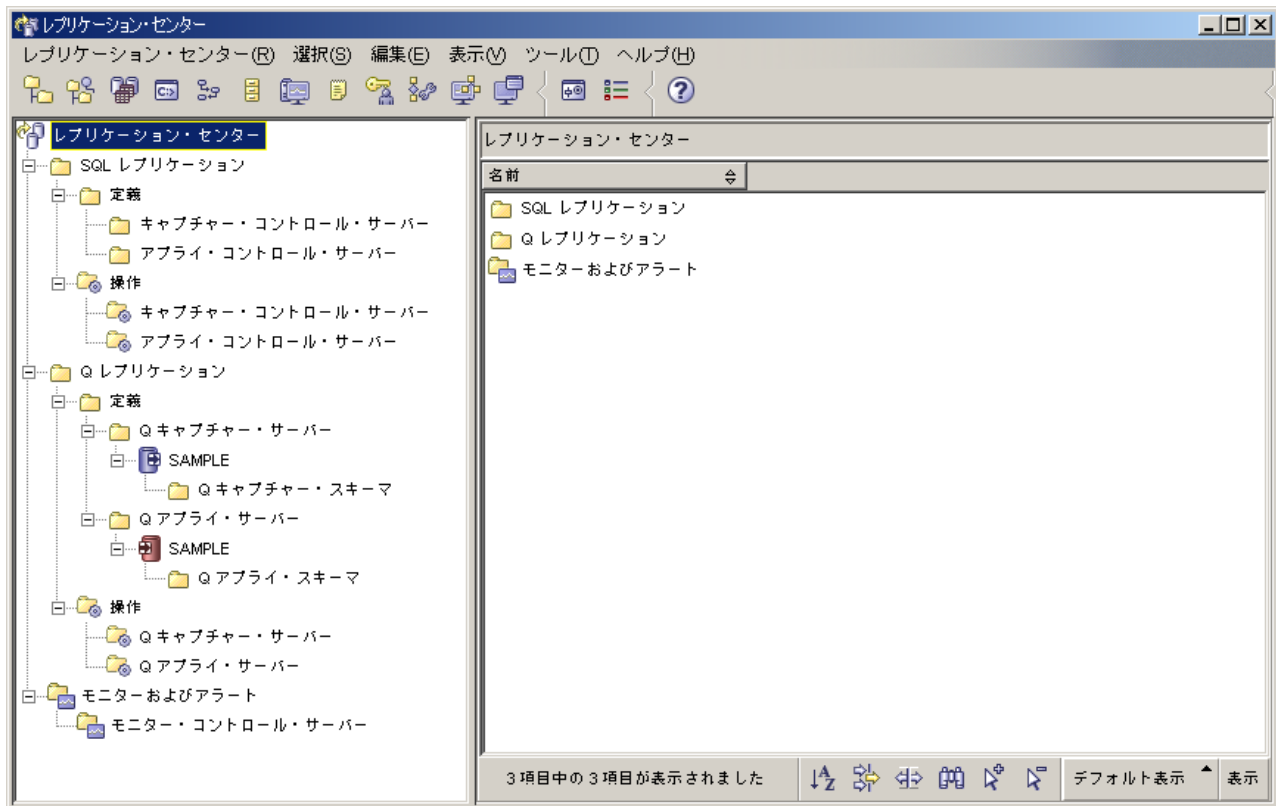


図7. レプリケーション・センター

SQL レプリケーションの場合のレプリケーション・センター・ランチパッドの使用

最初にレプリケーション・センターを始動すると、レプリケーション・センター・ランチパッドが表示されます。これを使用して、Q レプリケーション、イベント発行、または SQL レプリケーション環境のセットアップに必要な基本関数を実行することができます。またランチパッドは、さまざまなステップがお互いにどのように関係しているかをグラフィカルに表示します。

ランチパッドは、レプリケーション・センターのよく使用される関数のほとんどにアクセスできますが、これらの関数にはレプリケーション・センターのオブジェクト・ツリーからもアクセスできます。ランチパッドを使用して作成したものは、オブジェクト・ツリーを使用して、何でも表示または操作することができます。レプリケーション・センターでは、ランチパッドからはアクセスできない、他の多くの、より高度な関数を利用できます。

注: ランチパッドのイベント発行と Q レプリケーションのページに関する情報は、「レプリケーションとイベント・パブリッシング・ガイドおよびリファレンス」を参照してください。

SQL レプリケーションのページにアクセスするには、SQL レプリケーションの説明の横にあるボタンをクリックします。ランチパッドの SQL レプリケーション・ページから、以下のタスクを実行することができます。

- キャプチャー・コントロール表の作成

このオプションは「キャプチャー・コントロール表の作成」ウィンドウをオープンし、ここから、キャプチャー・プログラムのために必要なレプリケーション・コントロール表を、特定のデータベースに作成することができます。

- ソース表の登録

このオプションは「表の登録」ウィンドウをオープンし、ここから、登録したいソース表の登録情報（ソース列、CD 表の情報など）を定義することができます。

- アプライ・コントロール表の作成

このオプションは「アプライ・コントロール表の作成」ウィンドウをオープンし、ここから、アプライ・プログラムのために必要なレプリケーション・コントロール表を、特定のデータベース内に作成することができます。

- サブスクリプション・セットの作成

このオプションは「サブスクリプション・セットの作成」ウィンドウをオープンし、ここからサブスクリプション・セット情報を定義することができます。その情報とは、キャプチャー、ターゲットおよびアプライ・コントロール・サーバー、ソースからターゲットへのマッピング、各サブスクリプション・セット・メンバーのプロパティ、セットのスケジュール、およびセット用の SQL ステートメントなどです。

- キャプチャー・プログラムの始動

このオプションは「キャプチャーの開始」ウィンドウをオープンし、ここからキャプチャー・プログラムを始動し、始動パラメーターを指定することができます。

- アプライ・プログラムの始動

このオプションは「アプライの開始」ウィンドウをオープンし、ここからアプライ・プログラムを始動し、始動パラメーターを指定することができます。

推奨事項: ランチパッドではこれらのタスクを順番に実行する必要はありませんが、DB2 レプリケーションに慣れていないユーザーは、ランチパッドに示されている順序に従うことをお勧めします。必要なレプリケーションやデータベース・オブジェクトがすでに存在する場合は、ランチパッド内のステップを飛ばしたり、繰り返すこともできます。ランチパッドまたはレプリケーション・センターのオブジェクト・ツリーを使用して、必要なレプリケーションとデータベース・オブジェクトを作成してください。

「レプリケーション・センター」メニューから「ランチパッド」を選択するか、またはオブジェクト・ツリー内の「レプリケーション・センター」フォルダーを右クリックし、「ランチパッドの開始」を選択すれば、いつでもランチパッドを使用することができます。「概説」ページが表示されたら、「SQL レプリケーションの開始 (Getting Started with SQL Replication)」をクリックします。別のランチパ

ッド・ページが表示された場合は、「ランチパッド・ビューの選択」フィールドから、「SQL レプリケーション・ランチパッド (SQL replication launchpad)」を選択します。

レプリケーション・センターのユーザー ID およびパスワードの管理

レプリケーション・センターは、多くのデータベース・サーバー (ソース・サーバー、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、モニター・コントロール・サーバー、およびターゲット・サーバー) に接続できる必要があります。またレプリケーション・センターは、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・モニターを実行するシステムにも接続できる必要があります。リモートのデータベースおよびシステムはすべて、データベースに接続し、各システムにログオンするための有効なユーザー ID とパスワードが必要です。レプリケーション・センターは、各ユーザー ID とパスワードの指定が一度ですむようにしているので、レプリケーション・センターがリモート・データベースに接続するたびに、または各リモート・システムにログオンするたびに、プロンプトを求めることはありません。

デフォルトでは、レプリケーション・センターはユーザー ID とパスワード情報をメタデータ・ファイルに保管します。パスワードはこのファイル内で暗号化されないため、レプリケーション・センターがパスワードをメモリー内のみ保持し、このファイルにはパスワードを保管しないように指定することができます。DB2 でパスワードが変更された場合は、レプリケーション・センター内のパスワードも変更し、両方で同じパスワードを使用する必要があります。レプリケーション・センターは、アプライ・プログラム、レプリケーション・アナライザー、またはレプリケーション・アラート・モニターとパスワード情報を共有しません。

レプリケーション・センターのユーザー ID およびパスワードを管理するには、次のようにします。

1. 「レプリケーション・センター」アイコンを右クリックします。
2. 「パスワードおよび接続の管理」を選択します。

「パスワードおよび接続の管理」ウィンドウで、次のどのタスクでも実行することができます。

- レプリケーション環境で、使用を計画しているか使用しているサーバーまたはシステムのユーザー接続情報を追加する。
- レプリケーション環境で使用しているサーバーまたはシステムの接続情報を変更する。
- レプリケーション環境中のサーバーまたはシステムの接続情報を除去する。
- 指定されたユーザー ID とパスワードを使用して、サーバーまたはシステムの接続をテストする。

レプリケーション・センターは、接続情報を使用して次のアクションを実行します。

- ローカル・サーバーやリモート・サーバーに接続して、データを検索し、SQL スクリプトを実行する。
- リモート・システムにログオンし、コマンドを実行する。

レプリケーション・プロファイルの作成

レプリケーションのセットアップの一部として、多くの場合複数のデータベース内に、レプリケーション・コントロール表を作成します。また、多くのソース表およびビューを登録し、そのすべてが CD 表を持ちます。さらにサブスクリプション・セット作成の一部として、多くのターゲット表を定義します。これらのレプリケーション・オブジェクトのそれぞれについて、特定の命名規則を使用し、共通の属性を持たせることとなります (たとえば、CD 表用のすべての表スペースが同じページ・サイズを使用する可能性があります)。レプリケーション・センターでは、レプリケーション・オブジェクトを作成するたびにこれらの共通の定義を指定しなくてもすむように、これらの命名規則および共通の定義を反映するプロファイルがユーザーが作成できるようにしています。以下のレプリケーション・オブジェクト用のプロファイルを作成することができます。

- レプリケーション・コントロール表
- レプリケーション・ソース・オブジェクト (CD 表)
- レプリケーション・ターゲット・オブジェクト

各オブジェクト・プロファイルに、CD 表、索引、および表スペースといったデータベース・オブジェクト用の命名規則を指定し、これらのオブジェクト用の共通属性 (ページ・サイズやバッファ・プールなど) を指定します。各プロファイルに指定する値は、「コントロール表の作成」ウィンドウ、「表の登録」ウィンドウ、「ビューの登録」ウィンドウ、または「サブスクリプション・セットの作成」ウィンドウに表示されるデフォルト値になります。これらのデフォルト値は、特定のレプリケーション・オブジェクトの作成時にオーバーライドするか、または「OK」をクリックして、プロファイルに定義された値を受け入れることができます。

コントロール表プロファイルの作成

各レプリケーション・コントロール表 (たとえば、登録表 `IBMSNAP_REGISTER`) について、プロファイルに表スペース情報および索引情報を定義することができます。デフォルトにより、レプリケーション・センターはパフォーマンスを最適化するため、レプリケーション・コントロール表と一緒に表スペース内にまとめます。これらの表スペース用に、レプリケーション・センターが表スペースの作成時に使用する命名規則を定義するか、またはすでに存在する表スペースを指定することができます。また、コントロール表のための、その他のオペレーティング・システム特有の表スペース情報を定義することもできます。コントロール表の多くは、1 つまたは複数の索引も必要とします。レプリケーション・センターがこれらの索引を作成する時に使用する命名規則を定義するか、またはすでに存在する索引を指定することができます。

DB2 がサポートする各タイプのオペレーティング・システム用の、ユニークなコントロール表プロファイルを定義することができます。また、DB2 レプリケーションがサポートする DB2 以外のデータベースの各タイプ用のプロファイルを定義することもできます。レプリケーション・センターは OS/400[®] システム用のコントロール表プロファイルを提供しませんが、その理由は、DB2 DataPropagator[™] for iSeries[™] のインストール時にレプリケーション・コントロール表が作成されるためです。

コントロール表プロファイルを作成するには、次のようにします。

1. 「SQL レプリケーション」フォルダーを展開します。
2. 「定義」フォルダーを右クリックします。
3. 「コントロール表プロファイルの管理」を選択します。

「コントロール表プロファイルの管理」ウィンドウで、次のようにします。

- a. プロファイルを作成する、オペレーティング・システムまたは DB2 以外のデータベース環境を選択します。
- b. リストからレプリケーション・コントロール表を選択します。
- c. 選択したコントロール表の表スペースと索引の特性を定義します。

別の表スペースを作成するコントロール表を必ずすべて選択し、その特性を定義してください。

- d. 特定のオペレーティング・システム・プラットフォーム用の、または DB2 以外のリレーショナル・データベース・システム用の、コントロール表をすべて定義したら、「適用」をクリックします。「クローズ」をクリックし、「コントロール表プロファイルの管理」ウィンドウをクローズします。

ソース・オブジェクト・プロファイルの作成

それぞれの DB2 レプリケーション・ソース・オブジェクト (表またはビュー) ごとに、キャプチャー・プログラムは CD 表を必要とします。ユーザーはソース・オブジェクトを登録する時に、CD 表と CD 表用の索引の両方について、名前と特性を指定します。ソース・オブジェクト用のプロファイルを作成すれば、特定のソース・データベースから登録するすべてのソースについて、共通の特性を定義することができます。これらの共通の特性を使用して、多くの表またはビューを 1 つのアクションの一部として登録することができます。

レプリケーション・センターが CD 表の作成時に使用する命名規則、CD 表用の表スペース、および CD 表用の索引を定義することができます。また、ユーザーが指定した接頭部と接尾部をレプリケーション・センターが反映した後、これらのオブジェクトの名前がオペレーティング・システム特有の長さ制限 (たとえば、UNIX および Windows データベースの場合は 128 文字) を超えた場合に、オブジェクト名を切り捨てるための規則を定義することもできます。たとえば、CD 表の名前を “CD_sourcetablename” (sourcetablename は、登録されるソース表ごとに異なる) とし、これを “CD_repltablespace” という名前の表スペースに入れるプロファイルを作成することができます。

キャプチャー・コントロール・サーバー用のソース・オブジェクト・プロファイルを作成する前に、そのサーバーをレプリケーション・センターに追加する必要があります。

ソース・オブジェクト・プロファイルを作成するには、「ソース・オブジェクト・プロファイルの管理」ウィンドウを使用します。このウィンドウは次の 2 つの方法でオープンすることができます。

- 「定義」フォルダーからウィンドウをオープンする場合は、次のようにします。
 1. 「SQL レプリケーション」フォルダーを展開します。
 2. 「定義」フォルダーを右クリックします。
 3. 「ソース・オブジェクト・プロファイルの管理」を選択します。

「ソース・オブジェクト・プロファイルの管理」ウィンドウで、プロファイルを作成するソース・サーバーを選択します。

- ソース・サーバーからウィンドウをオープンする場合は、次のようにします。
 1. 「SQL レプリケーション」フォルダーを展開します。
 2. 「定義」フォルダーを展開します。
 3. 「キャプチャー・コントロール・サーバー」フォルダーを展開します。
 4. ソース・サーバーを右クリックして、「ソース・オブジェクト・プロファイルの管理」を選択します。

この場合、「ソース・オブジェクト・プロファイルの管理」ウィンドウで、プロファイルを作成するソース・サーバーを選択する必要はありません。

「ソース・オブジェクト・プロファイルの管理」ウィンドウで、CD 表の特性、CD 表用の表スペース、および CD 表用の索引を定義します。これらのオブジェクトごとに切り捨て規則を選択することもできます。

ターゲット・オブジェクト・プロファイルの作成

サブスクリプション・セット・メンバーの作成時に、ソース・オブジェクト (表、ビュー、またはニックネーム) とターゲット表間のレプリケーションのマッピングを定義します。ターゲット表がまだ存在しない場合は、ターゲット表およびターゲット表用の索引の両方について、名前と特性を指定します。ターゲット・オブジェクト用のプロファイルを作成すれば、特定のターゲット・データベース内のすべてのターゲット表について、共通の特性を定義することができます。

レプリケーション・センターがターゲット表の作成時に使用する命名規則、ターゲット表用の表スペース、およびターゲット表用の索引を定義することができます。また、ユーザーが指定した接頭部と接尾部をレプリケーション・センターが反映した後、これらのオブジェクトの名前がオペレーティング・システム特有の長さ制限 (たとえば、UNIX および Windows データベースの場合は 128 文字) を超えた場合に、オブジェクト名を切り捨てるための規則を定義することもできます。たとえば、ターゲット表の名前を “TGsourcetablename” (sourcetablename は、登録されるソース表ごとに異なる) とし、これを “TS_targettablename” (targettablename は、ターゲット表ごとに異なる) という名前の表スペースに入れるプロファイルを作成することができます。

ターゲット・サーバー用のターゲット・オブジェクト・プロファイルを作成するには、その前にターゲット・サーバーをローカルの DB2 データベースにカタログしておく必要があります。ただし、ターゲット・サーバーをレプリケーション・センターに、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、またはモニター・コントロール・サーバーとして追加する必要はありません。

ターゲット・オブジェクト・プロファイルを作成するには、次のようにします。

1. 「SQL レプリケーション」フォルダーを展開します。
2. 「定義」フォルダーを右クリックします。
3. 「ターゲット・オブジェクト・プロファイルの管理」を選択します。
4. 「サーバーの選択」ウィンドウで、ターゲット表プロファイルを作成するデータベース・サーバーを選択し、「OK」をクリックします。

「ターゲット・オブジェクト・プロファイルの管理」ウィンドウで、プロファイルを作成するターゲット・サーバーを選択します。ターゲット表の特性、ターゲット表用の表スペース、およびターゲット表用の索引を定義します。これらのオブジェクトごとに切り捨て規則を選択することもできます。

レプリケーション・コントロール表の作成

レプリケーション・コントロール表には、レプリケーション環境のセットアップに関するすべての情報が保管され、キャプチャー・プログラムおよびアプライ・プログラムがレプリケーション中に使用する操作情報が保管されます。データベース・サーバーをレプリケーション・センターに追加する前に、レプリケーション・コントロール表を作成する必要があります。レプリケーション・コントロール表を特定のデータベース内に作成すると、レプリケーション・センターはそのデータベース・サーバーをレプリケーション・センターのオブジェクト・ツリーに自動的に追加します。

レプリケーション・センターを使用して OS/400 システム用のレプリケーション・コントロール表を作成することはできませんが、その理由は、DB2 DataPropagator for iSeries のインストール時にレプリケーション・コントロール表が作成されるためです。コントロール表を再作成する場合、または代替のキャプチャー・スキーマを使用してコントロール表を作成する場合は、OS/400 **CRTDPRTBL** コマンドを使用してください。

同じデータベース内にすべてのレプリケーション・コントロール表を作成すれば、キャプチャー・コントロール・サーバーはアプライ・コントロール・サーバーとしての役割も果たします。同様に、キャプチャー・コントロール・サーバーまたはアプライ・コントロール・サーバーは、モニター・コントロール・サーバーとしての役割も果たします。

キャプチャー・コントロール表の作成

キャプチャー・コントロール・サーバー用のコントロール表は、次の 2 つの方法で作成することができます。

- 「コントロール表の作成」 - 「クイック」 - 「サーバー情報」ウィンドウをオープンします。
 1. 「SQL レプリケーション」フォルダーを展開します。
 2. 「定義」フォルダーを展開します。
 3. 「キャプチャー・コントロール・サーバー」フォルダーを右クリックし、「キャプチャー・コントロール表の作成」→「クイック」を選択します。
 4. 「サーバーの選択」ウィンドウで、キャプチャー・コントロール表を作成するサーバーを選択し、「OK」をクリックします。

「コントロール表の作成」 - 「クイック」 - 「サーバー情報」ウィンドウでは、ユーザーのレプリケーション環境についていくつか簡単な質問をします。レプリケーション・センターはユーザーの答えに基づいて、該当のサイズの特定の表スペース内にレプリケーション・コントロール表を作成し、さらにパフォーマンスを最適にするため、これらの表スペース内のコントロール表をグループにまとめます。

- 「キャプチャー・コントロール表の作成」ウィンドウをオープンします。
 1. 「SQL レプリケーション」フォルダーを展開します。
 2. 「定義」フォルダーを展開します。
 3. 「キャプチャー・コントロール・サーバー」フォルダーを右クリックし、「キャプチャー・コントロール表の作成」→「カスタム」を選択します。
 4. 「サーバーの選択」ウィンドウで、キャプチャー・コントロール表を作成するサーバーを選択し、「OK」をクリックします。このサーバーは、キャプチャー・プログラムを実行するサーバーです。データベースが、DB2 以外のリレーショナル・ソース用のゲートウェイとして働くフェデレーテッド・データベースである場合は、そのフェデレーテッド・データベースを選択し、そのデータベースからサーバー・マッピングを検索し、そのフェデレーテッド・データベース用に定義された DB2 以外のリレーショナル・サーバーのリストを表示します。

「キャプチャー・コントロール表の作成」ウィンドウで、各コントロール表の特性を定義します。

- a. リストからレプリケーション・コントロール表を選択します。
- b. 選択したコントロール表の表スペースと索引の特性を定義します。

別の表スペースを作成するコントロール表を必ずすべて選択し、そのコントロール表の表スペースと索引の特性を定義してください。

ヒント: 選択したデータベースのオペレーティング・システム・プラットフォーム用のコントロール表プロファイルを作成した場合は、そのプロファイルから設定値を受け入れるか、あるいはオーバーライドすることができます。

キャプチャー・コントロール表用にユニークなスキーマ名を指定することができます。デフォルトは ASN です。選択したデータベースに複数のキャプチャー・プログラムを実行する場合は、別のスキーマ名が必要です。

- c. すべてのコントロール表を定義したら、「OK」をクリックします。

アプライ・コントロール表の作成

アプライ・コントロール・サーバー用のコントロール表は、次の 2 つの方法で作成することができます。

- 「コントロール表の作成」 - 「クイック」 - 「サーバー情報」ウィンドウをオープンします。
 1. 「SQL レプリケーション」フォルダーを展開します。
 2. 「定義」フォルダーを展開します。
 3. 「アプライ・コントロール・サーバー」フォルダーを右クリックし、「アプライ/キャプチャー・コントロール表 (Apply Capture Control Tables)」→「クイック」を選択します。
 4. 「サーバーの選択」ウィンドウで、アプライ・コントロール表を作成するサーバーを選択し、「OK」をクリックします。

「コントロール表の作成」 - 「クイック」 - 「サーバー情報」ウィンドウでは、ユーザーのレプリケーション環境についていくつか簡単な質問をします。レプ

リケーション・センターはユーザーの答えに基づいて、該当のサイズの特定の表スペース内にレプリケーション・コントロール表を作成し、さらにパフォーマンスを最適にするため、これらの表スペース内のコントロール表をグループにまとめます。

- 「アプライ・コントロール表の作成」ウィンドウをオープンします。
 1. 「**SQL レプリケーション**」フォルダーを展開します。
 2. 「**定義**」フォルダーを展開します。
 3. 「**アプライ・コントロール・サーバー**」フォルダーを右クリックし、「**アプライ/キャプチャー・コントロール表 (Apply Capture Control Tables)**」→「**カスタム**」を選択します。
 4. 「**サーバーの選択**」ウィンドウで、アプライ・コントロール表を作成するサーバーを選択し、「**OK**」をクリックします。

「アプライ・コントロール表の作成」ウィンドウで、各コントロール表の特性を定義します。

- a. リストからレプリケーション・コントロール表を選択します。
- b. 選択したコントロール表の表スペースと索引の特性を定義します。

別の表スペースを作成するコントロール表を必ずすべて選択し、そのコントロール表の表スペースと索引の特性を定義してください。

ヒント: 選択したデータベースのオペレーティング・システム・プラットフォーム用のコントロール表プロファイルを作成した場合は、そのプロファイルから設定値を受け入れるか、あるいはオーバーライドすることができます。

- c. すべてのコントロール表を定義したら、「**OK**」をクリックします。

モニター・コントロール表の作成

モニター・コントロール表は、Linux、UNIX、Windows、VM/VSE、または z/OS データベースに作成することができます。OS/400 データベースまたは DB2 以外のリレーショナル・データベースにモニター・コントロール表を作成することはできません。モニター・コントロール・サーバーを使用すれば、OS/400 データベースを含め、レプリケーション・ネットワーク内の任意の DB2 データベースのレプリケーション・アクティビティをモニターすることができます。

モニター・コントロール・サーバー用のコントロール表を作成するには、次のようにします。

1. 「**モニターおよびアラート**」フォルダーを展開します。
2. 「**モニター・コントロール・サーバー**」フォルダーを右クリックし、「**モニター・コントロール表の作成**」を選択します。
3. 「**サーバーの選択**」ウィンドウで、モニター・コントロール表を作成するサーバーを選択し、「**OK**」をクリックします。

「モニター・コントロール表の作成」ウィンドウで、各コントロール表の特性を定義します。

- a. リストからレプリケーション・コントロール表を選択します。

- b. 選択したコントロール表の表スペースと索引の特性を定義します。

別の表スペースを作成するコントロール表を必ずすべて選択し、そのコントロール表の表スペースと索引の特性を定義してください。

- c. すべてのコントロール表を定義したら、「OK」をクリックします。

レプリケーション・センターへのサーバーの追加

レプリケーション・コントロール表を特定のデータベース内に作成すると、レプリケーション・センターはそのデータベース・サーバーをレプリケーション・センターのオブジェクト・ツリーに自動的に追加します。また、オブジェクト・ツリーにデータベース・サーバーを追加または除去する場合、そのデータベース内に作成した他のレプリケーション・オブジェクトに影響を与えずに行うことができ、また、そのサーバー上で実行されている可能性のあるキャプチャー・プログラム、キャプチャー・トリガー、アプライ・プログラム、またはレプリケーション・アラート・モニターに影響を与えずに行うことができます。レプリケーション・センターは以下の理由から、ローカルでカタログされているデータベースすべてを自動的にリストしません。

- レプリケーション・センターは有効なレプリケーション・オブジェクトのみを表示します。ローカルでカタログされたデータベースにレプリケーション・コントロール表が含まれない場合、レプリケーション・センターはそのデータベースをオブジェクト・ツリーに表示しません。
- ユーザーのレプリケーション環境がレプリケーション・コントロール表の作成権限を管理者に限定している場合でも、他の人々が自分に必要なデータベース内でレプリケーション・オブジェクト（たとえば、登録されたソースやサブスクリプション・セット）を管理することを許可できます。
- レプリケーション・チームの全員が同じ権限を持っている場合であっても、それぞれの人にとってみれば、特定のレプリケーション・サーバーにのみ焦点をあてたい場合があります。レプリケーション・センターが表示する以上のものがレプリケーション環境に含まれている場合でも、それぞれの人は、自分が管理したいデータベース・サーバーだけを追加することができます。

重要: レプリケーション・センターにデータベース・サーバーを追加する前に、まず、そのサーバーをローカル DB2 データベースにカタログし、データベース内に必ずレプリケーション・コントロール表が存在するようにしなければなりません。

レプリケーション・センターに以下のサーバーを追加することができます。

- キャプチャー・コントロール・サーバー

キャプチャー・コントロール・サーバーをレプリケーション・センターに追加するには、

1. 「キャプチャー・コントロール・サーバー」フォルダーを右クリックし、「追加」を選択します。「キャプチャー・コントロール・サーバーの追加」ウィザードがオープンします。
2. 「始めに」ページを読み終えたら、「次へ」をクリックして、「レプリケーション・センターへ追加するキャプチャー・コントロール・サーバーの指定」ページを表示します。

3. 「**サーバー別名**」フィールド上のプッシュボタンをクリックします。「サーバーの選択」ウィンドウがオープンします。
4. レプリケーション・センターに追加したいサーバーを選択して、「**OK**」をクリックします。「サーバーの選択」ウィンドウがクローズします。
5. 「**ユーザー ID**」および「**パスワード**」フィールドに、サーバーのユーザー ID とパスワードを入力します。
6. 「**次へ**」をクリックして、「サマリー」ページを表示します。
7. 情報が正しい場合は、「**完了 (Finish)**」をクリックします。情報が誤っている場合は、「**戻る**」をクリックして、必要な変更を加えます。

DB2 以外のリレーショナル・サーバーをレプリケーション・センターにキャプチャー・サーバーとして追加することもでき、これを行うには、「キャプチャー・コントロール・サーバーの追加」ウィンドウに表示された特定のフェデレーテッド・データベースを右クリックし、「**非 DB2 サーバーの検索**」を選択します。レプリケーション・センターは、そのフェデレーテッド・データベースに定義された、DB2 以外のリレーショナル・サーバーを表に追加します。

- **アプライ・コントロール・サーバー**

アプライ・コントロール・サーバーをレプリケーション・センターに追加するには、

1. 「**アプライ・コントロール・サーバー**」フォルダーを右クリックし、「**追加**」を選択します。「アプライ・コントロール・サーバーの追加」ウィザードがオープンします。
2. 「**始めに**」ページを読み終えたら、「**次へ**」をクリックして、「レプリケーション・センターへ追加するアプライ・コントロール・サーバーの指定」ページを表示します。
3. 「**サーバー別名**」フィールド上のプッシュボタンをクリックします。「サーバーの選択」ウィンドウがオープンします。
4. レプリケーション・センターに追加したいサーバーを選択して、「**OK**」をクリックします。「サーバーの選択」ウィンドウがクローズします。
5. 「**ユーザー ID**」および「**パスワード**」フィールドに、サーバーのユーザー ID とパスワードを入力します。
6. 「**次へ**」をクリックして、「サマリー」ページを表示します。
7. 情報が正しい場合は、「**完了 (Finish)**」をクリックします。情報が誤っている場合は、「**戻る**」をクリックして、必要な変更を加えます。

- **モニター・コントロール・サーバー**

モニター・コントロール・サーバーをレプリケーション・センターに追加するには、

1. 「**モニター・コントロール・サーバー**」フォルダーを右クリックし、「**追加**」を選択します。「モニター・コントロール・サーバーの追加」ウィザードがオープンします。
2. 「**始めに**」ページを読み終えたら、「**次へ**」をクリックして、「レプリケーション・センターへ追加するモニター・コントロール・サーバーの指定」ページを表示します。

3. 「**サーバー別名**」フィールド上のプッシュボタンをクリックします。「サーバーの選択」ウィンドウがオープンします。
4. レプリケーション・センターに追加したいサーバーを選択して、「**OK**」をクリックします。「サーバーの選択」ウィンドウがクローズします。
5. 「**ユーザー ID**」および「**パスワード**」フィールドに、サーバーのユーザー ID とパスワードを入力します。
6. 「**次へ**」をクリックして、「**サマリー**」ページを表示します。
7. 情報が正しい場合は、「**完了 (Finish)**」をクリックします。情報が誤っている場合は、「**戻る**」をクリックして、必要な変更を加えます。

データベースの変更キャプチャーの使用可能化 (UNIX および Windows)

UNIX または Windows システム上の DB2 データベースのデフォルトのロギングは循環ロギングであり、これは固定サイズのファイルを使用し、ログがいっぱいになると再利用されます。レプリケーションはアーカイブ・ロギングを必要とし、これは無限に増える 1 つまたは複数のログ・ファイルを使用し、決して再利用されることはありません (もちろん、DB2 ユーティリティーを使用してアーカイブ・ログを管理し、ディスク・スペースがすべて一杯にならないようにできます)。

DB2 データベースのアーカイブ・ロギングを使用可能にするには、次のようにします。

1. 「**SQL レプリケーション**」フォルダーを展開します。
2. 「**定義**」フォルダーを展開します。
3. 「**キャプチャー・コントロール・サーバー**」フォルダーを展開します。
4. アーカイブ・ロギングを可能にしたいデータベースを右クリックし、「**データベースのレプリケーションを使用可能にする**」を選択します。
5. 「データベースのレプリケーションを使用可能にする」ウィンドウで「**OK**」をクリックし、データベース構成を変更 (LOGRETAIN を RECOVERY に) し、データベース・バックアップを開始します。

アーカイブ・ログを管理する出口ルーチンも使用したい場合は、USEREXIT データベース構成パラメーターを手作業で設定する必要があります。

「**キャプチャー・コントロール・サーバー**」フォルダーの内容ペインで複数のデータベースを選択することもでき、選択したデータベースすべてのアーカイブ・ロギングを同時に使用可能にすることができます。

その他のオペレーティング・システム上のデータベースにアーカイブ・ロギングを使用可能にする必要はありません。その理由は、それらの環境のデフォルトのロギングはアーカイブであるためです。また、DB2 以外のリレーショナル・データベース用にアーカイブ・ロギングを使用可能にする必要もありません。その理由はキャプチャー・トリガーがデータベース・ロギングに依存してしていないからです。

ソースの登録

レプリケーション用に 1 つまたは複数の表を登録するには、次のようにします。

1. 「キャプチャー・コントロール・サーバー」フォルダーを展開します。
2. 登録したいソース表を含むデータベース・サーバーを展開します。
3. 「キャプチャー・スキーマ」フォルダーを展開します。
4. 登録したいソース表を含むスキーマを展開します。
5. 「登録済み表」フォルダーを右クリックし、「表の登録」を選択します。「登録可能な表の追加」ウィンドウがオープンします。

データベースには数百の表が含まれることがあるので、表のリストをあらかじめフィルターに掛けておけば、「表の登録」ウィンドウには必要な表だけを表示することができます。

6. 「登録可能な表の追加」ウィンドウから、検索基準があれば指定し、「検索」をクリックします。すべての表を含めたい場合は、「すべて検索」をクリックします。
7. フィルターに掛けられたリストから、レプリケーション・ソースとして登録したい 1 つまたは複数の表を選択し、「OK」をクリックします。「表の登録」ウィンドウはオープンされたままです。
8. 「選択済み表」リストから、レプリケーション・ソースとして登録したい最初の表を選択します。レプリケーション・ソースには、次の情報を定義することができます。
 - いつキャプチャー・プログラムが CD 表に行を書き込むか (または、いつキャプチャー・トリガーが整合変更データ (CCD) 表に行を書き込むか) を指定する行キャプチャー規則。
 - レプリケーションに使用したい特定の列 (変更前イメージの列および、変更後イメージの列を含む)。

登録しない列は、サブスクリプション・セットで使用できません。

推奨事項: この表をソースとして使用して定義する可能性のあるターゲット表があるかどうかを検討してください。ターゲット表のキー列をソースで更新することができる場合は、ターゲットでキー列を構成する列の変更前イメージ値をソースで登録します。(サブスクリプション・セットを作成して) どのターゲットをこのソースにサブスクライブするのかを定義するときに、アプライ・プログラムを使用して、これらの変更前イメージ値を使用して、ターゲット・キー列に特殊な更新を実行することができます。

登録しない列は、サブスクリプション・セットで使用できません。

- 登録された変更前イメージの列を変更後イメージの列と関連付けるために使用する接頭部。
- このソース表に基づいて、アプライ・プログラムにターゲット表をリフレッシュさせるかどうか。
- 変更を、削除と挿入の対としてキャプチャーするかどうか (パーティション化キーへの変更に便利)。

- Update-anywhere シナリオにおいて、従属するレプリカ内で変更を再度取り込むかどうか。
- Update-anywhere シナリオ用の、競合検出のレベル。

対等 (peer-to-peer) のシナリオの場合は、「**検出せず**」を選択する必要があります。

これらのオプションの詳細については、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。

登録されるソース表ごとに、CD 表と CD 表用の索引についての情報も指定します。このデータベース・サーバー用のソース・オブジェクト・プロファイルを作成した場合は、そのプロファイルに定義されたデフォルトを受け入れることも、デフォルトをオーバーライドすることもできます。

ビューを登録する場合は、「**登録済みビュー**」フォルダーを右クリックし、「**ビューの登録**」を選択します。ビューをレプリケーション・ソースとして登録するには、ビューは既存のものでなければなりません。ビューが存在しない場合は、「**ビューの登録**」ウィンドウ内の「**ビューの作成**」をクリックします。「**ビューの作成**」ウィンドウで、ビュー名およびビューを定義する SQL ステートメントを指定します。ビューを定義する SQL ステートメントを作成するには、「**SQL Assist**」をクリックし、「**SQL Assist**」ウィンドウを使用することができます。

リモートでジャーナルに記録されている OS/400 の表を、他の表を登録するのと同様に登録することができます。他の表の登録と異なる点は、ソース表の名前に加えて、ジャーナル・ライブラリーおよびジャーナル・レシーバー名を指定する必要があることです。ただし、ジャーナル・ライブラリーとジャーナル名が、ソース表またはソース・ファイルで使用されるジャーナル・ライブラリーおよびジャーナル名と同じ場合は、指定する必要はありません。

表の登録と同様に、ニックネームを登録できます。ただし、実際の表名 (DB2 以外のデータベースに保管されている名前) ではなく、表のニックネーム (DB2 データベースに保管されている名前) を指定する必要があります。

サブスクリプション・セットの作成

1 つまたは複数の表、ニックネーム、またはビューを登録した後、これらのソースにサブスクライブする必要があります。つまり、サブスクリプション・セットを作成し、このセットにメンバーを追加します。空のサブスクリプション・セットを作成し、後からこれにメンバーを追加することも、サブスクリプション・セットの作成中にすべてのメンバーを追加することもできます。

サブスクリプション・セットを作成するには、次のようにします。

1. 「**キャプチャー・コントロール・サーバー**」フォルダーを展開します。
2. サブスクリプション・セットを作成したいソース表を含むデータベース・サーバーを展開します。
3. 「**キャプチャー・スキーマ**」フォルダーを展開します。
4. サブスクリプション・セットを作成したいソース表を含むスキーマを展開します。

5. 「登録済み表」フォルダーをクリックします。
6. 「登録済み表」フォルダーの内容ペインで、ソース表を右クリックし、「サブスクリプション・セットの作成」を選択します。「サブスクリプション・セットの作成」ウィンドウがオープンします。

別の方法として、次のステップを使用してサブスクリプション・セットを作成することもできます。

1. 「アプライ・コントロール・サーバー」フォルダーを展開します。
2. 特定のアプライ・コントロール・サーバーを展開します。
3. 「サブスクリプション・セット」フォルダーを右クリックし、「作成」を選択します。「サブスクリプション・セットの作成」ウィンドウがオープンします。

サブスクリプション・セットの作成は、主に次の 4 つのサブタスクからなります。

- 『サブスクリプション・セットの情報の定義』
- 284 ページの『ソースからターゲットへのマッピング』
- 285 ページの『サブスクリプション・セットのスケジューリング』
- 286 ページの『サブスクリプション・セットへの SQL ステートメントまたはストアド・プロシージャの追加』

サブスクリプション・セットを作成した後、サブスクリプション・セットの編集、サブスクリプション・セットのメンバーの追加や除去、ステートメントやプロシージャの追加や除去、サブスクリプション・セットの活動化、そのメンバーの強制フル・リフレッシュ、あるいは、それを別のデータベースにプロモートする、といった作業を行うことができます。

サブスクリプション・セット作成の詳細については、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。

サブスクリプション・セットの情報の定義

「サブスクリプション・セットの作成」ウィンドウで、サブスクリプション・セットに次の情報を定義することができます。

- アプライ・コントロール・サーバーの別名
- サブスクリプション・セット名
- アプライ修飾子
- キャプチャー・コントロール・サーバーの別名
- キャプチャー・コントロール表 (サブスクリプション・セットに登録されたソースを定義する表) のセットを識別する、キャプチャー・スキーマ
- ターゲット・サーバーの別名
- サブスクリプション・セットを作成したらすぐにアクティブにするかどうか

デフォルトでは、新しいサブスクリプション・セットが作成されると、すぐに非アクティブになります。サブスクリプション・セットを即時にアプライ・プログラムの処理対象にするか、あるいは 1 回のアプライ・サイクルだけアクティブにするかを選択できます。

- サブスクリプション・セットの処理プロパティ

ソースからターゲットへのマッピング

サブスクリプション・セット情報を定義した後、ソース表およびビューをターゲット表にマッピングすることができます。「サブスクリプション・セットの作成」ウィンドウの「ソースからターゲットへのマッピング」ページで、次のようにします。

1. 「追加」をクリックして、「登録済みソースの追加」ウィンドウを表示します。このウィンドウから、選択されたソース・データベースの登録済みソースのリストをフィルターに掛けることができます。
2. フィルターに掛けられたリストから、サブスクリプション・セット・メンバーのソースとして追加したい 1 つまたは複数の表を選択し、「OK」をクリックします。「サブスクリプション・セットの作成」ウィンドウは「ソースからターゲットへのマッピング」ページでオープンされたままです。
3. 「ソースからターゲットへのマッピング」ページの表で、ターゲットのスキーマ、名前、またはターゲットのタイプを選択し、これらの値のどれでも、まだ存在しないターゲット表用の値に変更します。ソースを選択した後、ターゲット・スキーマおよびターゲット名は、選択されたターゲット・サーバーのターゲット・オブジェクト・プロファイル (ある場合) に基づいて、自動的に生成されます。
4. 「ソースからターゲットへのマッピング」ページの表で、表内のソースとターゲットの対を選択し、「変更」をクリックして「メンバー・プロパティ」ウィンドウを表示します。このウィンドウから、以下のような情報を含む、ソース表とターゲット表の正確なマッピングを指定することができます。
 - ターゲットがサブスクライブするソース列の選択
 - ソース列からターゲット列へのマッピング (算出列の作成を含む)
 - ターゲット表の索引の指定

重要: ターゲットの索引を構成する列をソースで更新できる場合は、ターゲット・キー変更オプションを選択します。このオプションはアプライ・プログラムに、ターゲット・キーが変更されるときには必ず、ターゲット・キー列に特殊な更新を加えるように指示します。アプライ・プログラムはソースのその列の変更前イメージ値を使用して、ターゲットにそれらの特殊な更新を加えます。ターゲットでキーを構成する列の変更前イメージ値をキャプチャーするためのソース登録を定義していない場合は、ターゲット・キー変更オプションを選択する前に、登録を変更して変更前イメージ値を組み込まなければなりません。

- オプションとして、WHERE 文節を使用してソースの行をフィルターに掛け、ターゲット表にはソース・データのサブセットだけを含めるようにする
- Linux、UNIX、Windows、および z/OS システムの場合は、ターゲット表用の表スペースを指定する

レプリカのターゲット・タイプの場合、レプリカ定義 (行のキャプチャー規則、変更を再キャプチャーするかどうか、および更新をどのように扱うか)、レプリカ表の CD 表、および CD 表の索引も指定します。

CCD 表の場合は CCD 表のプロパティも指定します。このプロパティには、CCD 表が完全であるか不完全であるか、圧縮されているか、圧縮されていないか、およびこれをレプリケーション・ソースとして登録したいかどうか、が含まれます。

空のサブスクリプション・セットを作成する場合は、「ソースからターゲットへのマッピング」ページを空のままにしておきます。後から、サブスクリプション・セット・メンバーをサブスクリプション・セットに追加することができます。次のノートブックの 1 つを使用して、既存のサブスクリプション・セットにメンバーを追加することができます。

- サブスクリプション・セット・プロパティ。すでにサブスクリプション・セットを作成済みで、このセットに 1 つ以上のサブスクリプション・セット・メンバーを追加する場合は、このノートブックを使用します。

「サブスクリプション・セット」フォルダーの内容ペインからサブスクリプション・セットを右クリックし、「プロパティ」を選択します。

- サブスクリプション・セットにメンバーを追加する。1 つのメンバーを複数のサブスクリプション・セットに追加するには、このノートブックを使用します。たとえば、このノートブックを開いて 4 つのサブスクリプション・セットを選択した場合、1 つのメンバーをそれぞれのセットに追加できます。各メンバーは同じソースを使用する必要があります。

「登録済み表」フォルダーの内容ペインで、ソース表を右クリックし、「メンバーの追加」を選択します。

「サブスクリプション・セットにメンバーを追加する」ノートブックを使用すると、複数のサブスクリプション・セットに登録済みソースを追加することができます。これを使用すれば、いくつかの空のサブスクリプション・セットを作成し、これらのセットに同じ「ソースからターゲットへのマッピング」を移植することができます。「サブスクリプション・セットにメンバーを追加する」ノートブックで選択されたすべてのサブスクリプション・セットは、同じキャプチャー・サーバーおよびキャプチャー・スキーマを使用する必要があります。

サブスクリプション・セットのスケジューリング

ソースからターゲットへのマッピングを行った後 (または空のサブスクリプション・セットを作成した後)、サブスクリプション・セットのタイミング情報を定義します。「サブスクリプション・セットの作成」ウィンドウの「スケジュール」ページで、サブスクリプション・セットを最初に処理の対象とするのはいつかを指定します。デフォルトは、ローカル・マシンの現在の日付と時刻です。また、サブスクリプション・セットをどのような頻度で処理の対象にするかも指定します。次のような頻度を指定できます。

- 時間に基づくレプリケーション

アプライ・プログラムは、定期的な時間間隔を使用してこのサブスクリプション・セットを処理します。

- イベント・ベースのレプリケーション

アプライ・プログラムは、あるイベントが起こるたびにこのサブスクリプション・セットを処理します。

- ・ 時間に基づくレプリケーションとイベント・ベースのレプリケーションの両方

アプライ・プログラムは、定期的な時間インターバルおよびイベントが起こる程度の両方を使用して、このサブスクリプション・セットを処理します。この場合、サブスクリプション・セットは、スケジュールされた時刻および、イベントが起こった時の両方で処理の対象になります。

サブスクリプション・セットへの SQL ステートメントまたはストアード・プロシージャの追加

サブスクリプション・セットのタイミング情報を定義した後、オプションとして、サブスクリプション・セットに SQL ステートメントまたはストアード・プロシージャを追加することができます。「サブスクリプション・セットの作成」ウィンドウの「ステートメント」ページで、アプライ・プログラムがサブスクリプション・セットを処理する時に実行する、SQL ステートメントまたはストアード・プロシージャを追加することができます。「追加」をクリックし、サブスクリプション・セットにステートメントまたはプロシージャを追加します。

「SQL ステートメントまたはプロシージャ呼び出しの追加」ウィンドウで、SQL ステートメントを入力するか、または SQL Assist を使用してステートメントを定義することができます。これらのステートメントまたはプロシージャを、ターゲット・サーバーでサブスクリプション・セットを処理する前または後に実行すべきこと、またはキャプチャー・コントロール・サーバーでサブスクリプション・セットを処理する前に実行すべきことを指定できます。また、アプライ・プログラムが正常終了として受け入れる SQLSTATE 値 (たとえば、存在しない行を削除しようとした場合の 02000 など) を追加することもできます。これらの SQLSTATE 値は正常として受け入れられるので、アプライ・トレール表 (IBMSNAP_APPLYTRAIL) にエラー条件は示されず、したがってレプリケーション・アラート・モニターはこれらについてアラートを生成しません。

サブスクリプション・セットの活動化または非活動化

通常、サブスクリプション・セットはアクティブにして、アプライ・プログラムで処理できるようにしておきます。しかし、一時的に、または無期限に、サブスクリプション・セットを非アクティブにしておきたい場合があります。サブスクリプション・セットの作成時に非アクティブにした場合は、いずれアクティブにすることが必要になります。

アクティブなサブスクリプション・セットを非アクティブにするには、次のようにします。

1. 「サブスクリプション・セット」フォルダーをクリックし、内容ペインにサブスクリプション・セットを表示します。
2. アクティブなサブスクリプション・セットを右クリックし、「非アクティブ化」を選択します。レプリケーション・センターは、サブスクリプション・セットを即時に非活動化します。

非アクティブなサブスクリプション・セットをアクティブにするには、次のようにします。

1. 「サブスクリプション・セット」フォルダーをクリックし、内容ペインにサブスクリプション・セットを表示します。
2. 非アクティブなサブスクリプション・セットを右クリックし、次のオプションの1つを選択します。
 - サブスクリプション・セットをアクティブにする場合は、「活動化」→「無期限」
 - サブスクリプション・セットを1回のアプライ・サイクルのみアクティブにする場合は、「活動化」→「1回のみ」

レプリケーション・センターは、サブスクリプション・セットを即時にアクティブにします。

レプリケーション・オブジェクトのプロモート

ソースを登録し、データベース・サーバーにサブスクリプション・セットを作成した後、ソースを再度登録したり、サブスクリプション・セットを再び作成したりせずに、レプリケーション定義を別のデータベースにコピーしたい場合があります(たとえば、テスト・システムから実動システムに)。レプリケーション・センターはプロモート関数を提供し、レプリケーション定義をあるデータベースから別のデータベースにコピーする作業を支援します。

制約事項:

- プロモート関数を使用してレプリケーション定義をコピーできるのは、類似のシステム間のみです。たとえば、ある DB2 (UNIX および Windows) システムから、別の DB2 (UNIX および Windows) システムへはコピー可能ですが、DB2 (UNIX および Windows) システムから DB2 for z/OS システムへはコピーできません。システムのオペレーティング・システム・プラットフォームがすべて同じであれば、OS/400、Linux、UNIX、Windows、または z/OS システム用にプロモート関数を使用できます。
- プロモート関数を使用して、DB2 以外のデータベースまたはフェデレーテッド・データベース・オブジェクト用のレプリケーション定義をコピーすることはできません。
- プロモート関数を使用して、OS/400 リモート・ジャーナルを含むレプリケーション定義をコピーすることはできません。

登録済みの表またはビューのプロモート

登録済みの表をプロモートするには、次のようにします。

1. 「登録済みの表」フォルダーをクリックし、内容ペインに登録済みのソース表を表示します。
2. ソース表を右クリックし、「プロモート」を選択します。「登録済みの表のプロモート」ウィンドウがオープンします。
3. 「登録済みの表のプロモート」ウィンドウで、登録情報のコピー先のデータベース・サーバーについて、次の情報を指定します。
 - キャプチャー・コントロール・サーバーの別名

登録済みソース表の新しいキャプチャー・コントロール・サーバーを選択します。

- キャプチャー・スキーマ

登録済みソース表の新しいキャプチャー・スキーマを指定します。

- CD 表のスキーマ

ソース表に関連する CD 表の新しいスキーマ名を指定します。

- 表スキーマ

表の新しいスキーマ名を指定します。プロモート関数を使用して、新しいデータベース内にソース表を作成することができます。

登録済みのビューをプロモートするには、「登録済みのビュー」フォルダーをクリックし、内容ペインに登録済みのソース・ビューを表示して、ソース・ビューを右クリックし、「プロモート」を選択します。

サブスクリプション・セットのプロモート

サブスクリプション・セットをプロモートするには、次のようにします。

1. 「サブスクリプション・セット」フォルダーをクリックし、内容ペインにサブスクリプション・セットを表示します。
2. サブスクリプション・セットを右クリックし、「プロモート」を選択します。「サブスクリプション・セットのプロモート」ウィンドウがオープンします。
3. 「サブスクリプション・セットのプロモート」ウィンドウで、サブスクリプション・セット情報のコピー先のデータベース・サーバーについて、次の情報を指定します。

- アプライ・コントロール・サーバーの別名

サブスクリプション・セットの新しいアプライ・コントロール・サーバーを選択します。サブスクリプション・セットにすでに定義されているアプライ・コントロール・サーバーを選択することができます。

- キャプチャー・コントロール・サーバーの別名

サブスクリプション・セットの新しいキャプチャー・サーバーを選択します。サブスクリプション・セットにすでに定義されているキャプチャー・コントロール・サーバーを選択することができます。

- ターゲット・サーバーの別名

サブスクリプション・セットの新しいターゲット・サーバーを選択します。サブスクリプション・セットにすでに定義されているターゲット・サーバーを選択することができます。

- アプライ修飾子

サブスクリプション・セットの新しいアプライ修飾子を入力します。

- サブスクリプション・セット名

サブスクリプション・セットの新しい名前を入力します。

- キャプチャー・スキーマ

サブスクリプション・セット内のソース表の新しいキャプチャー・スキーマを入力します。

- ソース表またはビューのスキーマ名

サブスクリプション・セット内のソース表の新しいスキーマ名を入力します。

- ターゲット表またはビューのスキーマ名

サブスクリプション・セット内のターゲット表の新しいスキーマ名を入力します。

現行のサブスクリプション・セット定義の値を使用したい場合は、すべてのフィールドをブランクのままにできます。

ターゲット表の強制フル・リフレッシュ

ターゲット表の再ロードが必要になる場合があります。たとえば、ソース・データベースのログまたはジャーナルに不一致があると、キャプチャー・プログラムは停止し、コールド・スタートが要求され、その結果、すべてのターゲット表をそのソース・データベースに基づいてすべてリフレッシュ (フル・リフレッシュ) することが必要になります。小さな表の場合は、アプライ・プログラムに自動的にフル・リフレッシュを行わせてもかまわないでしょう。しかし大きな表の場合は、ASNLOAD 出口ルーチンを使用する必要があります。

レプリケーション・センターを使用して、通常はアプライ・プログラムが実行するフル・リフレッシュをバイパスし、ユーザーが、ソース表からのアンロードまたは抽出および、ターゲット表へのロードを実行することができます。レプリケーション・センターは、レプリケーション・コントロール表に必要な変更を行い、ロードが完了した後、レプリケーションが確実に継続できるようにします。

手動フル・リフレッシュを実行するには、次のようにします。

1. 「サブスクリプション・セット」フォルダーをクリックし、内容ペインにサブスクリプション・セットを表示します。
2. サブスクリプション・セットを右クリックし、「フル・リフレッシュ」→「手動」を選択します。
3. 「フル・リフレッシュ - 手動の概要」ウィンドウにあるテキストを読み、「次へ」をクリックします。
4. 続いて表示されるウィンドウで「次へ」をクリックし、タスクを完了します。

「フル・リフレッシュ - 手動」ウィンドウでは、以下のステップを実行することができます。

1. 選択したサブスクリプション・セットの現行のサブスクリプションを使用不可にする。

サブスクリプション・セットを使用不可にした後、ソース表をアンロードし、ターゲット表にロードすることができます。

2. 選択したサブスクリプション・セットのサブスクリプションを、再び使用可能にする。

ステップ 1 とステップ 2 用に生成された SQL スクリプトは、即時に、または後で実行することができます。これらのステップは、必ず「フル・リフレッシュ - 手動」ウィンドウで指定された順序で実行してください。そうしない場合、レプリケーション環境は予測不能な結果になる可能性があります。

自動フル・リフレッシュを行うため、アプライ・プログラムが次のアプライ・サイクルでフル・リフレッシュを開始するようにするには、次のようにします。

1. 「サブスクリプション・セット」フォルダーをクリックし、内容ペインにサブスクリプション・セットを表示します。
2. サブスクリプション・セットを右クリックし、「フル・リフレッシュ」→「自動」を選択します。

レプリケーション定義の除去または削除

レプリケーション・センターを使用して、作成済みのレプリケーション定義をどれでも除去または削除することができます。以下のタスクのいずれかを実行することができます。

- レプリケーション・センターからユーザー ID を除去する。
- 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、またはモニター・コントロール・サーバーから、レプリケーション・コントロール表をドロップする (OS/400 システムには該当しない)。
- 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、またはモニター・コントロール・サーバーを、レプリケーション・センターから除去する。
- ソース表またはビューの登録を削除する。
- サブスクリプション・セットを削除する。
- サブスクリプション・セットからメンバーを削除する。
- サブスクリプション・セットからステートメントを削除する。
- サブスクリプション・セットからストアード・プロシージャを除去する。

各タスクの詳細については、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。

キャプチャー・プログラムの操作

レプリケーション・センターから、レプリケーションのための日常の操作タスクを実行することができます。たとえば、キャプチャー・プログラムを始動または停止することができます。キャプチャー・プログラムを操作するには、「操作」フォルダーを展開して、「キャプチャー・コントロール・サーバー」フォルダーをクリックし、現在定義されているキャプチャー・コントロール・サーバーを内容ペインに表示します。キャプチャー・コントロール・サーバーの 1 つを右クリックし、次の操作の 1 つを選択します。

- キャプチャー・プログラムを始動する。
- キャプチャー・プログラムを停止する。
- キャプチャー・プログラムを中断する。
- キャプチャー・プログラムを再開する (延期の後)

- キャプチャーの整理処理を開始し、CD 表、UOW 表、キャプチャー・モニター表、キャプチャー・トレース表、およびシグナル表の整理を行う。
- キャプチャー・プログラムを再初期化し、登録表を再度読み取る。
- キャプチャー・パラメーター表に保管された値を表示または変更する。
- キャプチャー・プログラムが使用している現行のパラメーターを表示または変更する。
- キャプチャー・プログラムが出したメッセージを表示する。
- キャプチャー・プログラムが収集した統計を表示する。
 - キャプチャー・プログラムが CD 表に挿入またはスキップした行数
 - キャプチャー・プログラムが CD 表から除去した行数
 - キャプチャー・プログラムがコミットしたトランザクションの数
 - キャプチャー・プログラムが使用しているメモリーの量
- キャプチャー・プログラムの平均待ち時間を表示する。
- キャプチャー・プログラムの状況を照会する。

レプリケーション・ネットワーク内で実行中のどのキャプチャー・プログラムに対しても、これらのタスクのどれでも実行することができます。

アプライ・プログラムの操作

レプリケーション・センターを使用して、アプライ・プログラムを操作することもできます。アプライ・プログラムを操作するには、「操作」フォルダーを展開して、「アプライ・コントロール・サーバー」フォルダーをクリックし、アプライ・コントロール・サーバーの 1 つを展開してから、「アプライ修飾子」フォルダーをクリックして、現在定義されているアプライ修飾子を内容ペインに表示します。アプライ修飾子の 1 つを右クリックし、次の操作の 1 つを選択します。

- アプライ・プログラムを始動する。
- アプライ・プログラムを停止する。
- サブスクリプション・セット・アクティビティのレポートを表示する。
 - すべてのサブスクリプション・セットを表示する。
 - 失敗したサブスクリプション・セットを表示する。
 - 成功したサブスクリプション・セットを表示する。
 - 失敗したサブスクリプション・セットのエラー・サマリー・レポートを表示する。
- アプライ・プログラムのパフォーマンス情報を表示する。
 - CD 表からフェッチした行数を表示する。
 - サブスクリプション・セットごとの経過時間を表示する。
- サブスクリプション・セットごとのエンドツーエンドの待ち時間のレポートを表示する。
- アプライ・プログラムの状況を照会する。

レプリケーション・ネットワーク内で実行中のどのアプライ・プログラムに対しても、これらのタスクのどれでも実行することができます。

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・センターを使用して、レプリケーション・アラート・モニター用の連絡先およびアラート条件を定義することができます。

レプリケーション・アラート・モニターが、指定されたアラート条件のいずれかを検出した時に知らせる連絡先を作成するには、次のようにします。

1. 「モニターおよびアラート」フォルダーを展開します。
2. 「モニター・コントロール・サーバー」フォルダーを展開します。
3. 「モニター・コントロール・サーバー」を展開します。
4. 「連絡先」フォルダーを右クリックし、「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。
5. 「連絡先の作成」ウィンドウで、担当者の名前および E メールまたはページャーのアドレスを指定します。「連絡先グループの作成」ウィンドウで、グループの名前とグループのメンバーを指定します。

キャプチャー・プログラムのアラート条件を選択するには、次のようにします。

1. 「モニターおよびアラート」フォルダーを展開します。
2. 「モニター・コントロール・サーバー」フォルダーを展開します。
3. 「モニター・コントロール・サーバー」を展開します。
4. 「モニター」フォルダーを右クリックし、「作成」を選択します。「モニターの作成ウィザード」がオープンします。
5. 「開始」ページで、レプリケーション・アラート・モニターのインスタンスに関連付けられたモニター修飾子を指定します。
6. 「SQL レプリケーション」の下で、「キャプチャー・プログラム」チェックボックスを選択します。
7. 「次へ」をクリックし、「キャプチャー・プログラムのアラート条件の選択」ページを表示します。
8. 「追加」をクリックします。「キャプチャー・スキーマのアラート条件の選択 (Select alert conditions for Capture schemas)」ウィンドウがオープンします。
9. 次の情報を指定します。
 - モニターしたいキャプチャー・コントロール・サーバー
 - モニターしたいキャプチャー・スキーマ
 - 任意のアラート条件
10. 「OK」をクリックして、ノートブックをクローズします。
11. 「次へ」をクリックして、「サマリー」ページを表示します。
12. 情報が正しい場合は、「完了 (Finish)」をクリックします。情報が誤っている場合は、「戻る」をクリックして、必要な変更を加えます。

アプライ修飾子別にアプライ・プログラムのアラート条件を選択するには、次のようにします。

1. 「モニターおよびアラート」フォルダーを展開します。
2. 「モニター・コントロール・サーバー」フォルダーを展開します。
3. 「モニター・コントロール・サーバー」を展開します。

4. 「モニター」フォルダーを右クリックし、「作成」を選択します。「モニターの作成ウィザード」がオープンします。
5. 「開始」ページで、レプリケーション・アラート・モニターのインスタンスに関連付けられたモニター修飾子を指定します。
6. 「SQL レプリケーション」の下で、「アプライ修飾子によるアプライ・プログラム」チェック・ボックスを選択します。
7. 「アプライ修飾子 (処理するすべてのサブスクリプション・セットを含む) のアラート条件の選択 (Select alert conditions for Apply qualifiers, including all subscription sets that they process)」ページが表示されるまで、「次へ」をクリックします。
8. 「追加」をクリックします。「アプライ修飾子のアラート条件の選択 (Select alert conditions for Apply qualifiers)」ウィンドウがオープンします。
9. 次の情報を指定します。
 - モニターしたいアプライ・コントロール・サーバー
 - モニターしたいアプライ修飾子
 - 任意のアラート条件
10. 「OK」をクリックして、ノートブックをクローズします。
11. 「次へ」をクリックして、「サマリー」ページを表示します。
12. 情報が正しい場合は、「完了 (Finish)」をクリックします。情報が誤っている場合は、「戻る」をクリックして、必要な変更を加えます。

サブスクリプション・セット別にアプライ・プログラムのアラート条件を選択するには、次のようにします。

1. 「モニターおよびアラート」フォルダーを展開します。
2. 「モニター・コントロール・サーバー」フォルダーを展開します。
3. 「モニター・コントロール・サーバー」を展開します。
4. 「モニター」フォルダーを右クリックし、「作成」を選択します。「モニターの作成ウィザード」がオープンします。
5. 「開始」ページで、レプリケーション・アラート・モニターのインスタンスに関連付けられたモニター修飾子を指定します。
6. 「SQL レプリケーション」の下で、「サブスクリプション・セットによるアプライ・プログラム」チェック・ボックスを選択します。
7. 「次へ」をクリックし、「個別のサブスクリプション・セットのアラート条件の選択」ページを表示します。
8. 「追加」をクリックします。「サブスクリプション・セットのアラート条件の選択」ウィンドウがオープンします。
9. 次の情報を指定します。
 - モニターしたいアプライ・コントロール・サーバー
 - モニターしたいサブスクリプション・セット
 - 任意のアラート条件
10. 「OK」をクリックして、ノートブックをクローズします。
11. 「次へ」をクリックして、「サマリー」ページを表示します。

12. 情報が正しい場合は、「**完了 (Finish)**」をクリックします。情報が誤っている場合は、「**戻る**」をクリックして、必要な変更を加えます。

モニター修飾子用のレプリケーション・アラート・モニターを開始するには、次のようにします。

1. 「**モニターおよびアラート**」フォルダーを展開します。
2. 「**モニター・コントロール・サーバー**」フォルダーを展開します。
3. 「**モニター・コントロール・サーバー**」を展開します。
4. 「**モニター**」フォルダーを展開します。
5. モニター修飾子を右クリックし、「**モニターの開始**」を選択します。

レプリケーション・ネットワーク内で実行中のいずれのレプリケーション・アラート・モニターに対しても、これらのタスクのどれでも実行することができます。

第 16 章 基本的な SQL レプリケーションのシナリオ: DB2 for Windows

この章ではシナリオを使って、レプリケーション・センターとキャプチャー・プログラムおよびアプライ・プログラムを実際に操作してみます。ここに示す簡単なシナリオでは、DB2® レプリケーション・ソースの変更を、DB2 for Windows® Enterprise Server Edition (ESE) または Workgroup Server Edition (WSE) のデータベース内のターゲット表にコピーしているのので、そのステップをたどってみてください。

このシナリオは、以下の部分からなります。

1. 『開始する前に』
2. 296 ページの『このシナリオの計画』
3. 298 ページの『このシナリオ用のレプリケーション環境のセットアップ』
4. 312 ページの『レプリケーション環境における操作』

開始する前に

このシナリオをご使用のコンピューター上で実行する場合は、以下のステップでシステムをセットアップしてください。

1. コンピューターに DB2 for Windows がインストールされていることを確認します。
2. デフォルトの DB2 インスタンスを作成したことを確認します。このシナリオは、すべてのデータベースが同じインスタンス内にあると想定しています。
3. SAMPLE データベースへのアクセス権があることを確認します。このシナリオでは、このデータベースはソース・サーバーとキャプチャー・コントロール・サーバーの両方の役割を果たします。

SAMPLE データベースを作成するには、「ファースト・ステップ」アプリケーションを使用します（「スタート」→「プログラム」→「IBM DB2」→「セットアップ・ツール」→「ファースト・ステップ」を選択します）。データベースを作成したら、「ファースト・ステップ」ウィンドウをクローズします。

DB2 のインストール時にファースト・ステップをインストールしなかった場合は、DB2 コマンド・ウィンドウを開き、**db2sampl** コマンドを発行して、SAMPLE データベースを作成してください。

4. DB2 コントロール・センターを使って COPYDB という新しいデータベースを作成します。このデータベースは、ターゲット・サーバーとして、またアプライ・コントロール・サーバーとして使用します。データベースを作成するには、「すべてのデータベース」フォルダーを右クリックし、「作成」→「デフォルトを使用するデータベース (Database Using Default)」を選択し、指示にしたがって、デフォルト・オプションを使用して新しいデータベースを作成します。データベースの名前と別名は両方とも COPYDB にしてください。

この章のステップでは、SAMPLE データベースにある DEPARTMENT 表のデータを使用します。完全修飾名は *schema*.DEPARTMENT であり、*schema* は、表を作成したユーザー ID です。表 20 に DEPARTMENT 表を示します。

表 20. DEPARTMENT 表

DEPTNO	DEPTNAME	MGRNO	ADMRDEPT	LOCATION
A00	SPIFFY COMPUTER SERVICE DIV.	000010	A00	-
B01	PLANNING	000020	A00	-
C01	INFORMATION CENTER	000030	A00	-
D01	DEVELOPMENT CENTER	-	A00	-
D11	MANUFACTURING SYSTEMS	000060	D01	-
D21	ADMINISTRATION SYSTEMS	000070	D01	-
E01	SUPPORT SERVICES	000050	A00	-
E11	OPERATIONS	000090	E01	-
E21	SOFTWARE SUPPORT	000100	E01	-

このシナリオの残りの部分では、SAMPLE および COPYDB データベースの作成に使用したユーザー ID を使用してください。このユーザー ID はデータベースの作成者なので、レプリケーション・タスクを実行するために必要な権限 (DBADM または SYSADM) があります。

このシナリオの計画

ここでは、自分のグループがレポートを生成するアプリケーションを使用するとします。このアプリケーションでは、SAMPLE データベースの DEPARTMENT 表にある情報が必要になります。そこで、ソース表のデータを直接は使用せずに、変更をターゲット表にコピーし、レポート生成アプリケーションだけがターゲット表を読み取れるようにします。管理を簡単にするため、ターゲット表をソース表と同じマシンに保管します。

必要なのは簡単なデータ分散構成であり、変更は、1 つのレプリケーション・ソースから 1 つの読み取り専用コピーに複製されます。ここでは、レプリケーション・タスクを実行する前に考える必要がある、設計と計画の問題を扱います。

レプリケーション・ソース

レプリケーション・ソースは、SAMPLE データベースの *schema*.DEPARTMENT 表です。環境のセットアップに先立って、その表から何を複製したいかを決める必要があります。このシナリオでは、すべての列を登録し、すべての列をサブスクライブすることになります。

レプリケーション・ターゲット

レプリケーション・ターゲットを COPYDB データベースにすることを決め、これは 295 ページの『開始する前に』で作成しました。現時点では、このデータベー

スにはターゲット表がありません。ターゲット表は、ユーザーの指定に従ってレプリケーション・センターに作成させることにします。このように、ターゲット表を自動的に生成する方法は、レプリケーション・ソースとの正確なマッピングができるので、望ましい方法です。既存のターゲット表を使用することもできますが、このシナリオではターゲット表は存在していないと想定しています。

COPYDB 内のターゲット表に、表 21 に示された情報の列を含めることにします。

表 21. COPYDB 表の列

列	説明
DEPTNO	レプリケーション・ソース表の DEPTNO 列にある情報。この列は、ターゲット表の主キーになります。
DEPTNAME	レプリケーション・ソース表の DEPTNAME 列にある情報。
MGRNO	レプリケーション・ソース表の MGRNO 列にある情報。
ADMRDEPT	レプリケーション・ソース表の ADMRDEPT 列にある情報。
LOCATION	レプリケーション・ソース表の LOCATION 列にある情報。

ターゲット表の列はソース表のデータを単純に反映したものであり、ソース表の行ごとにターゲット表には 1 行だけが存在するので、ターゲット表のタイプとして、ユーザー・コピーを使用できます。

レプリケーション・オプション

このシナリオの目的から、ここでは、CD 表、ターゲット表、およびレプリケーション・コントロール表を、それぞれの該当のデフォルト表スペースに保管することを決めます (表 22 参照)。SAMPLE と COPYDB のデータベースは同じマシンにありますが、それぞれの表スペースは別のコンテナにあります。

表 22. このシナリオで使用される表および表スペース

データベース	表	表スペース	内容
SAMPLE	<i>schema</i> .DEPARTMENT	USERSPACE1	ソース表
	<i>schema</i> .CDDEPARTMENT	TSCDDEPARTMENT	DEPARTMENT 表の CD 表
	キャプチャー・コントロール表	TSASNCA および TSASNUOW	キャプチャー・プログラムのレプリケーション・コントロール表
COPYDB	<i>schema</i> .TGDEPTCOPY	TSTGDEPTCOPY	ターゲット表
	アプライ・コントロール表	TSASNAA	アプライ・プログラムのレプリケーション・コントロール表
	モニター・コントロール表	REPLMONT1、 REPLMONT2、および REPLMONT3	レプリケーション・アラート・モニターのレプリケーション・コントロール表

表スペース・レベルで起こりうる競合を減らすため、通常、CD 表はソース表とは別の表スペースに作成する必要があります。レプリケーション・コントロール表の表スペースについては、デフォルトを受け入れるか、またはレプリケーション・センター内にプロファイルを定義してください。実稼働環境の場合、起こりうる競合を避けるには、それぞれの表スペースを別の装置に作成するのがベストです。

レプリケーションのスケジューリングについては、DB2 レプリケーションを使用して、ソース表での変更を 1 分ごとに調べ、その変更をターゲット表に複製することになります。レポート生成アプリケーションの場合、1 分ごとに変更を反映するというような応答時間は必要ないでしょうが、レプリケーション環境をテストして、すべての機能が正確に働いているか確認することになります。

また、レプリケーション・サイクルが終わるごとに、1 週間 (7 日) 以上経過したアプライ・トレール表のレコードを削除することになります。このような整理操作により、表が大きくなりすぎるのを防ぐことができます。

このシナリオ用のレプリケーション環境のセットアップ

レプリケーション・モデルの計画を立てれば、レプリケーション環境をセットアップする準備ができたこととなります。以下のステップのほとんどはレプリケーション・センターを使用するので、レプリケーション・センターが実行されていることを確認してください (Windows の「スタート」メニューから、「プログラム」→「IBM DB2」→「汎用管理ツール」→「レプリケーション・センター」を選択します)。

ステップ 1: キャプチャー・プログラムのレプリケーション・コントロール表を作成する

キャプチャー・プログラムは、レプリケーション・コントロール表から現行の登録情報を読み取り、またキャプチャー・プログラムの状況をこの表に保管します。キャプチャー・コントロール・サーバーとして働くデータベースには、キャプチャー・コントロール表を含める必要があります。

キャプチャー・コントロール表を作成するには、次のようにします。

1. 「SQL レプリケーション」フォルダーを展開します。
2. 「定義」フォルダーを展開します。
3. 「キャプチャー・コントロール・サーバー」フォルダーを右クリックし、「キャプチャー・コントロール表の作成」→「クイック」を選択します。あるいは別の方法として、「キャプチャー・コントロール表の作成」→「カスタム」を選択し、キャプチャー・コントロール表をカスタマイズすることもできます。
4. 「サーバーの選択」ウィンドウで、SAMPLE データベースを選択します。このデータベースはキャプチャー・コントロール・サーバーになります。「OK」をクリックします。
5. 「コントロール表の作成」- 「クイック」- 「サーバー情報」ウィンドウで、「レプリケーション用のソースの受け入れとそれらのソースに対する変更のキャプチャー」を選択します。その後、「次へ」をクリックします。
6. 「コントロール表の作成」- 「クイック」- 「レプリケーションの詳細」ウィンドウで、「次へ」をクリックします。このウィンドウの情報は何も変更する必要はありません。
7. 「コントロール表の作成」- 「クイック」- 「表スペース」ウィンドウで、TSASNCA 表スペースの表スペース仕様を入力します。たとえば、バッファークラスを IBMDEFAULTBP に設定します。このシナリオでは、デフォルトのキャプチャー・スキーマ ASN を受け入れます。

8. 「コントロール表の作成」 - 「クイック」 - 「表スペース」 ウィンドウで、TSASNUOW 表スペースの表スペース仕様を入力します。
9. 「コントロール表の作成」 - 「クイック」 - 「表スペース」 ウィンドウで、両方の表スペースの情報を入力した後、「OK」をクリックします。
10. 「メッセージ・ダイアログ」 ウィンドウで「クローズ」をクリックします。このウィンドウは、キャプチャー・コントロール表を作成する SQL スクリプトの生成結果を表示します。エラーがある場合は、このウィンドウに表示されません。
11. 「今実行するか、SQL を保管する」 ウィンドウで有効なユーザー ID とパスワードを入力し、「OK」をクリックすれば、SQL スクリプトは即時に実行されます。
12. 「DB2 メッセージ」 ウィンドウに、スクリプトが正常に実行されたというメッセージが表示されます。「クローズ」をクリックします。
13. 「キャプチャー・コントロール・サーバー」 フォルダを展開します。フォルダの下に SAMPLE データベースが表示されます。

ステップ 2: ソース・データベースのレプリケーションを使用可能にする

キャプチャー・プログラムは DB2 ログを読み取り、登録された表への変更を含むログ・レコードを探します。キャプチャー・プログラムがログを読み取る前に DB2 がログ・ファイルを再利用することがないように、ログはアーカイブ・ログでなければなりません。UNIX[®] および Windows 環境の場合、DB2 のデフォルトは循環ロギングなので、この設定をアーカイブ・ロギングに変更する必要があります。

ソース・データベースのレプリケーションを使用可能にするには、次のようにします。

1. 「キャプチャー・コントロール・サーバー」フォルダを展開します。
2. SAMPLE データベースを右クリックし、「レプリケーション用データベースを使用可能にする」を選択します。
3. 「レプリケーション用データベースを使用可能にする」ウィンドウで「OK」をクリックし、SAMPLE データベースにアーカイブ・ロギングを使用して、データベースのバックアップを開始します。
4. 「バックアップ」ウィンドウで、データベース・バックアップ用の情報を指定し、「今バックアップする」をクリックします。

データベースをバックアップした後、キャプチャー・プログラムを始動することもできますが、ここではまだ始動しません。キャプチャー・プログラムを始動したい場合は、310 ページの『ステップ 7: シナリオのデータを複製する』を参照してください。

ステップ 3: レプリケーション・ソースを登録する

キャプチャー・コントロール表を作成し、データベースをレプリケーションの対象として使用可能にした後、DEPARTMENT 表をレプリケーション・ソースとして登録します。

表をレプリケーション・ソースとして登録するには、次のようにします。

1. 「SQL レプリケーション」フォルダーを展開します。
2. 「定義」フォルダーを展開します。
3. 「キャプチャー・コントロール・サーバー」フォルダーを展開します。
4. 「SAMPLE データベース」を展開します。
5. 「キャプチャー・スキーマ」フォルダーを展開します。
6. 「ASN スキーマ (ASN schema)」を展開します。
7. 「登録済み表」フォルダーを右クリックし、「表の登録」を選択します。
8. 「登録可能な表の追加」ウィンドウで、「すべて検索」をクリックし、レプリケーション・ソースとして登録できる SAMPLE データベース内のすべての表をリストします。DEPARTMENT 表を選択し、「OK」をクリックします。「表の登録」ウィンドウが、図 8 に示すように表示されます。

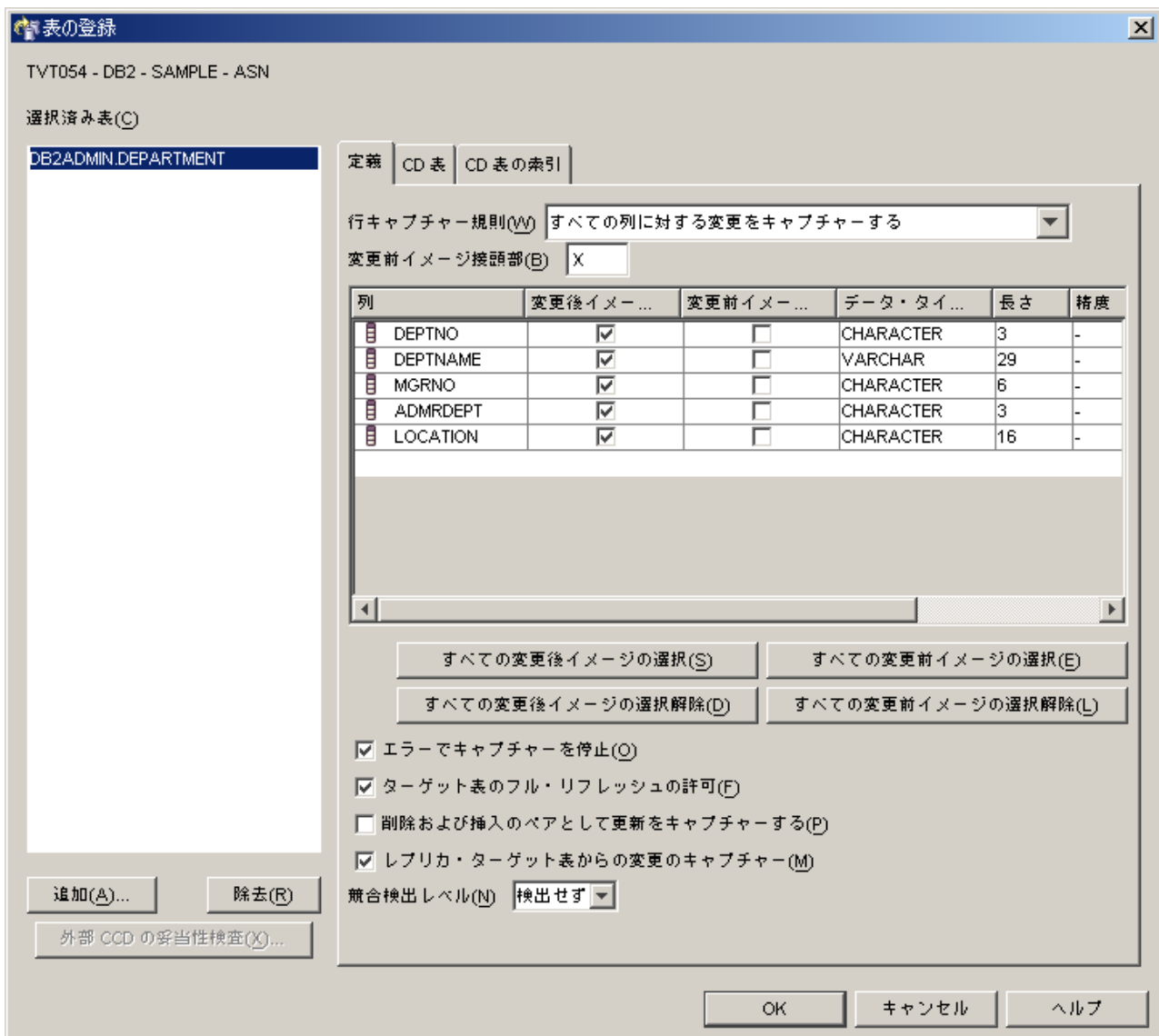


図 8. 「表の登録」ウィンドウ

9. 「表の登録」ウィンドウで、「CD 表」ノートブック・タブをクリックします。CD 表スペースについて、以下の情報を指定します。
 - 「表スペースの仕様」エリアで、「コンテナ名」フィールドをクリックし、TSCDDEPARTMENT 表スペースのコンテナ名を指定します。
 - 「表スペースの仕様」エリアで、「サイズ」フィールドを 1 に変更します。
 - 「表スペースの仕様」エリアで、「単位」フィールドを MB に変更します。
 - この新しい表スペースについてのその他の情報 (バッファ・プールを IBMDEFAULTBP に設定するなど) を指定します。

表スペース情報を入力した後、「OK」をクリックします。

10. 「メッセージ・ダイアログ」ウィンドウで「クローズ」をクリックします。このウィンドウは、ソース表を登録する SQL スクリプトの生成結果を表示します。エラーがある場合は、このウィンドウに表示されます。
11. 「今実行するか、SQL を保管する」ウィンドウで「OK」をクリックすれば、SQL スクリプトは即時に実行されます。
12. 「DB2 メッセージ」ウィンドウに、スクリプトが正常に実行されたというメッセージが表示されます。「クローズ」をクリックします。
13. SAMPLE データベース・フォルダーの内容ペインには、登録済みの表として DEPARTMENT 表が表示されます。図 9 に、SAMPLE データベース・フォルダーの内容ペインに、DEPARTMENT 表が登録済みの表として表示されている例が示されています。

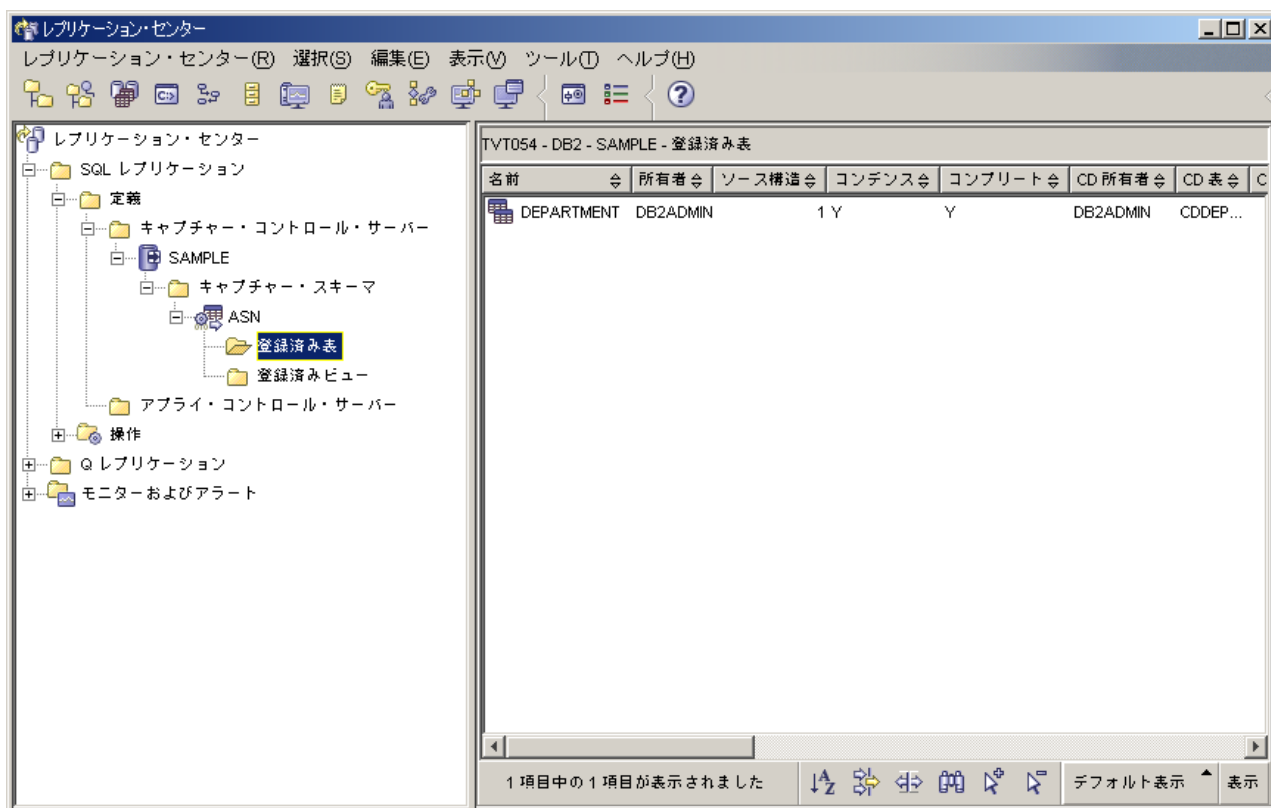


図 9. DEPARTMENT 表が SAMPLE データベースの登録済みの表としてリストされています。

これで、DEPARTMENT 表はレプリケーション・ソースとして定義されました。SQL スクリプトを実行した時に、レプリケーション・センターがこのレプリケーション・ソースの CD 表および CD 表の索引を作成し、キャプチャー・コントロール表を更新しました。

ステップ 4: アプライ・プログラムのレプリケーション・コントロール表を作成する

アプライ・プログラムは、レプリケーション・コントロール表から現行のサブスクリプション・セット情報を読み取り、またアプライ・プログラムの状況をこの表に保管します。アプライ・コントロール・サーバーとして働くデータベースには、アプライ・コントロール表を含める必要があります。

アプライ・コントロール表を作成するには、次のようにします。

1. 「SQL レプリケーション」フォルダーを展開します。
2. 「定義」フォルダーを展開します。
3. 「アプライ・コントロール・サーバー」フォルダーを右クリックし、「アプライ・コントロール表の作成」→「クイック」を選択します。あるいは別の方法として、「アプライ・コントロール表の作成」→「カスタム」を選択し、アプライ・コントロール表をカスタマイズすることもできます。
4. 「サーバーの選択」ウィンドウで、COPYDB データベースを選択します。このデータベースはアプライ・コントロール・サーバーになります。「OK」をクリックします。
5. 「コントロール表の作成」-「クイック」-「サーバー情報」ウィンドウで、「キャプチャー済み変更をターゲット表に適用」を選択します。その後、「次へ」をクリックします。
6. 「コントロール表の作成」-「クイック」-「レプリケーションの詳細」ウィンドウで、「次へ」をクリックします。このウィンドウの情報は何も変更する必要はありません。
7. 「コントロール表の作成」-「クイック」-「表スペース (Table Space)」ウィンドウで、TSASNAA 表スペースの表スペース仕様を入力します。たとえば、バッファ・プールを IBMDEFAULTBP に設定します。「OK」をクリックします。
8. 「メッセージ・ダイアログ」ウィンドウで「クローズ」をクリックします。このウィンドウは、アプライ・コントロール表を作成する SQL スクリプトの生成結果を表示します。エラーがある場合は、このウィンドウに表示されます。
9. 「今実行するか、SQL を保管する」ウィンドウで有効なユーザー ID とパスワードを入力し、「OK」をクリックすれば、SQL スクリプトは即時に実行されます。
10. 「DB2 メッセージ」ウィンドウに、スクリプトが正常に実行されたというメッセージが表示されます。「クローズ」をクリックします。
11. 「アプライ・コントロール・サーバー」フォルダーを展開します。フォルダーの下に COPYDB データベースが表示されます。

ステップ 5: 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

ソース表を登録したら、サブスクリプション・セットを作成する必要があります。サブスクリプション・セットは、レプリケーション・ソース・データベース (このシナリオでは SAMPLE) とターゲット・データベース (このシナリオでは COPYDB) のリレーションシップを定義します。サブスクリプション・セット・メンバーは、レプリケーション・ソース表 (このシナリオでは DEPARTMENT) と、1 つまたは複数のターゲット表 (このシナリオでは 1 つしかなく、名前は DEPTCOPY) のリレーションシップを定義します。

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には、次のようにします。

1. 「SQL レプリケーション」フォルダーを展開します。
2. 「定義」フォルダーを展開します。
3. 「アプライ・コントロール・サーバー」フォルダーを展開します。
4. 「COPYDB データベース」を展開します。
5. 「サブスクリプション・セット」フォルダーを右クリックし、「作成」を選択します。

別の方法として、SAMPLE データベースの「登録済み表」フォルダーを選択して、内容ペインの DEPARTMENT 表を右クリックし、「サブスクリプション・セットの作成」を選択しても、サブスクリプション・セットを作成することができます。

6. 「サブスクリプション・セットの作成」ウィンドウの「セット情報」ページに、以下の情報を入力します。
 - a. 「セット名」フィールドに、DEPTSUB と入力します。このストリングはサブスクリプション・セットを識別するものであり、特定のアプライ修飾子にユニークでなければなりません。
 - b. 「アプライ修飾子」フィールドに、DEPTQUAL と入力します。このストリングは、このレプリケーション定義を識別するものであり、このサブスクリプション・セットを実行するアプライ・プログラムのインスタンスにユニークです。

ヒント: アプライ修飾子は大文字小文字が区別されます。アプライ修飾子を小文字で指定する場合は、入力時にアプライ修飾子を区切る必要があります (たとえば "deptqual")。単に deptqual と入力すると、デフォルトにより、レプリケーション・センターはその値を大文字に変換します。

- c. 「キャプチャー・コントロール・サーバーの別名」フィールドのブラウズ・ボタンをクリックします。「キャプチャー・コントロール・サーバーの選択」ウィンドウで SAMPLE データベースを選択し、「OK」をクリックします。
- d. 「ターゲット・サーバーの別名」フィールドのブラウズ・ボタンをクリックします。「ターゲット・サーバーの選択」ウィンドウで COPYDB データベ

ースを選択し、「OK」をクリックします。COPYDB データベースは、ターゲット・サーバーであり、またアプライ・コントロール・サーバーでもあり、その両方です。

- e. 「サブスクリプション・セットの活動化」チェック・ボックスを選択します。

「セット情報」ページのその他のフィールドの設定値を変更する必要はありません。「サブスクリプション・セットの作成」ウィンドウは、図 10 と類似したウィンドウになります。

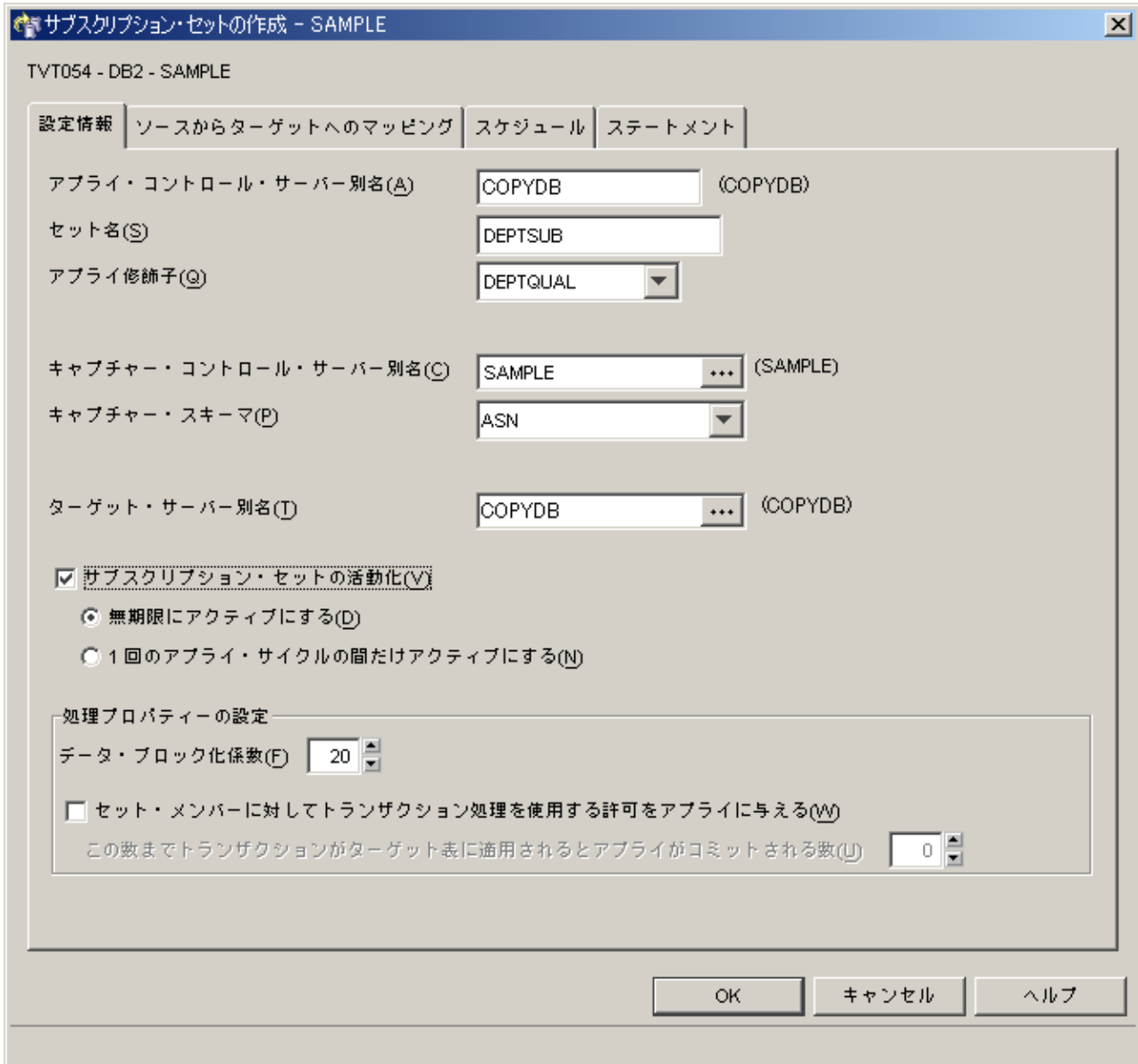


図 10. 「サブスクリプション・セットの作成」ウィンドウ

- 7. 「サブスクリプション・セットの作成」ウィンドウの「ソースからターゲットへのマッピング」ページに、以下の情報を入力します。

- a. 「追加」をクリックし、登録済みのソースをサブスクリプション・セット・メンバーに追加します。
- b. 「登録済みソースの追加」ウィンドウで、「すべて検索」をクリックし、SAMPLE データベース内の登録済みソースをすべて表示します。
- c. 「登録済みソースの追加」ウィンドウで、DEPARTMENT 表を選択し、「OK」をクリックします。
- d. 「サブスクリプション・セットの作成」ウィンドウの「ソースからターゲットへのマッピング」ページで、ターゲット表の名前を TGDEPARTMENT から TGDEPTCOPY に変更します。サブスクリプション・セット・メンバー表の「ターゲット名」列にある TGDEPARTMENT を選択し、デフォルト名を TGDEPTCOPY で上書きします。

ここではユーザー・コピー・ターゲット表を作成するので、ターゲット・タイプは変更しないでください。

- e. 「変更」をクリックし、「メンバー・プロパティ」ウィンドウをオープンします。このウィンドウから、サブスクリプション・セット・メンバーのプロパティを定義することができます。

このシナリオでは、すべての列を複製して、ソース表と同じ列をターゲット表に作成したいので、「メンバー・プロパティ」ウィンドウの「列の選択」または「列のマッピング」ページに変更を加える必要はありません。デフォルトでは、ソースに登録されたすべての列がターゲット表に含まれます。

フェデレーテッド・システムのためのヒント: 貴社のレプリケーション構成がソース表を既存のターゲット表にマップし、少なくとも 1 つの表が非 DB2 リレーショナル・データベースに由来する場合は、ソース列を既存のターゲット表の列にマップする方法について、「フェデレーテッド・システム・ガイド」を参照してください。

「メンバー・プロパティ」ウィンドウの例が 306 ページの図 11 に示されています。

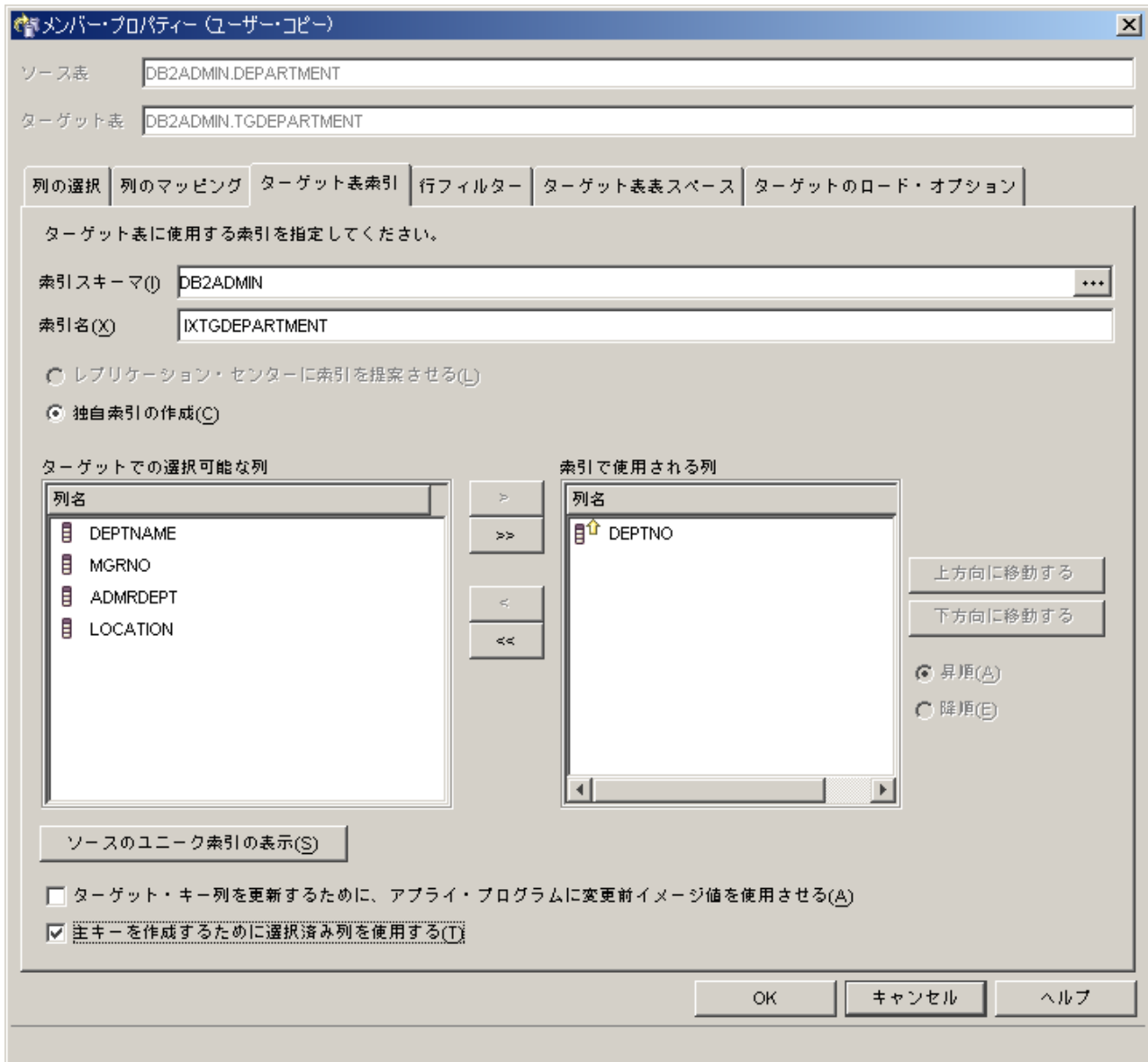


図 11. 「メンバー・プロパティ」ウィンドウ

8. 「メンバー・プロパティ」ウィンドウの「ターゲット表索引」ページで、次のようにします。
 - a. 「ターゲット内の選択可能な列」のリストから、DEPTNO 列を選択します。
 - b. 「移動」ボタン (>) をクリックし、DEPTNO 列を「索引で使用される列」リストに移動します。
 - c. 「主キーを作成するために選択済み列を使用する」を選択し、DEPTNO 列をターゲット表の主キーとして使用します。
9. 「メンバー・プロパティ」ウィンドウの「行フィルター」ページで、「WHERE ステートメント」フィールドに以下の文節を入力します。


```
DEPTNO >='E00'
```

この WHERE 文節は、特定の基準を満たす行だけを複製することを示します。ここでは、部門番号が“E00”より大きいか等しいという基準を満たす行が複製されます。この WHERE 文節により、ターゲット表には 9 行すべてではなく、3 行が含まれることとなります。

10. 「メンバー・プロパティ」ウィンドウの「ターゲット表の表スペース」ページで、新しい TSTGDEPTCOPY 表スペースについて以下の情報を指定します。
 - 「表スペースの仕様」エリアで、「コンテナ名」フィールドをクリックし、TSTGDEPTCOPY 表スペースのコンテナ名を指定します。
 - 「表スペースの仕様」エリアで、「サイズ」フィールドを 1 に変更します。
 - 「表スペースの仕様」エリアで、「単位」フィールドを MB に変更します。
 - この新しい表スペースについてのその他の情報 (バッファ・プールを IBMDEFAULTBP に設定するなど) を指定します。

この新しい表スペースについてのその他の情報 (バッファ・プールを IBMDEFAULTBP に設定するなど) を指定することもできます。

11. 「OK」をクリックし、「メンバー・プロパティ」ウィンドウをクローズします。このシナリオの場合、「ターゲット・ロード・オプション」タブでは何もする必要はありません。
12. 「サブスクリプション・セットの作成」ウィンドウの「スケジュール」ページで、分を 1 に変更し、アプライ・プログラムがこのサブスクリプション・セットを 1 分おきに処理するようにします。「レプリケーションの頻度」エリア内の「分」フィールドのスピン・ボタンを使って、1 分のインターバルを選択します (またはフィールドに 1 と入力します)。

「開始日付」、「開始時刻」、「時間に基づく」、および「相対タイミングの使用」のデフォルト値を保管します。

13. 「サブスクリプション・セットの作成」ウィンドウの「ステートメント」ページで「追加」をクリックし、「SQL ステートメントまたはプロシージャ呼び出しの追加」ウィンドウをオープンします。このウィンドウを使用して、サブスクリプション・セットの実行時に処理される SQL ステートメントを定義します。「SQL ステートメントまたはプロシージャ呼び出しの追加」ウィンドウで、以下の情報を入力します。
 - a. 「SQL ステートメント」フィールドに、次のように入力します。

```
DELETE FROM ASN.IBMSNAP_APPLYTRAIL WHERE LASTRUN  
< (CURRENT TIMESTAMP - 7 DAYS)
```

このステートメントは、アプライ・トレール表にある 7 日より前のレコードをすべて削除します。

アプライ・プログラムは、追加された SQL ステートメントを、サブスクリプション・セットを処理した後にターゲット・サーバーで実行します。

SQL ステートメントはターゲット・サーバーで実行する必要があります。

その理由は、アプライ・コントロール・サーバーとターゲット・サーバーは一緒の場所に置かれており、アプライ・トレール表がアプライ・コントロール・サーバー内にあるためです。

ヒント: アプライ・プログラムは、サブスクリプション・セットに追加された SQL ステートメントまたはプロシージャを、サブスクリプシ

ョン・サイクルごとに実行します。この例では、アプライ・プログラムがこのステートメントを 1 分おきに実行するので非効率です。実際に APPLYTRAIL 表からデータが削除されるのは、24 時間ごとになります (7 日より前のデータを削除)。

- b. 「SQLSTATE」フィールドに 02000 を入力し、「追加」をクリックします。この SQL 状態は、行が見つからない (row not found) というエラーが許容できるものであり、アプライ・プログラムがこのようなエラーを無視することを示します。

ヒント: このサブスクリプション・セットに関して、アプライ・プログラムに無視させたい SQL 状態を 10 個まで定義することができます。

- c. 「OK」をクリックして、「SQL ステートメントまたはプロシージャ呼び出しの追加」ウィンドウをクローズします。
14. 「OK」をクリックし、「サブスクリプション・セットの作成」ウィンドウをクローズします。
 15. 「メッセージ・ダイアログ」ウィンドウで「クローズ」をクリックします。このウィンドウは、アプライ・コントロール表を更新し、ターゲット表を作成する SQL スクリプトの生成結果を表示します。エラーがある場合は、このウィンドウに表示されます。
 16. 「今実行するか、SQL を保管する」ウィンドウで「OK」をクリックすれば、SQL スクリプトは即時に実行されます。

SQL スクリプトをファイルに保管して後から使用することも、即時に実行することもできます。

- a. 「ファイルに保管」を選択します。
- b. 「仕様の保管」エリアに、ファイル名などの情報を記入します。
- c. 「適用」をクリックしてファイルに保管します。スクリプトが複数の部分から成り、かつ「複数のスクリプトを 1 つのファイルに保管」チェック・ボックスを選択していない場合は、指定した名前に番号を追加した名前の付いた個別のファイルにそれぞれの部分が保管されます。「今実行するか、SQL を保管する」ウィンドウはオープンされたままです。
- d. 「今実行する」を選択します。
- e. 「OK」をクリックしてスクリプトを実行し、「今実行するか、SQL を保管する」ウィンドウをクローズします。

SQL スクリプトをファイルに保管して後から使用することも、SQL スクリプトを保管して実行することもできます。

17. 「DB2 メッセージ」ウィンドウに、スクリプトが SAMPLE サーバーと COPYDB サーバーの両方で正常に実行されたというメッセージが表示されます。「クローズ」をクリックします。
18. 「アプライ・コントロール・サーバー」フォルダーと COPYDB データベースを展開し、次に「サブスクリプション・セット」フォルダーをクリックします。「サブスクリプション・セット」フォルダーの内容ペインに、DEPTSUB サブスクリプション・セットが表示されます。309 ページの図 12 に、「サブ

「スクリプション・セット」フォルダーの内容ペインに、DEPTSUB サブスクリプション・セットが表示された例が示されています。

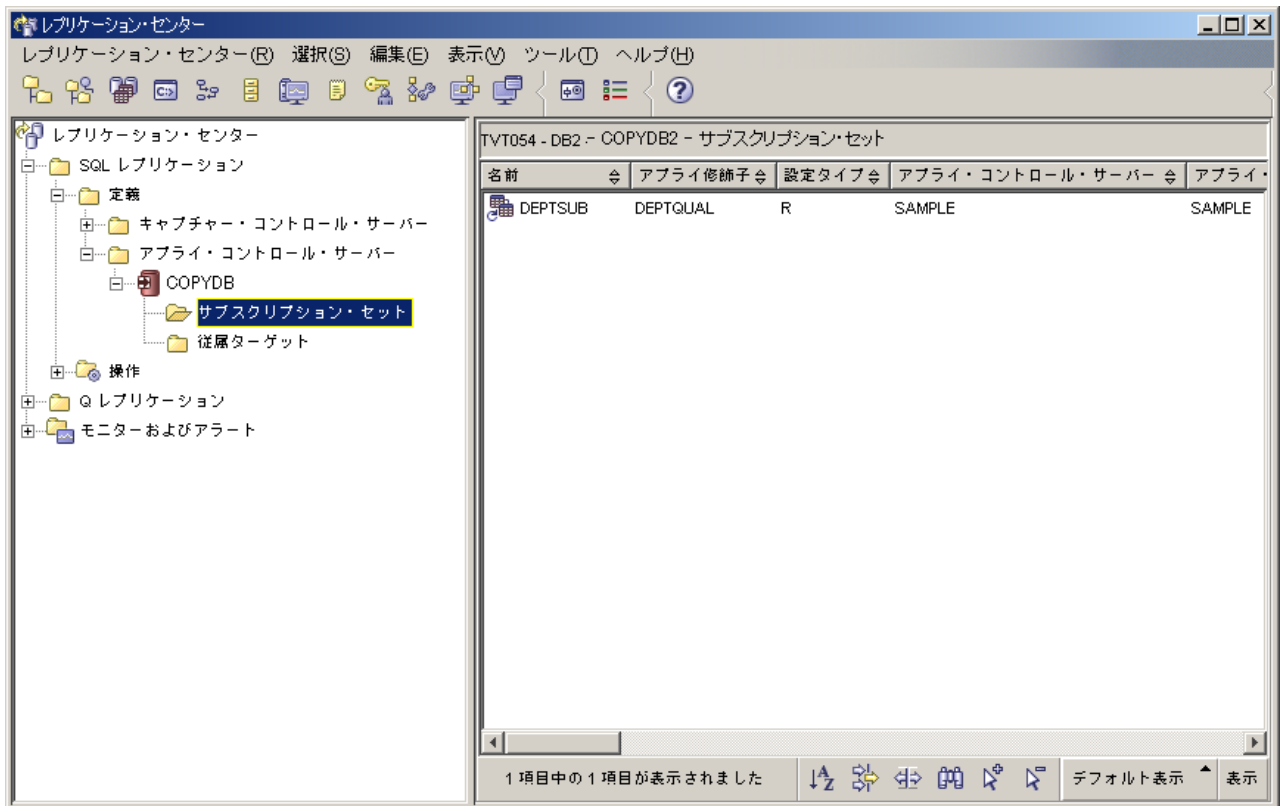


図 12. COPYDB データベースの DEPTSUB サブスクリプション・セットがリストされます。

ステップ 6: アプライ・パスワード・ファイルを作成する

アプライ・プログラムは、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、およびターゲット・サーバーと接続する必要があります。ユーザー認証のためのパスワード・ファイルを作成する必要があります。パスワード・ファイルの内容は暗号化されているため、ユーザーはファイルの変更はできませんが、ファイルの読み取りができるのはアプライ・プログラムだけです。

パスワード・ファイルを作成するには、次のようにします。

1. Windows の「コマンド・プロンプト」ウィンドウをオープンし、`C:\%sqllib%\bin` ディレクトリに変更します。
2. 次のコマンドを入力し、デフォルトのパスワード・ファイルを作成します。

```
asnpwd init using "path"
```

`path` は、パスワード・ファイルの作成に使用する、完全修飾されたディレクトリー・パスおよびファイル名です。コマンドが正常に完了したことを示すメッセージ `ASN1981I` が表示されます。

たとえば、パスワード・ファイルを `c:\%sqllib%\repl` ディレクトリに保管し、ファイル名を `asnpwd.aut` にする場合は、次のコマンドを入力します。

```
asnpwd init using "c:¥sqllib¥repl¥asnpwd.aut"
```

ヒント: パスワード・ファイルは、アプライ・プログラムを始動するディレクトリーに作成してください。アプライ・プログラムの始動時に、パスワード・ファイルのファイル名 (PWDFILE キーワードを使用して) およびアプライ・プログラムが自分のログと作業ファイルを保管するディレクトリーの値を (APPLY_PATH キーワードを使用して) 指定します。アプライ・プログラムの作業ファイルの 1 つはパスワード・ファイルです。

- 以下のコマンドを入力し、アプライ・プログラムが接続する必要のあるデータベースごとに、ユーザー ID とパスワード情報を追加します。

```
asnpwd add alias SAMPLE id userid password password using "path"
```

userid は、キャプチャー・コントロール表とアプライ・コントロール表を更新するための権限を持つ、有効な DB2 ユーザー ID です。コマンドが正常に完了したことを示すメッセージ ASN198II が表示されます。

ステップ 7: シナリオのデータを複製する

レプリケーション・ソースを登録し、サブスクリプション・セットを作成した後、キャプチャー・プログラムとアプライ・プログラムを始動し、ターゲット表の初期フル・リフレッシュを実行し、変更キャプチャー・レプリケーションを開始します。

キャプチャー・プログラムを始動するには、次のようにします。

- 「SQL レプリケーション」フォルダーを展開します。
- 「操作」フォルダーを展開します。
- 「キャプチャー・コントロール・サーバー」フォルダーを選択します。キャプチャー・コントロール・サーバーの内容ペインに SAMPLE データベースが表示されます。
- SAMPLE データベースを右クリックし、「キャプチャーの開始」を選択します。
- 「キャプチャーの開始」ウィンドウで、「キャプチャー・スキーマ」フィールド内で ASN を選択します。
- 「キャプチャーの開始」ウィンドウで「OK」をクリックします。
- 「今実行するか、コマンドを保管する」ウィンドウで「OK」をクリックすれば、コマンドは即時に実行されます。
- 「DB2 メッセージ」ウィンドウに、コマンドが正常に実行されたというメッセージが表示されます。「クローズ」をクリックします。これでキャプチャー・プログラムが実行されますが、アプライ・プログラムがすべての登録済みの表のフル・リフレッシュを完了するまでは、登録済みの表の変更キャプチャーを開始しません。

アプライ・プログラムを始動するには、次のようにします。

- 「SQL レプリケーション」フォルダーを展開します。
- 「操作」フォルダーを展開します。
- 「アプライ・コントロール・サーバー」フォルダーを展開します。

4. 「COPYDB データベース」を展開します。
5. 「**アプライ修飾子**」フォルダーを選択します。アプライ修飾子の内容ペインに、サブスクリプション・セット DEPTSUB のアプライ修飾子 DEPTQUAL が表示されます。
6. DEPTQUAL アプライ修飾子を右クリックし、「**アプライの開始**」を選択します。
7. 「アプライの開始」ウィンドウで、「**アプライが実行されている場所**」フィールドのブラウズ・ボタンをクリックし、アプライ・プログラムを実行している場所のシステム・アドレスか IP アドレスを選択します。
8. 「アプライの開始」ウィンドウで「**OK**」をクリックします。
9. 必要ならば、アプライ・プログラムをそこで実行するシステムの、有効なユーザー ID とパスワードを、「今実行するか、コマンドを保管する」ウィンドウに入力します。
10. 「今実行するか、コマンドを保管する」ウィンドウで「**OK**」をクリックすれば、コマンドは即時に実行されます。
11. 「DB2 メッセージ」ウィンドウに、コマンドが正常に実行されたというメッセージが表示されます。「**クローズ**」をクリックします。これでアプライ・プログラムが実行されます。

1 回のレプリケーション・サイクル後に TGDEPTCOPY ターゲット表を表示すると、表 23 に示すデータと一致する結果が表示されます。表の内容を表示するには、次のどの方式でも使用することができます。

• レプリケーション・センターを使用する場合:

1. 「**SQL レプリケーション**」フォルダーを展開します。
2. 「**定義**」フォルダーを展開します。
3. 「**アプライ・コントロール・サーバー**」フォルダーを展開します。
4. 「COPYDB データベース」を展開します。
5. 「**従属ターゲット**」フォルダーを右マウス・ボタンでクリックして、「**選択した内容を表示**」を選択します。

• DB2 コントロール・センターを使用して、以下のようになります。

1. DB2 インスタンスの「データベース」フォルダーを展開します。
2. 「COPYDB データベース」を展開します。
3. 「**表**」フォルダーを選択します。
4. 内容ペインの TGDEPTCOPY 表を右クリックし、「**サンプル内容**」を選択します。

• DB2 コマンド・センターまたは DB2 コマンド・ウィンドウを使用して、次の SQL ステートメントを発行します。

```
SELECT * FROM schema.TGDEPTCOPY
```

表 23. TGDEPTCOPY 表

DEPTNO	DEPTNAME	MGRNO	ADMRDEPT	LOCATION
E01	SUPPORT SERVICES	000050	A00	-
E11	OPERATIONS	000090	E01	-
E21	SOFTWARE SUPPORT	000100	E01	-

レプリケーション環境における操作

レプリケーション環境の起動後、レプリケーション・ソース表に加えられる変更はターゲット表に複製されます。キャプチャー・プログラムとアプライ・プログラムの両方の状況を表示すれば、レプリケーションの待ち時間やレプリケーション環境についてのその他の情報を見ることができます。キャプチャー・プログラムとアプライ・プログラムは連続して実行できますが、停止したい場合もあります (たとえば、コントロール表を含む表スペースを使用するユーティリティを実行する場合など)。

ステップ 1: ソース表を更新する

Spiffy Computer Service 社で新しい 2 つの部門が発足したとします。1 つは技術情報部門 (technical writing)、もう 1 つは広報部門 (public relations) です。ターゲット表にはこの両方の部門が含まれます。

ソース表を次のように更新します。

1. 「スタート」→「プログラム」→「IBM DB2」→「コマンド・ウィンドウ」を選択し、DB2 コマンド・ウィンドウをオープンします。
2. 次のように入力して、ソース・サーバーに接続します。

```
DB2 CONNECT TO SAMPLE
```

3. 次のコマンドを入力し、それぞれ Enter キーを押して、2 つの新しい行 (各部門ごとに 1 行) を追加します。

```
DB2 INSERT INTO DEPARTMENT
VALUES ('F01','TECHNICAL WRITING','000110','F01',NULL)
DB2 INSERT INTO DEPARTMENT
VALUES ('G01','PUBLIC RELATIONS','000120','G01',NULL)
DB2 COMMIT
```

4. 次のように入力して、ターゲット・サーバーに接続します。

```
DB2 CONNECT TO COPYDB
```

5. 少なくとも 1 分ほど待ってから、次のコマンドを入力し、ターゲット・データベースに新しい行が複製されていることを確認します。

```
DB2 SELECT * FROM TGDEPTCOPY
```

サブスクリプション・セットは 1 分おきにレプリケーションの対象として適格になるので、1 分待つ必要があります。データが大量にある場合は、アプライ・プログラムがそのデータをターゲット表に適用するのにもう少し時間がかかることがあります。

表 24 にレプリケーションの結果を示します。2 つの新しい行が表に追加されています。

表 24. 変更を複製した後の TGDEPTCOPY 表

DEPTNO	DEPTNAME	MGRNO	ADMRDEPT	LOCATION
E01	SUPPORT SERVICES	000050	A00	-
E11	OPERATIONS	000090	E01	-
E21	SOFTWARE SUPPORT	000100	E01	-

表 24. 変更を複製した後の TGDEPTCOPY 表 (続き)

DEPTNO	DEPTNAME	MGRNO	ADMRDEPT	LOCATION
F01	TECHNICAL WRITING	000110	F01	-
G01	PUBLIC RELATIONS	000120	G01	-

ステップ 2: キャプチャー・プログラムの状況を表示する

レプリケーション・センターを使用して、キャプチャー・プログラムについて次のような状況情報を表示します。

- キャプチャー・プログラムが出したエラー・メッセージ
- キャプチャー・プログラムのスループットの分析
- キャプチャー・プログラムの現在の待ち時間のサマリー
- キャプチャー・プログラムの現在の操作状況

このような状況から、キャプチャー・プログラムがどのように実行されているかをスナップショットとして見ることができます。

キャプチャー・プログラムの状況を照会するには、次のようにします。

1. 「SQL レプリケーション」フォルダーを展開します。
2. 「操作」フォルダーを展開します。
3. 「キャプチャー・コントロール・サーバー」フォルダーを選択します。
4. 内容ペインの SAMPLE データベースを右クリックし、「状況のチェック」を選択します。
5. 黄色の矢印アイコンをクリックし、現行情報を表示します。

キャプチャー・プログラムのスループット分析を表示するには、次のようにします。

1. 「SQL レプリケーション」フォルダーを展開します。
2. 「操作」フォルダーを展開します。
3. 「キャプチャー・コントロール・サーバー」フォルダーを選択します。
4. 内容ペインの SAMPLE データベースを右クリックし、「キャプチャー・スループット分析の表示」を選択します。
5. 「キャプチャー・スループット分析」ウィンドウで、次の情報を見ることができます。
 - DB2 ログから CD 表に挿入された行数、または CHGONLY キーワードの設定などの理由からスキップされた行数
 - CD 表から整理された行数
 - キャプチャー・プログラムによりコミットされたトランザクションの数
 - 特定の時間間隔内にキャプチャー・プログラムが使用したメモリー
6. 「検索」をクリックし、現行情報を表示します。

注: データの検索を初めて試行した後に、ボタンのラベルが「リフレッシュ」に変わります。

キャプチャー・プログラムの現在の待ち時間のサマリーを表示するには、次のようにします。

1. 「SQL レプリケーション」フォルダーを展開します。
2. 「操作」フォルダーを展開します。
3. 「キャプチャー・コントロール・サーバー」フォルダーを選択します。
4. 内容ペインの SAMPLE データベースを右クリックし、「キャプチャー待ち時間の表示」を選択します。
5. 「キャプチャー待ち時間」ウィンドウに、特定の時間間隔内のキャプチャー・プログラムの待ち時間の平均、最小、および最大が表示されます。
6. 「検索」をクリックし、現行情報を表示します。

注: データの検索を初めて試行した後に、ボタンのラベルが「リフレッシュ」に変わります。

ステップ 3: アプライ・プログラムの状況を表示する

レプリケーション・センターを使用して、アプライ・プログラムについて次のような状況情報を表示します。

- サブスクリプション・セット情報のサマリー (正常終了したサブスクリプション・セットと、失敗したサブスクリプション・セットを含む)
- アプライ・プログラムのパフォーマンスのサマリー
- レプリケーションのエンドツーエンド待ち時間のサマリー
- アプライ・プログラムの現在の操作状況

このような状況から、アプライ・プログラムがどのように実行されているかをスナップショットとして見ることができます。

アプライ・プログラムの状況を照会するには、次のようにします。

1. 「SQL レプリケーション」フォルダーを展開します。
2. 「操作」フォルダーを展開します。
3. 「アプライ・コントロール・サーバー」フォルダーを選択します。
4. 「COPYDB データベース」を展開します。
5. 「アプライ修飾子」フォルダーを選択します。
6. 内容ペインの DEPTQUAL アプライ修飾子を右クリックし、「状況のチェック」を選択します。
7. 「アプライが実行されている場所」フィールドが空の場合は、アプライ・プログラムを実行している場所のシステム・アドレスか IP アドレスを選択します。
8. 黄色の矢印アイコンをクリックし、現行情報を表示します。

アプライ・プログラムのパフォーマンスのサマリーを表示するには、次のようにします。

1. 「SQL レプリケーション」フォルダーを展開します。
2. 「操作」フォルダーを展開します。
3. 「アプライ・コントロール・サーバー」フォルダーを展開します。
4. 「COPYDB データベース」を展開します。

5. 「**アプライ修飾子**」フォルダーを選択します。
6. 内容ペインの DEPTQUAL アプライ修飾子を右クリックし、「**アプライ・スルー
プット分析**」を選択します。
7. 「アプライ・スループット分析」ウィンドウに、次の情報が表示されます。
 - アプライ・プログラムが CD 表からフェッチした行数
 - それぞれのサブスクリプション・セットごとの経過時間
8. 「**検索**」をクリックし、現行情報を表示します。

注: データの検索を初めて試行した後に、ボタンのラベルが「リフレッシュ」に変わります。

エンドツーエンドのレプリケーション待ち時間のサマリーを表示するには、次のようにします。

1. 「**SQL レプリケーション**」フォルダーを展開します。
2. 「**操作**」フォルダーを展開します。
3. 「**アプライ・コントロール・サーバー**」フォルダーを展開します。
4. 「**COPYDB データベース**」を展開します。
5. 「**アプライ修飾子**」フォルダーを選択します。
6. 内容ペインの DEPTQUAL アプライ修飾子を右クリックし、「**エンドツーエンド
待ち時間の表示**」を選択します。
7. 「エンドツーエンド待ち時間の表示」ウィンドウに、特定の時間間隔内のサブスクリプション・セットごとの待ち時間の平均が表示されます。
8. 「**検索**」をクリックし、現行情報を表示します。

注: データの検索を初めて試行した後に、ボタンのラベルが「リフレッシュ」に変わります。

ステップ 4: キャプチャー・プログラムとアプライ・プログラムを停止する

レプリケーション環境の保守における 1 つの重要な部分は、定期的なデータベース保守です。そのような保守のために、キャプチャー・プログラムおよびアプライ・プログラムを停止しなければならない場合があります。たとえば、キャプチャー・プログラムおよびアプライ・プログラムで使用される表スペースを直接使用するユーティリティを実行するには、まずそれらのプログラムを停止しなければなりません。

キャプチャー・プログラムを停止するには、次のようにします。

1. 「**SQL レプリケーション**」フォルダーを展開します。
2. 「**操作**」フォルダーを展開します。
3. 「**キャプチャー・コントロール・サーバー**」フォルダーを選択します。
4. 内容ペインの SAMPLE データベースを右クリックし、「**キャプチャーの停止**」を選択します。
5. 「キャプチャーの停止」ウィンドウで「**OK**」をクリックします。

6. 「今実行するか、コマンドを保管する」ウィンドウで「**OK**」をクリックすれば、コマンドは即時に実行されます。
7. 「DB2 メッセージ」ウィンドウに、コマンドが正常に実行されたというメッセージが表示されます。「**クローズ**」をクリックします。これでキャプチャー・プログラムは停止しました。

アプライ・プログラムを停止するには、次のようにします。

1. 「**SQL レプリケーション**」フォルダーを展開します。
2. 「**操作**」フォルダーを展開します。
3. 「**アプライ・コントロール・サーバー**」フォルダーを展開します。
4. 「**COPYDB データベース**」を展開します。
5. 「**アプライ修飾子**」フォルダーを選択します。
6. 内容ペインの DEPTQUAL アプライ修飾子を右クリックし、「**アプライの停止**」を選択します。
7. 「アプライの停止」ウィンドウで、「**OK**」をクリックします。これでアプライ・プログラムは停止しました。

キャプチャー・プログラムとアプライ・プログラムが停止したら、データベースに対して DB2 ユーティリティを実行することができます。ユーティリティの実行はこのシナリオの有効範囲外です。

レプリケーションのモニター

レプリケーション環境の起動後、キャプチャー・プログラムとアプライ・プログラムが正常に実行されているかどうかを知りたい場合があります。また、ある種のレプリケーション・エラーについては、エラーの発生時に自動的に通知するようにセットアップしておくことも必要です。

レプリケーション・センターを使用して、キャプチャー・プログラムとアプライ・プログラムの状況を照会し、ある種の統計を表示すれば、この両方のプログラムが正常に実行されているかどうかを知ることができます。また、キャプチャー・プログラムまたはアプライ・プログラムがある種のレプリケーション・エラーを検出した場合に、レプリケーション・アラート・モニターがユーザーに通知するようにセットアップすることもできます。

ステップ 1: モニター・プログラムのレプリケーション・コントロール表を作成する

レプリケーション・アラート・モニター・プログラムは、レプリケーション・モニター・コントロール表から現行のモニター情報を読み取り、またモニター・プログラムの状況をこの表に保管します。モニター・サーバーとして機能するデータベースには、モニター・コントロール表を含める必要があります。

モニター・コントロール表を作成するには、次のようにします。

1. 「**モニターおよびアラート**」フォルダーを展開します。
2. 「**モニター・コントロール・サーバー**」フォルダーを右クリックし、「**モニター・コントロール表の作成**」を選択します。

3. 「サーバーの選択」ウィンドウで、COPYDB データベースを選択します。このデータベースはモニター・コントロール・サーバーになります。
4. 「モニター・コントロール表の作成」ウィンドウで、IBMSNAP_CONTACTS コントロール表を選択し、REPLMONTS1 表スペース・プロパティの情報を記入します。コンテナ名の隣りにあるブラウザ・ボタンをクリックし、この表スペースのロケーションをカスタマイズします。この表スペースについてのその他の情報 (バッファ・プールを IBMDEFAULTBP に設定するなど) を指定することもできます。デフォルトでは、IBMSNAP_ALERTS と IBMSNAP_MONTRACE 以外のすべてのモニター管理テーブルは IBMSNAP_CONTACTS 表と同じ表スペースを共有します。「モニター・コントロール表の作成」ウィンドウは、図 13 と類似したウィンドウになります。

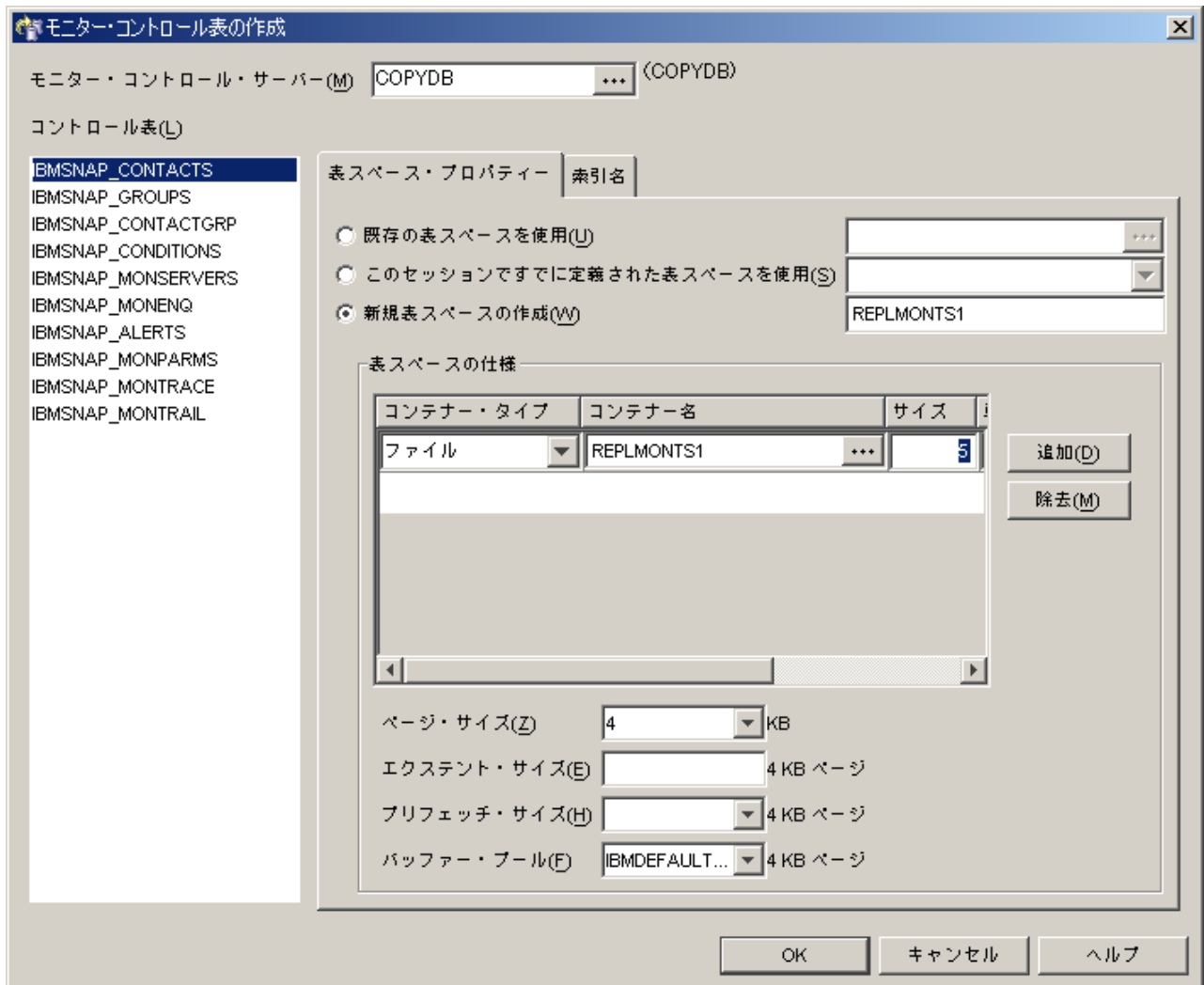


図 13. 「モニター・コントロール表の作成」ウィンドウ

5. 「モニター・コントロール表の作成」ウィンドウで、IBMSNAP_ALERTS コントロール表を選択し、REPLMONTS2 表スペース・プロパティの情報を記入します。

6. 「モニター・コントロール表の作成」ウィンドウで、IBMSNAP_MONTRACE コントロール表を選択し、RELPMONTS3 表スペース・プロパティの情報を記入します。
7. 「モニター・コントロール表の作成」ウィンドウで「OK」をクリックし、その他のコントロール表情報 (索引名など) についてデフォルト値を受け入れます。
8. 「メッセージ・ダイアログ」ウィンドウで「クローズ」をクリックします。このウィンドウは、モニター・コントロール表を作成する SQL スクリプトの生成結果を表示します。エラーがある場合は、このウィンドウに表示されます。
9. 「今実行するか、SQL を保管する」ウィンドウで「OK」をクリックすれば、SQL スクリプトは即時に実行されます。
10. 「DB2 メッセージ」ウィンドウに、スクリプトが正常に実行されたというメッセージが表示されます。「クローズ」をクリックします。
11. 「モニター・コントロール・サーバー」フォルダーを展開します。フォルダーの下に COPYDB データベースが表示されます。

ステップ 2: レプリケーション・アラートの連絡先を作成する

レプリケーション・アラート・モニター・プログラムは、キャプチャー・プログラムおよびアプライ・プログラムの特定のアクティビティを検出した時に、アラートを出すことができます。特定のアラート条件について、個別の連絡先を作成するか、またはレプリケーション・アラート・モニターが複数の人にアラートを出す必要がある場合には、連絡先のグループを作成することができます。

連絡先を作成するには、次のようにします。

1. 「モニターおよびアラート」フォルダーを展開します。
2. 「モニター・コントロール・サーバー」フォルダーを展開します。
3. 「COPYDB データベース」を展開します。
4. 「連絡先」フォルダーを右クリックし、「連絡先の作成」→「担当者」を選択します。
5. 「連絡先の作成」ウィンドウで、ここでは担当者として自分の名前と E メールアドレスを指定してください。「OK」をクリックして、ノートブックをクローズします。
6. 「メッセージと SQL スクリプト」ウィンドウで「クローズ」をクリックします。このウィンドウは、モニター・コントロール表を更新する SQL スクリプトの生成結果を表示します。エラーがある場合は、このウィンドウに表示されません。
7. 「今実行するか、SQL を保管する」ウィンドウで「OK」をクリックすれば、SQL スクリプトは即時に実行されます。
8. 「DB2 メッセージ」ウィンドウに、スクリプトが正常に実行されたというメッセージが表示されます。「クローズ」をクリックします。
9. 「連絡先」フォルダーをクリックします。定義した連絡先が「連絡先」の内容ページに表示されます。

ステップ 3: キャプチャー・プログラムのアラート条件を選択する

レプリケーション・アラート・モニター・プログラムは、キャプチャー・プログラムの特定のアクティビティをモニターすることができます。どのアクティビティ

ーをモニターするかを選択する必要があります。これらのアクティビティーごとに、アラート条件を選択します。キャプチャー・プログラムがこの条件を検出すると、レプリケーション・アラート・モニターは、アラート条件に定義された連絡先にアラートを送信します。

キャプチャー・プログラムのモニター定義を作成するには、次のようにします。

1. 「モニターおよびアラート」フォルダーを展開します。
2. 「モニター・コントロール・サーバー」フォルダーを展開します。
3. 「COPYDB データベース」を展開します。
4. 「モニター」フォルダーを右クリックし、「作成」を選択します。
5. 「モニターの作成ウィザード」に、次の情報を入力します。
 - a. 「モニター修飾子」フィールドに、MON1 と入力します。
 - b. 「SQL レプリケーション」の下で、「キャプチャー・プログラム」チェック・ボックスを選択します。
 - c. 「次へ (Next)」をクリックします。
 - d. 「キャプチャー・プログラムのアラート条件の選択」ページで、「追加」をクリックします。
 - e. 「キャプチャー・コントロール・サーバー」フィールドのブラウズ・ボタンをクリックし、モニターしたいキャプチャー・コントロール・サーバーを選択します。「キャプチャー・コントロール・サーバーの選択」ウィンドウでSAMPLE データベースを選択し、「OK」をクリックします。
 - f. 表で、「CAPTURE_ERRORS」チェック・ボックスを選択します。
 - g. 「CAPTURE_ERRORS」と同じ行にある「連絡先」フィールドにカーソルを置き、ブラウズ・ボタンをクリックします。
 - h. 「連絡先または連絡先グループの選択」ウィンドウで、318 ページの『ステップ 2: レプリケーション・アラートの連絡先を作成する』で作成した連絡先を選択し、「OK」をクリックしてウィンドウをクローズします。
 - i. 「OK」をクリックし、「キャプチャー・スキーマ用のアラート条件を選択」ウィンドウをクローズします。
 - j. 「完了 (Finish)」をクリックします。
6. 「メッセージと SQL スクリプト」ウィンドウで「クローズ」をクリックします。このウィンドウは、モニター・コントロール表を更新する SQL スクリプトの生成結果を表示します。エラーがある場合は、このウィンドウに表示されません。
7. 「今実行するか、SQL を保管する」ウィンドウで「OK」をクリックすれば、SQL スクリプトは即時に実行されます。
8. 「DB2 メッセージ」ウィンドウに、スクリプトが正常に実行されたというメッセージが表示されます。「クローズ」をクリックします。
9. 「COPYDB」データベースを展開して、「モニター」フォルダーを展開し、「MON1」フォルダーを選択します。定義したアラート条件が、モニター修飾子の内容ペインに表示されます。

ステップ 4: アプライ・プログラムのアラート条件を選択する

レプリケーション・アラート・モニター・プログラムは、アプライ・プログラムの特定のアクティビティをモニターすることができます。どのアクティビティをモニターするかを選択する必要があります。これらのアクティビティごとに、アラート条件を選択します。アプライ・プログラムがこの条件を検出すると、レプリケーション・アラート・モニターは、アラート条件に定義された連絡先にアラートを送信します。

アプライ・プログラムのモニター定義を作成するには、次のようにします。

1. 「モニターおよびアラート」フォルダーを展開します。
2. 「モニター・コントロール・サーバー」フォルダーを展開します。
3. 「COPYDB データベース」を展開します。
4. 「モニター」フォルダーを右クリックし、「作成」を選択します。
5. 「モニターの作成ウィザード」に、次の情報を入力します。
 - a. 「モニター修飾子」フィールドに MON1 と入力します (キャプチャー・プログラムのアラート条件を作成しなかった場合)。
 - b. 「SQL レプリケーション」の下で、「アプライ修飾子によるアプライ・プログラム」チェック・ボックスを選択します。
 - c. 「次へ (Next)」をクリックします。
 - d. 「アプライ修飾子 (処理するすべてのサブスクリプション・セットを含む) のアラート条件の選択 (Select alert conditions for Apply qualifiers, including all subscription sets that they process)」ページで、「追加」をクリックします。
 - e. 「アプライ・コントロール・サーバー」フィールドのブラウズ・ボタンをクリックし、モニターしたいアプライ・コントロール・サーバーを選択します。「アプライ・コントロール・サーバーの選択」ウィンドウで COPYDB データベースを選択し、「OK」をクリックします。
 - f. 表で、「APPLY_FULLREFRESH」チェック・ボックスを選択します。
 - g. 「APPLY_FULLREFRESH」と同じ行にある「連絡先」フィールドにカーソルを置き、ブラウズ・ボタンをクリックします。
 - h. 「連絡先または連絡先グループの選択」ウィンドウで、318 ページの『ステップ 2: レプリケーション・アラートの連絡先を作成する』で作成した連絡先を選択し、「OK」をクリックしてウィンドウをクローズします。
 - i. 「アプライ修飾子またはサブスクリプション・セットのアラート条件を選択」ウィンドウで、「OK」をクリックします。
 - j. 「完了 (Finish)」をクリックします。
6. 「メッセージと SQL スクリプト」ウィンドウで「クローズ」をクリックします。このウィンドウは、モニター・コントロール表を更新する SQL スクリプトの生成結果を表示します。エラーがある場合は、このウィンドウに表示されません。
7. 「今実行するか、SQL を保管する」ウィンドウで「OK」をクリックすれば、SQL スクリプトは即時に実行されます。
8. 「DB2 メッセージ」ウィンドウに、スクリプトが正常に実行されたというメッセージが表示されます。「クローズ」をクリックします。

- 「COPYDB」データベースを展開して、「モニター」フォルダーを展開し、「MON1」フォルダーを選択します。定義したアラート条件が、モニター修飾子の内容ペインに表示されます。図 14 に、「モニター修飾子」フォルダーに MON1 モニター修飾子が表示された内容ペインの例を示します。

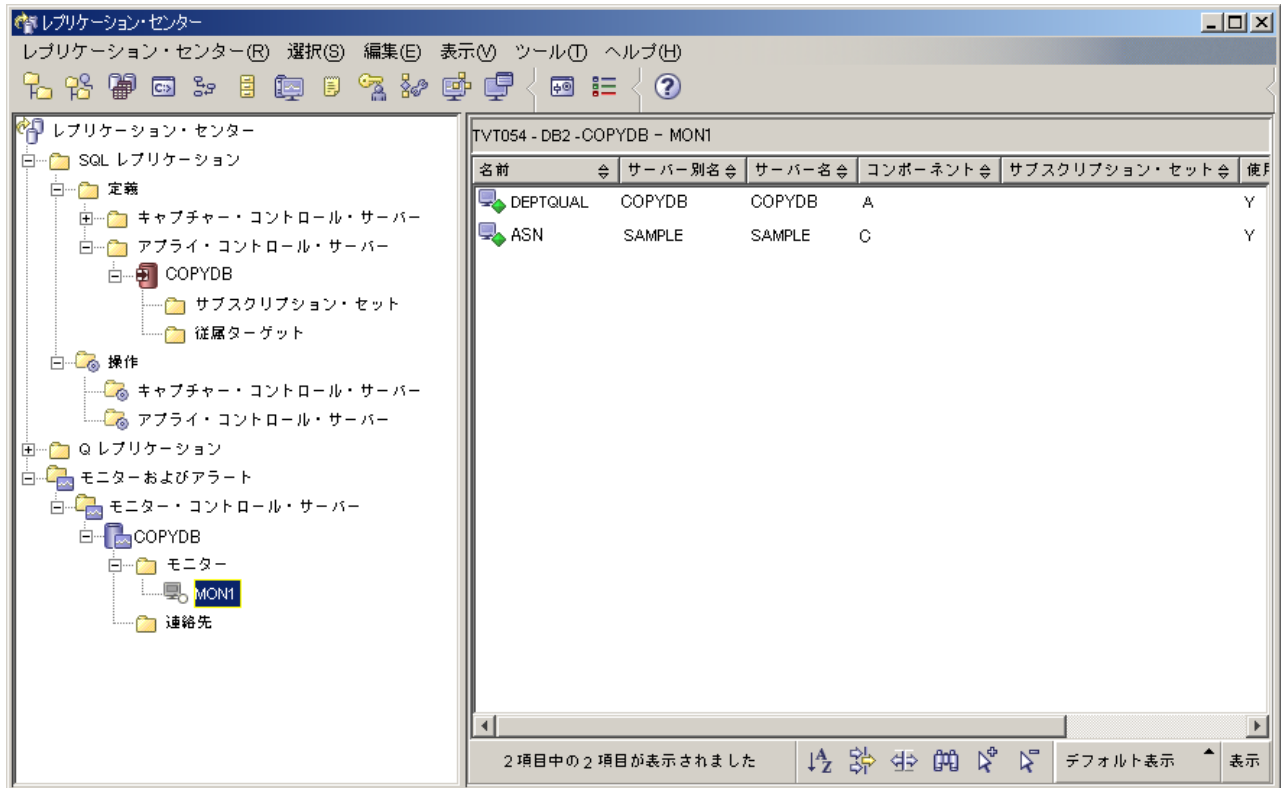


図 14. モニター修飾子 MON1 は COPYDB データベースにリストされます。

ステップ 5: モニター修飾子のレプリケーション・アラート・モニターを開始する

キャプチャーまたはアプライ・プログラムのアラート条件を選択した後、レプリケーション・アラート・モニター・プログラムを始動して、キャプチャーおよびアプライ・プログラムのアクティビティを、モニター修飾子に関連付けられた特定の条件についてモニターすることができます。キャプチャーまたはアプライ・プログラムが指定された条件の 1 つを検出すると、レプリケーション・アラート・モニターは、アラート条件に定義された連絡先にアラートを送信します。

レプリケーション・アラート・モニターを開始するには、次のようにします。

- 「モニターおよびアラート」フォルダーを展開します。
- 「モニター・コントロール・サーバー」フォルダーを展開します。
- 「COPYDB データベース」を展開します。
- 「モニター」フォルダーを展開します。
- モニター修飾子 MON1 を右クリックし、「モニターの開始」を選択します。
- 「モニターの開始」ウィンドウに、次の情報を入力します。

- a. MONITOR_PATH キーワードを選択します。レプリケーション・アラート・モニターが自分のログと作業ファイルを保管するディレクトリーの値を入力します。

ヒント: MONITOR_PATH キーワードの値は、APPLY_PATH キーワードの値と同じにし、レプリケーション・アラート・モニターとアプライ・プログラムが同じパスワード・ファイルを使用できるようにします。

- b. EMAIL_SERVER キーワードを選択します。自分の E メール・サーバー名を入力してください。
 - c. MONITOR_ERRORS キーワードを選択します。指定した条件を検出したときにレプリケーション・アラート・モニターが自分に通知するようにしたい場合は、自分の E メール・アドレスを入力します。レプリケーション・アラート・モニターが他の人に通知するようにしたい場合は、ブラウズ・ボタンをクリックして「連絡先または連絡先グループを選択」ウィンドウを開き、連絡先または連絡先グループを選択します。
 - d. 「OK」をクリックして、「モニターの開始」ウィンドウをクローズします。
7. 「今実行するか、コマンドを保管する」ウィンドウで「OK」をクリックすれば、コマンドは即時に実行されます。
 8. 「DB2 メッセージ」ウィンドウに、コマンドが正常に開始されたというメッセージが表示されます。「クローズ」をクリックします。

レプリケーション・アラート・モニターがモニターしたアラートを表示するには、次のようにします。

1. 「モニターおよびアラート」フォルダーを展開します。
2. 「モニター・コントロール・サーバー」フォルダーを展開します。
3. 「COPYDB データベース」を展開します。
4. 「モニター」フォルダーを展開します。
5. モニター修飾子 MON1 を選択します。
6. モニター修飾子の内容ペインでアラート条件の 1 つを右クリックし、「アラート条件」を選択します。
7. 「アラートの表示」ウィンドウで、時刻範囲を指定し、「検索」をクリックします。

注: データの検索を初めて試行した後に、ボタンのラベルが「リフレッシュ」に変わります。

レプリケーション・アラート・モニターを停止するには、次のようにします。

1. 「モニターおよびアラート」フォルダーを展開します。
2. 「モニター・コントロール・サーバー」フォルダーを展開します。
3. 「COPYDB データベース」を展開します。
4. 「モニター」フォルダーを展開します。
5. モニター修飾子 MON1 を右クリックし、「モニターの停止」を選択します。
6. 「モニターの停止」ウィンドウで、「OK」をクリックします。

7. 「今実行するか、コマンドを保管する」ウィンドウで、レプリケーション・アラート・モニター・プログラムを開始したディレクトリーを「ディレクトリー」フィールドに入力するか、またはブラウズ・ボタンでパスを選択します。「OK」をクリックします。

ヒント: 「ディレクトリー」フィールドは、ユーザー ID とパスワードを指定するまで使用可能にはなりません。

8. 「今実行するか、コマンドを保管する」ウィンドウで「OK」をクリックすれば、コマンドは即時に実行されます。
9. 「DB2 メッセージ」ウィンドウに、コマンドが正常に実行されたというメッセージが表示されます。「クローズ」をクリックします。

第 3 部 レプリケーションのリファレンス

本書の第 3 部は、以下の章で構成されています。

327 ページの『第 17 章 SQL レプリケーション・オブジェクトの命名規則』では、レプリケーション・オブジェクトに有効な名前の指定方法について説明します。

329 ページの『第 18 章 SQL レプリケーション用のシステム・コマンド (Linux、UNIX、Windows、z/OS)』では、DB2 レプリケーションの経験のあるユーザーが、Linux、UNIX、Windows、および z/OS オペレーティング・システムでレプリケーション操作をするために、レプリケーション・センターの代わりに使用することができるコマンドについて説明します。

385 ページの『第 19 章 SQL レプリケーション用のシステム・コマンド (OS/400)』では、OS/400 オペレーティング・システムで、ローカルに、レプリケーションをセットアップ、管理、および保守するためのコマンドについて説明します。

483 ページの『第 20 章 SQL レプリケーション・プログラムの操作 (z/OS)』では、z/OS で JCL やシステム起動タスクを使用して、レプリケーション・プログラムを開始および操作する方法について説明します。

489 ページの『第 21 章 Windows Service Control Manager を使用して SQL レプリケーションのシステム・コマンドを発行する方法 (Windows)』では、Windows オペレーティング・システムで、レプリケーション・プログラムをサービスとして開始する方法について説明します。

493 ページの『第 22 章 各種オペレーティング・システムでの SQL レプリケーション・プログラムのスケジューリング』では、さまざまなオペレーティング・システムで、レプリケーション・プログラムをスケジューリングする方法について説明します。

495 ページの『第 23 章 SQL レプリケーション・コンポーネントの通信方法』では、レプリケーション・コンポーネントが相互通信のためにコントロール表を使用する方法について説明します。

501 ページの『第 24 章 SQL レプリケーションの表構造』では、さまざまなレプリケーション・サーバー上にあるレプリケーション表の構造について説明します。

第 17 章 SQL レプリケーション・オブジェクトの命名規則

以下の表は、レプリケーション・オブジェクトの名前の制限を示しています。

表 25. レプリケーション・オブジェクトの名前の制限

オブジェクト	名前の制限
ソース表とターゲット表	UNIX、Windows、z/OS: データベース管理システムの命名規則に従ってください。 OS/400: 名前に、ブランク、アスタリスク (*)、疑問符 (?)、単一引用符 (')、二重引用符 (")、またはスラッシュ (/) を含めることはできません。
ソース列とターゲット列	データベース管理システムの命名規則に従ってください。(すべての変更前イメージ列には、1 文字の接頭部が追加されます。変更前イメージ列の名前が未確定にならないようにするため、ソース列名が 29 文字までユニークなものになるようにし、列名に変更前イメージ文字接頭部を追加するときに、変更前イメージ列名が既存の列名と競合しないようにしてください。)
サブスクリプション・セット	サブスクリプション・セット名には、DB2 で可変文字 (VARCHAR) 列に許可されているすべての文字を使用できます。 推奨: DB2 表および列名の命名規則に従ってください。DB2 レプリケーションは、サブスクリプション・セット名を各レプリケーション・コントロール・サーバーに保管するため、名前を 3 つのすべてのサーバーのコード・ページと互換性のあるものにしてください。
キャプチャー・スキーマ	UNIX、Windows: キャプチャー・スキーマには 30 文字以下のストリングを使用できます ¹ 。 OS/390、z/OS: キャプチャー・スキーマには 18 文字以下のストリングを使用できます。DB2 UDB for z/OS バージョン 8 の新規関数モードのサブシステムでは、128 文字以下のストリングを使用できます ¹ 。 OS/400: キャプチャー・スキーマ (CAPCTLLIB) には 10 文字以下の英数字を使用できます ¹ 。
アプライ修飾子	UNIX、Windows、z/OS の場合: アプライ修飾子は、18 文字以下のストリングにできます ¹ 。 OS/400 の場合: アプライ修飾子には 18 文字以下のストリングを使用できますが、アプライ・ジョブの長さは 10 文字までにしかできないため、特定のアプライ修飾子について、最初の 10 文字はユニークでなければなりません ¹ 。
モニター修飾子	UNIX、Windows、z/OS の場合: モニター修飾子には 18 文字以下のストリングを使用できます ¹ 。

表 25. レプリケーション・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
<p>注:</p> <p>1. キャプチャー・スキーマ、アプライ修飾子、およびモニター修飾子では、これらのオブジェクトの名前に、以下の有効文字のみを使用してください。</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>空白は使用できません。また、コロン ":" および正符号 "+" などの特殊文字も使用できません。</p>	

レプリケーション・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。

Windows オペレーティング・システムの場合は、同じ名前を区別するためにユニークなパスを使用する必要があります。たとえば、3 つのアプライ修飾子、`myqual`、`MyQual`、および `MYQUAL` 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 つの修飾子が同じアプライ・パスにあると、名前の競合が発生します。

重要: 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・モニター用に Windows サービスをセットアップするときには、キャプチャー・スキーマ、アプライ修飾子、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。

第 18 章 SQL レプリケーション用のシステム・コマンド (Linux、UNIX、Windows、z/OS)

この章では、以下の 1 つ以上のオペレーティング・システムの下で実行する、レプリケーション・コマンドについて説明します。

- Linux
- UNIX
- Windows
- z/OS

これらのコマンドはすべて `asn` という接頭部を持ち、オペレーティング・システムのコマンド・プロンプトか、シェル・スクリプトに入力されます。コマンドの 1 つ、**asnanalyze** は、OS/400 オペレーティング・システムにあるリモート・データにも使用できます。

この章には、コマンド別のセクションがあります。各セクションには、コマンドの簡単な記述、構文図、およびパラメーターとそれに対応する定義の表があります。各セクションの最後には、コマンドの使用例と、関連情報への相互参照があります。

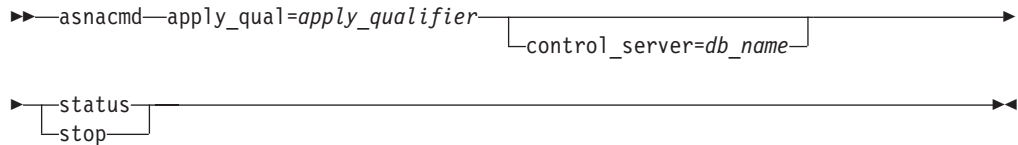
コマンドには、以下のものがあります。

- 330 ページの『asnacmd: アプライの操作』
- 331 ページの『asnanalyze: アナライザーの操作』
- 334 ページの『asnapply: アプライの始動』
- 342 ページの『asnacp: キャプチャーの始動』
- 349 ページの『asnccmd: キャプチャーの操作』
- 354 ページの『asnmcmd: 実行中のレプリケーション・アラート・モニターの処理』
- 357 ページの『asnmon: レプリケーション・アラート・モニターの始動』
- 362 ページの『asnpswd: パスワード・ファイルの作成および保守』
- 366 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』
- 370 ページの『asnsdrop: DB2 レプリケーション・サービスのドロップ』
- 371 ページの『asnslst: DB2 レプリケーション・サービスのリスト』
- 372 ページの『asntdiff: ソース表とターゲット表とのデータの比較』
- 374 ページの『asntrc: レプリケーション・トレース機能の操作』
- 382 ページの『asntrep: ソース表とターゲット表の間の違いの修復』

asnacmd: アプライの操作

Linux、UNIX、Windows、および z/OS 上の UNIX System Services (USS) でアプライ・プログラムを操作するには、**asnacmd** 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

表 26 は、呼び出しパラメーターを定義します。

表 26. Linux、UNIX、Windows、および z/OS オペレーティング・システム用 *asnacmd* 呼び出しパラメーター定義

パラメーター	定義
apply_qual=apply_qualifier	<p>アプライ・プログラムが、処理されるサブスクリプション・セットの識別に使用するアプライ修飾子を指定します。</p> <p>アプライ修飾子を指定してください。入力する値は、サブスクリプション・セット (IBMSNAP_SUBS_SET) 表の APPLY_QUAL 列の値と一致する必要があります。アプライ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。</p>
control_server=db_name	<p>サブスクリプション定義とアプライ・コントロール表が存在するアプライ・コントロール・サーバーの名前を指定します。</p> <p>Linux、UNIX および Windows の場合: アプライ・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。</p> <p>z/OS の場合: コントロール・サーバー・パラメーターは、コントロール・サーバーに接続するデータベース・サーバーの名前です。</p>
status	アプライ内の各スレッド (管理およびワーカー) 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	アプライ・プログラムを通常の方法で停止することを指定します。

asnacmd の例

次の例は、**asnacmd** コマンドの使用法を示しています。

例 1

それぞれのアプライ・スレッドの状態についてメッセージを受け取る場合のコマンド例です。

```
asnacmd apply_qual=AQ1 control_server=dbx status
```

例 2

アプライ・プログラムを停止するには、次のようにします。

```
asnacmd apply_qual=AQ1 control_server=dbx stop
```

関連タスク:

- 483 ページの『第 20 章 SQL レプリケーション・プログラムの操作 (z/OS)』

関連資料:

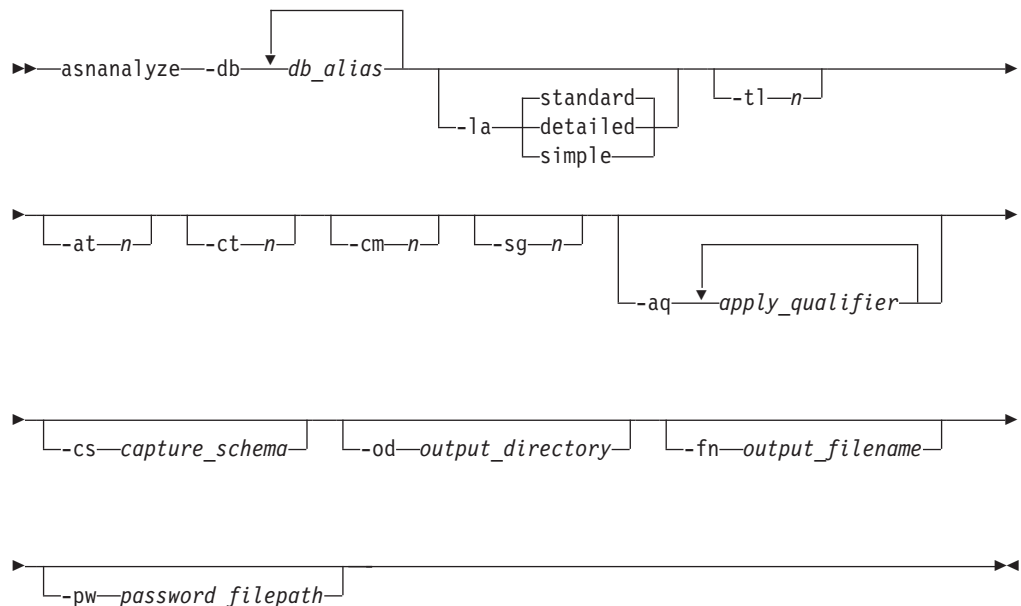
- 431 ページの『ENDDPRAPY: アプライ・プログラムの停止 (OS/400)』
- 460 ページの『STRDPRAPY: アプライ・プログラムの始動 (OS/400)』

asnanalyze: アナライザーの操作

レプリケーション・コントロール表の状態についてのレポートを生成するには、**asnanalyze** コマンドを使用します。このコマンドは、任意のオペレーティング・システムにあるレプリケーション・コントロール表を分析します。OS/400 オペレーティング・システムでも可能ですが、コマンドは Linux、UNIX または Windows から呼び出す必要があります。

コマンドを呼び出す場合、**asnanalyze** コマンドと最初のパラメーターの間にはスペースを 1 つ入れる必要があります。パラメーターなしでコマンドを発行すると、画面にコマンド・ヘルプが表示されます。

構文



パラメーター

表 27 は、呼び出しパラメーターを定義します。

表 27. Linux、UNIX および Windows オペレーティング・システム用 asnanalyze 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>	<p>キャプチャー・コントロール・サーバー、ターゲット・サーバー、およびアプライ・コントロール・サーバーを指定します。</p> <p>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。</p>
-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および、キャプチャー・プログラムおよびアプライ・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"> 変更データ (CD) 表および作業単位 (UOW) 表の整理情報。 DB2 for z/OS 表スペースのパーティション化および圧縮情報 サブスクリプション・キーのためのターゲット索引の分析 <p>simple 標準レポートで情報を生成しますが、サブスクリプション列 (IBMSNAP_SUBS_COLS) 表からの詳細は含まれません。</p>
-tl <i>n</i>	アプライ・トレール (IBMSNAP_APPLYTRAIL) 表から検索する項目の日付範囲 (0 から 30 日) を指定します。デフォルトは 3 日です。
-at <i>n</i>	アプライ・トレース (IBMSNAP_APPLYTRACE) 表から検索する項目の日付範囲 (0 から 30 日) を指定します。デフォルトは 3 日です。
-ct <i>n</i>	キャプチャー・トレース (IBMSNAP_CAPTRACE) 表から検索する項目の日付範囲 (0 から 30 日) を指定します。デフォルトは 3 日です。
-cm <i>n</i>	キャプチャー・モニター (IBMSNAP_CAPMON) 表から検索する項目の日付範囲 (0 から 30 日) を指定します。デフォルトは 3 日です。
-sg <i>n</i>	シグナル (IBMSNAP_SIGNAL) 表から検索する項目の日付範囲 (0 から 30 日) を指定します。デフォルトは 3 日です。

表 27. Linux、UNIX および Windows オペレーティング・システム用 *asnanalyze* 呼び出しパラメーター定義 (続き)

パラメーター	定義
-aq <i>apply_qualifier</i>	分析する特定のサブスクリプション・セットを識別するアプライ修飾子を指定します。 複数のアプライ修飾子を指定することができます。複数のアプライ修飾子がある場合は、ブランク・スペースを使用して値を区切ります。アプライ修飾子を指定しない場合、指定されたデータベース別名のすべてのサブスクリプション・セットが分析されます。
-cs <i>capture_schema</i>	分析するキャプチャー・スキーマの名前を指定します。 このパラメーターを使用する場合、指定できるキャプチャー・スキーマは 1 つだけです。
-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>asnanalyze.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。

asnanalyze の例

以下の例は、**asnanalyze** コマンドの使用法を示しています。

例 1

proddb1 という名前のデータベース上のレプリケーション・コントロール表を分析する例です。

```
asnanalyze -db proddb1
```

例 2

proddb1 と *proddb2* のデータベース上のレプリケーション・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。

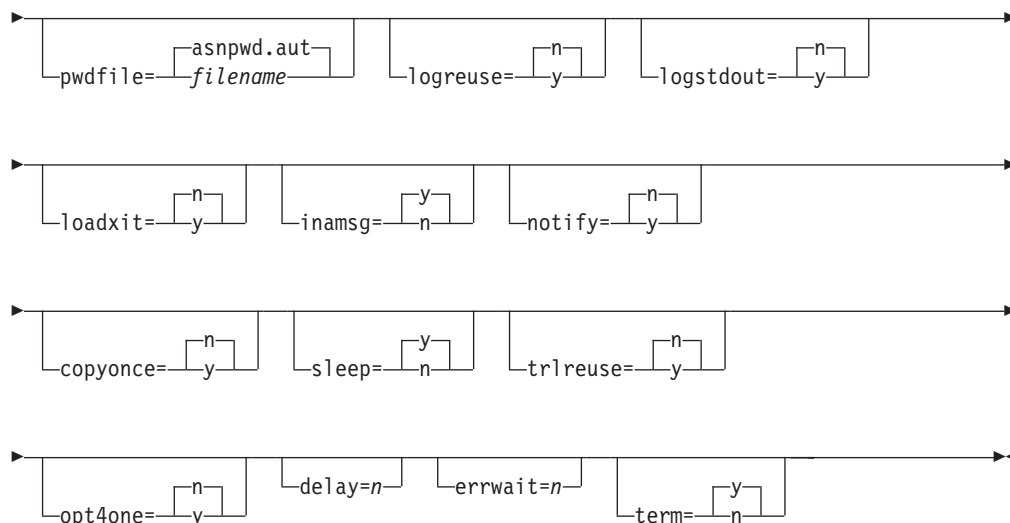
```
asnanalyze -db proddb1 proddb2 -la detailed
```

例 3

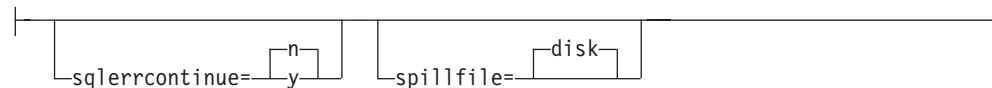
proddb1 と *proddb2* のデータベース上の、*IBMSNAP_APPLYTRAIL*、*IBMSNAP_APPLYTRACE*、*IBMSNAP_CAPTRACE*、*IBMSNAP_CAPMON*、および *IBMSNAP_SIGNAL* 表から、最新の 2 日の情報を分析する例です。

```
asnanalyze -db proddb1 proddb2 -tl 2 -at 2 -ct 2 -cm 2 -sg 2
```

例 4



UNIX、Windows および z/OS パラメーター:



z/OS パラメーター:



注:

- 1 `db2_subsystem` パラメーターは z/OS オペレーティング・システムでのみ使用します。

パラメーター

表 28 は、呼び出しパラメーターを定義します。

表 28. Linux、UNIX、Windows、および z/OS オペレーティング・システム用 `asnapply` 呼び出しパラメーター定義

パラメーター	定義
<code>apply_qual=apply_qualifier</code>	<p>アプライ・プログラムが、処理されるサブスクリプション・セットの識別に使用するアプライ修飾子を指定します。このパラメーターは必須です。</p> <p>入力する値は、サブスクリプション・セット (IBMSNAP_SUBS_SET) 表の <code>APPLY_QUAL</code> 列の値と一致する必要があります。アプライ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。</p>
<code>db2_subsystem=name</code>	<p>z/OS の場合のみ: 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。入力する DB2 サブシステム名は最大 4 文字です。このパラメーターにはデフォルトはありません。このパラメーターは必須です。</p>

表 28. Linux、UNIX、Windows、および z/OS オペレーティング・システム用 *asnapply* 呼び出しパラメーター定義 (続き)

パラメーター	定義
control_server = <i>db_name</i>	<p>サブスクリプション定義とアプライ・プログラム・コントロール表が存在する、アプライ・コントロール・サーバーの名前を指定します。</p> <p>Linux、UNIX および Windows の場合: アプライ・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。</p> <p>z/OS の場合: アプライ・コントロール・サーバーのロケーション名を指定します。</p>
apply_path = <i>pathname</i>	<p>アプライ・プログラムが使用する作業ファイルのロケーションを指定します。デフォルトは、asnapply コマンドが呼び出されたディレクトリーです。</p>
pwdfile = <i>filename</i>	<p>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 asnpwd.aut です。</p> <p>このコマンドは、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。</p> <p>apply_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>
logreuse = <i>y/n</i>	<p>アプライ・プログラムが、ログ・ファイル (<i>db2instance.control_server.apply_qualifier.APP.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) アプライ・プログラムは、アプライ・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> <p>y アプライ・プログラムは、ログ・ファイルを削除し、アプライ・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>z/OS の場合: ログ・ファイルには DB2 インスタンス名は含まれません (<i>control_server.apply_qualifier.APP.log</i>)。</p>
logstdout = <i>y/n</i>	<p>アプライ・プログラムがログ・ファイル・メッセージを送信する場所を指定します。</p> <p>n (デフォルト) アプライ・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、ログ・ファイルと標準出力 (STDOUT) の両方に送られます。</p> <p>y アプライ・プログラムは、ログ・ファイルと標準出力 (stdout) の両方にログ・ファイル・メッセージを送信します。</p>

表 28. Linux、UNIX、Windows、および z/OS オペレーティング・システム用 *asnapply* 呼び出しパラメーター定義 (続き)

パラメーター	定義
loadxit=y/n	<p>アプライ・プログラムが ASNLOAD を呼び出すかどうかを指定します。ASNLOAD は IBM 提供の出口ルーチンであり、エクスポートおよびロード・ユーティリティーを使用して、ターゲット表をリフレッシュします。</p> <p>n (デフォルト) アプライ・プログラムは ASNLOAD を呼び出しません。</p> <p>y アプライ・プログラムは ASNLOAD を呼び出します。</p>
inamsg=y/n	<p>アプライ・プログラムを非アクティブにしたとき、アプライ・プログラムからメッセージを出すかどうかを指定します。</p> <p>y (デフォルト) アプライ・プログラムは非アクティブ時にメッセージを出します。</p> <p>n アプライ・プログラムは非アクティブ時にメッセージを出しません。</p>
notify=y/n	<p>アプライ・プログラムが ASNDONE を呼び出すかどうかを指定します。ASNDONE は、アプライ・プログラムがサブスクリプション・セットのコピーを終了した時に、ユーザーにコントロールを戻すための出口ルーチンです。</p> <p>n (デフォルト) アプライ・プログラムは ASNDONE を呼び出しません。</p> <p>y アプライ・プログラムは ASNDONE を呼び出します。</p>

表 28. Linux、UNIX、Windows、および z/OS オペレーティング・システム用 *asnapply* 呼び出しパラメーター定義 (続き)

パラメーター	定義
copyonce=y/n	<p>アプライ・プログラムが呼びだされた時点で適格と見なされたサブスクリプション・セットごとに、アプライ・プログラムがコピー・サイクルを 1 回実行するかどうかを指定します。その後、アプライ・プログラムは終了します。適格と見なされるサブスクリプション・セットとは、以下の基準を満たすものです。</p> <ul style="list-style-type: none"> サブスクリプション・セット (IBMSNAP_SUBS_SET) 表の中で (ACTIVATE > 0)。ACTIVATE 列の値がゼロより大きい場合、そのサブスクリプション・セットは無期限にアクティブであるか、または 1 回のみサブスクリプション処理に使用されています。 (REFRESH_TYPE = R または B) または (REFRESH_TYPE = E であり、指定されたイベントが発生)。REFRESH_TYPE 列の値は IBMSNAP_SUBS_SET 表に保管されます。 <p>サブスクリプション・セット表の MAX_SYNCH_MINUTES 限度および、サブスクリプション・イベント (IBMSNAP_SUBS_EVENT) 表の END_OF_PERIOD タイム・スタンプが指定されている場合は、これに従います。</p> <p>n (デフォルト) アプライ・プログラムは、適格なサブスクリプション・セットごとにコピー・サイクルを 1 回実行しません。</p> <p>y アプライ・プログラムは、適格なサブスクリプション・セットごとにコピー・サイクルを 1 回実行します。</p>
sleep=y/n	<p>処理の対象として適格となる新しいサブスクリプションがない場合に、アプライ・プログラムがどうするかを指定します。</p> <p>y (デフォルト) アプライ・プログラムはスリープ状態に入ります。</p> <p>n アプライ・プログラムは停止します。</p>
trlreuse=y/n	<p>アプライ・プログラムの始動時に、アプライ・プログラムがアプライ・トレール (IBMSNAP_APPLYTRAIL) 表を空にするかどうかを指定します。</p> <p>n (デフォルト) アプライ・プログラムは IBMSNAP_APPLYTRAIL 表に項目を付加します。アプライ・プログラムは表を空にしません。</p> <p>y アプライ・プログラムはプログラム始動時に IBMSNAP_APPLYTRAIL 表を空にします。</p>

表 28. Linux、UNIX、Windows、および z/OS オペレーティング・システム用 *asnapply* 呼び出しパラメーター定義 (続き)

パラメーター	定義
opt4one=y/n	<p>アプライ・プログラムに定義されているサブスクリプション・セットが 1 つだけの場合、アプライ・プログラムのパフォーマンスを最適化するかどうかを指定します。</p> <p>n (デフォルト) サブスクリプション・セットが 1 つの場合、アプライ・プログラムのパフォーマンスを最適化しません。</p> <p>y サブスクリプション・セットが 1 つの場合、アプライ・プログラムのパフォーマンスを最適化します。</p> <p>最適化を y に設定すると、アプライ・プログラムはサブスクリプション・セット・メンバーの情報をキャッシュに入れて再利用します。このようにサブスクリプション・セット・メンバーの情報を再利用すると、CPU 使用率が減り、スループットが向上します。</p>
delay=n	<p>連続レプリケーションを使用する場合に、それぞれのアプライ・サイクルが終了した後、何秒待つかを示す遅延時間 (秒単位) を指定します。 <i>n</i> は、0、1、2、3、4、5、または 6 です。デフォルトは 6 で、これは連続レプリケーションで使用されます (つまりサブスクリプション・セットが sleep=0 分を使用する場合)。copyonce が指定されている場合、このパラメーターは無視されます。</p>
errwait=n	<p>アプライ・プログラムがエラー状態になった後、何秒待つてから再試行するかを示す秒数 (1 から 65535) を指定します。デフォルト値は 300 秒 (5 分) です。</p> <p>重要: アプライ・プログラムはほとんど切れ目なく稼働しており、アプライ・トレール (IBMSNAP_APPLYTRAIL) 表に多くの行を生成するので、ここにあまり小さい数を指定しないでください。</p>

表 28. Linux、UNIX、Windows、および z/OS オペレーティング・システム用 *asnapply* 呼び出しパラメーター定義 (続き)

パラメーター	定義
term=y/n	<p>DB2 の状況がアプライ・プログラムの動作にどのように影響するかを指定します。</p> <p>y (デフォルト) DB2 が終了すると、アプライ・プログラムは終了します。</p> <p>n DB2 がアクティブでない場合、アプライ・プログラムは DB2 の始動を待ちます。</p> <p>Linux、UNIX および Windows の場合: DB2 が静止し、アプライ・プログラムがアクティブの場合、アプライ・プログラムはアクティブのままであり、DB2 が静止モードでなくなるまで再接続しません。</p> <p>z/OS の場合: DB2 が静止し、アプライ・プログラムがアクティブの場合、アプライ・プログラムはアクティブのままであり、DB2 が再度開始されるまで再接続しません。</p>
sqlerrcontinue=y/n	<p>アプライ・プログラムがある種の SQL エラーを検出した場合、アプライ・プログラムが処理を継続するかどうかを指定します。</p> <p>アプライ・プログラムは、失敗した SQLSTATE を SQLSTATE ファイルに指定された値に照らしてチェックします。この SQLSTATE ファイルは、アプライ・プログラムの実行前にユーザーが作成します。ファイルの内容と一致すれば、アプライ・プログラムは失敗した行についての情報をエラー・ファイル (<i>apply_qualifier.ERR</i>) に書き込み、処理を継続します。SQLSTATE ファイルには 5 バイト値を 20 個まで含めることができます。</p> <p>n (デフォルト) アプライ・プログラムは SQLSTATE ファイルをチェックしません。</p> <p>y アプライ・プログラムは処理中に SQLSTATE ファイルをチェックします。</p>

表 28. Linux、UNIX、Windows、および z/OS オペレーティング・システム用 *asnapply* 呼び出しパラメーター定義 (続き)

パラメーター	定義
spillfile= <i>filetype</i>	フェッチした応答セットをどこに保管するかを指定します。 Linux、UNIX および Windows の場合、有効な値は以下のとおりです。 disk (デフォルト) ディスク・ファイル。 z/OS の場合、有効な値は以下のとおりです。 mem (デフォルト) メモリー・ファイル。応答セット用の十分なメモリーがない場合、アプライ・プログラムは失敗します。 disk ディスク・ファイル。

戻りコード

asnapply 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。

asnapply の例

以下の例は、**asnapply** コマンドの使用法を示しています。

例 1

アプライ修飾子 **AQ1** を使用し、コントロール・サーバー名は **dbx** で、**/home/files/apply/** ディレクトリーにある作業ファイルを使用して、アプライ・プログラムを始動する例です。

```
asnapply apply_qual=AQ1 control_server=dbx apply_path=/home/files/apply/
pwdfile=pass1.txt
```

アプライ・プログラムは **/home/files/apply/** ディレクトリーで **pass1.txt** という名前のパスワード・ファイルを探します。

例 2

ASNLOAD 出口ルーチン呼び出すアプライ・プログラムを始動します。

```
asnapply apply_qual=AQ1 control_server=dbx pwdfile=pass1.txt loadxit=y
```

この例では、アプライ・プログラムは **pass1.txt** という名前のパスワード・ファイルを現行ディレクトリーで探します。

例 3

適格なサブスクリプション・セットごとにコピー・サイクルを 1 回実行するアプライ・プログラムを始動します。

```
asnapply apply_qual=AQ1 control_server=dbx apply_path=/home/files/apply/
copyonce=y
```

この例では、アプライ・プログラムは /home/files/apply/ ディレクトリーで、デフォルトのパスワード・ファイル (asnpwd.aut) を探します。

関連タスク:

- 483 ページの『第 20 章 SQL レプリケーション・プログラムの操作 (z/OS)』

関連資料:

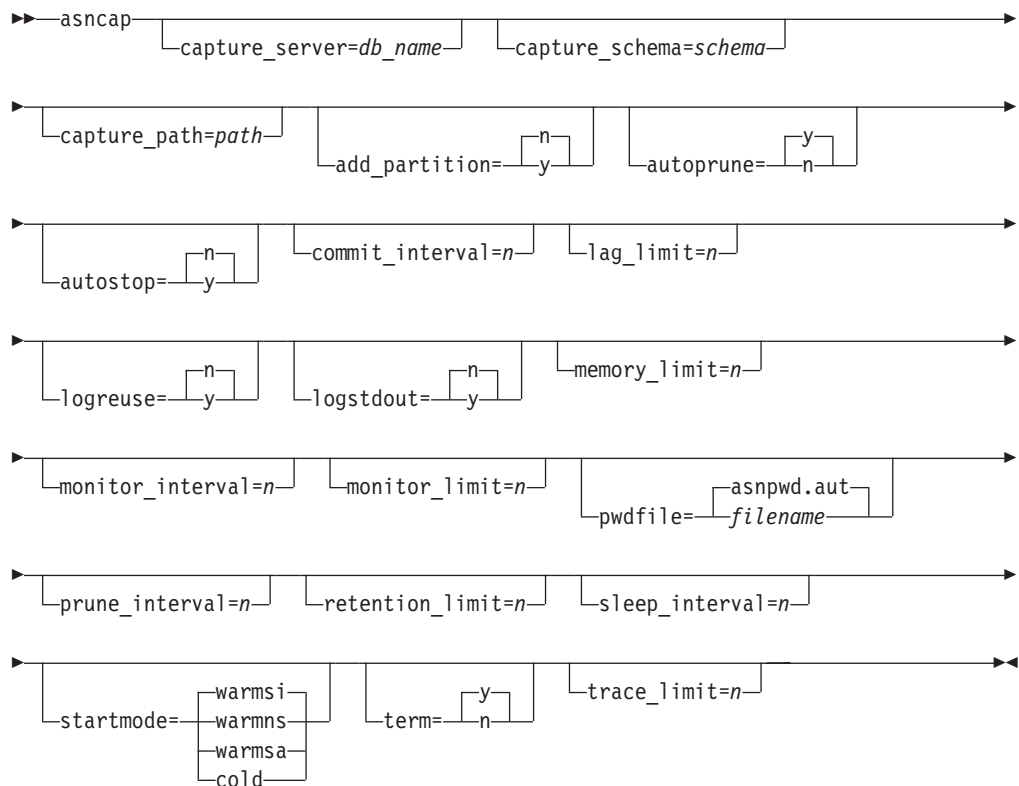
- 460 ページの『STRDPRAPY: アプライ・プログラムの始動 (OS/400)』

asnscap: キャプチャーの始動

Linux、UNIX、Windows、および z/OS 上の UNIX System Services (USS) でキャプチャー・プログラムを始動するには、**asnscap** コマンドを使用します。このコマンドは、レプリケーション・センターではなく、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。

キャプチャー・プログラムを始動すると、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 29 は、呼び出しパラメーターを定義します。

表 29. Linux、UNIX、Windows、および z/OS オペレーティング・システム用 *asncap* 呼び出しパラメーター定義

パラメーター	定義
capture_server=db_name	<p>キャプチャー・コントロール・サーバーの名前を指定します。</p> <p>Linux、UNIX および Windows の場合: キャプチャー・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。</p> <p>z/OS の場合: 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>
add_partition=y/n	<p>Linux、UNIX および Windows の場合のみ: キャプチャー・プログラムが最後に再始動されてから新たに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キャプチャー・プログラムが開始する場所を指定します。</p> <p>n (デフォルト) 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> <p>y 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、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>
capture_schema=schema	<p>特定のキャプチャー・プログラムを識別するために使用するキャプチャー・スキーマの名前を指定します。入力するスキーマ名の長さは 1 から 30 文字でなければなりません。デフォルトは ASN です。</p>
capture_path=path	<p>キャプチャー・プログラムが使用する作業ファイルのロケーションを指定します。デフォルトは、asncap コマンドが呼び出されたディレクトリーです。</p>

表 29. Linux、UNIX、Windows、および z/OS オペレーティング・システム用 asncap 呼び出しパラメーター定義 (続き)

パラメーター	定義
autoprune=y/n	<p>変更データ (CD) 表、作業単位 (UOW) 表、キャプチャー・モニター (IBMSNAP_CAPMON) 表、キャプチャー・トレース (IBMSNAP_CAPTRACE) 表、およびシグナル (IBMSNAP_SIGNAL) 表の中の行の自動整理を可能にするかどうかを指定します。</p> <p>y (デフォルト) キャプチャー・プログラムは、キャプチャー・パラメーター (IBMSNAP_CAPPARMS) 表で指定されたインターバルで、適格な行の自動整理を行います。キャプチャー・プログラムは、行が複製されたものか否かに関係なく、保持制限より古い CD、UOW、および IBMSNAP_SIGNAL の行を削除します。</p> <p>n 自動整理は使用不可になります。</p>
autostop=y/n	<p>キャプチャー・プログラムの始動前にログに記録されたすべてのトランザクションを検索した後、キャプチャー・プログラムを終了するかどうかを指定します。</p> <p>n (デフォルト) キャプチャー・プログラムは、トランザクションを検索した後、終了しません。</p> <p>y キャプチャー・プログラムは、トランザクションを検索した後、終了します。</p>
commit_interval=n	<p>キャプチャー・プログラムが、何秒待ってから、作業単位 (UOW) 表および変更データ (CD) 表に行をコミットするかを示す秒数を指定します。デフォルトは 30 秒です。</p>
lag_limit=n	<p>キャプチャー・プログラムがログ・レコードを処理するときに、許される遅れの分数を指定します。デフォルトは 10,080 分 (7 日) です。キャプチャー・プログラムは、ウォーム・スタートの場合にのみ、このパラメーターの値をチェックします。この限界を超えると、キャプチャー・プログラムは開始されません。</p>

表 29. Linux、UNIX、Windows、および z/OS オペレーティング・システム用 asncap 呼び出しパラメーター定義 (続き)

パラメーター	定義
logreuse=y/n	<p>キャプチャー・プログラムがログ・ファイルを再利用するか、またはログ・ファイルにメッセージを付加するかを指定します (ログ・ファイル名は <code>db2instance.capture_server.capture_schema.CAP.log</code>)。</p> <p>n (デフォルト) キャプチャー・プログラムは、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> <p>y キャプチャー・プログラムは、まず現行のログ・ファイルを切り捨て、次にキャプチャー・プログラムの再始動時に新しいログを開始して、ログ・ファイルを再利用します。</p> <p>z/OS の場合: ログ・ファイル名には DB2 インスタンス名は含まれません (<code>capture_server.capture_schema.CAP.log</code>)。</p>
logstdout=y/n	<p>キャプチャー・プログラムがログ・ファイル・メッセージを誘導する場所を指定します。</p> <p>n (デフォルト) 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・ファイルにのみ送ります。初期化メッセージは、ログ・ファイルと標準出力 (STDOUT) の両方に送られます。</p> <p>y キャプチャー・プログラムは、ログ・ファイルと標準出力 (stdout) の両方にメッセージを送信します。</p>
memory_limit=n	<p>トランザクションを作成するためにキャプチャー・プログラムが使用できるメモリーの最大サイズ (MB 単位) を指定します。このメモリー限度に達すると、キャプチャー・プログラムはトランザクションをファイルに書き出します。デフォルトは 32 MB です。</p>
monitor_interval=n	<p>キャプチャー・プログラムが、キャプチャー・モニター (IBMSNAP_CAPMON) 表に行を挿入する頻度 (秒単位) を指定します。デフォルトは 300 秒 (5 分) です。</p>
monitor_limit=n	<p>キャプチャー (IBMSNAP_CAPMON) 表内の行は、何分たったら、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。monitor_limit パラメーターの値よりも古い、すべての IBMSNAP_CAPMON 行は、次の整理サイクルで削除されません。デフォルトは 10,080 分 (7 日) です。</p>
pwdfile=filename	<p>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 <code>asnpwd.aut</code> 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>

表 29. Linux、UNIX、Windows、および z/OS オペレーティング・システム用 asncap 呼び出しパラメーター定義 (続き)

パラメーター	定義
prune_interval=n	変更データ (CD) 表、作業単位 (UOW) 表、キャプチャー・モニター (IBMSNAP_CAPMON) 表、キャプチャー・トレース (IBMSNAP_CAPTRACE) 表、およびシグナル (IBMSNAP_SIGNAL) 表の整理の頻度 (秒数) を指定します。autoprune パラメーターを n に設定した場合は、このパラメーターは無視されます。デフォルトは 300 秒 (5 分) です。
retention_limit=n	変更データ (CD) 表、作業単位 (UOW) 表、またはシグナル (IBMSNAP_SIGNAL) 表の行は、何分たったら、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。 retention_limit パラメーターの値よりも古い行は、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。
sleep_interval=n	キャプチャー・プログラムが、アクティブ・ログの処理を終了し、バッファーが空であると判断するまで、何秒スリープするかを示す秒数を指定します。デフォルトは 5 秒です。 z/OS の場合: バッファーが半分以上空いていることを戻した後、キャプチャー・プログラムが何秒スリープするかを指定します。

表 29. Linux、UNIX、Windows、および z/OS オペレーティング・システム用 asncap 呼び出しパラメーター定義 (続き)

パラメーター	定義
startmode=mode	<p>キャプチャー始動時にキャプチャー・プログラムが使用する、処理プロシーチャーを指定します。</p> <p>warmsi (デフォルト) ウォーム・スタート情報を入手できる場合、キャプチャー・プログラムは、直前の実行の終了時点から処理を再開します。これがキャプチャー・プログラムの最初の始動である場合は、自動的にコールド・スタートに切り替えます。</p> <p>ウォーム・スタート中は、キャプチャー・プログラムは、キャプチャー・トレース (IBMSNAP_CAPTRACE) 表、変更データ (CD) 表、作業単位 (UOW) 表、および再始動 (IBMSNAP_RESTART) 表に手を付けずそのままにしておきます。キャプチャー・プログラムの始動後にエラーが起こった場合、キャプチャー・プログラムは終了します。</p> <p>warmns ウォーム・スタート情報を入手できる場合、キャプチャー・プログラムは、直前の実行の終了時点から処理を再開します。キャプチャー・プログラムの始動後にエラーが起こった場合、キャプチャー・プログラムは終了します。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。</p> <p>warmsa ウォーム・スタート情報を入手できる場合、キャプチャー・プログラムは、直前の実行の終了時点から処理を再開します。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 キャプチャー・プログラムを始動すると、CD、UOW 表内のすべての行を削除します。ほとんどの登録はリセットされ、それらのソースに対するすべてのサブスクリプションは、次のアプライ処理のサイクルですべてリフレッシュされます。外部 CCD の登録とそれらのサブスクリプション (そのターゲットが非コンプリート CCD である) は、完全には更新されません。</p>

表 29. Linux、UNIX、Windows、および z/OS オペレーティング・システム用 asncap 呼び出しパラメーター定義 (続き)

パラメーター	定義
term=y/n	<p>DB2 が終了した場合にキャプチャー・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 が終了した場合、キャプチャー・プログラムは終了します。</p> <p>n DB2 が MODE(QUIESCE) で終了した場合、キャプチャー・プログラムは実行を継続します。DB2 が始動されると、キャプチャー・プログラムは warm モードで始動し、DB2 が終了した時点からの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 にしてもキャプチャー・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、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>キャプチャー・トレース (IBMSNAP_CAPTRACE) 表の行は、何分たったら、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_CAPTRACE 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。</p>

戻りコード

asncap 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。

asncap の例

次の例は、**asncap** コマンドの使用法を示しています。

例 1

db という名前のキャプチャー・コントロール・サーバーおよび、キャプチャー・スキーマ ASN を使用し、 /home/files/capture/logs/ ディレクトリーにある作業ファイルを使用して、初めてキャプチャー・プログラムを始動する例です。

```
asncap capture_server=db capture_schema=ASN
capture_path=/home/files/capture/logs/ startmode=cold
```

例 2

キャプチャー・プログラムが停止した後、整理を行わずにキャプチャー・プログラムを再始動する例です。

```
asncap capture_server=db autoprune=n sleep_interval=10 startmode=warmsa
```

この例では、キャプチャー・プログラムは、対応するコントロール表内のすべての行を保存し、アクティブ・ログの処理を終了し、バッファが空であると判断した後、10 秒スリープします。ウォーム・スタート情報を入手できない場合、キャプチャー・プログラムは、直前の実行の終了時点から処理を再開し、コールド・スタートに切り替えます。

例 3

warmns startmode で、変更されたパラメーター設定値を使用して、キャプチャー・プログラムを再始動します。

```
asncap capture_server=db autoprune=y prune_interval=60 retention_limit=1440
startmode=warmns
```

このコマンドは、キャプチャー・プログラムを再始動し、新しいパラメーター設定値を使用するようにします。新しいパラメーター設定値は、CD、UOW、および IBMSNAP_SIGNAL 表が整理の対象として適格になるまでの時間を減らし、整理の頻度をデフォルトのパラメーター設定値よりも増やしています。ウォーム・スタート情報を入手できない場合、キャプチャー・プログラムは、直前の実行の終了時点から処理を再開しますが、自動的にコールド・スタートに切り替えることはしません。

例 4

その作業ファイルをすべて、capture_files という新しいサブディレクトリーに送信するキャプチャー・プログラムを始動します。

1. 該当のディレクトリーに行き、 capture_files という新しいサブディレクトリーを作成します。

```
cd /home/db2inst
mkdir capture_files
```

2. キャプチャー・プログラムを始動し、たった今作成した新しいサブディレクトリーにあるキャプチャー・パスを指定します。

```
asncap capture_server=db capture_schema=ASN
capture_path=/home/db2inst/capture_files startmode=warmns
```

関連タスク:

- 483 ページの『第 20 章 SQL レプリケーション・プログラムの操作 (z/OS)』

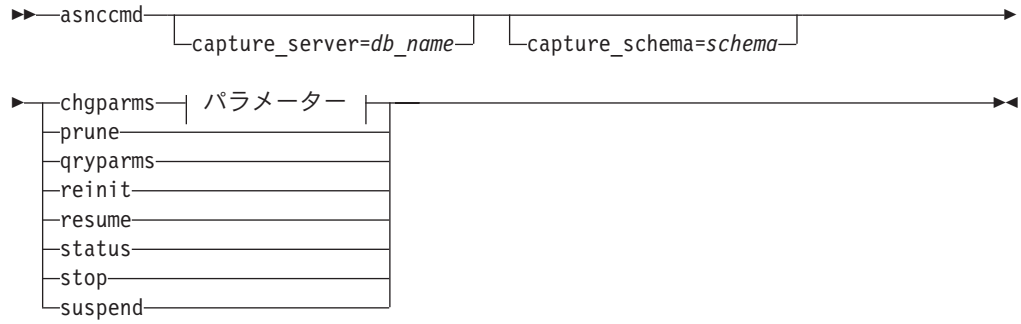
関連資料:

- 468 ページの『STRDPRCAP: キャプチャー・プログラムの始動 (OS/400)』

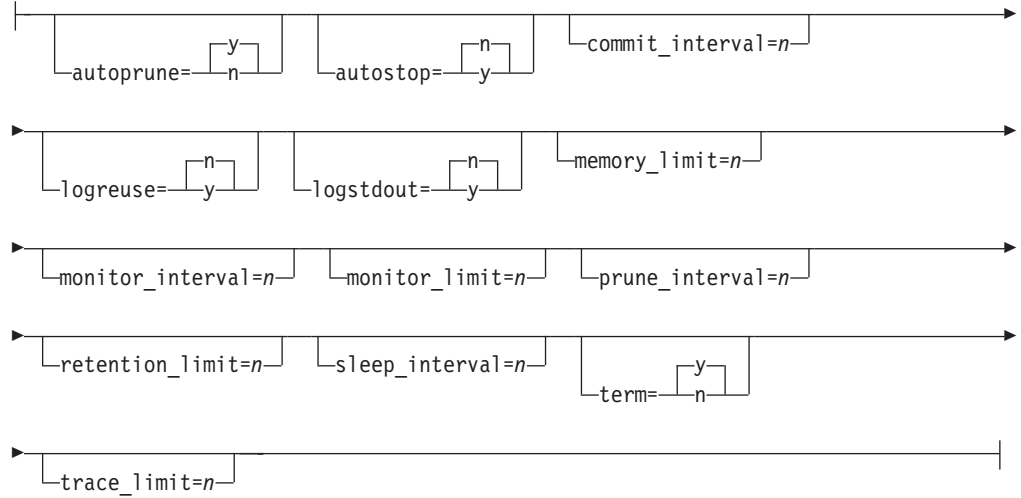
asncmd: キャプチャーの操作

Linux、UNIX、Windows、および z/OS 上の UNIX System Services (USS) でキャプチャー・プログラムを操作するには、**asncmd** 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

| このコマンドの呼び出しパラメーターは、 342 ページの『asncap: キャプチャーの
 | 始動』中のパラメーターと同一です。これらのパラメーターの定義については、
 | 343 ページの表 29 を参照してください。

表 30 は、chgparms 呼び出しパラメーターを定義しています。

表 30. Linux、UNIX、Windows、および z/OS オペレーティング・システム用 *asnccmd* *chgparms* パラメーター定義

パラメーター	定義
autoprun =y/n	<p>変更データ (CD) 表、作業単位 (UOW) 表、キャプチャー・モニター (IBMSNAP_CAPMON) 表、キャプチャー・トレース (IBMSNAP_CAPTRACE) 表、およびシグナル (IBMSNAP_SIGNAL) 表の中の行の自動整理を可能にするかどうかを指定します。</p> <p>y (デフォルト) キャプチャー・プログラムは、キャプチャー・パラメーター (IBMSNAP_CAPPARMS) 表で指定されたインターバルで、適格な行の自動整理を行います。キャプチャー・プログラムは、行が複製されたものか否かに関係なく、保持制限より古い CD、UOW、および IBMSNAP_SIGNAL の行を削除します。</p> <p>n 自動整理は使用不可になります。</p>
autostop =y/n	<p>キャプチャー・プログラムの始動前にログに記録されたすべてのトランザクションを検索した後、キャプチャー・プログラムを終了させるかどうかを指定します。</p> <p>n (デフォルト) キャプチャー・プログラムは、トランザクションを検索した後、終了しません。</p> <p>y キャプチャー・プログラムは、トランザクションを検索した後、終了します。</p>
commit_interval =n	<p>キャプチャー・プログラムが、何秒待ってから、作業単位 (UOW) 表および変更データ (CD) 表に行をコミットするかを示す秒数を指定します。デフォルトは 30 秒です。</p>
logreuse =y/n	<p>キャプチャー・プログラムがログ・ファイルを再利用するか、またはログ・ファイルにメッセージを付加するかを指定します (ログ・ファイル名は <code>db2instance.capture_server.capture_schema.CAP.log</code>)。</p> <p>n (デフォルト) キャプチャー・プログラムは、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> <p>y キャプチャー・プログラムは、まず現行のログ・ファイルを切り捨て、次にキャプチャー・プログラムの再始動時に新しいログを開始して、ログ・ファイルを再利用します。</p> <p>chgparms パラメーターを使用してこのパラメーターを y に変更すると、ログは即時に切り捨てられ、再利用されます。この場合の logreuse パラメーターの変更は、キャプチャー・プログラムの次の始動には影響しません。</p> <p>z/OS の場合: ログ・ファイル名には DB2 インスタンス名は含まれません (<code>capture_server.capture_schema.CAP.log</code>)。</p>

表 30. Linux、UNIX、Windows、および z/OS オペレーティング・システム用 *asnccmd* *chgparms* パラメーター定義 (続き)

パラメーター	定義
logstdout=y/n	<p>キャプチャー・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) キャプチャー・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 キャプチャー・プログラムは、ログ・ファイルと標準出力 (stdout) の両方にメッセージを送信します。</p>
memory_limit=n	<p>トランザクションを作成するためにキャプチャー・プログラムが使用できるメモリーの最大サイズ (MB 単位) を指定します。このメモリー限度に達すると、キャプチャー・プログラムはトランザクションをファイルに書き出します。デフォルトは 32 MB です。</p>
monitor_interval=n	<p>キャプチャー・プログラムが、キャプチャー・モニター (IBMSNAP_CAPMON) 表に行を挿入する頻度 (秒単位) を指定します。デフォルトは 300 秒 (5 分) です。</p>
monitor_limit=n	<p>キャプチャー (IBMSNAP_CAPMON) 表内の行は、何分たったら、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。monitor_limit パラメーターの値よりも古い、すべての IBMSNAP_CAPMON 行は、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。</p>
prune_interval=n	<p>変更データ (CD) 表、作業単位 (UOW) 表、キャプチャー・モニター (IBMSNAP_CAPMON) 表、キャプチャー・トレース (IBMSNAP_CAPTRACE) 表、およびシグナル (IBMSNAP_SIGNAL) 表の整理の頻度 (秒数) を指定します。autoprune パラメーターを n に設定した場合は、このパラメーターは無視されます。デフォルトは 300 秒 (5 分) です。</p>
retention_limit=n	<p>変更データ (CD) 表、作業単位 (UOW) 表、またはシグナル (IBMSNAP_SIGNAL) 表の行は、何分たったら、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。retention_limit パラメーターの値よりも古い行は、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。</p>
sleep_interval=n	<p>キャプチャー・プログラムが、アクティブ・ログの処理を終了し、バッファーが空であると判断するまで、何秒スリープするかを示す秒数を指定します。デフォルトは 5 秒です。</p> <p>z/OS の場合: バッファーが半分以上空いていることを戻した後、キャプチャー・プログラムが何秒スリープするかを指定します。</p>

表 30. Linux、UNIX、Windows、および z/OS オペレーティング・システム用 *asnccmd* *chgparms* パラメーター定義 (続き)

パラメーター	定義
term=y/n	<p>DB2 が終了した場合にキャプチャー・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 が終了した場合、キャプチャー・プログラムは終了します。</p> <p>n DB2 が MODE(QUIESCE) で終了した場合、キャプチャー・プログラムは実行を継続します。DB2 が始動されると、キャプチャー・プログラムは warm モードで始動し、DB2 が終了した時点からの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 にしてもキャプチャー・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、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>キャプチャー・トレース (IBMSNAP_CAPTRACE) 表の行は、何分たったら、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_CAPTRACE 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。</p>

asnccmd の例

以下の例は、**asnccmd** コマンドの使用法を示しています。

例 1

新しく追加したレプリケーション・ソースを、実行中のキャプチャー・プログラムに認識させるには、次のように入力します。

```
asnccmd capture_server=db capture_schema=ASN reinit
```

例 2

CD、UOW、IBMSNAP_CAPMON、IBMSNAP_CAPTRACE、および IBMSNAP_SIGNAL 表の整理を 1 回行う例です。

```
asnccmd capture_server=db capture_schema=ASN prune
```

例 3

それぞれのキャプチャー・スレッドの状態についてメッセージを受け取る場合の例です。

```
asnccmd capture_server=db capture_schema=ASN status
```

例 4

asnccmd

キャプチャー・プログラムの現行操作値を標準出力に送信する例です。

```
asnccmd capture_server=db capture_schema=ASN qryparms
```

例 5

キャプチャー・プログラムの実行中に自動整理を行わないようにするには、次のように入力します。

```
asnccmd capture_server=db capture_schema=ASN chgparms autoprunen
```

例 6

キャプチャー・プログラムの実行を停止するには、次のように入力します。

```
asnccmd capture_server=db capture_schema=ASN stop
```

関連タスク:

- 483 ページの『第 20 章 SQL レプリケーション・プログラムの操作 (z/OS)』

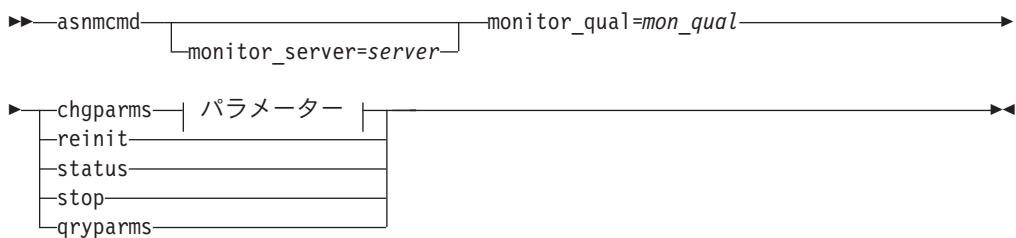
関連資料:

- 448 ページの『OVRDPRCAPA: DPR キャプチャー属性のオーバーライド (OS/400)』

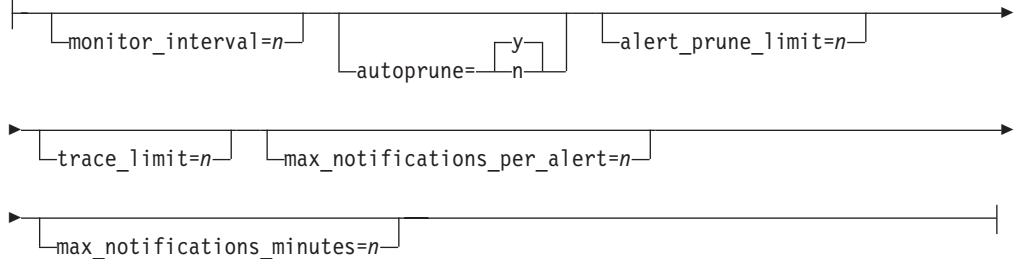
asnmcmd: 実行中のレプリケーション・アラート・モニターの処理

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:



パラメーター

表 31 は、 **asnmcmd** コマンドの呼び出しパラメーターを定義します。

表 31. 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	<p>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。</p> <p>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。</p> <p>z/OS: デフォルトは DSN です。</p>
monitor_qual=mon_qual	<p>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。</p> <p>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。</p>
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。 asnmcmd: 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>

表 31. 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
reinit	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。
status	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。
qryparms	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。
stop	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。

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 qrypargs
```

関連資料:

- 357 ページの『asnmon: レプリケーション・アラート・モニターの始動』

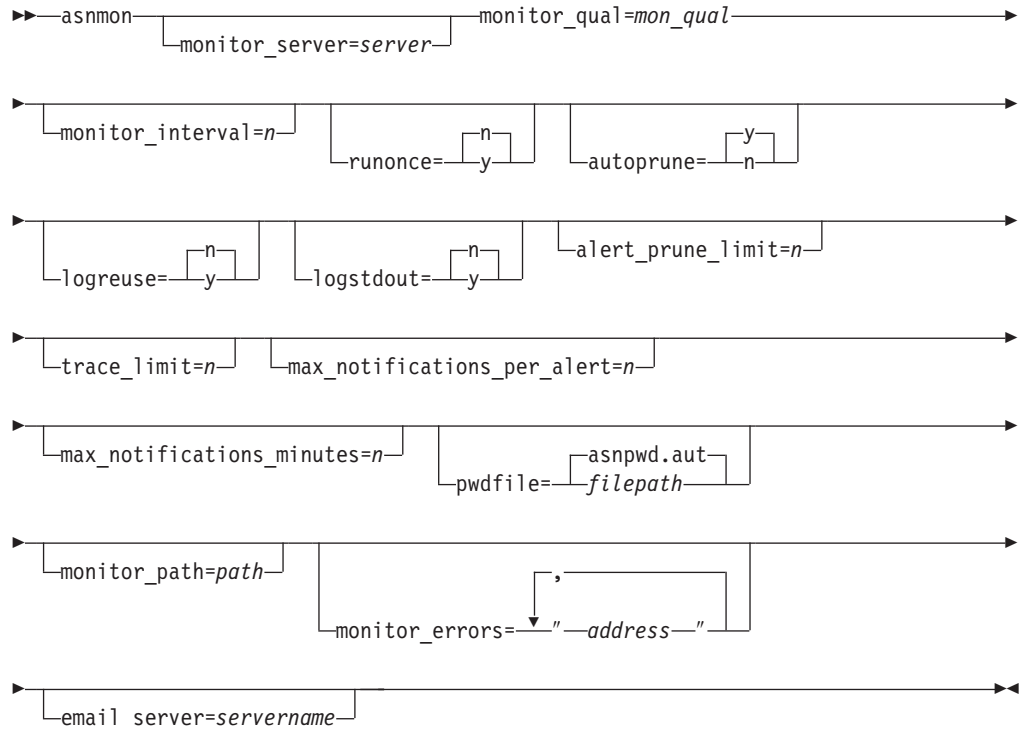
asnmon: レプリケーション・アラート・モニターの始動

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 32 は、**asnmon** コマンドの呼び出しパラメーターを定義します。

表 32. *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 32. 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 (IBMSNAP_ALERTS) 表内の行の自動整理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 自動整理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 32. 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 (IBMSNAP_ALERTS) 表に、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 (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:

- 354 ページの『asnmcmd: 実行中のレプリケーション・アラート・モニターの処理』

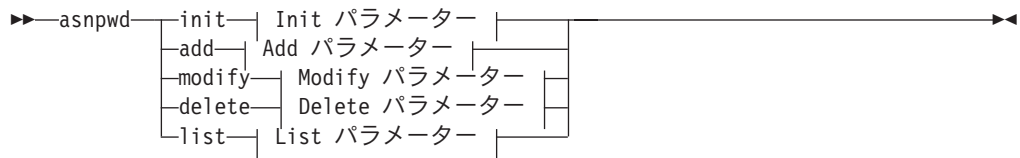
asnpwd: パスワード・ファイルの作成および保守

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。

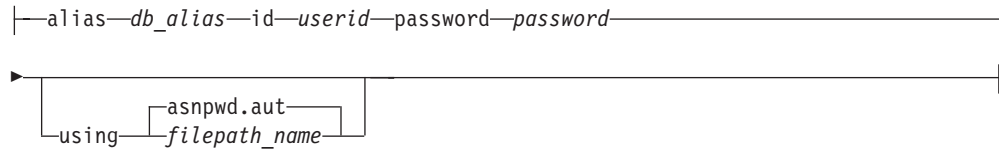
構文



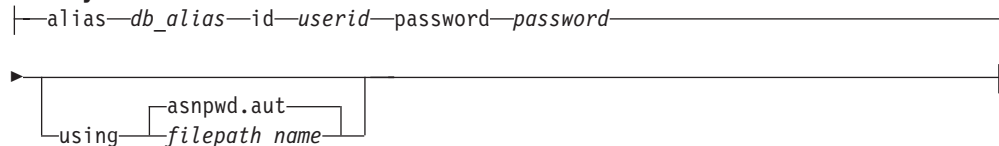
Init パラメーター:



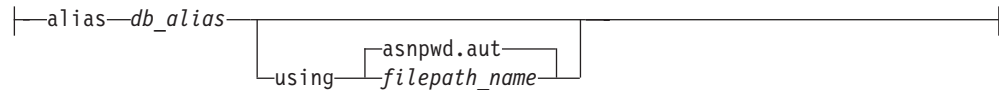
Add パラメーター:



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 33 は、**asnpwd** コマンドの呼び出しパラメーターを定義します。

表 33. 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できます。パスワードを表示することはできません。

表 33. 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.
```

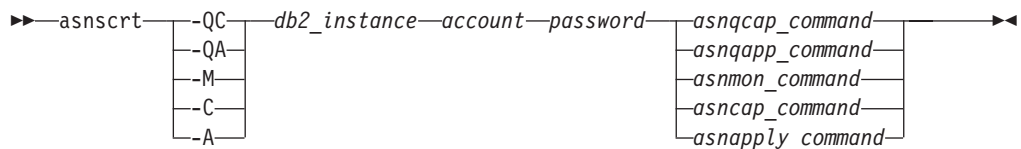
- **encrypt all** パラメーターを使用して初期設定されなかった場合、以下の詳細がリストされます。

```
asnpwd LIST USING pass1.aut
Alias: SAMPLE ID: chglocalid
Number of Entries: 1
```

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、**asnsqcap**、**asnsqapp**、**asnsmon**、**asnsqcap**、および **asnsapply** 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



パラメーター

表 34 は、**asnscri** コマンドの呼び出しパラメーターを定義します。

表 34. 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。

表 34. 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>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>asnscri_command</i>	<p>Q キャプチャー・プログラムを始動するための完全な asnscri コマンドを指定します。文書化されている asnscri コマンド構文と該当の asnscri 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、asnscri 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 asnscri パラメーターの構文の妥当性をチェックしません。</p>
<i>asnscriapp_command</i>	<p>Q アプライ・プログラムを始動するための完全な asnscriapp コマンドを指定します。文書化されている asnscriapp コマンド構文と該当の asnscriapp 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、asnscriapp 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 asnscriapp パラメーターの構文の妥当性をチェックしません。</p>

表 34. 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・プログラムを始動するための完全な 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>asncap_command</i>	<p>キャプチャー・プログラムを始動するための完全な asncap コマンドを指定します。文書化されている asncap コマンド構文および該当の asncap 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、asncap 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 asncap 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・プログラムを始動するための完全な 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
```

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 **asnsCRT** コマンドを使用します。)

構文

```
▶▶ asnsdrop service_name
      ALL
```

パラメーター

表 35 は、**asnsdrop** コマンドの呼び出しパラメーターを定義します。

表 35. 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んでください。
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

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
```

asnslist: 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、**asnslist** 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。**asnslist** 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

```
asnslist [DETAILS]
```

パラメーター

表 36 は、**asnslist** コマンドの呼び出しパラメーターを定義します。

表 36. Windows オペレーティング・システム用の *asnslist* 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。

asnslist の例

次の例は、**asnslist** 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。

```
asnslist
```

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。

```
asnslist 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

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 37 は、**asntdiff** 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、および Windows オペレーティング・システム用 **asntdiff** 呼び出しパラメーター定義

パラメーター	定義
db=server	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。 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。

表 37. Linux、UNIX、および Windows オペレーティング・システム用 `asntdiff` 呼び出しパラメーター定義 (続き)

パラメーター	定義
<code>where=WHERE_clause</code>	<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。</p> <p>Q レプリケーション</p> <p>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</p> <p>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>
<code>diff=table_name</code>	<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.ASNTDIFF</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:

- 「*IBM DB2 Information Integrator レプリケーションとイベント・パブリッシングガイド*および*リファレンス*」の『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』

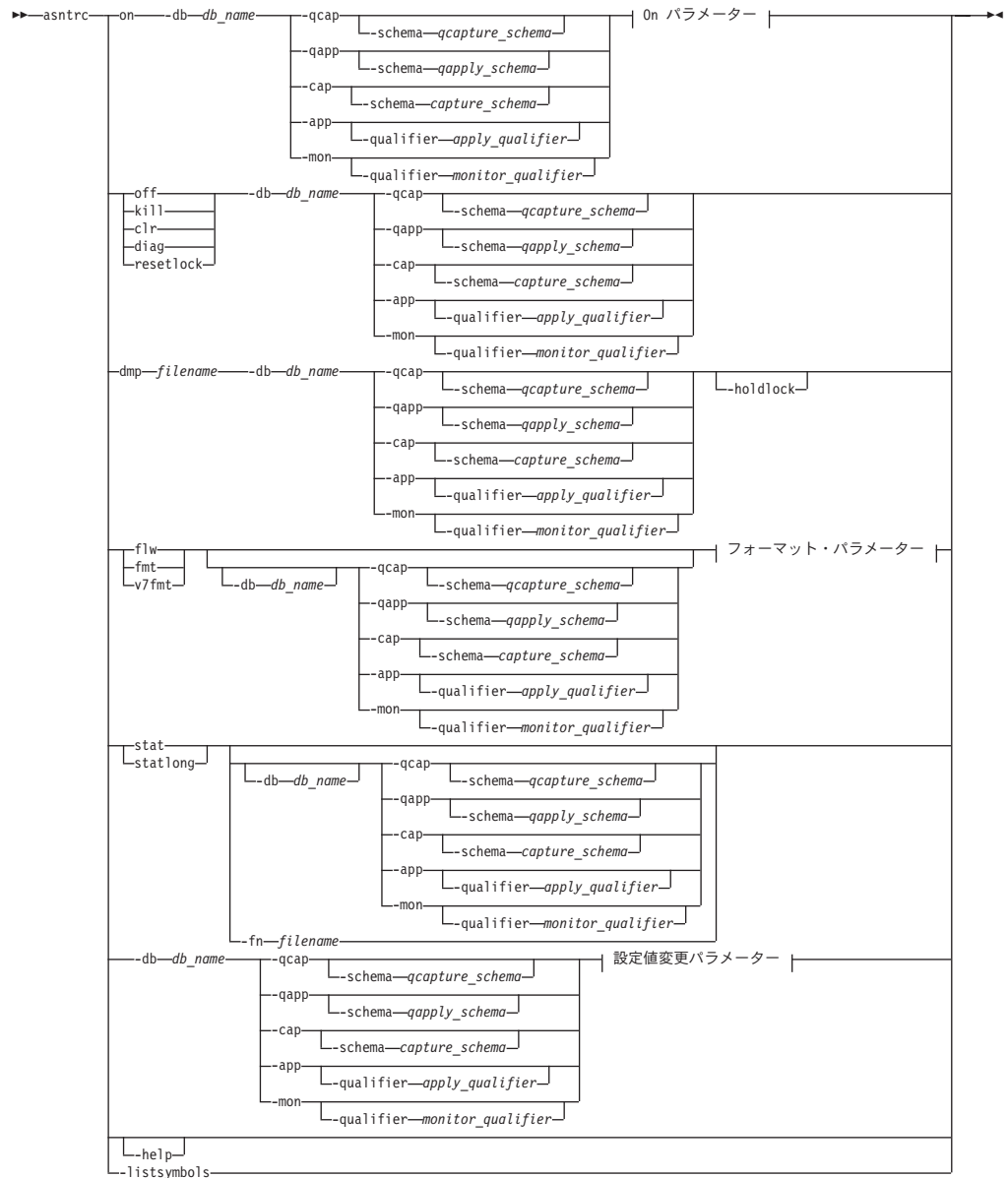
関連資料:

- 「*IBM DB2 Information Integrator レプリケーションとイベント・パブリッシングガイド*および*リファレンス*」の『ロードマップ: Q レプリケーションとイベント発行のシステム・コマンド』
- 382 ページの『asntrep: ソース表とターゲット表の間の違いの修復』

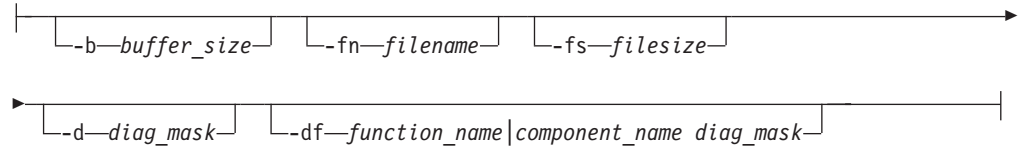
asntrc: レプリケーション・トレース機能の操作

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。

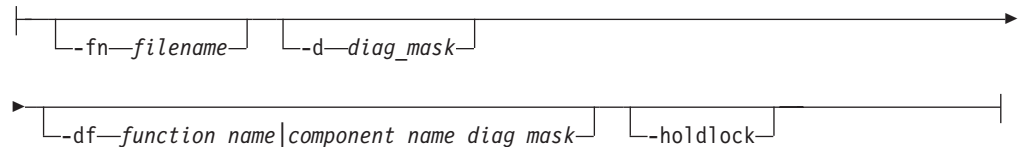
構文



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:

```
┌───┴───┐ ┌───┴───┐ ┌───┴───┐ ┌───┴───┐
└─d─diag_mask┘ └─df─function_name|component_name diag_mask┘
```

パラメーター

表 38 は、**asntrc** 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*、*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します。 • 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します。 • 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します。 • 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します。 • 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します。
-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 38. 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
-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 38. 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 38. 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> <p>1 フロー・データ。関数の入口点と出口点が含まれます。</p> <p>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。</p> <p>3 詳細データ。主要なイベントとその記述が含まれます。</p> <p>4 パフォーマンス・データ。</p> <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 つまたは複数入力し、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 つまたは複数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。

- 最大のバッファ・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・プログラムを始動し、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。

```
asntrc 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。

```
asntrc 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。

```
asntrc 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。

```
asntrc flw -fn myfile.trc
```

および

```
asntrc 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。

```
asntrc 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。

```
asntrc on -db mydb -app -qualifier aq1 -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。

```
asntrc 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ー・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・プログラムを始動し、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、指定する必要が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、**asnlrcap** 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
asnlrcap 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、**asnlrapp** 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
asnlrapp apply_server=TSN3 apply_schema=ELB
```

asnlrc

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、**asncap** 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 DSN6 -schema JAY -cap
asncap capture_server=DSN6 capture_schema=JAY
```

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、**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。

```
asnlrc -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、**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、**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。

```
asnlrc -db DSN6 -qualifier MONQUAL -mon
asnmon monitor_server=DSN6 monitor_qual=MONQUAL
```

asnlrep: 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、**asnlrep** コマンドを使用します。**asnlrep** 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

```
▶▶ asnlrep—db=server—schema=schema—where=WHERE_clause—diff=table_name▶▶
```

パラメーター

表 39 は、**asntrep** コマンドの呼び出しパラメーターを定義します。

表 39. 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 (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:

- 「*IBM DB2 Information Integrator レプリケーションとイベント・パブリッシングガイドおよびリファレンス*」の『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:

- 「*IBM DB2 Information Integrator レプリケーションとイベント・パブリッシングガイドおよびリファレンス*」の『ロードマップ: Q レプリケーションとイベント発行のシステム・コマンド』
- 372 ページの『asntdiff: ソース表とターゲット表とのデータの比較』

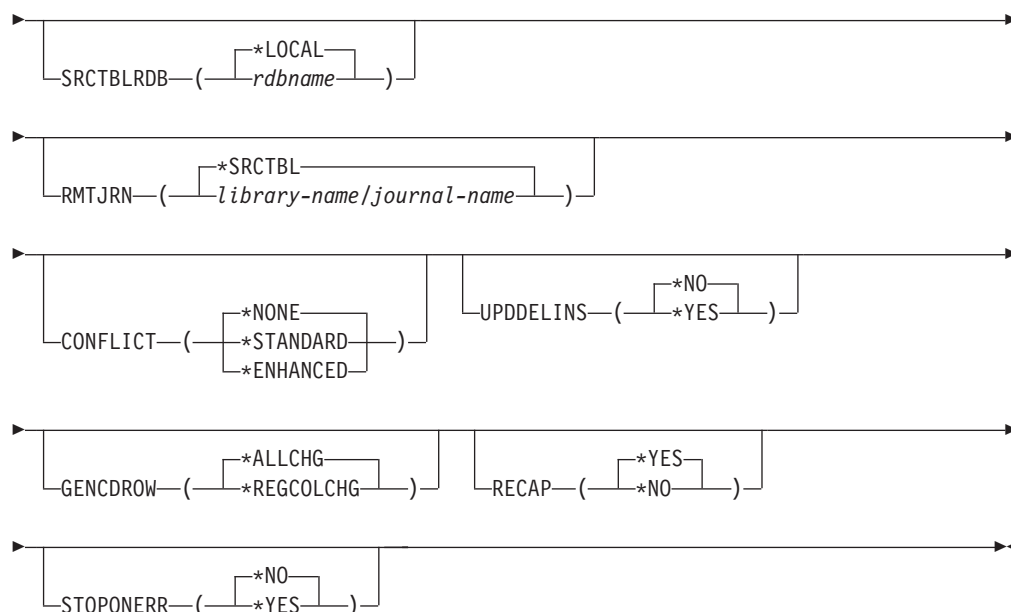
第 19 章 SQL レプリケーション用のシステム・コマンド (OS/400)

この章では、iSeries サーバーの OS/400 オペレーティング・システムの下で実行するレプリケーション・コマンドについて説明します。これらのコマンドは、オペレーティング・システムのコマンド・プロンプトか、またはコマンド行プログラムを使用して入力することができます。

この章には、コマンド別のセクションがあります。各セクションには、コマンドの簡単な記述、構文図、およびパラメーターとそれに対応する定義の表があります。各セクションの最後には、コマンドの使用例と、関連情報への相互参照があります。

コマンドには、以下のものがあります。

- 386 ページの『ADDDPRREG: DPR 登録の追加 (OS/400)』
- 395 ページの『ADDDPRSUB: DPR サブスクリプション・セットの追加 (OS/400)』
- 411 ページの『ADDDPRSUBM: DPR サブスクリプション・セットのメンバーの追加 (OS/400)』
- 422 ページの『ANZDPR: アナライザーの操作 (OS/400)』
- 425 ページの『CHGDPRCAPA: DPR キャプチャー属性の変更 (OS/400)』
- 430 ページの『CRTDPRTBL: レプリケーション・コントロール表の作成 (OS/400)』
- 431 ページの『ENDDPRAPY: アプライ・プログラムの停止 (OS/400)』
- 434 ページの『ENDDPRCAP: キャプチャー・プログラムの停止 (OS/400)』
- 436 ページの『GRDPRAUT: ユーザーの許可 (OS/400)』
- 446 ページの『INZDPRCAP: DPR キャプチャー・プログラムの再初期化 (OS/400)』
- 448 ページの『OVRDPRCAPA: DPR キャプチャー属性のオーバーライド (OS/400)』
- 453 ページの『RMVDPRREG: DPR 登録の除去 (OS/400)』
- 454 ページの『RMVDPRSUB: DPR サブスクリプション・セットの除去 (OS/400)』
- 456 ページの『RMVDPRSUBM: DPR サブスクリプション・セット・メンバーの除去 (OS/400)』
- 458 ページの『RVKDPRAUT: 権限の取り消し (OS/400)』
- 460 ページの『STRDPAPY: アプライ・プログラムの始動 (OS/400)』
- 468 ページの『STRDPRCAP: キャプチャー・プログラムの始動 (OS/400)』
- 476 ページの『WRKDPTRC: DPR トレース機能の使用方法 (OS/400)』



注:

1 列名は 300 個まで指定できます。

表 40 では、呼び出しパラメーターをリストしています。

表 40. ADDDPRREG コマンド・パラメーター定義 (OS/400 版)

パラメーター	定義およびプロンプト
SRCTBL	<p>ソース表として登録する表を指定します。キャプチャー・プログラムは、OS/400 ライブラリー内の任意の物理ファイルまたは、外部で定義され、1つのフォーマットを持つ集合をサポートします。このパラメーターは必須です。</p> <p><i>library-name/file-name</i> 登録する表の修飾名を表します。</p>
CAPCTLLIB	<p>キャプチャー・スキーマ (キャプチャー・コントロール表が存在するライブラリーの名前) を指定します。</p> <p>ASN (デフォルト) キャプチャー・コントロール表は ASN ライブラリー内にあります。</p> <p><i>library-name</i> キャプチャー・コントロール表を含むライブラリーの名前。 CRTDPRTBL コマンドに CAPCTLLIB パラメーターを指定して、このライブラリーを作成することができます。</p>
CDLIB	<p>この登録されたソースの変更データ (CD) 表を作成するライブラリーを指定します。</p> <p>*SRCTBL (デフォルト) ソース表が存在するライブラリー内に CD 表を作成します。</p> <p><i>library-name</i> 指定したライブラリー名に CD 表を作成します。</p>

表 40. ADDDPRREG コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
CDNAME	<p>変更データ (CD) 表の名前を指定します。</p> <p>*DEFAULT (デフォルト) デフォルト名 (現行タイム・スタンプを基にしたもの) を使用して CD 表を作成します。たとえば、現行のタイム・スタンプが 2002 年 1 月 23 日 9 時 58 分 26 秒の場合、デフォルト名は ASN020123095826CD となります。</p> <p><i>cdname</i> ここに指定した名前 で CD 表を作成します。</p>
SRCTYPE	<p>登録するソース表のタイプを指定します。ご使用のレプリケーション構成に基づいてソース・タイプを選択します。</p> <ul style="list-style-type: none"> • 基本データ分散またはデータ統合構成の場合は、デフォルトの USERTABLE を使用します。 • Update-anywhere 構成の場合は REPLICA を使用します。 • 複数の階層からなる構成を持ち、ターゲット表を、レプリケーション構成内の下位の層のためのソースにしたい場合は、POINTINTIME、BASEAGR、CHANGEAGR、USERCOPY、または CCD を使用します。 <p>既存のターゲット表をソースとして登録する場合、指定されたソース・タイプに示された IBMSNAP 表の列がターゲット表に含まれていないと、登録は失敗します。</p>

表 40. ADDDPRREG コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
SRCTYPE (続き)	<p>*USERTABLE (デフォルト) ユーザー・データベース表。登録された表の最もよくあるタイプです。表には、DB2 DataPropagator for iSeries の列 ID である、IBMSNAP または IBMQSQ で始まる列を含めることはできません。</p> <p>*POINTINTIME ポイント・イン・タイム・コピー表。これは、ソース表の内容の一部またはすべてと一致する内容を含み、さらに、DB2 DataPropagator for iSeries システム列を保持します。このシステム列は、特定の行がソース・システムで最後に挿入または更新されたポイント・イン・タイムを示します。表には IBMSNAP_LOGMARKER タイム・スタンプ列を含める必要があり、オプションとして IBMQSQ_RRN と呼ばれる INTEGER 列を含めることができます。</p> <p>*BASEAGR 基本集約コピー。ユーザー表またはポイント・イン・タイム表からインターバルごとに集約されたデータが入ります。基礎集約表には、IBMSNAP_HLOGMARKER および IBMSNAP_LLOGMARKER タイム・スタンプ列を含める必要があります。</p> <p>*CHANGEAGR 変更を集約したコピー表。ソース表に記録された変更に基づくデータの集約が入ります。この表には、IBMSNAP_HLOGMARKER および IBMSNAP_LLOGMARKER タイム・スタンプ列を含める必要があります。</p> <p>*REPLICA レプリカ・サブスクリプションのターゲット表。ターゲット表からの変更が複製されて、元のソース表に戻せるようにするには、このタイプの表を登録します。この表には、DB2 DataPropagator for iSeries のシステム列または、DataPropagator for iSeries の列 ID である、IBMSNAP または IBMQSQ で始まる列を含めることはできません。この表は、元のソース表からのすべての列を保持します。</p> <p>*USERCOPY ソース表の内容のすべてまたは一部と一致する内容を持つ、ターゲット表です。ユーザー・コピー表はユーザー・データ列のみ保持します。</p>
SRCTYPE (続き)	<p>*CCD 整合変更データ (CCD) 表。ソース表からのトランザクション整合性のあるデータが入ります。表には、以下のように定義された列を含める必要があります。</p> <ul style="list-style-type: none"> • IBMSNAP_INTENTSEQ CHAR(10) FOR BIT DATA NOT NULL • IBMSNAP_OPERATION CHAR(1) NOT NULL • IBMSNAP_COMMITSEQ CHAR(10) FOR BIT DATA NOT NULL • IBMSNAP_LOGMARKER TIMESTAMP NOT NULL

表 40. ADDDPRREG コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
REFRESH	<p>フル・リフレッシュ機能を使用可能にするかどうかを指定します。この値を使用して、ソース・データベースからのフル・リフレッシュを実行するアプライ・プログラムの機能をオフにすることができます。</p> <p>*YES (デフォルト) フル・リフレッシュを使用可能にします。</p> <p>*NO フル・リフレッシュは使用不可です。 ターゲット表が基礎集約表または変更集約表の場合、このパラメーターは *NO にする必要があります。</p>
TEXT	<p>この登録に関する記述テキストを指定します。</p> <p>*NONE (デフォルト) 項目には記述がありません。</p> <p><i>description</i> この登録を記述するテキスト。最大 50 文字を入力でき、テキストは単一引用符で囲む必要があります。</p>
CAPCOL	<p>この登録済み表について、どの列の変更をキャプチャーするかを指定します。</p> <p>*ALL (デフォルト) すべての列の変更をキャプチャーします。</p> <p>*NONE この表の変更をキャプチャーしません。この値は、この表をフル・リフレッシュ用としてのみ登録する場合に指定します。登録されたこの表については変更データ (CD) 表は作成されず、キャプチャー・プログラムは表の変更をキャプチャーしません。</p> <p><i>column-name</i> その列の変更をキャプチャーする列名。列名は 300 個まで指定できます。列名はスペースで区切ります。</p>
CAPRRN	<p>変更されたレコードの相対レコード番号 (RRN) をキャプチャーするかどうかを指定します。</p> <p>*NO (デフォルト) 相対レコード番号をキャプチャーしません。</p> <p>*YES 相対レコード番号をキャプチャーします。変更データ (CD) 表に IBMQSQ_RRN と呼ばれる追加の列が作成されます。 ソース表にユニーク・キーがない場合のみ、このパラメーターを *YES にします。</p>

表 40. ADDDPRREG コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
IMAGE	<p>変更データ (CD) 表に、ソース表の変更前イメージと変更後イメージの両方を含めるかどうかを指定します。この値は、キャプチャー列パラメーター (CAPCOL) に指定したすべての列にグローバルに適用されます。</p> <p>この IMAGE パラメーターは、CAPCOL パラメーターが *NONE の場合は無効です。</p> <p>このパラメーターに *AFTER を指定した場合でも、ソース表は *BOTH イメージでジャーナルに記録する必要があります。</p> <p>*AFTER (デフォルト) キャプチャー・プログラムはソース表の変更後イメージのみ CD 表に記録します。</p> <p>*BOTH キャプチャー・プログラムはソース表の変更前イメージと変更後イメージの両方を CD 表に記録します。</p>
PREFIX	<p>変更データ (CD) 表の変更前イメージの列を識別する接頭部文字を指定します。ソース表の登録された列名に、この接頭部文字で始まる列名がないことを確認してください。</p> <p>*DEFAULT (デフォルト) デフォルトの接頭部 (@) が使用されます。</p> <p>*NULL 変更前イメージはキャプチャーされません。この値は、IMAGE パラメーターを *BOTH にした場合は無効です。</p> <p><i>character</i> オブジェクト名に使用できる任意の 1 つの英字。</p>
CONDENSED	<p>ソース表が圧縮されているかどうかを指定します。コンデンス表には、現行データが含まれ、1 つの主キーの値に対して複数の行が含まれることはありません。</p> <p>*YES (デフォルト) ソース表は圧縮されています。</p> <p>*NO ソース表は圧縮されていません。</p> <p>*AGGREGATE ソース表のタイプは *BASEAGR (基本集約) または *CHANGEAGR (変更集約) のいずれかです。この値を使用する場合、COMPLETE パラメーターを *NO にする必要があります。</p>
COMPLETE	<p>ソース表が完全であるかどうかを指定します。これは、表に、対象となる主キー値ごとに 1 行が含まれていることを意味します。</p> <p>*YES (デフォルト) ソース表は完全です。</p> <p>*NO ソース表は完全ではありません。</p>

表 40. ADDDPRREG コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
SRCTBLRDB	<p>リモート・ジャーナリングを使用するかどうかを指定します。リモート・ジャーナリングでは、ソース表とリモート・ジャーナルは別のシステムにあります。このパラメーターを使用して、ソース表のロケーションを指定します。</p> <p>*LOCAL (デフォルト) ソース表はローカルにあります (ADDDPRREG コマンドを実行するマシン上に)。</p> <p><i>rdname</i> ソース表が存在するリレーショナル・データベースの名前。 RDB ディレクトリー項目の作業 (WRKRDBDIRE) コマンドを使用して、このリレーショナル・データベース名を見つけることができます。</p>
RMTJRN	<p>このジャーナルの名前とソース・システム上のジャーナルの名前が異なる場合に、リモート・ジャーナルの名前を指定します。このコマンドは、リモート・ジャーナルが存在するシステムから発行する必要があります。</p> <p>*SRCTBL (デフォルト) リモート・ジャーナル名は、ソース表のジャーナル名と同じです。</p> <p><i>library-name/journal-name</i> このシステムにあり、リモート・ソース表のジャーナリングに使用される、修飾されたライブラリー名とジャーナル名。</p> <p>リモート・ジャーナル名は、SRCTBLRDB パラメーターでリモート・ソース表のロケーションを指定した場合のみ指定できます。</p>
CONFLICT	<p>レプリカ・サブスクリプション内の競合を検出するときにアプライ・プログラムが使用する、競合レベルを指定します。</p> <p>*NONE (デフォルト) 競合検出を行いません。</p> <p>*STANDARD 適度な競合検出。アプライ・プログラムは、レプリカ変更データ (CD) 表内のすでにキャプチャーされた行について、競合を探します。</p> <p>*ENHANCED 拡張競合検出を行います。このオプションは、すべてのレプリカとソース表の間に最良のデータ保全性を提供します。</p>
UPDDELINS	<p>キャプチャー・プログラムが、更新されたソース・データを、変更データ (CD) 表にどのように保管するかを決めます。</p> <p>*NO (デフォルト) キャプチャー・プログラムは、それぞれのソース変更を CD 表内の 1 つの行に保管します。</p> <p>*YES キャプチャー・プログラムは、それぞれのソース変更を CD 表内に 2 行を使用して保管し、その 1 行は削除用、もう 1 行は挿入用です。アプライ・プログラムは最初に削除用の行を処理し、2 番目に挿入用の行を処理します。</p>

表 40. ADDDPRREG コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
GENCDROW	<p>キャプチャー・プログラムがソース表のすべての行から変更をキャプチャーするかどうかを指定します。</p> <p>*ALLCHG (デフォルト) キャプチャー・プログラムはソース表のすべての行から変更をキャプチャーし (登録されていない列の変更も含む)、これらの変更を変更データ (CD) 表に追加します。</p> <p>*REGCOLCHG キャプチャー・プログラムは、登録された列に変更があった場合のみ変更をキャプチャーします。その後、キャプチャー・プログラムはこれらの行を CD 表に追加します。</p> <p>*REGCOLCHG は、CAPCOL パラメーターが *ALL または *NONE の場合には指定できません。</p>
RECAP	<p>アプライ・プログラムが行った変更を、キャプチャー・プログラムが再度キャプチャーするかどうかを指定します。</p> <p>*YES (デフォルト) アプライ・プログラムがソース表を変更した場合に、その変更をキャプチャーし、変更データ (CD) 表に入れます。</p> <p>*NO アプライ・プログラムがソース表を変更した場合に、その変更はキャプチャーせず、したがって、変更データ (CD) 表には入れません。このオプションは、REPLICA タイプの表を登録する場合に使用してください。</p>
STOPONERR	<p>キャプチャー・プログラムがエラーを検出した場合に、キャプチャー・プログラムを停止するかどうかを指定します。¹</p> <p>*NO (デフォルト) キャプチャー・プログラムはエラーを検出しても停止しません。キャプチャー・プログラムはメッセージを出し、エラーの原因となった登録を非活動化してから、処理を継続します。</p> <p>*YES キャプチャー・プログラムはエラーを検出した場合に、メッセージを出してから停止します。</p>

表 40. ADDDPRREG コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
注:	
1.	このパラメーターを Yes (Y) に設定すると、他のジャーナル・ジョブが実行を継続している間、キャプチャー・ジャーナル・ジョブは停止します。このパラメーターを No (N) に設定すると、キャプチャー・プログラムはエラーを含む登録ファイルを停止します。
	さらに、このパラメーターは登録表の行に列を設定します。 STATE 列が 'S' に設定され、STATE_INFO 列は 200Axxxx に設定されます (xxxx は理由コード)。登録の設定をアクション ('A') 状態に戻すには、以下のステップを実行します。
	<ul style="list-style-type: none"> • ASN200A メッセージを訂正します。修正されたアクションについては、該当する OS/400 資料を参照してください。 • レプリケーション・センターまたは OS/400 コマンド STRSQL を使用して、IBMSNAP_REGISTER 表の行に列を設定します。 STATE 列を 'A' に設定し、STATE_INFO 列を NULL に設定します。 • キャプチャーを実行中の場合は、 INZDPRCAP コマンドを出して、そのジャーナルのデータ・レプリケーションを再初期化します。

ADDDPRREG の例

以下の例は、ADDDPRREG コマンドの使用法を示しています。

例 1

デフォルトのキャプチャー・スキーマの下に、HR ライブラリーから EMPLOYEE という名前のソース表を登録します。

```
ADDDPRREG SRCTBL(HR/EMPLOYEE)
```

例 2

BSN キャプチャー・スキーマの下に、HR ライブラリーから EMPLOYEE という名前のソース表を登録し、HRCDLIB ライブラリーの下に CDEMPLOYEE という名前の CD 表を作成します。

```
ADDDPRREG SRCTBL(HR/EMPLOYEE) CAPCTLLIB(BSN) CDLIB(HRCDLIB) CDNAME(CDEMPLOYEE)
```

例 3

BSN キャプチャー・スキーマの下に、DEPT ライブラリーから SALES という名前のポイント・イン・タイムのソース・タイプ付きソース表を登録します。

```
ADDDPRREG SRCTBL(DEPT/SALES) CAPCTLLIB(BSN) SRCTYPE(*POINTINTIME)
```

例 4

DEPT ライブラリーから SALES という名前のソース表を登録し、CD 表には、ソース表の変更について変更前イメージと変更後イメージの両方を含めます。

```
ADDDPRREG SRCTBL(DEPT/SALES) IMAGE(*BOTH)
```

例 5

DEPT ライブラリーから SALES という名前のソース表 (RMTRDB1 という名前のリレーショナル・データベースで、リモート・ジャーナルを使用する) を登録します。

ADDDPRREG SRCTBL(DEPT/SALES) SRCTBLRDB(RMTRDB1) RMTJRN(RMTJRNLIB/RMTJRN)

例 6

HR ライブラリーから EMPLOYEE ソース表を登録し、EMPNO、NAME、 DEPT、および NETPAY の列についてのみ、変更をキャプチャーします。

ADDDPRREG SRCTBL(HR/EMPLOYEE) CAPCOL(EMPNO NAME DEPT NETPAY)

関連タスク:

- 43 ページの『第 3 章 表およびビューを SQL レプリケーション・ソースとして登録する』

ADDDPRSUB: DPR サブスクリプション・セットの追加 (OS/400)

1 つのメンバーを持つ、またはメンバーを持たないサブスクリプション・セットを作成するには、DPR サブスクリプション・セットの追加 (ADDDPRSUB) コマンドを使用します。

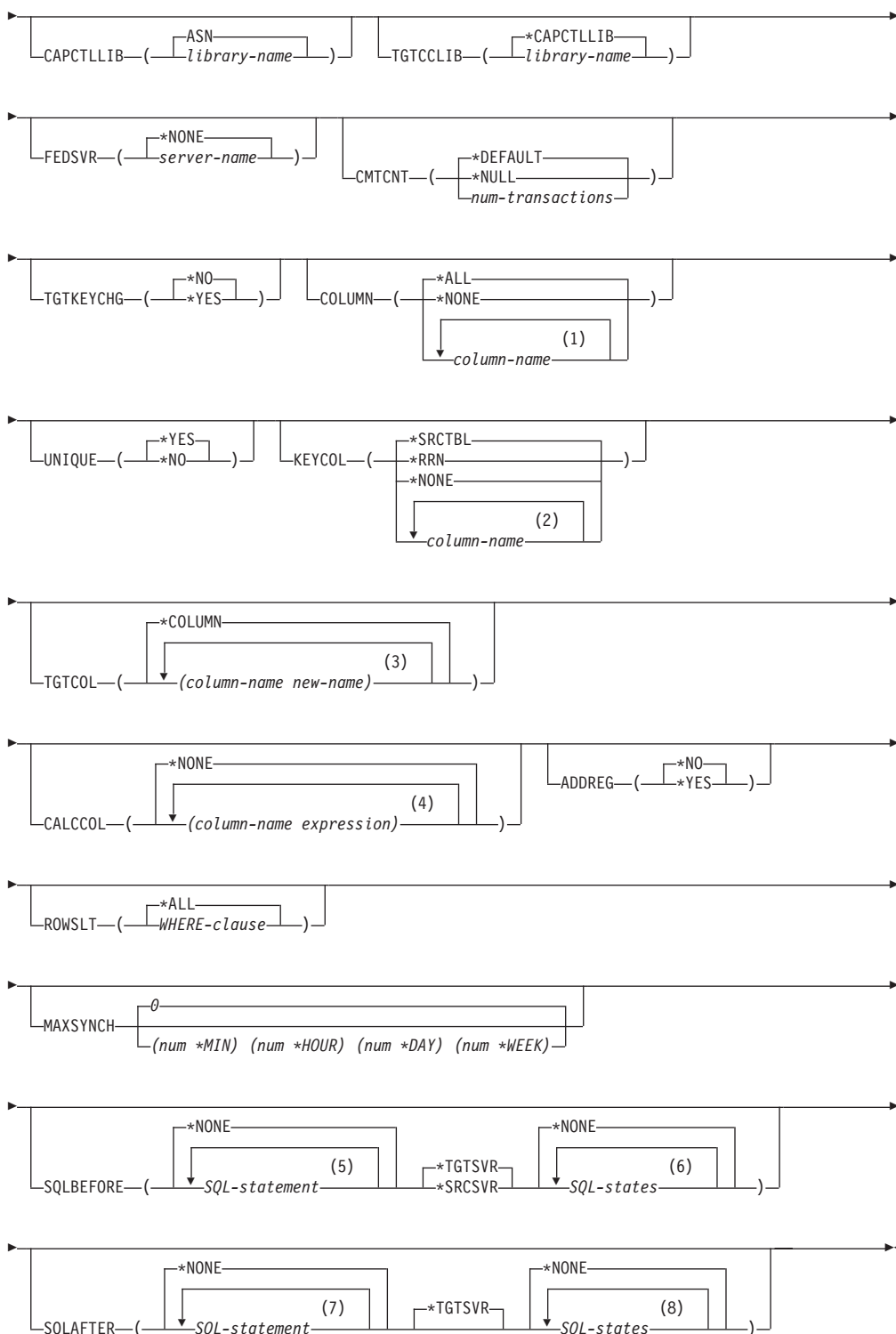
コマンド行にコマンド名を入力してから、F4 キーを押してコマンド構文を表示することができます。

このコマンド、およびこのコマンドのすべてのパラメーターの詳しい記述を表示するには、画面の一番上のコマンドにカーソルを移動し、F1 キーを押します。特定のパラメーターの記述を表示するには、そのパラメーター上にカーソルを移動し、F1 キーを押します。

ADDDPRSUB コマンドを使用してサブスクリプション・セットを作成するための構文



ADDDPRSUB



注:

- 1 列名は 300 個まで指定できます。
- 2 列名は 120 個まで指定できます。
- 3 列名は 300 個まで指定できます。
- 4 100 個までの列名および式を指定できます。

- 5 SQL ステートメントは 3 個まで指定できます。
- 6 10 個までの SQLSTATES を指定できます。
- 7 SQL ステートメントは 3 個まで指定できます。
- 8 10 個までの SQLSTATES を指定できます。

表 41 では、呼び出しパラメーターをリストしています。

表 41. ADDDPRSUB コマンド・パラメーター定義 (OS/400 版)

パラメーター	定義およびプロンプト
APYQUAL	<p>どのアプライ・プログラムがこのサブスクリプション・セットを処理するかを示すアプライ修飾子を指定します。アプライ修飾子の下のサブスクリプション・セットは別のジョブで実行されます。このパラメーターは必須です。</p> <p><i>apply-qualifier</i> アプライ修飾子の名前。</p>
SETNAME	<p>サブスクリプション・セットの名前を指定します。このパラメーターは必須です。</p> <p><i>set-name</i> サブスクリプション・セットの名前。入力するサブスクリプション・セット名は、指定されたアプライ修飾子についてユニークでなければなりません。ユニークでない場合、ADDDPRSUB コマンドはエラーになります。アプライ・プログラムはターゲット表のセットをグループとして扱うので、何らかの理由で 1 つのターゲット表が失敗すると、サブスクリプション・セット全体が失敗します。</p>
SRCTBL	<p>情報をサブスクリプション・セットにコピーするために使用する、ソース表の名前を指定します。この表をサブスクリプション・セットのメンバーにするには、この表をキャプチャー・コントロール・サーバーにあらかじめ登録しておく必要があります。このパラメーターは必須です。</p> <p>*NONE (デフォルト) このサブスクリプション・セットはソース・メンバーを持ちません。メンバーのないサブスクリプション・セットを作成する場合に使用します。</p> <p><i>library-name/file-name</i> ソース表の修飾名。1 つのメンバーを持つサブスクリプション・セットを作成する場合に使用します。</p>
TGTTBL	<p>ターゲット表の名前を指定します。CRITGTTBL パラメーターを *YES に設定し、ターゲット表が存在しない場合、ターゲット表は自動的に作成されます。このパラメーターは必須です。</p> <p>*NONE (デフォルト) このサブスクリプション・セットはターゲット・メンバーを持ちません。メンバーのないサブスクリプション・セットを作成する場合に使用します。</p> <p><i>library-name/file-name</i> ターゲット表の修飾名。1 つのメンバーを持つサブスクリプション・セットを作成する場合に使用します。</p>

表 41. ADDDPRSUB コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
CTLSVR	<p>アプライ・コントロール表を含むシステムのリレーショナル・データベース名を指定します。</p> <p>*LOCAL (デフォルト) アプライ・コントロール表はローカルに存在します (ADDDPRSUB コマンドを実行するマシン上)。</p> <p><i>rdb-name</i> アプライ・コントロール表が置かれているリレーショナル・データベースの名前。RDB ディレクトリー項目の作業 (WRKRDBDIRE) コマンドを使用して、この名前を検出することができます。</p>
SRCSVR	<p>キャプチャー・コントロール表を含むシステムのリレーショナル・データベース名を指定します。</p> <p>*LOCAL (デフォルト) ソース表はローカル・マシンに登録されています (ADDDPRSUB コマンドを実行するマシン)。</p> <p><i>rdb-name</i> キャプチャー・コントロール表が存在するリレーショナル・データベースの名前。RDB ディレクトリー項目の作業 (WRKRDBDIRE) コマンドを使用して、この名前を検出することができます。</p>
TGTTYPE	<p>ターゲット表のタイプを指定します。これらのいずれかのタイプとしてターゲット表を作成した後、DPR 登録の追加 (ADDDPRREG) コマンドの SRCTBL パラメーターにこのパラメーター値を使用し、このターゲット表を multi-tier レプリケーションのソース表として登録することができます。</p> <p>*USERCOPY (デフォルト) ターゲット表はユーザー・コピーであり、これはソース表の内容のすべてまたは一部と一致する内容を持つ、ターゲット表です。ユーザー・コピーはポイント・イン・タイムのコピーのように扱われますが、ポイント・イン・タイムのターゲット表に存在する、DB2 DataPropagator for iSeries システム列は 1 つも含まれていません。</p> <p>この値は、KEYCOL パラメーターに *RRN の値が指定されている場合は無効です。</p> <p>SRCTBL パラメーターで指定した表は、ユーザー・データベース、ポイント・イン・タイム・コピー、または整合変更データ (CCD) のいずれかでなければなりません。</p> <p>重要: ターゲット表がすでに存在する場合、DB2 DataPropagator for iSeries は、これに対する変更を自動的にジャーナルに記録しません。ジャーナリングは、DB2 DataPropagator for iSeries の外側で開始する必要があります。</p>

表 41. ADDDPRSUB コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
TGTTYPE (続き)	<p>*POINTINTIME</p> <p>ターゲット表はポイント・イン・タイム・コピーです。ポイント・イン・タイム・コピーは、ソース表の内容の一部またはすべてと一致する内容を持つターゲット表であり、DB2 DataPropagator for iSeries システム列 (IBMSNAP_LOGMARKER) を保持します。この列は、特定の行がキャプチャー・コントロール・サーバーでいつ挿入または変更されたかを示します。</p> <p>*BASEAGR</p> <p>ターゲット表は基本集約コピーであり、これはソース表から集約された (算出された) データを含む、ターゲット表です。基本集約ターゲットのソース表は、ユーザー表またはポイント・イン・タイム表のどちらかでなければなりません。このターゲット表には、システム・タイム・スタンプ列の IBMSNAP_HLOGMARKER および IBMSNAP_LLOGMARKER が含まれます。</p> <p>*CHANGEAGR</p> <p>表は変更集約コピーであり、これは、変更データ (CD) 表の内容を基に集約された (算出された) データを含むターゲット表です。このターゲット表は、システム・タイム・スタンプ列 IBMSNAP_HLOGMARKER および IBMSNAP_LLOGMARKER を使用して作成されます。</p> <p>*CCD</p> <p>表は整合変更データ (CCD) 表であり、これは、変更データ (CD) 表と作業単位 (UOW) 表内のデータを結合したものから作成されたターゲット表です。CCD 表は、アプライ・プログラムにトランザクション整合性のあるデータを提供し、次の列を含む必要があります。</p> <ul style="list-style-type: none"> • IBMSNAP_INTENTSEQ • IBMSNAP_OPERATION • IBMSNAP_COMMITSEQ • IBMSNAP_LOGMARKER <p>*REPLICA</p> <p>ターゲット表はレプリカ表であり、これは Update-anywhere レプリケーションにのみ使用されます。レプリカ・ターゲット表はマスター・ソース表から変更を受信し、またレプリカ・ターゲット表への変更は、マスター・ソース表に戻して伝搬されます。レプリカ表はソース表として自動的に登録されます。</p>

表 41. ADDDPRSUB コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
TIMING	<p>アプライ・プログラムがサブスクリプション・セットの処理に使用するタイミング (スケジューリング) のタイプを指定します。</p> <p>*INTERVAL (デフォルト) アプライ・プログラムは、サブスクリプション・セットを特定の時間間隔 (たとえば、1 日に 1 度) で処理します。</p> <p>*EVENT アプライ・プログラムは、特定のイベントが起こった時にサブスクリプション・セットを処理します。</p> <p>*BOTH アプライ・プログラムは、特定の時間インターバル、またはイベントが起こった時のどちらでも、最初に起こった時にサブスクリプション・セットを処理します。</p>
EVENT	<p>イベントを指定します。入力するイベントは、サブスクリプション・イベント表 (IBMSNAP_SUBS_EVENT) のイベント名と一致する必要があります。</p> <p>*NONE (デフォルト) イベントは使用しません。</p> <p><i>event-name</i> IBMSNAP_SUBS_EVENT 表に記述されているイベントを表す、ユニークな文字ストリング。</p>
INTERVAL	<p>ターゲット・コピーのリフレッシュを行う時間間隔を指定します。これは開始時刻から開始時刻までの時間間隔であり、週、日、時間、および分で指定します。これは 2 つの部分からなる値です。最初の部分は数値、2 番目の部分は時間の単位です。</p> <p>*MIN 分</p> <p>*HOUR 時間</p> <p>*DAY 日</p> <p>*WEEK 週</p> <p>時間の単位と一緒に、数値の組み合わせを指定することができます。たとえば、((2 *WEEK) (3 *DAY) (35 *MIN)) は、2 週、3 日、および 35 分の時間間隔を指定します。同じ時間単位で複数を指定すると、最後の指定が使用されます。</p>
ACTIVATE	<p>サブスクリプション・セットがアクティブかどうかを指定します。アプライ・プログラムは、このパラメーターが *YES でない場合、このサブスクリプション・セットを処理しません。</p> <p>*YES (デフォルト) サブスクリプション・セットはアクティブです。</p> <p>*NO サブスクリプション・セットはアクティブではありません。</p>

表 41. ADDDPRSUB コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
CRTTGTBL	<p>ターゲット表 (またはビュー) を作成するかどうかを指定します。</p> <p>*YES (デフォルト) ターゲット表 (またはビュー) が存在しなければ、作成します。存在する場合は、既存の表またはビューがターゲットになり、この既存の表またはビューのフォーマットの CHKFMT パラメーターが *YES に設定されているかどうかをチェックします。 UNIQUE および KEYCOL パラメーターに指定された値を使用して、(そのような索引がまだ存在しなければ) ターゲット表に追加の索引が作成されます。既存のターゲット表に、追加索引の条件に違反するような行が含まれていると、コマンドは失敗します。</p> <p>*NO ターゲット表またはビューを作成しません。アプライ・プログラムを始動する前に、正しい属性を使用して表またはビューを作成する必要があります。</p> <p>表またはビューが存在する場合に CHKFMT を *YES にすると、ADDDPRSUB コマンドは、既存の表のフォーマットが、設定されたサブスクリプション・セット定義と一致することを確認します。 CHKFMT を *NO にする場合は、ユーザーは既存の表のフォーマットがサブスクリプション・セット定義と一致することを確認しておく必要があります。</p> <p>重要: 表またはビューがすでに存在する場合、DB2 DataPropagator for iSeries は、既存のオブジェクトへの変更を自動的にジャーナルに記録しません。ジャーナリングは、DB2 DataPropagator for iSeries の外側で開始する必要があります。</p>
CHKFMT	<p>DB2 DataPropagator for iSeries が、サブスクリプション・セットとターゲット表をチェックし、列が一致することを確認するかどうかを指定します。このパラメーターは、CRTTGTBL パラメーターが *YES の場合は無視され、また CRTTGTBL パラメーターが *NO でターゲット表が存在しない場合も無視されます。</p> <p>*YES (デフォルト) DB2 DataPropagator for iSeries は、このサブスクリプション・セットに定義された列がターゲット表内の列と一致するか検証します。両者が一致しない場合、このコマンドは失敗します。</p> <p>*NO DB2 DataPropagator for iSeries は、サブスクリプション・セットと既存のターゲット表間の相違を無視します。ユーザーはターゲット表がサブスクリプション・セットと互換性があることを確認する必要があります。</p>
CAPCTLLIB	<p>キャプチャー・スキーマ (キャプチャー・コントロール表が存在するライブラリーの名前) を指定します。これらのキャプチャー・コントロール表は、このサブスクリプション・セットのソースを処理します。</p> <p>ASN (デフォルト) キャプチャー・コントロール表は ASN ライブラリー内にあります。</p> <p><i>library-name</i> キャプチャー・コントロール表を含むライブラリーの名前。これは、ソース表が登録されたライブラリーです。</p>

表 41. ADDDPRSUB コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
TGTCCLIB	<p>ターゲット・コントロール・ライブラリーを指定します。</p> <p>*CAPCTLLIB (デフォルト) ターゲット・コントロール・ライブラリーは、キャプチャー・コントロール表が存在するライブラリーと同じです。</p> <p><i>library-name</i> ターゲット・コントロール表を含むライブラリーの名前。</p> <p>ターゲット表を別のサブスクリプション・セット (たとえば外部 CCD 表など) のソースとして使用する場合、このパラメーター値は、この表がソースとして使用される時のキャプチャー・スキーマです。</p>
FEDSVR	<p>このサブスクリプション・セットのソースがフェデレーテッド・データベース・システムであるかどうかを指定します。</p> <p>*NONE (デフォルト) ソース・サーバーはフェデレーテッド・データベース・システムではありません。</p> <p><i>server-name</i> このサブスクリプション・セットのフェデレーテッド・データベース・システムの名前 (DB2 以外のリレーショナル・ソースの場合)。</p>
CMTCNT	<p>コミットメント・カウントを指定します。これは、アプライ・プログラムがトランザクションをいくつ処理したらコミットするかを示す数です。</p> <p>*DEFAULT (デフォルト) 使用する値をコマンドが決めます。 TGTTYPE が *REPLICA の場合、 CMTCNT はゼロ (0) です。 TGTTYPE が *REPLICA 以外の場合、 CMTCNT は NULL です。</p> <p>*NULL サブスクリプション・セットは読み取り専用です。アプライ・プログラムは、サブスクリプション・セット・メンバーの応答セットを 1 度に 1 メンバーずつフェッチし、すべてのデータの処理を終了した後、サブスクリプション・セット全体について 1 つのコミットを出します。</p> <p><i>num-transactions</i> いくつのトランザクション処理を処理したらアプライ・プログラムが変更をコミットするかを示す数を指定します。このパラメーターは、 TGTTYPE パラメーターが *REPLICA の場合のみ有効です。</p>

表 41. ADDDPRSUB コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
TGTKEYCHG	<p>ターゲット表のターゲット・キー列の一部であるソース列に変更があった場合、アプライ・プログラムがその更新をどのように扱うかを指定します。このパラメーターは、ADDDPRREG コマンドの USEDELINS パラメーターと組み合わせて働きます。</p> <ul style="list-style-type: none"> • USEDELINS が YES で TGTKEYCHG が YES の場合、更新はできません。 • USEDELINS が YES で TGTKEYCHG が NO の場合、更新は削除と挿入の対になります。 • USEDELINS が NO で TGTKEYCHG が YES の場合、アプライ・プログラムは特別な論理を使用してこの条件を扱います。 • USEDELINS が NO で TGTKEYCHG が NO の場合、アプライ・プログラムは変更を通常の更新として処理します。 <p>*NO (デフォルト) ソース表に対する更新は、キャプチャー・プログラムによりステージ化され、アプライ・プログラムによりターゲット表に処理が行われず。</p> <p>*YES アプライ・プログラムは、ターゲット・キー列の変更前イメージに基づいてターゲット表を更新します。つまり、アプライ・プログラムは述部を新しい値ではなく、古い値に変更します。</p>
COLUMN	<p>ターゲット表に含める列を指定します。列名は修飾できません。列名は、ソース表の登録時に CAPCOL パラメーターに指定した列名のリストから選択してください。</p> <p>この表の登録時に IMAGE パラメーターを *BOTH に設定した場合は、変更前イメージ列名を指定することができます。変更前イメージ列名は、接頭部を持つオリジナルの列名です。この接頭部は、ADDDPRREG コマンドの PREFIX パラメーターに指定した文字です。</p> <p>*ALL (デフォルト) ソースに登録した列のすべてがターゲット表に含まれます。</p> <p>*NONE ソース表からの列は 1 つもターゲット表に含まれません。 *NONE は、算出列だけをターゲット表に含めたい場合に使用します。この値は、CALCCOL パラメーターに合計関数が含まれているが、GROUP BY が実行されない、という場合に必要です。</p> <p><i>column-name</i> ターゲット表に含めたいソース列の名前を 300 個まで指定できます。列名はスペースで区切ります。</p>

表 41. ADDDPRSUB コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
UNIQUE	<p>ターゲット表が KEYCOL パラメーターで示されたユニーク・キーを持つかどうかを指定します。</p> <p>*YES (デフォルト) ターゲット表はキーごとに正味 1 つの変更をサポートします。つまり、キーに対していかに多くの変更がなされたとしても、そのキーについてはターゲット表に 1 つの行しか存在しないということです。</p> <p>この値は、表がデータの変更の履歴ではなく、現行のデータを含むことを指定します。コンデンス表には、1 つの主キー値に対して複数の行が含まれることはなく、リフレッシュ用の最新情報を提供するために使用できます。</p> <p>*NO ターゲット表はキーごとに複数の変更をサポートします。変更はターゲット表に付加されます。</p> <p>この値は、表が現行のデータではなく、変更の履歴を含むことを指定します。非コンデンス表には、それぞれのキー値に対して複数の行が含まれ、データの変更履歴を提供するために使用できます。ただし、非コンデンス表はリフレッシュ用の最新データは提供できません。</p>

表 41. ADDDPRSUB コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
KEYCOL	<p>ターゲット表のキーを記述する列を指定します。列名は修飾できません。 *POINTINTIME、*REPLICA、および *USERCOPY ターゲット表 (TGTYPE パラメーターで指定されている) の場合、ターゲット表に 1 つまたは複数の列をターゲット・キーとして指定する必要があります。アプライ・プログラムはこのターゲット・キーを使用して、変更キャプチャー・レプリケーション中に、変更された個々のユニークな行を識別します。</p> <p>*SRCTBL (デフォルト) ターゲット表のキー列は、ソース表のキー列と同じです。 ADDDPRREG コマンドは、ソース表がキー付きの場合、ソース表に指定されたキーを使用します。以下のキー列が使用されます。</p> <ul style="list-style-type: none"> 物理ファイル作成コマンド (CRTPF) を使用して表を作成した時に、DDS を使用して定義したキー列 CREATE TABLE および ALTER TABLE SQL ステートメントを使用して定義した、主キーおよびユニーク・キー CREATE INDEX SQL ステートメントを使用して定義したユニーク・キー <p>1 つの列を、キーとして、異なる順序付けで複数回使用すると、ターゲット表のキーは昇順で定義されます。</p> <p>*RRN ターゲット表のキー列は IBMQSQ_RRN 列です。ターゲット表は IBMQSQ_RRN 列を使用して作成され、この列がキーとして使用されます。アプライ・プログラムの実行時に、ソース表がユーザー表であり、ターゲット表がポイント・イン・タイムまたはユーザー・コピーの場合、ターゲット表の IBMQSQ_RRN 列が、ソース表内の関連するレコードの RRN (相対レコード番号) で更新されます。それ以外では、ターゲット表の IBMQSQ_RRN 列は、ソース表内の IBMQSQ_RRN 列の値で更新されます。</p> <p>*NONE ターゲット・コピーはターゲット・キーを含みません。ターゲット表のタイプが *POINTINTIME、*REPLICA、または *USERCOPY の場合、*NONE は指定できません。</p> <p><i>column-name</i> ターゲット・キー列として使用したいターゲット列の名前。列名は 120 個まで指定できます。列名はスペースで区切ります。</p>

表 41. ADDDPRSUB コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
TGTCOL	<p>アプライン・プログラムがターゲット表内で更新するすべての列の新しい名前を指定します。これらの名前は、ソース表から取られた列名をオーバーライドします。列名は修飾できません。 COLUMN パラメーターに *NONE を指定した場合はこのパラメーターを使用しないでください。</p> <p>このパラメーターを使用して、ターゲット表の列により分かりやすい名前を付けることができます。それぞれのソース列の名前と、ターゲット表の対応する列の名前を指定します。</p> <p>*COLUMN (デフォルト) ターゲット列は、COLUMN パラメーターに指定した列と同じです。</p> <p><i>column-name</i> ターゲットではその名前を変更したい、ソース表からの列名。列名は 300 個まで指定できます。</p> <p><i>new-name</i> ターゲット列の新しい名前。新しい列名を 300 個まで指定できます。このパラメーターを使用しない場合、ターゲット表の列名はソースの列名と同じになります。</p>
CALCCOL	<p>ターゲット表のユーザー定義の列または算出された列のリストを指定します。列名は修飾できません。それぞれの列名と式の対を括弧で囲みます。</p> <p>各 SQL 式には列名を指定する必要があります。GROUP BY ステートメントのない SQL 式として列を定義する場合は、COLUMN パラメーターを *NONE にする必要があります。</p> <p>*NONE (デフォルト) ターゲット表はユーザー定義の列または算出された列を含みません。</p> <p><i>column-name</i> ターゲット表のユーザー定義の列または算出された列の列名。列名は 100 個まで指定できます。</p> <p><i>expression</i> ターゲット表のユーザー定義の列または算出された列の式。SQL 列の式は 100 個まで指定できます。</p>

表 41. ADDDPRSUB コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
ADDREG	<p>ターゲット表をソース表として自動的に登録するかどうかを指定します。このパラメーターは CCD ターゲット・タイプの表を登録する場合に使用します。</p> <p>*NO (デフォルト) ターゲット表はソース表として登録されません。 DB2 DataPropagator for iSeries は、ターゲット・タイプが *REPLICA の場合、このパラメーター値を無視します。レプリカ・ターゲット表は必ず、ソース表として自動的に登録されます。</p> <p>*YES ターゲット表はソース表として登録されます。ターゲット表をすでにユーザーが登録していると、このコマンドは失敗します。</p> <p>ターゲット表のタイプが *USERCOPY、*POINTINTIME、*BASEAGR、または *CHANGEAGR の場合、このパラメーターを *YES にしないでください。</p> <p>CRTTGTTBL パラメーターを *NO にした場合、これをソースとして登録する前に、ターゲット表を作成する必要があります。</p>
ROWSLT	<p>SQL の WHERE 文節に入れる述部を指定します。アプライ・プログラムはこれらの述部を使用して、ソースの変更データ (CD) 表のどの行をターゲット表に適用するかを決めます。ソースの変更のサブセットだけをターゲット表に複製したい場合に、このパラメーターを使用します。</p> <p>*ALL (デフォルト) アプライ・プログラムは、CD 表内のすべての変更をターゲット表に適用します。</p> <p><i>WHERE-clause</i> アプライ・プログラムが CD 表からどの行をターゲット表に適用するかを指定する SQL の WHERE 文節です。WHERE キーワードはこのパラメーターで暗黙に想定されているので、WHERE キーワードを含めないでください。この WHERE 文節は、この文節を実行するデータ・サーバー上で有効なものでなければなりません。</p> <p>注: このパラメーターの WHERE 文節は、SQLBEFORE または SQLAFTER パラメーターに指定された WHERE 文節とはまったく関係ありません。</p>

表 41. ADDDPRSUB コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
MAXSYNCH	<p>最大の同期化の分数を指定します。このパラメーターは、時間しきい値の制限を指定し、サブスクリプションのサイクル中に、キャプチャー・プログラムおよびアプライ・プログラムが処理する変更データの量を規制するために使用されます。時間しきい値の制限は 2 つの部分からなる値で指定します。最初の部分は数値、2 番目の部分は時間の単位です。</p> <p>*MIN 分</p> <p>*HOUR 時間</p> <p>*DAY 日</p> <p>*WEEK 週</p> <p>時間の単位と一緒に、数値の組み合わせを指定することができます。たとえば、((1 *WEEK) (2 *DAY) (35 *MIN)) は、1 週、2 日、および 35 分の時間間隔を指定します。同じ時間単位で複数を指定すると、最後の指定が使用されます。</p> <p>デフォルトはゼロ (0) であり、変更データのすべてを適用することを示します。</p>

表 41. ADDDPRSUB コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
SQLBEFORE	<p>アプリ・プログラムがターゲット表をリフレッシュする前に実行する、SQL ステートメントを指定します。このパラメーターは以下の 3 つの要素からなります。</p> <p>エレメント 1: SQL コード</p> <p>*NONE (デフォルト) SQL ステートメントを指定しません。</p> <p><i>SQL-statement</i> 実行したい SQL ステートメント。SQL ステートメントの構文が正しいことを確認してください。DB2 DataPropagator for iSeries は構文の妥当性を検査しません。また、適切な SQL 命名規則を使用する必要があります。SQL ファイル参照は、システムの命名規則 (LIBRARY/FILE) ではなく、LIBRARY.FILE の形式でなければなりません。3 つまでの SQL ステートメントを指定できます。</p> <p>エレメント 2: 実行するサーバー</p> <p>*TGTSVR (デフォルト) SQL ステートメントは、ターゲット表が存在するターゲット・サーバーで実行されます。</p> <p>*SRCSVR SQL ステートメントは、ソース表が存在するキャプチャー・コントロール・サーバーで実行されます。</p> <p>エレメント 3: 許される SQLSTATE 値</p> <p>*NONE (デフォルト) SQLSTATE 値 00000 のみが正常と見なされます。</p> <p><i>SQL-states</i> 1 から 10 個までの許容される SQLSTATE 値のリスト。 SQLSTATE 値はスペースで区切ります。SQLSTATE 値は、00000 から FFFFF の範囲の 5 桁の 16 進数です。</p> <p>SQL ステートメントは、SQLSTATE 値 00000 または、リストされた許容可能な SQLSTATE 値の 1 つで完了した場合に、正常終了となります。</p>

表 41. ADDDPRSUB コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
SQLAFTER	<p>アプライ・プログラムがターゲット表をリフレッシュした後に実行する、SQL ステートメントを指定します。このパラメーターは以下の 3 つのエレメントからなります。</p> <p>エレメント 1: SQL コード</p> <p>*NONE (デフォルト) SQL ステートメントを指定しません。</p> <p><i>SQL-statement</i> 実行したい SQL ステートメント。SQL ステートメントの構文が正しいことを確認してください。DB2 DataPropagator for iSeries は構文の妥当性を検査しません。また、適切な SQL 命名規則を使用する必要があります。SQL ファイル参照は、システムの命名規則 (LIBRARY/FILE) ではなく、LIBRARY.FILE の形式でなければなりません。3 つまでの SQL ステートメントを指定できます。</p> <p>エレメント 2: 実行するサーバー</p> <p>*TGTSVR (デフォルト) SQL ステートメントは、ターゲット表が存在するターゲット・サーバーで実行されます。</p> <p>エレメント 3: 許される SQLSTATE 値</p> <p>*NONE (デフォルト) SQLSTATE 値 00000 のみが正常と見なされます。</p> <p><i>SQL-states</i> 1 から 10 個までの許容される SQLSTATE 値のリスト。 SQLSTATE 値はスペースで区切ります。SQLSTATE 値は、00000 から FFFFF の範囲の 5 桁の 16 進数です。</p> <p>SQL ステートメントは、SQLSTATE 値 00000 または、リストされた許容可能な SQLSTATE 値の 1 つで完了した場合に、正常終了となります。</p>

ADDDPRSUB の例

以下の例は、ADDDPRSUB コマンドの使用法を示しています。

例 1

AQHR アプライ修飾子の下に SETHR という名前のサブスクリプション・セットを作成します。

```
ADDDPRSUB APYQUAL(AQHR) SETNAME(SETHR) SRCTBL(HR/EMPLOYEE)
TGTTBL(TGTLIB/TGTEMPL)
```

このサブスクリプション・セットには 1 つのサブスクリプション・セット・メンバーが含まれ、HR ライブラリー下の EMPLOYEE という名前の登録済みソース表から、TGTLIB ライブラリー下の TGTEMPL という名前のターゲット表にデータを複製します。

例 2

2 つだけの列 (EMPNO (キー) および NAME) を持つ、SETHR という名前のサブスクリプション・セットを、EMPLOYEE という名前の登録済みソース表から作成し、これらの列を TGTEMPL という名前の既存のターゲット表に複製します。

```
ADDDPRSUB APYQUAL(AQHR) SETNAME(SETHR) SRCTBL(HR/EMPLOYEE)
      TGTTBL(TGTLIB/TGTEMPL) CRTTGTTBL(*NO) COLUMN(EMPNO NAME) KEYCOL(EMPNO)
```

例 3

SETHR という名前のサブスクリプション・セットを、EMPLOYEE という名前の登録済みソース表からのデータを使用して作成し、このデータを TGTREPL という名前のレプリカ・タイプのターゲット表に複製します。

```
ADDDPRSUB APYQUAL(AQHR) SETNAME(SETHR) SRCTBL(HR/EMPLOYEE)
      TGTTBL(TGTLIB/TGTREPL) TGTTYPE(*REPLICA)
```

例 4

NOMEM という名前のサブスクリプション・セットを、サブスクリプション・セット・メンバーなしで作成します。

```
ADDDPRSUB APYQUAL(AQHR) SETNAME(NOMEM) SRCTBL(*NONE) TGTTBL(*NONE)
```

関連タスク:

- 69 ページの『第 4 章 SQL レプリケーションのソースのサブスクライブ』

ADDDPRSUBM: DPR サブスクリプション・セットのメンバーの追加 (OS/400)

既存のサブスクリプション・セットにメンバーを追加するには、DPR サブスクリプション・セット・メンバーの追加 (ADDDPRSUBM) コマンドを使用します。サブスクリプション・セットは、ADDDPRSUB コマンドを使用して、UNIX、Windows、または z/OS 上のシステム・コマンドを使用して、またはレプリケーション・センターから作成することができます。サブスクリプション・セット内のすべてのソース表は、すでにジャーナルに記録済みであり、登録済みでなければならず、その後でないとこのコマンドは使用できません。

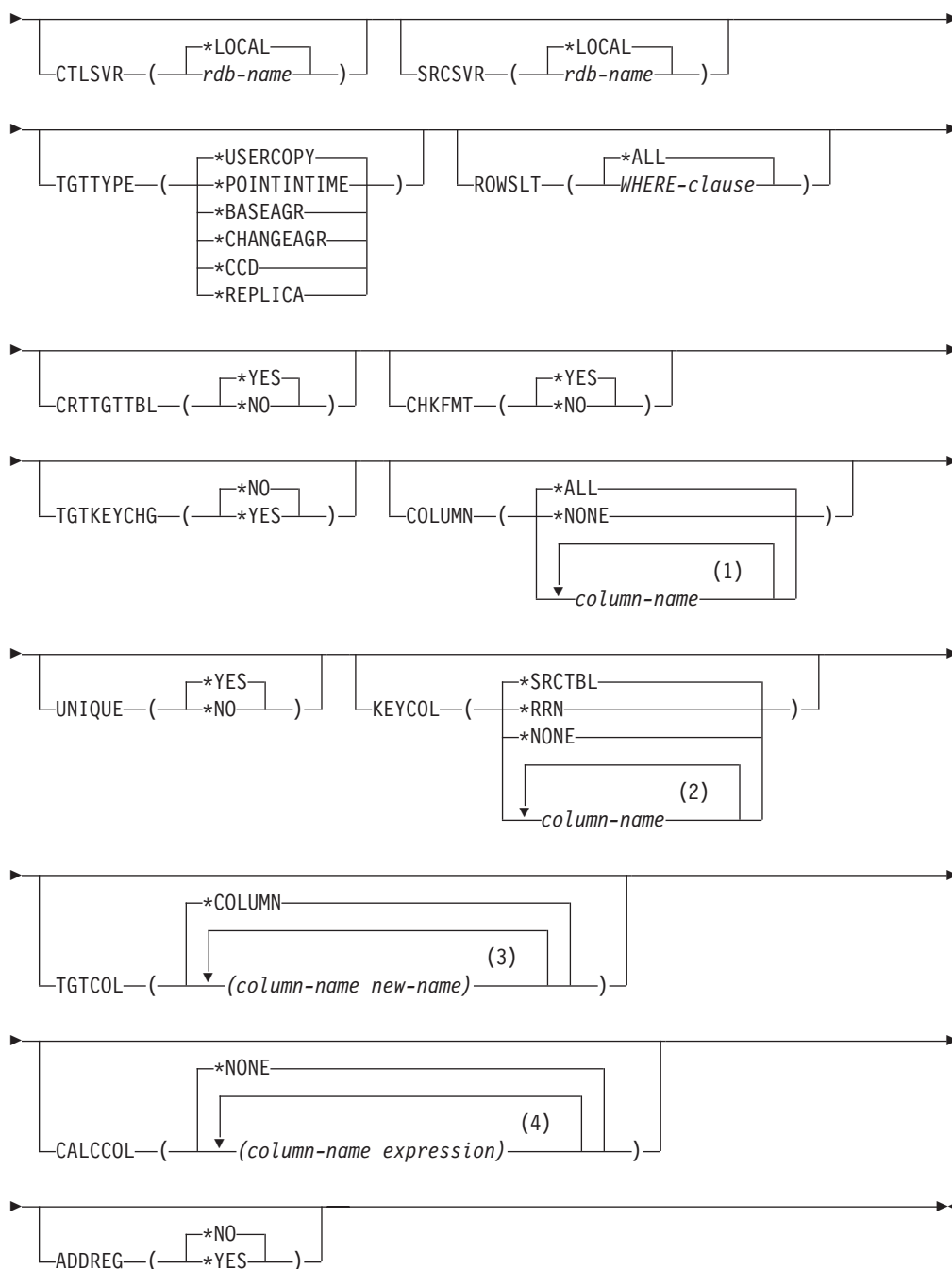
コマンド行にコマンド名を入力してから、F4 キーを押してコマンド構文を表示することができます。

このコマンド、およびこのコマンドのすべてのパラメーターの詳しい記述を表示するには、画面の一番上のコマンドにカーソルを移動し、F1 キーを押します。特定のパラメーターの記述を表示するには、そのパラメーター上にカーソルを移動し、F1 キーを押します。

ADDDPRSUBM コマンドを使用してサブスクリプション・セットにメンバーを追加するための構文

```
►►—ADDDPRSUBM—APYQUAL—(—apply-qualifier—)—SETNAME—(—set-name—)————►
►—SRCTBL—(—library-name/file-name—)—TGTTBL—(—library-name/file-name—)————►
```

ADDDPRSUBM



注:

- 1 列名は 300 個まで指定できます。
- 2 列名は 120 個まで指定できます。
- 3 列名は 300 個まで指定できます。
- 4 100 個までの列名および式を指定できます。

413 ページの表 42 では、呼び出しパラメーターをリストしています。

表 42. ADDDPRSUBM コマンド・パラメーター定義 (OS/400 版)

パラメーター	定義およびプロンプト
APYQUAL	<p>どのアプライ・プログラムがこのサブスクリプション・セットを処理するかを示すアプライ修飾子を指定します。アプライ修飾子の下のサブスクリプション・セットは別のジョブで実行されます。このパラメーターは必須です。</p> <p><i>apply-qualifier</i> アプライ修飾子の名前。</p>
SETNAME	<p>サブスクリプション・セットの名前を指定します。このパラメーターは必須です。</p> <p><i>set-name</i> サブスクリプション・セットの名前。入力するサブスクリプション・セット名は、指定されたアプライ修飾子についてユニークでなければなりません。ユニークでない場合、ADDDPRSUBM コマンドはエラーになります。アプライ・プログラムはターゲット表のセットをグループとして扱うので、何らかの理由で 1 つのターゲット表が失敗すると、そのセット全体が失敗します。</p>
SRCTBL	<p>このサブスクリプション・セット・メンバーのソースである表の名前を指定します。この表をサブスクリプション・セットのメンバーにするには、この表をキャプチャー・コントロール・サーバーにあらかじめ登録しておく必要があります。このパラメーターは必須です。</p> <p><i>library-name/file-name</i> ソース表の修飾名。</p>
TGTTBL	<p>このサブスクリプション・セット・メンバーのターゲット表の名前を指定します。CRITGTTBL パラメーターを *YES に設定し、ターゲット表が存在しない場合、ターゲット表は自動的に作成されます。このパラメーターは必須です。</p> <p><i>library-name/file-name</i> ターゲット表の修飾名。</p>
CTLSVR	<p>アプライ・コントロール表を含むシステムのリレーショナル・データベース名を指定します。</p> <p>*LOCAL (デフォルト) アプライ・コントロール表はローカルに存在します (ADDDPRSUBM コマンドを実行するマシン上)。</p> <p><i>rdb-name</i> アプライ・コントロール表が置かれているリレーショナル・データベースの名前。RDB ディレクトリー項目の作業 (WRKRDBDIRE) コマンドを使用して、この名前を検出することができます。</p>
SRCSVR	<p>キャプチャー・コントロール表を含むシステムのリレーショナル・データベース名を指定します。</p> <p>*LOCAL (デフォルト) ソース表はローカル・マシンに登録されています (ADDDPRSUBM コマンドを実行するマシン)。</p> <p><i>rdb-name</i> キャプチャー・コントロール表が存在するリレーショナル・データベースの名前。RDB ディレクトリー項目の作業 (WRKRDBDIRE) コマンドを使用して、この名前を検出することができます。</p>

表 42. ADDDPRSUBM コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
TGTTYPE	<p>ターゲット表のタイプを指定します。これらは、ターゲット表の内容を記述する DB2 レプリケーション用語です。これらのいずれかのタイプとしてターゲット表を作成した後、DPR 登録の追加 (ADDDPRREG) コマンドの SRCTBL パラメーターにこのパラメーター値を使用し、このターゲット表をソース表として登録することができます。</p> <p>*USERCOPY (デフォルト)</p> <p>ターゲット表はユーザー・コピーであり、これはソース表の内容のすべてまたは一部と一致する内容を持つ、ターゲット表です。ユーザー・コピーはポイント・イン・タイムの表のように扱われますが、ポイント・イン・タイムのターゲット表に存在する、DB2 DataPropagator for iSeries システム列は 1 つも含まれていません。</p> <p>この値は、KEYCOL パラメーターに *RRN の値が指定されている場合は無効です。</p> <p>SRCTBL パラメーターで指定した表は、ユーザー・データベース、ポイント・イン・タイム表、または整合変更データ (CCD) のいずれかでなければなりません。</p> <p>重要: ターゲット表がすでに存在する場合、DB2 DataPropagator for iSeries は、これに対する変更を自動的にジャーナルに記録しません。ジャーナリングは、DB2 DataPropagator for iSeries の外側で開始する必要があります。</p>

表 42. ADDDPRSUBM コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
TGTTYPE (続き)	<p>*POINTINTIME</p> <p>ターゲット表はポイント・イン・タイム表です。ポイント・イン・タイム表は、ソース表の内容の一部またはすべてと一致する内容を持つターゲット表であり、DB2 DataPropagator for iSeries システム列 (IBMSNAP_LOGMARKER) を保持します。この列は、特定の行がキャプチャー・コントロール・サーバーでいつ挿入または変更されたかを示します。</p> <p>*BASEAGR</p> <p>ターゲット表は基本集約表であり、これはソース表から集約された (算出された) データを含む、ターゲット表です。基本集約ターゲットのソース表は、ユーザー表またはポイント・イン・タイム表のどちらかでなければなりません。このターゲット表には、システム・タイム・スタンプ列の IBMSNAP_HLOGMARKER および IBMSNAP_LLOGMARKER が含まれます。</p> <p>*CHANGEAGR</p> <p>表は変更集約表であり、これは、変更データ (CD) 表の内容を基に集約された (算出された) データを含むターゲット表です。このターゲット表は、システム・タイム・スタンプ列 IBMSNAP_HLOGMARKER および IBMSNAP_LLOGMARKER を使用して作成されます。</p> <p>*CCD</p> <p>表は整合変更データ (CCD) 表であり、これは、変更データ (CD) 表と作業単位 (UOW) 表内のデータを結合したものから作成されたターゲット表です。CCD 表は、アプライ・プログラムにトランザクション整合性のあるデータを提供し、次の列を含む必要があります。</p> <ul style="list-style-type: none"> • IBMSNAP_INTENTSEQ • IBMSNAP_OPERATION • IBMSNAP_COMMITSEQ • IBMSNAP_LOGMARKER <p>*REPLICA</p> <p>ターゲット表はレプリカ表であり、これは Update-anywhere レプリケーションにのみ使用されます。レプリカ・ターゲット表はマスター・ソース表から変更を受信し、またレプリカ・ターゲット表への変更は、マスター・ソース表に戻して伝搬されます。レプリカ表はソース表として自動的に登録されます。</p>

表 42. ADDDPRSUBM コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
ROWSLT	<p>SQL の WHERE 文節に入れる述部を指定します。アプライ・プログラムはこれらの述部を使用して、ソースの変更データ (CD) 表のどの行をターゲット表に適用するかを決めます。ソースの変更のサブセットだけをターゲット表に複製したい場合に、このパラメーターを使用します。</p> <p>*ALL (デフォルト) アプライ・プログラムは、CD 表内のすべての変更をターゲット表に適用します。</p> <p><i>WHERE-clause</i> アプライ・プログラムが CD 表からどの行をターゲット表に適用するかを指定する SQL の WHERE 文節です。WHERE キーワードはこのパラメーターで暗黙に想定されているので、WHERE キーワードを含めないでください。この WHERE 文節は、この文節を実行するデータ・サーバー上で有効なものでなければなりません。</p> <p>注: このパラメーターの WHERE 文節は、SQLBEFORE または SQLAFTER パラメーターに指定された WHERE 文節とはまったく関係ありません。</p>
CRITGTBL	<p>ターゲット表 (またはビュー) を作成するかどうかを指定します。</p> <p>*YES (デフォルト) ターゲット表 (またはビュー) が存在しなければ、作成します。存在する場合は、既存の表またはビューがターゲットになり、この既存の表またはビューのフォーマットの CHKFMT パラメーターが *YES に設定されているかどうかをチェックします。 UNIQUE および KEYCOL パラメーターに指定された値を使用して、(そのような索引がまだ存在しなければ) ターゲット表に追加の索引が作成されます。既存のターゲット表に、追加索引の条件に違反するような行が含まれていると、コマンドは失敗します。</p> <p>*NO ターゲット表またはビューを作成しません。アプライ・プログラムを開始する前に、正しい属性を使用して表またはビューを作成する必要があります。</p> <p>表またはビューが存在する場合に CHKFMT を *YES に設定すると、ADDDPRSUBM コマンドは、既存の表のフォーマットが、設定されたサブスクリプション・セット定義と一致することを確認します。 CHKFMT を *NO にする場合は、ユーザーは既存の表のフォーマットがサブスクリプション・セット定義と一致することを確認しておく必要があります。</p> <p>重要: 表またはビューがすでに存在する場合、DB2 DataPropagator for iSeries は、既存のオブジェクトへの変更を自動的にジャーナルに記録しません。ジャーナリングは、DB2 DataPropagator for iSeries の外側で開始する必要があります。</p>

表 42. ADDDPRSUBM コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
CHKFMT	<p>DB2 DataPropagator for iSeries が、サブスクリプション・セット・メンバーの定義を既存のターゲット表に照らし合わせ、列が一致することをチェックするかどうかを指定します。このパラメーターは、CRTTGTTBL パラメーターが *YES の場合は無視され、また CRTTGTTBL パラメーターが *NO でターゲット表が存在しない場合も無視されます。</p> <p>*YES (デフォルト) DB2 DataPropagator for iSeries は、このサブスクリプション・セット・メンバーに定義された列が、ターゲット表内の列と一致するか検証します。両者が一致しない場合、このコマンドは失敗します。</p> <p>*NO DB2 DataPropagator for iSeries は、サブスクリプション・セット・メンバーと既存のターゲット表間の相違を無視します。ユーザーはターゲット表がサブスクリプション・セット・メンバーと互換性があることを確認する必要があります。</p>
TGTKEYCHG	<p>ターゲット表のターゲット・キー列の一部であるソース列に変更があった場合、アプライ・プログラムがその更新をどのように扱うかを指定します。このパラメーターは、ADDDPRREG コマンドの USEDELINS パラメーターと組み合わせて働きます。</p> <ul style="list-style-type: none"> • USEDELINS が YES で TGTKEYCHG が YES の場合、更新はできません。 • USEDELINS が YES で TGTKEYCHG が NO の場合、更新は削除と挿入の対になります。 • USEDELINS が NO で TGTKEYCHG が YES の場合、アプライ・プログラムは特別な論理を使用してこの条件を扱います。 • USEDELINS が NO で TGTKEYCHG が NO の場合、アプライ・プログラムは変更を通常の更新として処理します。 <p>*NO (デフォルト) ソース表に対する更新は、キャプチャー・プログラムによりステージ化され、アプライ・プログラムによりターゲット表に処理が行われます。</p> <p>*YES アプライ・プログラムは、ターゲット・キー列の変更前イメージに基づいてターゲット表を更新します。つまり、アプライ・プログラムは述部を新しい値ではなく、古い値に変更します。</p>

表 42. ADDDPRSUBM コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
COLUMN	<p>ターゲット表に含める列を指定します。列名は修飾できません。列名は、ソース表の登録時に CAPCOL パラメーターに指定した列名のリストから選択してください。</p> <p>この表の登録時に IMAGE パラメーターを *BOTH に設定した場合は、変更前イメージ列名を指定することができます。変更前イメージ列名は、接頭部を持つオリジナルの列名です。この接頭部は、ADDDPRREG コマンドの PREFIX パラメーターに指定した文字です。</p> <p>*ALL (デフォルト) ソースに登録した列のすべてがターゲット表に含まれます。</p> <p>*NONE ソース表からの列は 1 つもターゲット表に含まれません。 *NONE は、算出列だけをターゲット表に含めたい場合に使用します。この値は、CALCCOL パラメーターに合計関数が含まれているが、グループ化が実行されない、という場合に必要です。</p> <p><i>column-name</i> ターゲット表に含めたいソース列の名前を 300 個まで指定できます。列名はスペースで区切ります。</p>
UNIQUE	<p>ターゲット表が KEYCOL パラメーターで示されたユニーク・キーを持つかどうかを指定します。</p> <p>*YES (デフォルト) ターゲット表はキーごとに正味 1 つの変更をサポートします。つまり、キーに対していかに多くの変更がなされたとしても、そのキーについてはターゲット表に 1 つの行しか存在しないということです。</p> <p>この値は、表がデータの変更の履歴ではなく、現行のデータを含むことを指定します。コンデンス表には、1 つの主キー値に対して複数の行が含まれることはなく、リフレッシュ用の最新情報を提供するために使用できます。</p> <p>*NO ターゲット表はキーごとに複数の変更をサポートします。変更はターゲット表に付加されます。</p> <p>この値は、表が現行のデータではなく、変更の履歴を含むことを指定します。非コンデンス表には、それぞれのキー値に対して複数の行が含まれ、データの変更履歴を提供するために使用できます。ただし、非コンデンス表はリフレッシュ用の最新データは提供できません。</p>

表 42. ADDDPRSUBM コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
KEYCOL	<p>ターゲット表のキーを記述する列を指定します。列名は修飾できません。 *POINTINTIME、*REPLICA、および *USERCOPY ターゲット表 (TGTTYPE パラメーターで指定されている) の場合、ターゲット表に 1 つまたは複数の列をターゲット・キーとして指定する必要があります。ア プライ・プログラムはこのターゲット・キーを使用して、変更キャプチャ ・レプリケーション中に、変更された個々のユニークな行を識別しま す。</p> <p>*SRCTBL (デフォルト) ターゲット表のキー列は、ソース表のキー列と同じです。 ADDDPRREG コマンドは、ソース表にキーがある場合、ソース表に 指定されたキーを使用します。以下のキー列が使用されます。</p> <ul style="list-style-type: none"> • 物理ファイル作成コマンド (CRTPF) を使用して表を作成した時 に、DDS を使用して定義したキー列 • CREATE TABLE および ALTER TABLE SQL ステートメントを 使用して定義した、主キーおよびユニーク・キー • CREATE INDEX SQL ステートメントを使用して定義したユニーク ・キー <p>1 つの列を、キーとして、異なる順序付けで複数回使用すると、ター ゲット表のキーは昇順で定義されます。</p> <p>*RRN ターゲット表のキー列は IBMQSQ_RRN 列です。ターゲット表は IBMQSQ_RRN 列を使用して作成され、この列がキーとして使用され ます。アプライ・プログラムの実行時に、ソース表がユーザー表であ り、ターゲット表がポイント・イン・タイム表またはユーザー・コピ ーの場合、ターゲット表の IBMQSQ_RRN 列が、ソース表内の関連 するレコードの RRN (相対レコード番号) で更新されます。それ以 外では、ターゲット表の IBMQSQ_RRN 列は、ソース表内の IBMQSQ_RRN 列の値で更新されます。</p> <p>*NONE ターゲット・コピーはターゲット・キーを含みません。ターゲット表 のタイプが *POINTINTIME、*REPLICA、または *USERCOPY の場 合、*NONE は指定できません。</p> <p><i>column-name</i> ターゲット・キー列として使用したいターゲット列の名前。列名は 120 個まで指定できます。列名はスペースで区切ります。</p>

表 42. ADDDPRSUBM コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
TGTCOL	<p>アプライ・プログラムがターゲット表内で更新するすべての列の新しい名前を指定します。これらの名前は、ソース表から取られた列名をオーバーライドします。列名は修飾できません。COLUMN パラメーターに *NONE を指定した場合は、TGTCOL パラメーターを使用しないでください。</p> <p>このパラメーターを使用して、ターゲット表の列により分かりやすい名前を付けることができます。それぞれのソース列の名前と、ターゲット表の対応する列の名前を指定します。</p> <p>*COLUMN (デフォルト) ターゲット列は、COLUMN パラメーターに指定した列と同じです。</p> <p><i>column-name</i> ターゲットではその名前を変更したい、ソース表からの列名。列名は 300 個まで指定できます。</p> <p><i>new-name</i> ターゲット列の新しい名前。新しい列名を 300 個まで指定できます。このパラメーターを使用しない場合、ターゲット表の列名はソースの列名と同じになります。</p>
CALCCOL	<p>ターゲット表のユーザー定義の列または算出された列のリストを指定します。列名は修飾できません。それぞれの列名と式の対を括弧で囲みます。</p> <p>各 SQL 式には列名を指定する必要があります。GROUP BY 文節のない SQL 式として列を定義する場合は、COLUMN パラメーターを *NONE にする必要があります。</p> <p>*NONE (デフォルト) ターゲット表はユーザー定義の列または算出された列を含みません。</p> <p><i>column-name</i> ターゲット表のユーザー定義の列または算出された列の列名。列名は 100 個まで指定できます。</p> <p><i>expression</i> ターゲット表のユーザー定義の列または算出された列の式。SQL 列の式は 100 個まで指定できます。</p>

表 42. ADDDPRSUBM コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
ADDREG	<p>ターゲット表をソース表として自動的に登録するかどうかを指定します。このパラメーターは CCD ターゲット・タイプの表を登録する場合に使用します。</p> <p>*NO (デフォルト) ターゲット表はソース表として登録されません。 DB2 DataPropagator for iSeries は、ターゲット・タイプが *REPLICA の場合、このパラメーター値を無視します。レプリカ・ターゲット表は必ず、ソース表として自動的に登録されます。</p> <p>*YES ターゲット表はソース表として登録されます。ターゲット表をすでにユーザーが登録している場合、このコマンドは失敗します。</p> <p>ターゲット表のタイプが *USERCOPY、*POINTINTIME、*BASEAGR、または *CHANGEAGR の場合、このパラメーターを *YES にしないでください。</p> <p>CRTTGTTBL パラメーターを *NO にした場合、これをソースとして登録する前に、ターゲット表を作成する必要があります。</p>

ADDDPRSUBM の例

以下の例は、ADDDPRSUBM コマンドの使用法を示しています。

例 1

サブスクリプション・セット・メンバーを AQHR アプライ修飾子の下の SETHR という名前のサブスクリプション・セットに追加します。

```
ADDDPRSUBM APYQUAL(AQHR) SETNAME(SETHR) SRCTBL(HR/YTD TAX) TGTTBL(TGTHR/TGTTAX)
```

例 2

2 つだけの列 (AMOUNT および NAME) を持つサブスクリプション・セット・メンバーを、YTD TAX という名前の登録済みソース表から追加し、これらの列を TGTTAX という名前の既存のターゲット表に複製します。

```
ADDDPRSUBM APYQUAL(AQHR) SETNAME(SETHR) SRCTBL(HR/YTD TAX) TGTTBL(TGTLIB/TGTTAX)
  CRTTGTTBL(*NO) COLUMN(AMOUNT NAME) CHKfmt(*YES)
```

このコマンドは、このサブスクリプション・セット・メンバーに定義された AMOUNT 列と NAME 列が、ターゲット表内の列と一致するか検証します。

例 3

サブスクリプション・セット・メンバーを SETHR という名前のサブスクリプション・セットに追加し、このデータを TGTYTD という名前の整合変更データ・ターゲット表に複製します。

```
ADDDPRSUBM APYQUAL(AQHR) SETNAME(SETHR) SRCTBL(HR/YTD TAX) TGTTBL(TGTLIB/TGTYTD)
  TGTTYPE(*CCD) ADDREG (*YES)
```

このコマンドは、ターゲット表を DB2 DataPropagator for iSeries のソース表として登録します。

表 43. ANZDPR コマンド・パラメーター定義 (OS/400 版)

パラメーター	定義およびプロンプト
RDB	<p>分析対象のデータベースを指定します。</p> <p>*LOCAL (デフォルト) ローカル・システム上のデータベース。</p> <p><i>rdb-name</i> データベースを示す、RDB ディレクトリー項目名。</p> <p>最大 10 のデータベースを入力できます。ローカル・システム上のデータベースを含めて複数のデータベースを分析したい場合は、リストの最初の項目を *LOCAL にしてください。また、現行システムからこれらすべてのデータベースに接続できることを確認してください。</p>
OUTFILE	<p>アナライザー出力の保管に使用されるライブラリーとファイル名を指定します。このコマンドは、出力を HTML ファイルに書き込みます。</p> <p>*CURLIB (デフォルト) 現行ライブラリー。</p> <p><i>library-name</i> ライブラリーの名前。</p> <p>ANZDPR (デフォルト) 出力は、ANZDPR という名前の HTML ファイルに書き込まれます。</p> <p><i>file-name</i> HTML 出力ファイルの名前。</p> <p>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。ファイル名が存在しない場合は、RCDLEN(512) および SIZE(*NOMAX) という属性のファイルがコマンドにより作成されます。</p>
ANZLVL	<p>報告される分析のレベルを指定します。分析のレベルは次のとおりです。</p> <p>*STANDARD (デフォルト) コントロール表の内容と、キャプチャー・プログラムおよびアプ ライ・プログラムの状況情報を含むレポートが生成されます。</p> <p>*SIMPLE 標準レポートで情報を生成しますが、サブカラムの詳細は含まれ ません。システム・リソースの使用量を抑えて小さなレポートを 生成する場合は、このオプションを使用してください。</p> <p>*DETAILED 最大限に詳細な分析を含むレポートを生成します。詳細レポート には、サブスクリプション・セット情報に加えて、標準レポート の情報が含まれます。</p>
CAPTRC	<p>キャプチャー・トレース (IBMSNAP_CAPTRACE) 表から報告される項目 の日付範囲 (0 から 30 日) を指定します。デフォルトは 3 です。</p> <p><i>no-of-days</i> 報告される日数。</p>

表 43. ANZDPR コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
APYTRC	<p>アプライ・トレース (IBMSNAP_APPLYTRACE) 表から報告される項目の日付範囲 (0 から 30 日) を指定します。デフォルトは 3 です。</p> <p><i>no-of-days</i> 報告される日数。</p>
APYTRAIL	<p>アプライ・トレール (IBMSNAP_APPLYTRAIL) 表から報告される項目の日付範囲 (0 から 30 日) を指定します。デフォルトは 3 です。</p> <p><i>no-of-days</i> 報告される日数。</p>
SIGTBL	<p>シグナル (IBMSNAP_SIGNAL) 表から報告される項目の日付範囲 (0 から 30 日) を指定します。デフォルトは 3 です。</p> <p><i>no-of-days</i> 報告される日数。</p>
CAPMON	<p>キャプチャー・モニター (IBMSNAP_CAPMON) 表から報告される項目の日付範囲 (0 から 30 日) を指定します。デフォルトは 3 です。</p> <p><i>no-of-days</i> 報告される日数。</p>
APYQUAL	<p>分析対象のアプライ修飾子を指定します。</p> <p>*ALL (デフォルト) すべてのアプライ修飾子が分析されます。</p> <p><i>apply-qualifier</i> 分析対象のアプライ修飾子の名前。最大 10 のアプライ修飾子を入力できます。</p>
CAPCTLLIB	<p>分析対象のキャプチャー・コントロール・ライブラリーの名前である、キャプチャー・スキーマを指定します。特定のキャプチャー・コントロール・ライブラリーを分析するか、デフォルトの *ALL を選択してすべてのキャプチャー・コントロール・ライブラリーを分析できます。</p> <p>*ALL (デフォルト) すべてのキャプチャー・コントロール・ライブラリーが分析されます。</p> <p><i>library-name</i> 分析対象の、特定のキャプチャー・コントロール・ライブラリーの名前。</p>

ANZDPR の例

次の例は、**ANZDPR** コマンドの使用方を示しています。

例 1

標準レベルの分析を使用して、ローカル・データベースと、**RMTRDB1** という名前のリモート・データベースの両方に対してアナライザーを実行するには、次のようにします。

```
ANZDPR RDB(*LOCAL RMTRDB1) OUTFILE(MYLIB/ANZDPR) ANZLVL(*STANDARD) CAPTRC(1)
APYTRC(1) APYTRAIL(1) SIGTBL(1) CAPMON(1) APYQUAL(*ALL)
```


この例では、すべてのアプライ修飾子に関して、IBMSNAP_CAPTRACE、IBMSNAP_APPLYTRACE、IBMSNAP_APPLYTRAIL、IBMSNAP_SIGNAL、およびIBMSNAP_CAPMON 表から 1 日分の項目が生成され、MYLIB という名前のライブラリー内の ANZDPR という名前の HTML ファイルに出力が書き込まれます。

例 2

すべてのデフォルト値を使用してアナライザーを実行するには、次のようにします。

ANZDPR

関連資料:

- 331 ページの『asnanalyze: アナライザーの操作』

CHGDPRCAPA: DPR キャプチャー属性の変更 (OS/400)

DPR キャプチャー属性の変更 (**CHGDPRCAPA**) コマンドは、キャプチャー・プログラムにより使用され、キャプチャー・パラメーター (IBMSNAP_CAPPARMS) 表に保管されるグローバル操作パラメーターを変更するために使用されます。これらのパラメーターに対して行われた変更は、次のアクションのいずれかが実行されるまでは有効になりません。

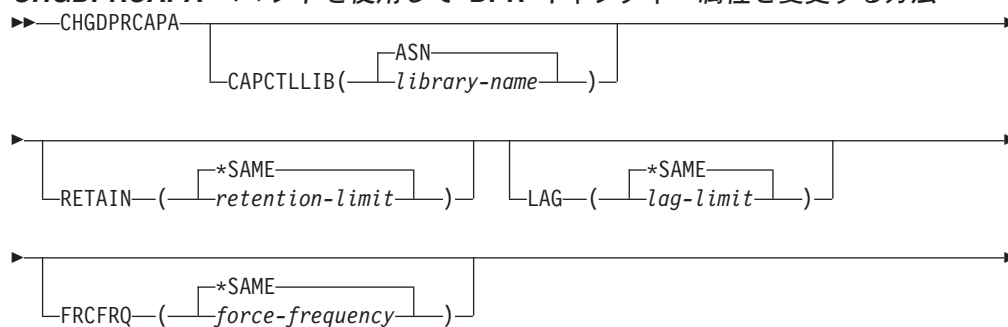
- **INZDPRCAP** コマンドの発行。
- キャプチャー・プログラムの終了と再始動。

実行中のキャプチャー・プログラムの動作を変更するには、448 ページの『OVRDPRCAPA: DPR キャプチャー属性のオーバーライド (OS/400)』を参照してください。

コマンド行にコマンド名を入力してから、F4 キーを押してコマンド構文を表示することができます。

このコマンド、およびこのコマンドのすべてのパラメーターの詳しい記述を表示するには、画面の一番上のコマンドにカーソルを移動し、F1 キーを押します。特定のパラメーターの記述を表示するには、そのパラメーター上にカーソルを移動し、F1 キーを押します。

CHGDPRCAPA コマンドを使用して **DPR** キャプチャー属性を変更する方法



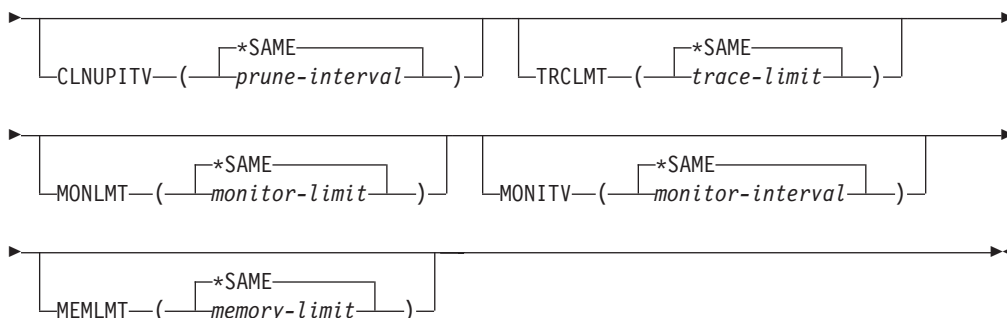


表 44 では、呼び出しパラメーターをリストしています。

表 44. CHGDPRCAPA コマンド・パラメーター定義 (OS/400 版)

パラメーター	定義およびプロンプト
CAPCTLLIB	<p>キャプチャー・スキーマ (キャプチャー・コントロール表が存在するライブラリーの名前) を指定します。</p> <p>ASN (デフォルト) キャプチャー・コントロール表は ASN ライブラリー内にありません。</p> <p><i>library-name</i> キャプチャー・コントロール表を含むライブラリーの名前。</p>
RETAIN	<p>データが除去されずに、変更データ (CD) 表、作業単位 (UOW) 表、シグナル (IBMSNAP_SIGNAL) 表、およびアプライ修飾子相互参照 (IBMSNAP_AUTHTKN) 表に保存される分数である、新規の保持制限を指定します。この値は、キャプチャー・パラメーター (IBMSNAP_CAPPARMS) 表の RETENTION_LIMIT 列に保管されません。</p> <p>この値は、CLNUPITV パラメーターの値と共同で処理を行います。CLNUPITV の値に達すると、CD、UOW、IBMSNAP_SIGNAL、および IBMSNAP_AUTHTKN データのうちで、保持制限よりも古いデータが削除されます。</p> <p>表内のデータに矛盾が生じないように、データがこの RETAIN パラメーター値に達する前に変更情報がコピーされるようにアプライ・インターバルを設定してください。データの矛盾が発生した場合、アプライ・プログラムはフル・リフレッシュを実行します。</p> <p>デフォルトは 10,080 分 (7 日) です。最大値は、35000000 分です。</p> <p>*SAME (デフォルト) この値は変更されません。</p> <p><i>retention-limit</i> 新しい保持制限値。</p>

表 44. CHGDPRCAPA コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
LAG	<p>キャプチャー・プログラムの処理が遅れても、その間は再始動が行われない分数である、新規の遅延限度を指定します。この値は、キャプチャー・パラメーター (IBMSNAP_CAPPARMS) 表の LAG_LIMIT 列に保管されます。</p> <p>遅延限度に達すると (つまり、ジャーナル項目のタイム・スタンプが現在の時間から遅延限度を引いたものより古くなると)、キャプチャー・プログラムは、そのジャーナルで処理中の表のコールド・スタートを開始します。アプライ・プログラムはその後フル・リフレッシュを実行し、キャプチャー・プログラムに新しい開始点を提供します。</p> <p>デフォルトは 10,080 分 (7 日) です。最大値は、35000000 分です。</p> <p>*SAME (デフォルト) この値は変更されません。</p> <p><i>lag-limit</i> 新しい遅延限度値。</p>
FRCFRQ	<p>キャプチャー・プログラムが変更データ (CD) 表および作業単位 (UOW) 表に変更を書き込む頻度 (30 秒から 600 秒) を指定します。この値は、キャプチャー・パラメーター (IBMSNAP_CAPPARMS) 表の COMMIT_INTERVAL 列に保管されます。</p> <p>キャプチャー・プログラムは、バッファがフルになるか、FRCFRQ 時間制限が満了するか、いずれか先に発生した時点で、これらの変更をアプライ・プログラムから使用可能にします。</p> <p>このパラメーターは、ソース表の変更率が少ないサーバー上で、アプライ・プログラムが変更をより早期に使用できるようにするために使用してください。FRCFRQ パラメーター値はグローバル値であり、すべての定義済みソース表で使用されます。FRCFRQ 値を低い数値に設定すると、システム・パフォーマンスが影響を受ける可能性があります。</p> <p>デフォルトは 30 秒です。</p> <p>*SAME (デフォルト) この値は変更されません。</p> <p><i>force-frequency</i> キャプチャー・プログラムが、アプライ・プログラムで変更を使用できるようにする前に、CD 表および UOW 表の変更をバッファ・スペース内に保持する秒数である、新規のコミット・インターバル値です。</p>

表 44. CHGDPRCAPA コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
CLNUPITV	<p>キャプチャー・プログラムが変更データ (CD) 表、作業単位 (UOW) 表、シグナル (IBMSNAP_SIGNAL) 表、キャプチャー・モニター (IBMSNAP_CAPMON) 表、キャプチャー・トレース (IBMSNAP_CAPTRACE) 表、およびアプライ修飾子相互参照 (IBMSNAP_AUTHTKN) 表から古いレコードを除去するまでの最大時間 (時間単位) を指定します。</p> <p>このパラメーターは、RETAIN パラメーターと組み合わせて、CD、UOW、IBMSNAP_SIGNAL、および IBMSNAP_AUTHTKN 表の整理、MONLMT パラメーターと組み合わせて、IBMSNAP_CAPMON 表の整理、TRCLMT パラメーターと組み合わせて、IBMSNAP_CAPTRACE 表の整理をコントロールします。(キャプチャー・プログラムの RETAIN、MONLMT、および TRCLMT パラメーターを設定するには、STRDPRCAP コマンドを使用します。)</p> <p>このパラメーターの値は、自動的に時間から秒に変換され、キャプチャー・パラメーター (IBMSNAP_CAPPARMS) 表の PRUNE_INTERVAL 列に保管されます。PRUNE_INTERVAL 列が (CHGDPRCAPA コマンドを使用せずに) 手動で変更された場合、F4 キーを使用してプロンプトを出すと、丸めによる変更が行われている場合があります。</p> <p>*SAME (デフォルト) このキャプチャー属性値は変更されません。</p> <p><i>prune-interval</i> 時間数 (1 から 100) で指定された、整理インターバル。</p>
TRCLMT	<p>トレース限度を指定します (分単位)。この値は、キャプチャー・パラメーター (IBMSNAP_CAPPARMS) 表の TRACE_LIMIT 列に保管されません。</p> <p>キャプチャー・プログラムは、トレース限度よりも古い IBMSNAP_CAPTRACE 行を削除します。デフォルトは 10,080 分 (トレース項目が 7 日間) です。</p> <p>*SAME (デフォルト) この値は変更されません。</p> <p><i>trace-limit</i> 整理後に、トレース・データが IBMSNAP_CAPTRACE 表に保持される分数。</p>

表 44. CHGDPRCAPA コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
MONLMT	<p>モニター限度を指定します (分単位)。この値は、キャプチャー・パラメーター (IBMSNAP_CAPPARMS) 表の MONITOR_LIMIT 列に保管されます。</p> <p>キャプチャー・プログラムは、モニター限度よりも古い IBMSNAP_CAPMON 行を削除します。</p> <p>デフォルトは 10,080 分 (モニター項目が 7 日間) です。</p> <p>*SAME (デフォルト) この値は変更されません。</p> <p><i>monitor-limit</i> 整理後に、モニター・データが IBMSNAP_CAPMON 表に保持される分数。</p>
MONITV	<p>キャプチャー・プログラムが、キャプチャー・モニター (IBMSNAP_CAPMON) 表に行を挿入する頻度 (秒単位) を指定します。この値は、キャプチャー・パラメーター (IBMSNAP_CAPPARMS) 表の MONITOR_INTERVAL 列に保管されます。</p> <p>デフォルトは 300 秒 (5 分) です。</p> <p>*SAME (デフォルト) この値は変更されません。</p> <p><i>monitor-interval</i> IBMSNAP_CAPMON 表への行挿入の間隔の秒数。モニター・インターバルは、少なくとも 120 秒 (2 分) あける必要があります。120 よりも小さい数値がユーザーから指定された場合、このコマンドは自動的にパラメーター値を 120 に設定します。</p>
MEMLMT	<p>キャプチャー・ジャーナル・ジョブが使用できるメモリの最大サイズ (MB 単位) を指定します。この値は、キャプチャー・パラメーター (IBMSNAP_CAPPARMS) 表の MEMORY_LIMIT 列に保管されます。</p> <p>デフォルトは 32 MB です。</p> <p>*SAME (デフォルト) この値は変更されません。</p> <p><i>memory-limit</i> メモリの最大値の MB 数。</p>

CHGDPRCAPA の例

次の例は、**CHGDPRCAPA** コマンドの使用方法を示しています。

例 1

キャプチャー・プログラムが IBMSNAP_CAPMON 表に行う行挿入の頻度を 6,000 秒 (100 分) 間隔に変更するには、次のようにします。

```
CHGDPRCAPA CAPCTLLIB(ASN) MONITV(6000)
```

この頻度の値は、デフォルト ASN ライブラリー内の IBMSNAP_CAPPARMS 表に保管されます。

例 2

LIB1 と呼ばれるキャプチャー・コントロール・ライブラリー内の
IBMSNAP_CAPPARMS 表で保持制限、遅延限度、トレース限度、およびモニター限
度を変更するには、次のようにします。

```
CHGDPRCAPA CAPCTLLIB(LIB1) RETAIN(6000) LAG(3000) TRCLMT(3000) MONLMT(6000)
```

例 3

キャプチャー・プログラムが CD 表および UOW 表に変更を書き込む頻度を示すコ
ミット・インターバルを変更するには、次のようにします。

```
CHGDPRCAPA CAPCTLLIB(ASN) FRCFRQ(360)
```

関連タスク:

- 125 ページの『第 9 章 SQL レプリケーションに関するキャプチャー・プログラムの操作』

CRTDPRTBL: レプリケーション・コントロール表の作成 (OS/400)

レプリケーション・コントロール表が誤って削除または破壊された場合は、DPR
表の作成 (CRTDPRTBL) コマンドを使用して手動で表を作成します。

重要: CRTDPRTBL コマンドは、OS/400 のコントロール表を作成するために使用
できる唯一のコマンドです。レプリケーション・センターを使用して、コントロ
ール表を作成しないでください。

制約事項: 代替キャプチャー・スキーマを作成する場合は、ASN ライブラリーが
存在するのと同じ補助記憶域プール (基本または独立のいずれか) 内に作成しなけれ
ばなりません。

コマンド行にコマンド名を入力してから、F4 キーを押してコマンド構文を表示す
ることができます。

このコマンド、およびこのコマンドのすべてのパラメーターの詳しい記述を表示す
るには、画面の一番上のコマンドにカーソルを移動し、F1 キーを押します。特定
のパラメーターの記述を表示するには、そのパラメーター上にカーソルを移動し、F1
キーを押します。

**CRTDPRTBL コマンドを使用してレプリケーション・コントロール表を作成する方
法**

```

▶▶ CRTDPRTBL
└─ CAPCTLLIB ( ┌─ ASN ─┐
                └─ library-name ─┘ )

```

431 ページの表 45 では、呼び出しパラメーターをリストしています。

表 45. CRTDPRTBL コマンド・パラメーター定義 (OS/400 版)

パラメーター	定義およびプロンプト
CAPCTLLIB	新しく作成されたキャプチャー・コントロール表が置かれるライブラリーの名前である、キャプチャー・スキーマを指定します。 ASN (デフォルト) キャプチャー・コントロール表は ASN ライブラリー内に置かれます。 <i>library-name</i> キャプチャー・コントロール表が置かれるライブラリーの名前。

CRTDPRTBL の例

次の例は、**CRTDPRTBL** コマンドの使用方法を示しています。

例 1

新しいレプリケーション・コントロール表をデフォルト ASN ライブラリー内に作成するには、次のようにします。

```
CRTDPRTBL CAPCTLLIB(ASN)
```

例 2

DPRSALES という名前のキャプチャー・スキーマの新しいレプリケーション・コントロール表を作成するには、次のようにします。

```
CRTDPRTBL CAPCTLLIB(DPRSALES)
```

関連タスク:

- 17 ページの『第 2 章 SQL レプリケーション用のサーバーの構成』

ENDDPRAPY: アプライ・プログラムの停止 (OS/400)

DPR アプライ・プログラムの終了 (**ENDDPRAPY**) コマンドは、ローカル・システム上のアプライ・プログラムを停止するために使用されます。

計画されているシステムのダウン時間より前に、アプライ・プログラムを停止する必要があります。また、システムの使用がピークになる間、アプライ・プログラムを終了することもできます。

コマンド行にコマンド名を入力してから、F4 キーを押してコマンド構文を表示することができます。

このコマンド、およびこのコマンドのすべてのパラメーターの詳しい記述を表示するには、画面の一番上のコマンドにカーソルを移動し、F1 キーを押します。特定のパラメーターの記述を表示するには、そのパラメーター上にカーソルを移動し、F1 キーを押します。

ENDDPRAPY コマンドを使用してアプライ・プログラムを停止する方法

```

▶—ENDDPRAPY—
      |-----|-----|
      | *CURRENT | *CNTRLD |
      |-----|-----|
      | USER( user-name ) | OPTION( *IMMED ) |
  
```

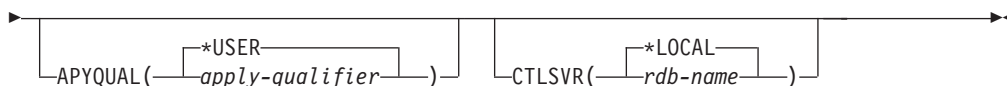


表 46 では、呼び出しパラメーターをリストしています。

表 46. ENDDPRAPY コマンド・パラメーター定義 (OS/400 版)

パラメーター	定義およびプロンプト
USER	<p>このパラメーターは、APYQUAL パラメーターの値が *USER である場合はアプライ・プログラムに関連付けられたアプライ修飾子を指定しますが、それ以外の場合は無視されます。</p> <p>*CURRENT (デフォルト) 現行ジョブに関連付けられたユーザーのアプライ・プログラムです。</p> <p><i>user-name</i> 指定したユーザーのアプライ・プログラム。</p> <p>ENDDPRAPY コマンドでプロンプトを出すと、 F4 キーを押して、サブスクリプションを定義したユーザーのリストを表示することができます。</p>
OPTION	<p>アプライ・プログラムを停止する方法を指定します。</p> <p>*CNTRLD (デフォルト) アプライ・プログラムは、停止前にすべてのタスクを完了します。アプライ・プログラムがサブスクリプション・セットを完了している場合、これらのタスクを終わらせるまでに、かなりの時間がかかる場合があります。</p> <p>*IMMED アプライ・プログラムは、ENDJOB OPTION(*IMMED) コマンドですべてのタスクを完了します。終結処理を行うことなく、このタスクはすぐに終了します。望ましくない結果を引き起こすことがあるため、コントロールされた終了が正常に行われなかった場合にのみ、このオプションを使用してください。(ENDDPRAPY コマンドの発行時に、アプライ・プログラムがスリープ状態でなければ、ターゲット表の目次を検査する必要があります。)</p> <p>アプライ・プログラムがターゲット表へのフル・リフレッシュを実行していた場合、その表がソース表の目次を使ってリフレッシュされる前にアプライ・プログラムが終了しているため、ターゲット表が空である可能性があります。ターゲット表が空の場合、このレプリケーション・ターゲットにフル・リフレッシュを行う必要があります。</p> <p>サブスクリプション・セットが使用中 (サブスクリプション・セット (IBMSNAP_SUBS_SET) にある STATUS 列の値が 1 である) と見なされている場合があります。この場合、この値を 0 または -1 に設定し直してください。こうすると、アプライ・プログラムはサブスクリプション・セットを再度実行することができます。</p>

表 46. ENDDPRAPY コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
APYQUAL	<p>アプライ・プログラムによって使われるアプライ修飾子を指定します。</p> <p>*USER (デフォルト) USER パラメーターで指定されたユーザー名がアプライ修飾子。</p> <p><i>apply-qualifier</i> このアプライ・プログラムが実行するサブスクリプション・セットをグループ化するのに使われる名前。アプライ修飾子名として最大 18 文字を指定できます。この名前は、リレーショナル・データベース名と同じ命名規則に準拠します。実行中のサブスクリプションは、APPLY_QUAL 列にあるこの値を使用して、サブスクリプション・セット (IBMSNAP_SUBS_SET) 表にあるレコードが識別します。</p> <p>ENDDPRAPY コマンドでプロンプトを出すと、F4 キーを押して、既存のサブスクリプションを使用したアプライ修飾子のリストを表示することができます。</p>
CTLSVR	<p>アプライ・コントロール表を含むシステムのリレーショナル・データベース名を指定します。</p> <p>*LOCAL (デフォルト) アプライ・コントロール表はローカルに (ENDDPRAPY コマンドが実行されたマシンに) 存在します。</p> <p><i>rdb-name</i> アプライ・コントロール表が置かれているリレーショナル・データベースの名前。RDB ディレクトリー項目の作業 (WRKRDBDIRE) コマンドを使用して、この名前を検出することができます。</p> <p>ENDDPRAPY コマンドでプロンプトを出すと、F4 キーを押して、RDB ディレクトリーにあるデータベースのリストから選択することができます。</p>

使用上の注意

ENDDPRAPY コマンドは、**APYQUAL** および **CTLSVR** パラメーターの値を使用して、参照済みのアプライ・プログラムのジョブ名、ジョブ番号、およびジョブ・ユーザーに応じたアプライ・ジョブ (IBMSNAP_APPLY_JOB) 表を検索し、そのジョブを終了します。

以下のいずれかの状態が生じると、**ENDDPRAPY** コマンドはエラー・メッセージを発行します。

- IBMSNAP_APPLY_JOB 表が存在していないか、または破壊された場合。
- アプライ修飾子およびコントロール・サーバー名の IBMSNAP_APPLY_JOB 表にレコードがない場合。
- アプライ・ジョブがすでに終了している場合。
- コマンドを実行しているユーザー ID が、アプライ・ジョブを終了する権限を所有していない場合。

ENDDPRAPY の例

次の例は、**ENDDPRAPY** コマンドの使用方法を示しています。

例 1

AQHR アプライ修飾子を使用するアプライ・プログラムを終了するには、次のようにします。

```
ENDDPRAPY OPTION(*CNTRLD) APYQUAL(AQHR)
```

アプライ・プログラムは、すべてのタスクが完了すると終了します。

例 2

アプライ・プログラムを即時に終了するには、次のようにします。

```
ENDDPRAPY OPTION(*IMMED) APYQUAL(AQHR)
```

終結処理を行うことなく、アプライ・プログラムのタスクはすぐに終了します。

例 3

DB1X という名前のリレーショナル・データベース上にあるアプライ・コントロール表を使用するアプライ・プログラムを終了するには、次のようにします。

```
ENDDPRAPY OPTION(*CNTRLD) APYQUAL(AQHR) CTLSVR(DB1X)
```

関連タスク:

- 147 ページの『第 10 章 SQL レプリケーションに関するアプライ・プログラムの操作』

ENDDPRCAP: キャプチャー・プログラムの停止 (OS/400)

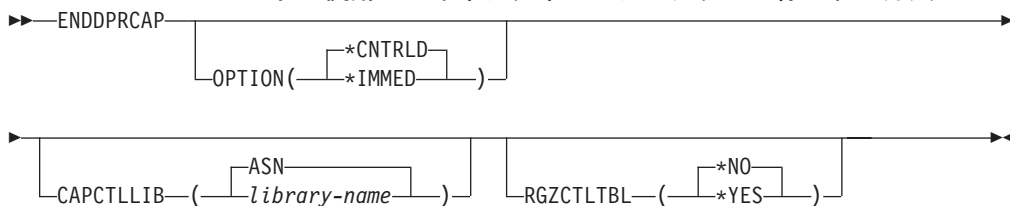
DPR キャプチャー・プログラムの終了 (ENDDPRCAP) コマンドを使用して、キャプチャー・プログラムを停止します。

このコマンドを使用して、システムをシャットダウンする前にキャプチャー・プログラムを停止します。また、システムで実行している他のプログラムのパフォーマンスを向上させるために、システムの使用がピークになる間、プログラムを停止することもできます。

コマンド行にコマンド名を入力してから、F4 キーを押してコマンド構文を表示することができます。

このコマンド、およびこのコマンドのすべてのパラメーターの詳しい記述を表示するには、画面の一番上のコマンドにカーソルを移動し、F1 キーを押します。特定のパラメーターの記述を表示するには、そのパラメーター上にカーソルを移動し、F1 キーを押します。

ENDDPRCAP コマンドを使用してキャプチャー・プログラムを停止する方法



435 ページの表 47 では、呼び出しパラメーターをリストしています。

表 47. ENDDPRCAP コマンド・パラメーター定義 (OS/400 版)

パラメーター	定義およびプロンプト
OPTION	<p>キャプチャー・プログラムを停止する方法を指定します。</p> <p>*CNTRLD (デフォルト) キャプチャー・プログラムは、すべてのタスクを完了した後、通常終了します。</p> <p>ENDDPRCAP コマンドは、*CNTRLD オプションを指定すると、処理が終了するまで長く時間がかかる場合があります。これは、停止する前に、キャプチャー・プログラムはそれに従属する処理をすべて完了させるからです。</p> <p>*IMMED キャプチャー・プログラムは、ENDJOB OPTION(*IMMED) コマンドですべてのタスクを完了してから通常終了します。</p>
CAPCTLLIB	<p>キャプチャー・コントロール表が置かれるライブラリーの名前である、キャプチャー・スキーマを指定します。このライブラリーには、ソース表の登録情報を保管する登録 (IBMSNAP_REGISTER) 表が入っています。</p> <p>ASN (デフォルト) キャプチャー・コントロール表は ASN ライブラリー内にあります。ASN ライブラリーはデフォルト・ライブラリーです。</p> <p><i>library-name</i> キャプチャー・コントロール表を含むライブラリーの名前。</p>
RGZCTLBL	<p>キャプチャー・プログラムの終了時に、コントロール表 (変更データ (CD) および作業単位 (UOW) 表を含む) に対して物理ファイル・メンバーの再編成 (RGZPFM) コマンドが実行されるかどうかを指定します。表に対して RGZPFM コマンド処理が実行されないかぎり、システムはディスク・スペースを回復しません。コントロール表がアプライ・プログラム、またはその他のアプリケーション・プログラムからアクセスされるときには、RGZPFM コマンドは実行されません。</p> <p>*NO (デフォルト) RGZPFM コマンドは実行されません。</p> <p>*YES RGZPFM コマンドは実行されます。</p>

使用上の注意

ENDJOB コマンドを使用すると、一時オブジェクトが QDP4 ライブラリーに残される場合があります。これらのオブジェクトのタイプは ***DTAQ** および ***USRSPC** で、QDP4 *nnnnnn* という名前です。ここで、*nnnnnn* は、オブジェクトを使用したジョブのジョブ番号です。これらのオブジェクトを使用したジョブ (オブジェクト名にあるジョブ番号で識別) がアクティブでないとき、オブジェクトを削除することができます。

このコマンドを発行してもキャプチャー・コントロール・ライブラリー下のジョブが終了しない場合は、***IMMED** オプションを指定して **ENDJOB** コマンドを使用して、このジョブ、および DB2 DataPropagator for iSeries サブシステムで実行中のす

ENDDPRCAP

すべてのジャーナル・ジョブを終了させます。キャプチャー・プログラムだけを終了させたい場合には、同じサブシステムで実行しているアプライ・ジョブを終了させないでください。

まれなことですが、キャプチャー・コントロール・ジョブが異常終了した場合、キャプチャー・コントロール・ジョブにより作成された (**CAPCTLLIB** パラメーターに従って命名される) ジャーナル・ジョブが実行中のまま残されることがあります。こうしたジョブを終了させる唯一の方法は、*IMMED または *CNTRLD オプションのいずれかを指定した **ENDJOB** コマンドを使用することです。

ENDDPRCAP の例

次の例は、**ENDDPRCAP** コマンドの使用法を示しています。

例 1

すべての処理タスクが完了した後、ASN ライブラリー内のキャプチャー・コントロール表を使用するキャプチャー・プログラムを終了するには、次のようにします。

```
ENDDPRCAP OPTION(*CNTRLD) CAPCTLLIB(ASN) RGZCTLTBL(*NO)
```

例 2

キャプチャー・スキーマ BSN のキャプチャー・プログラムを即時に終了するには、次のようにします。

```
ENDDPRCAP OPTION(*IMMED) CAPCTLLIB(BSN) RGZCTLTBL(*NO)
```

例 3

すべての処理タスクが完了した後、キャプチャー・プログラムを終了し、キャプチャー・コントロール表を再編成するには、次のようにします。

```
ENDDPRCAP OPTION(*CNTRLD) CAPCTLLIB(ASN) RGZCTLTBL(*YES)
```

関連タスク:

- 125 ページの『第 9 章 SQL レプリケーションに関するキャプチャー・プログラムの操作』

GRTDPRAUT: ユーザーの許可 (OS/400)

DPR 権限の GRANT (**GRTDPRAUT**) コマンドは、ユーザーがキャプチャー・プログラムおよびアプライ・プログラムを実行できるように、レプリケーション・コントロール表に対するユーザーのリストを許可します。たとえば、キャプチャー・プログラムおよびアプライ・プログラムを実行しているユーザーの許可要件が、レプリケーション・ソースおよびターゲットを定義するユーザーの許可要件とは異なる場合があります。

権限を付与するための *ALLOBJ 権限を所有している必要があります。

コマンド行にコマンド名を入力してから、F4 キーを押してコマンド構文を表示することができます。

このコマンド、およびこのコマンドのすべてのパラメーターの詳しい記述を表示するには、画面の一番上のコマンドにカーソルを移動し、F1 キーを押します。特定のパラメーターの記述を表示するには、そのパラメーター上にカーソルを移動し、F1 キーを押します。

GRTDPRAUT コマンドを使用して、レプリケーション・コントロール表に対してユーザーを許可する方法

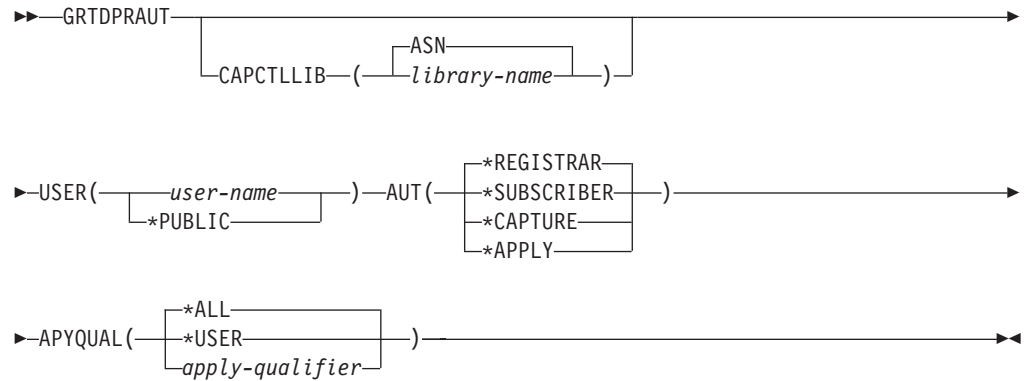


表 48 では、呼び出しパラメーターをリストしています。

表 48. GRTDPRAUT コマンド・パラメーター定義 (OS/400 版)

パラメーター	定義およびプロンプト
CAPCTLLIB	<p>ユーザーが権限を付与されているレプリケーション・コントロール表を含むライブラリーである、キャプチャー・スキーマを指定します。</p> <p>ASN (デフォルト) キャプチャー・コントロール表は ASN ライブラリー内にあります。</p> <p><i>library-name</i> レプリケーション・コントロール表を含むライブラリーの名前。</p>
USER	<p>権限を所有するユーザーを指定します。</p> <p><i>user-name</i> 権限を所有するユーザーの名前を最大 50 まで指定します。</p> <p>*PUBLIC ファイルに対して *PUBLIC 許可を認可するよう指定しますが、(その許可がタスクを行う上で不十分な場合には) 特定の許可を所有していないユーザー、ファイルに関連付けられた許可リストにないユーザー、およびグループ・プロファイルに許可がないユーザーに関してのみ使用されます。</p>

表 48. GRTDPRAUT コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
AUT	<p>付与されている権限のタイプを指定します。</p> <p>*REGISTRAR (デフォルト) ユーザーは、登録を定義、変更、および消去するための権限を付与されます。</p> <p>AUT(*REGISTRAR) 付き権限の完全なリストについては、439 ページの表 49 を参照してください。</p> <p>*SUBSCRIBER ユーザーは、サブスクリプション・セットを定義、変更、および消去するための権限を付与されます。</p> <p>AUT(*SUBSCRIBER) 付き権限の完全なリストについては、440 ページの表 50 を参照してください。</p> <p>*CAPTURE ユーザーは、キャプチャー・プログラムを実行するための権限を付与されます。</p> <p>AUT(*CAPTURE) を付与された権限の完全なリストについては、441 ページの表 51 を参照してください。</p> <p>*APPLY ユーザーは、アプライ・プログラムを実行するための権限を付与されます。</p> <p>このコマンドは、アプライ・プログラムがアクセスする他のデータベースにあるオブジェクトに権限を付与することはありません。</p> <p>アプライ・プログラムが起動されるとき、DRDA アプリケーション・サーバーのジョブに関連するユーザーは、*APPLY 権限も付与されている必要があります。ソースが iSeries サーバーの場合は、USER パラメーターで指定されたアプリケーション・サーバー・ジョブ・ユーザーと、APYQUAL パラメーターで指定されたアプライ修飾子を指定して、ソース・サーバー・システム上で GRTDPRAUT コマンドを実行する必要があります。</p> <p>ターゲット・サーバーがコントロール・サーバーと同じで、その両方がコマンドが実行されるシステムに常駐していない場合、権限がターゲット表に付与されることはありません。</p> <p>AUT(*APPLY) を付与された権限の完全なリストについては、443 ページの表 52 を参照してください。</p>

表 48. GRTPRAUT コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
APYQUAL	<p>USER パラメーターで指定された、ユーザーにより使用されるアプライ修飾子を指定します。このパラメーターは、AUT(*APPLY) または AUT(*SUBSCRIBER) が指定されるときだけ使われます。</p> <p>*ALL (デフォルト) ユーザーは、すべてのアプライ修飾子に関して、アプライ・プログラムを実行する、またはサブスクリプション・セットを定義または除去する権限を付与されます。</p> <p>*USER USER パラメーターで指定されたユーザーは、ユーザー名と同じアプライ修飾子を持つサブスクリプション・セットに対する権限を付与されます。</p> <p><i>apply-qualifier</i> ユーザーは、アプライ・プログラムの実行、または、このアプライ修飾子に関連するアプライ修飾子のサブスクリプション・セットの定義および除去の権限を付与されます。</p> <ul style="list-style-type: none"> ユーザーは、APYQUAL パラメーターを指定して入力された値と一致する APPLY_QUAL 列の中に値がある整理コントロール (IBMSNAP_PRUNCNTL) 表のレコードに関連付けられているすべてのレプリケーション・ソース、変更データ (CD) 表、および整合変更データ (CCD) 表に対する権限を付与されます。 ユーザーは、このシステム上にあるサブスクリプション・メンバー (IBMSNAP_SUBS_MEMBR) 表に記載されたサブスクリプション・セットに対する権限を付与されます。

使用上の注意

使用中のファイルでは許可を変更できないため、キャプチャー・プログラムまたはアプライ・プログラムの実行中や、ソース表を使用するアプリケーションがアクティブのときは、**GRTPRAUT** コマンドを使用することはできません。

以下の表では、

- AUT(*REGISTRAR)
- AUT(*SUBSCRIBER)
- AUT(*CAPTURE)
- AUT(*APPLY)

を、**GRTPRAUT** コマンドに対して指定すると付与される権限をリストします。

以下の表では、**GRTPRAUT** コマンドに対して AUT(*REGISTRAR) パラメーターを指定すると付与される権限をリストします。

表 49. GRTPRAUT AUT(*REGISTRAR) を指定して付与される権限

ライブラリー	オブジェクト	種類	許可
QSYS	capctlib	*LIB	*USE、 *ADD

表 49. GRTDPRAUT AUT(*REGISTRAR) を指定して付与される権限 (続き)

ライブラリー	オブジェクト	種類	許可
capctllib ¹	QSQRN	*JRN	*OBJOPR、 *OBJMGT
capctllib ¹	QZS8CTLBLK	*USRSPC	*CHANGE
capctllib ¹	IBMSNAP_REGISTER	*FILE	*OBJOPR、 *READ、 *ADD、 *UPDT、 *DLT
capctllib ¹	IBMSNAP_REGISTERX	*FILE	*OBJOPR、 *READ、 *ADD、 *UPDT、 *DLT
capctllib ¹	IBMSNAP_REGISTERX1	*FILE	*OBJOPR、 *READ、 *ADD、 *UPDT、 *DLT
capctllib ¹	IBMSNAP_REGISTERX2	*FILE	*OBJOPR、 *READ、 *ADD、 *UPDT、 *DLT
capctllib ¹	IBMSNAP_REG_EXT	*FILE	*OBJOPR、 *READ、 *ADD、 *UPDT、 *DLT
capctllib ¹	IBMSNAP_REG_EXTX	*FILE	*OBJOPR、 *READ、 *ADD、 *UPDT、 *DLT
capctllib ¹	IBMSNAP_PRUNCNTL	*FILE	*OBJOPR、 *READ
capctllib ¹	IBMSNAP_PRUNCNTLX	*FILE	*OBJOPR、 *READ
capctllib ¹	IBMSNAP_PRUNCNTLX1	*FILE	*OBJOPR、 *READ
capctllib ¹	IBMSNAP_PRUNCNTLX2	*FILE	*OBJOPR、 *READ
capctllib ¹	IBMSNAP_PRUNCNTLX3	*FILE	*OBJOPR、 *READ
ASN	ASN4B*	*SQLPKG	*USE
ASN	ASN4C*	*SQLPKG	*USE

注:

1. ライブラリー列の *capctllib* 項目は、**GRTDPRAUT** コマンドの **CAPCTLLIB** パラメーターに渡される値を表します。このコマンドは、同時には 1 つのキャプチャー・コントロール・ライブラリーに対する権限しか更新できません。

以下の表では、**GRTDPRAUT** コマンドに対して AUT(*SUBSCRIBER) パラメーターを指定すると付与される権限をリストします。

表 50. GRTDPRAUT AUT(*SUBSCRIBER) を指定して付与される権限

ライブラリー	オブジェクト	種類	許可
QSYS	ASN	*LIB	*OBJOPR、 *READ、 *ADD、 *EXECUTE
QSYS	capctllib	*LIB	*OBJOPR、 *READ、 *ADD、 *EXECUTE

表 50. GRTDPRAUT AUT(*SUBSCRIBER) を指定して付与される権限 (続き)

ライブラリー	オブジェクト	種類	許可
ASN	IBMSNAP_SUBS_SET	*FILE	*CHANGE
ASN	IBMSNAP_SUBS_COLS	*FILE	*CHANGE
ASN	IBMSNAP_SUBS_EVENT	*FILE	*CHANGE
ASN	IBMSNAP_SUBS_STMTS	*FILE	*CHANGE
ASN	IBMSNAP_SUBS_MEMBR	*FILE	*CHANGE
capctllib ¹	IBMSNAP_REGISTER	*FILE	*OBJOPR、 *READ、 *UPD、 *EXECUTE
capctllib ¹	IBMSNAP_REG_EXT	*FILE	*OBJOPR、 *READ、 *UPD、 *EXECUTE
capctllib ¹	IBMSNAP_PRUNCNTL	*FILE	*OBJOPR、 *READ、 *DLT、 *ADD、 *EXECUTE
capctllib ¹	IBMSNAP_PRUNCNTLX	*FILE	*USE
ASN	ASN4A*	*SQLPKG	*USE
ASN	ASN4U*	*SQLPKG	*USE

注:

1. ライブラリー列の *capctllib* 項目は、**GRTDPRAUT** コマンドの **CAPCTLLIB** パラメータに渡される値を表します。このコマンドは、同時には 1 つのキャプチャー・コントロール・ライブラリーに対する権限しか更新できません。

以下の表では、**GRTDPRAUT** コマンドに対して AUT(*CAPTURE) パラメータを指定すると付与される権限をリストします。

表 51. GRTDPRAUT AUT(*CAPTURE) を指定して付与される権限

ライブラリー	オブジェクト	種類	許可
QSYS	capctllib	*LIB	*OBJOPR、 *OBJMGT、 *READ、 *EXECUTE
QSYS	QDP4	*LIB	*OBJOPR、 *ADD、 *READ、 *EXECUTE
capctllib ¹	QZSN	*MSGQ	*CHANGE
capctllib ¹	IBMSNAP_REGISTER	*FILE	*OBJOPR、 *OBJMGT、 *READ、 *ADD、 *UPD、 *EXECUTE
capctllib ¹	IBMSNAP_REGISTERX	*FILE	*OBJOPR、 *OBJMGT、 *READ、 *ADD、 *UPD、 *EXECUTE

表 51. GRTDPRAUT AUT(*CAPTURE) を指定して付与される権限 (続き)

ライブラリー	オブジェクト	種類	許可
capctllib ¹	IBMSNAP_REGISTERX1	*FILE	*OBJOPR、 *OBJMGT、 *READ、 *ADD、 *UPD、 *EXECUTE
capctllib ¹	IBMSNAP_REGISTERX2	*FILE	*OBJOPR、 *OBJMGT、 *READ、 *ADD、 *UPD、 *EXECUTE
capctllib ¹	IBMSNAP_REG_EXT	*FILE	*OBJOPR、 *OBJMGT、 *READ、 *ADD、 *UPD、 *EXECUTE
capctllib ¹	IBMSNAP_REG_EXTX	*FILE	*OBJOPR、 *OBJMGT、 *READ、 *ADD、 *UPD、 *EXECUTE
capctllib ¹	IBMSNAP_PRUNCNTL	*FILE	*OBJOPR、 *OBJMGT、 *READ、 *UPD、 *EXECUTE
capctllib ¹	IBMSNAP_PRUNCNTLX	*FILE	*OBJOPR、 *OBJMGT、 *READ、 *UPD、 *EXECUTE
capctllib ¹	IBMSNAP_PRUNCNTLX1	*FILE	*OBJOPR、 *OBJMGT、 *READ、 *UPD、 *EXECUTE
capctllib ¹	IBMSNAP_PRUNCNTLX2	*FILE	*OBJOPR、 *OBJMGT、 *READ、 *UPD、 *EXECUTE
capctllib ¹	IBMSNAP_PRUNCNTLX3	*FILE	*OBJOPR、 *OBJMGT、 *READ、 *UPD、 *EXECUTE
capctllib ¹	IBMSNAP_CAPTRACE	*FILE	*CHANGE
capctllib ¹	IBMSNAP_CAPTRACEX	*FILE	*CHANGE
capctllib ¹	IBMSNAP_RESTART	*FILE	*CHANGE
capctllib ¹	IBMSNAP_RESTARTX	*FILE	*CHANGE
capctllib ¹	IBMSNAP_AUTHTKN	*FILE	*CHANGE
capctllib ¹	IBMSNAP_AUTHTKNX	*FILE	*CHANGE

表 51. GRTDPRAUT AUT(*CAPTURE) を指定して付与される権限 (続き)

ライブラリー	オブジェクト	種類	許可
capctllib ¹	IBMSNAP_UOW	*FILE	*OBJOPR、 *OBJMGT、 *READ、 *UPD、 *DLT、 *ADD、 *EXECUTE
capctllib ¹	IBMSNAP_UOW_IDX	*FILE	*CHANGE
capctllib ¹	IBMSNAP_PRUNE_SET	*FILE	*CHANGE
capctllib ¹	IBMSNAP_PRUNE_SETX	*FILE	*CHANGE
capctllib ¹	IBMSNAP_CAPPARMS	*FILE	*READ、 *EXECUTE
capctllib ¹	IBMSNAP_SIGNAL	*FILE	*CHANGE
capctllib ¹	IBMSNAP_SIGNALX	*FILE	*CHANGE
capctllib ¹	IBMSNAP_CAPMON	*FILE	*CHANGE
capctllib ¹	IBMSNAP_CAPMONX	*FILE	*CHANGE
capctllib ¹	IBMSNAP_PRUNE_LOCK	*FILE	*CHANGE
ASN	ASN4B*	*SQLPKG	*USE
ASN	ASN4C*	*SQLPKG	*USE
ASN	QZS8CTLBLK	*USRSPC	*CHANGE

注:

1. ライブラリー列の *capctllib* 項目は、**GRTDPRAUT** コマンドの **CAPCTLLIB** パラメーターに渡される値を表します。このコマンドは、同時には 1 つのキャプチャー・コントロール・ライブラリーに対する権限しか更新できません。

以下の表では、**GRTDPRAUT** コマンドに対して AUT(*APPLY) パラメーターを指定すると付与される権限をリストします。

表 52. GRTDPRAUT AUT(*APPLY) を指定して付与される権限

ライブラリー	オブジェクト	種類	許可
QSYS	ASN	*LIB	*OBJOPR、 *READ、 *EXECUTE
QSYS	capctllib	*LIB	*OBJOPR、 *READ、 *EXECUTE
QDP4	QZSNAPV2	*PGM	*OBJOPR、 *READ、 *OBMGT、 *OBJALTER、 *EXECUTE
capctllib ¹	IBMSNAP_REGISTER	*FILE	*OBJOPR、 *READ、 *UPD、 *EXECUTE

表 52. GRTDPRAUT AUT(*APPLY) を指定して付与される権限 (続き)

ライブラリー	オブジェクト	種類	許可
capctllib ¹	IBMSNAP_REGISTERX	*FILE	*OBJOPR、 *READ、 *UPD、 *EXECUTE
capctllib ¹	IBMSNAP_REGISTERX1	*FILE	*OBJOPR、 *READ、 *UPD、 *EXECUTE
capctllib ¹	IBMSNAP_REGISTERX2	*FILE	*OBJOPR、 *READ、 *UPD、 *EXECUTE
capctllib ¹	IBMSNAP_REGISTER_EXT	*FILE	*OBJOPR、 *READ、 *UPD、 *EXECUTE
capctllib ¹	IBMSNAP_REGISTER_EXTX	*FILE	*OBJOPR、 *READ、 *UPD、 *EXECUTE
capctllib ¹	IBMSNAP_SIGNAL	*FILE	*OBJOPR、 *READ、 *UPD、 *ADD、 *EXECUTE
capctllib ¹	IBMSNAP_SIGNALX	*FILE	*OBJOPR、 *READ、 *UPD、 *ADD、 *EXECUTE
capctllib ¹	IBMSNAP_PRUNE_LOCK	*FILE	*CHANGE
capctllib ¹	IBMSNAP_UOW	*FILE	*OBJOPR、 *READ、 *UPD、 *ADD、 *EXECUTE
capctllib ¹	IBMSNAP_PRUNCNTL	*FILE	*OBJOPR、 *READ、 *UPD、 *ADD、 *EXECUTE
capctllib ¹	IBMSNAP_AUTHTKN	*FILE	*OBJOPR、 *READ、 *UPD、 *ADD、 *EXECUTE
capctllib ¹	IBMSNAP_AUTHTKNX	*FILE	*OBJOPR、 *READ、 *UPD、 *ADD、 *EXECUTE
ASN	IBMSNAP_SUBS_SET	*FILE	*OBJOPR、 *READ、 *UPD、 *EXECUTE
ASN	IBMSNAP_SUBS_SETX	*FILE	*OBJOPR、 *READ、 *UPD、 *EXECUTE

表 52. GRDPRAUT AUT(*APPLY) を指定して付与される権限 (続き)

ライブラリー	オブジェクト	種類	許可
ASN	IBMSNAP_APPLYTRAIL	*FILE	*OBJOPR、 *READ、 *UPD、 *ADD、 *EXECUTE
ASN	IBMSNAP_APPLYTRACE	*FILE	*OBJOPR、 *READ、 *UPD、 *EXECUTE
ASN	IBMSNAP_APPLYTRACX	*FILE	*OBJOPR、 *READ、 *UPD、 *EXECUTE
ASN	IBMSNAP_SUBS_COLS	*FILE	*USE
ASN	IBMSNAP_SUBS_EVENT	*FILE	*USE
ASN	IBMSNAP_SUBS_STMTS	*FILE	*USE
ASN	IBMSNAP_SUBS_MEMBR	*FILE	*USE
ASN	ASN4A*	*SQLPKG	*USE
ASN	ASN4U*	*SQLPKG	*USE
ASN	IBMSNAP_APPLY_JOB	*FILE	*OBJOPR、 *READ、 *UPD、 *ADD、 *EXECUTE

注:

1. ライブラリー列の *capctllib* 項目は、GRDPRAUT コマンドの **CAPCTLLIB** パラメーターに渡される値を表します。このコマンドは、同時には 1 つのキャプチャー・コントロール・ライブラリーに対する権限しか更新できません。

GRDPRAUT の例

次の例は、GRDPRAUT コマンドの使用方法を示しています。

例 1

登録の定義および変更を行う権限を USER1 という名前のユーザーに許可するには、次のようにします。

```
GRDPRAUT CAPCTLLIB(ASN) USER(USER1) AUT(*REGISTRAR)
```

例 2

サブスクリプション・セットの定義および変更を行う権限を USER1 という名前のユーザーに許可するには、次のようにします。

```
GRDPRAUT CAPCTLLIB(ASN) USER(USER1) AUT(*SUBSCRIBER)
```

例 3

キャプチャー・プログラムを実行する権限を USER1 という名前のユーザーに許可するには、次のようにします。

```
GRDPRAUT CAPCTLLIB(ASN) USER(USER1) AUT(*CAPTURE)
```

例 4

アプライ修飾子 A1 に関連付けられた既存のサブスクリプション・セットの定義および変更を行う権限を USER1 という名前のユーザーに許可するには、次のようにします。

```
GRDPRAUT CAPCTLLIB(ASN) USER(USER1) AUT(*SUBSCRIBER) APYQUAL(A1)
```

例 5

ターゲット・サーバーとコントロール・サーバーが同じであるという条件で、アプライ修飾子 A1 に関連付けられたすべてのサブスクリプション・セットのコントロール・サーバー・システム上にあるアプライ・プログラムを実行するための権限をユーザーに与えるには、以下の操作を行ってください。

1. アプライ・プログラムが稼働するシステムで、以下のコマンドを実行します。

```
GRDPRAUT CAPCTLLIB(ASN) USER(USER1) AUT(*APPLY) APYQUAL(A1)
```

2. ソース・サーバー・システム上で適切な **GRDPRAUT** コマンドを実行します。

- アプライ・プログラムによって使われるソース・サーバーのアプリケーション・サーバー・ジョブが、ユーザー・プロファイル USER1 の下で実行される場合、ソース・サーバー・システムで以下のコマンドを実行します。

```
GRDPRAUT CAPCTLLIB(ASN) USER(USER1) AUT(*APPLY) APYQUAL(A1)
```

- アプライ・プログラムによって使われるソース・サーバーのアプリケーション・サーバー・ジョブが、別のユーザー・プロファイル (たとえば、QUSER) の下で実行される場合には、ソース・サーバー・システムで以下のコマンドを実行します。

```
GRDPRAUT CAPCTLLIB(ASN) USER(QUSER) AUT(*APPLY) APYQUAL(A1)
```

関連タスク:

- 17 ページの『第 2 章 SQL レプリケーション用のサーバーの構成』

関連資料:

- 362 ページの『asnpwd: パスワード・ファイルの作成および保守』
- 458 ページの『RVKDPRAUT: 権限の取り消し (OS/400)』

INZDPRCAP: DPR キャプチャー・プログラムの再初期化 (OS/400)

DPR キャプチャー・プログラムの初期化 (INZDPRCAP) コマンドを使用して、ソース表の更新済みリストを使用するようにキャプチャー・プログラムに指示し、キャプチャー・プログラムを初期化します。

キャプチャー・プログラムのコントロール下にあるソース表は、キャプチャー・プログラムの実行中に変更することができます。INZDPRCAP コマンドを使用して、キャプチャー・プログラムが必ず最新のレプリケーション・ソースを処理するようにします。

このコマンドを実行する前に、キャプチャー・プログラムを実行しておく必要があります。

コマンド行にコマンド名を入力してから、F4 キーを押してコマンド構文を表示することができます。

このコマンド、およびこのコマンドのすべてのパラメーターの詳しい記述を表示するには、画面の一番上のコマンドにカーソルを移動し、F1 キーを押します。特定のパラメーターの記述を表示するには、そのパラメーター上にカーソルを移動し、F1 キーを押します。

INZDPRCAP コマンドを使用してキャプチャー・プログラムを初期化する方法

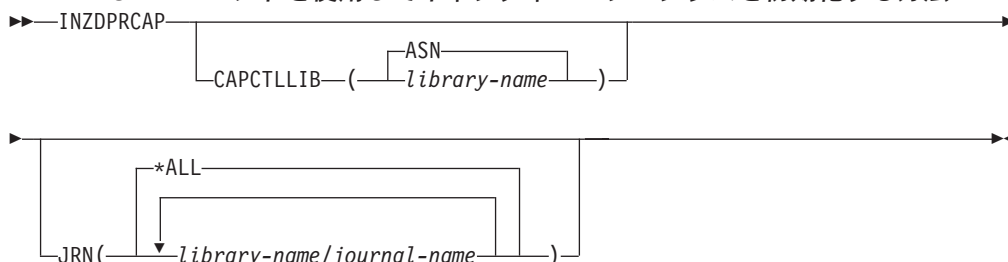


表 53 では、呼び出しパラメーターをリストしています。

表 53. INZDPRCAP コマンド・パラメーター定義 (OS/400 版)

パラメーター	定義およびプロンプト
CAPCTLLIB	<p>キャプチャー・スキーマ (キャプチャー・コントロール表が存在するライブラリーの名前) を指定します。</p> <p>ASN (デフォルト) キャプチャー・コントロール表は ASN ライブラリー内にあります。 ASN ライブラリーはデフォルト・ライブラリーです。</p> <p><i>library-name</i> キャプチャー・コントロール表を含むライブラリーの名前。</p>
JRN	<p>最大 50 のジャーナルのサブセットをキャプチャー・プログラムが使用して作動するように指定します。キャプチャー・プログラムは、現在このジャーナルに記録されているすべてのソース表の処理を開始します。</p> <p>*ALL (デフォルト) キャプチャー・プログラムは、すべてのジャーナルを使って動作します。</p> <p><i>library-name/journal-name</i> キャプチャー・プログラムが使用して作動するジャーナルの修飾名。</p>

INZDPRCAP の例

次の例は、INZDPRCAP コマンドの使用方法を示しています。

例 1

TRAINING という名前のライブラリーにある QSQJRN ジャーナルを使用して、キャプチャー・プログラムの初期化を行うには、次のようにします。

```
INZDPRCAP CAPCTLLIB(ASN) JRN(TRAINING/QSQJRN)
```

キャプチャー・コントロール表はデフォルト ASN スキーマ内にあります。

例 2

すべてのジャーナルを扱うキャプチャー・プログラムの初期化を行うには、次のようにします。

INZDPRCAP CAPCTLLIB(BSN) JRN(*ALL)

キャプチャー・コントロール表は BSN という名前のスキーマ内にあります。

関連タスク:

- 125 ページの『第 9 章 SQL レプリケーションに関するキャプチャー・プログラムの操作』

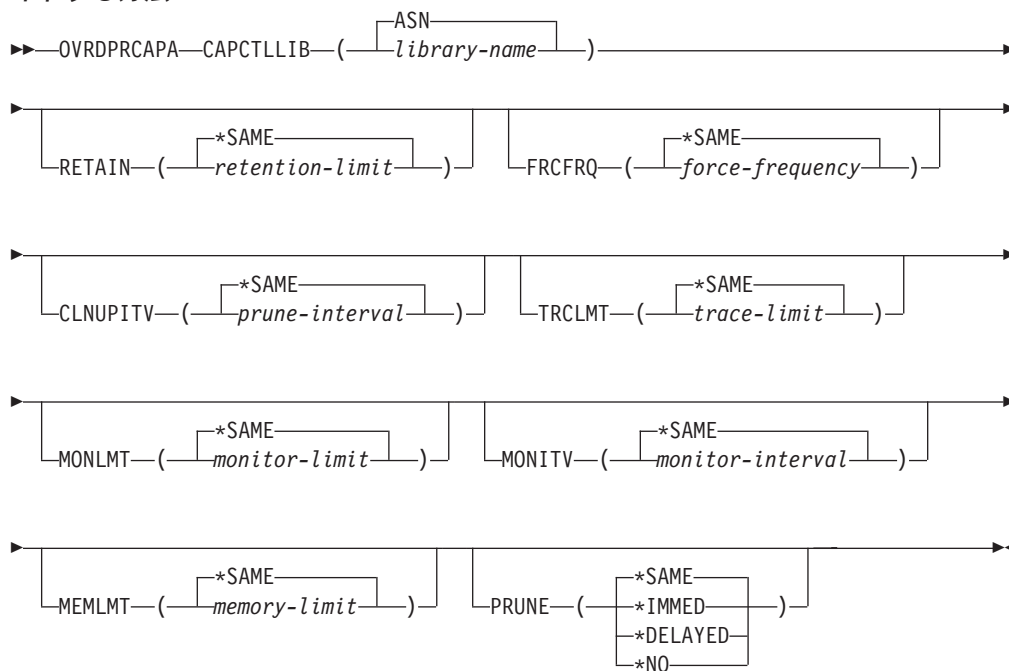
OVRDPRCAPA: DPR キャプチャー属性のオーバーライド (OS/400)

実行中のキャプチャー・プログラムの動作を変更するには、DPR キャプチャー属性のオーバーライド (**OVRDPRCAPA**) コマンドを使用します。このコマンドは、キャプチャー・プログラムの始動時にキャプチャー・パラメーター (IBMSNAP_CAPPARMS) 表、または **STRDPRCAP** コマンドからキャプチャー・プログラムに渡された値をオーバーライドすることにより、プログラムの動作を変更します。

コマンド行にコマンド名を入力してから、F4 キーを押してコマンド構文を表示することができます。

このコマンド、およびこのコマンドのすべてのパラメーターの詳しい記述を表示するには、画面の一番上のコマンドにカーソルを移動し、F1 キーを押します。特定のパラメーターの記述を表示するには、そのパラメーター上にカーソルを移動し、F1 キーを押します。

OVRDPRCAPA コマンドを使用してキャプチャー・プログラムの属性をオーバーライドする方法



449 ページの表 54 では、呼び出しパラメーターをリストしています。

表 54. OVRDPRCAPA コマンド・パラメーター定義 (OS/400 版)

パラメーター	定義およびプロンプト
CAPCTLLIB	<p>キャプチャー・スキーマ (キャプチャー・コントロール表が存在するライブラリーの名前) を指定します。このライブラリーには、ソース表の登録情報を保管する登録 (IBMSNAP_REGISTER) 表が入っています。このパラメーターは必須です。</p> <p>ASN (デフォルト) キャプチャー・コントロール表は ASN ライブラリー内にあります。</p> <p><i>library-name</i> キャプチャー・コントロール表を含むライブラリーの名前。</p> <p>CRTDPRTBL コマンドに CAPCTLLIB パラメーターを指定して、このライブラリーを作成することができます。</p>
RETAIN	<p>データが除去されずに、変更データ (CD)、作業単位 (UOW)、シグナル (IBMSNAP_SIGNAL)、およびアプライ修飾子相互参照 (IBMSNAP_AUTHTKN) 表に保存される分数を指定します。</p> <p>この値は、DPR キャプチャー・プログラムの始動 (STRDPRCAP) コマンドの CLNUPITV パラメーターと共同で処理を行います。最初にキャプチャー・プログラムは、現在実行中のアプライ・プログラムの最も古いものよりも古い、CD、UOW、IBMSNAP_SIGNAL、または IBMSNAP_AUTHTKN 行を削除します。その後、CD、UOW、IBMSNAP_SIGNAL、または IBMSNAP_AUTHTKN 表の新規の行、または残りの行は、経過時間が RETAIN パラメーターの値に達したときに削除されます。</p> <p>表内のデータに矛盾が生じないように、データがこの RETAIN パラメーター値に達する前に変更情報がコピーされるようにアプライ・インターバルを設定してください。データの矛盾が発生した場合、アプライ・プログラムはフル・リフレッシュを実行します。</p> <p>デフォルトは 10,080 分 (7 日) です。最大値は、35000000 分です。</p> <p>*SAME (デフォルト) この値は変更されません。</p> <p><i>retention-limit</i> 新しい保持制限値。</p>

表 54. OVRDPRCAPA コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
FRCFRQ	<p>キャプチャー・プログラムが変更データ (CD) 表および作業単位 (UOW) 表に変更を書き込む頻度 (30 秒から 600 秒) を指定します。</p> <p>キャプチャー・プログラムは、バッファがフルになるか、FRCFRQ 時間制限が満了するか、いずれか先に発生した時点で、これらの変更をアプライ・プログラムから使用可能にします。このパラメーター値は、キャプチャー・プログラムが DPR キャプチャー・プログラムの初期化 (INZDPRCAP) コマンドからの変更に応答するためにかかる時間の長さに影響します。</p> <p>このパラメーターは、ソース表の変更率が少ないサーバー上で、アプライ・プログラムが変更をより早期に使用できるようにするために使用してください。FRCFRQ パラメーター値はグローバル値であり、すべての登録済みソース表で使用されます。FRCFRQ 値を低い数値に設定すると、システム・パフォーマンスに影響を受ける可能性があります。</p> <p>デフォルトは 30 秒です。</p> <p>*SAME (デフォルト) この値は変更されません。</p> <p><i>force-frequency</i> キャプチャー・プログラムが、アプライ・プログラムで変更を使用できるようにする前に、CD 表および UOW 表の変更をバッファースペース内に保持する秒数の新規の値。</p>
CLNUPITV	<p>キャプチャー・プログラムが変更データ (CD) 表、作業単位 (UOW) 表、シグナル (IBMSNAP_SIGNAL) 表、キャプチャー・モニター (IBMSNAP_CAPMON) 表、キャプチャー・トレース (IBMSNAP_CAPTRACE) 表、およびアプライ修飾子相互参照 (IBMSNAP_AUTHTKN) 表から古いレコードを除去するまでの最大時間 (時間単位) を指定します。</p> <p>このパラメーターは、RETAIN パラメーターと組み合わせて、CD、UOW、IBMSNAP_SIGNAL、および IBMSNAP_AUTHTKN 表の整理、MONLMT パラメーターと組み合わせて、IBMSNAP_CAPMON 表の整理、TRCLMT パラメーターと組み合わせて、IBMSNAP_CAPTRACE 表の整理をコントロールします。</p> <p>(キャプチャー・プログラムの RETAIN、MONLMT、および TRCLMT パラメーターを設定するには、STRDPRCAP コマンドを使用します。)</p> <p>CLNUPITV パラメーターの値は、自動的に時間から秒に変換され、キャプチャー・パラメーター (IBMSNAP_CAPPARMS) 表の PRUNE_INTERVAL 列に保管されます。</p> <p>*SAME (デフォルト) このキャプチャー属性値は変更されません。</p> <p><i>prune-interval</i> 時間数 (1 から 100) で指定された、整理インターバル。</p>

表 54. OVRDPRCAPA コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
TRCLMT	<p>キャプチャー・トレース (IBMSNAP_CAPTRACE) 表の整理の頻度を示す、トレース限度を指定します。</p> <p>*SAME (デフォルト) キャプチャー・プログラムは、現行のトレース限度値を継続して使用します。</p> <p><i>trace-limit</i> IBMSNAP_CAPTRACE 表の各整理操作の間隔の分数。</p>
MONLMT	<p>キャプチャー・モニター (IBMSNAP_CAPMON) 表の整理の頻度を示す、モニター限度を指定します。</p> <p>*SAME (デフォルト) キャプチャー・プログラムは、現行のモニター限度値を継続して使用します。</p> <p><i>monitor-limit</i> IBMSNAP_CAPMON 表の各整理操作の間隔の分数。</p>
MONITV	<p>キャプチャー・プログラムがキャプチャー・モニター (IBMSNAP_CAPMON) 表に行を挿入する頻度を示すモニター・インターバル (秒単位) を指定します。</p> <p>*SAME (デフォルト) キャプチャー・プログラムは、現行のモニター・インターバル値の使用を継続します。</p> <p><i>monitor-interval</i> IBMSNAP_CAPMON 表への行挿入の間隔の秒数。モニター・インターバルは、少なくとも 120 秒 (2 分) あける必要があります。120 よりも小さい数値がユーザーから入力された場合、このコマンドは自動的にパラメーター値を 120 に設定します。</p>
MEMLMT	<p>キャプチャー・ジャーナル・ジョブが使用できるメモリーの最大サイズ (MB 単位) を指定します。</p> <p>*SAME (デフォルト) キャプチャー・プログラムは、現行のメモリー限度値を継続して使用します。</p> <p><i>memory-limit</i> メモリーの最大値の MB 数。</p>

表 54. OVRDPRCAPA コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
PRUNE	<p>キャプチャー・プログラムが変更データ (CD) 表、作業単位 (UOW) 表、シグナル (IBMSNAP_SIGNAL) 表、キャプチャー・モニター (IBMSNAP_CAPMON) 表、キャプチャー・トレース (IBMSNAP_CAPTRACE) 表、およびアプライ修飾子相互参照 (IBMSNAP_AUTHTKN) 表から行を削除する方法を変更するには、このパラメーターを使用します。</p> <p>*SAME (デフォルト) キャプチャー・プログラムは、STRDPRCAP コマンドの始動時にユーザーから指定された整理パラメーターを継続して使用します。</p> <p>*IMMED キャプチャー・プログラムは、STRDPRCAP コマンドの始動時にユーザーから指定された CLNUPITV パラメーターの値に関係なく、表の整理を即時に開始します。</p> <p>*DELAYED キャプチャー・プログラムは、指定された整理インターバルの最後に古い行を削除します。</p> <p>PRUNE(*DELAYED) は、STRDPRCAP コマンドの CLNUPITV パラメーターの 2 番目の部分が *IMMED または *DELAYED に設定された場合は、整理の頻度に影響を与えることはありません。しかし PRUNE(*DELAYED) は、STRDPRCAP コマンドの始動時に CLNUPITV パラメーターの 2 番目の部分が *NO に設定された場合は、整理を開始します。</p> <p>*NO キャプチャー・プログラムは整理を開始しません。この値は、STRDPRCAP コマンドの CLNUPITV パラメーターの設定値をオーバーライドします。</p>

OVRDPRCAPA の例

次の例は、**OVRDPRCAPA** コマンドの使用方法を示しています。

例 1

CD、UOW、IBMSNAP_SIGNAL、IBMSNAP_CAPMON、IBMSNAP_CAPTRACE、および IBMSNAP_AUTHTKN 表 (デフォルト ASN ライブラリー内にある) の整理パラメーターを変更し、実行中のキャプチャー・プログラムのキャプチャー・ジャーナル・ジョブの IBMSNAP_CAPMON モニター・インターバルおよびメモリー限度を変更するには、次のようにします。

```
OVRDPRCAPA CAPCTLLIB(ASN) CLNUPITV(12) MONITV(600) MEMLMT(64)
```

例 2

BSN ライブラリー内にある CD、UOW、IBMSNAP_SIGNAL、IBMSNAP_CAPMON、IBMSNAP_CAPTRACE、および IBMSNAP_AUTHTKN 表の整理を開始するには、次のようにします。

```
OVRDPRCAPA CAPCTLLIB(BSN) PRUNE(*IMMED)
```

関連タスク:

例 2

BSN という名前のキャプチャー・スキーマ内の DEPT ライブラリーの、SALES という名前のソース表の登録を除去するには、次のようにします。

```
RMVDPRREG SRCTBL(DEPT/SALES) CAPCTLLIB(BSN)
```

関連タスク:

- 209 ページの『第 13 章 SQL レプリケーション環境の変更』

RMVDPRSUB: DPR サブスクリプション・セットの除去 (OS/400)

サブスクリプション・セットを除去するには、DPR サブスクリプション・セットの除去 (**RMVDPRSUB**) コマンドを使用します。**RMVMBRS** パラメーターが *YES に設定された場合、このコマンドはサブスクリプション・セットとそのすべてのメンバーを除去します。

コマンド行にコマンド名を入力してから、F4 キーを押してコマンド構文を表示することができます。

このコマンド、およびこのコマンドのすべてのパラメーターの詳しい記述を表示するには、画面の一番上のコマンドにカーソルを移動し、F1 キーを押します。特定のパラメーターの記述を表示するには、そのパラメーター上にカーソルを移動し、F1 キーを押します。

RMVDPRSUB コマンドを使用してサブスクリプション・セットを除去する方法

▶▶ RMVDPRSUB—APYQUAL—(—*apply-qualifier*—)—SETNAME—(—*set-name*—)▶▶

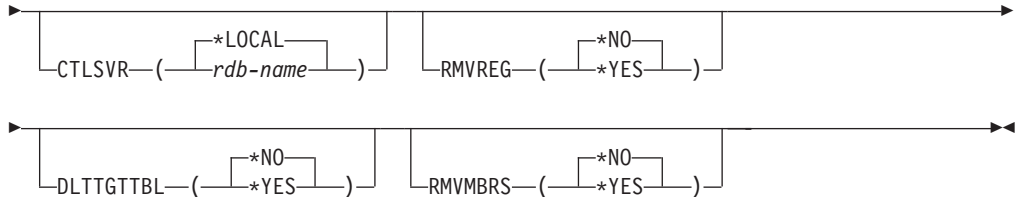


表 56 では、呼び出しパラメーターをリストしています。

表 56. RMVDPRSUB コマンド・パラメーター定義 (OS/400 版)

パラメーター	定義およびプロンプト
APYQUAL	<p>アプライ・プログラムでサブスクリプション・セットの識別に使用されるアプライ修飾子を指定します。このパラメーターは必須です。</p> <p><i>apply-qualifier</i> アプライ修飾子の名前。</p>
SETNAME	<p>サブスクリプション・セットの名前を指定します。このパラメーターは必須です。</p> <p><i>set-name</i> サブスクリプション・セットの名前。指定されたアプライ修飾子に対して存在していないサブスクリプション・セット名が指定されると、エラー・メッセージが戻されます。</p>

表 56. RMVDPRSUB コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
CTLSVR	<p>アプライ・コントロール表を含むシステムのリレーショナル・データベース名を指定します。</p> <p>*LOCAL (デフォルト) アプライ・コントロール表はローカルに (RMVDPRSUB コマンドが実行されたマシンに) 存在します。</p> <p><i>rdb-name</i> アプライ・コントロール表が置かれているリレーショナル・データベースの名前。RDB ディレクトリー項目の作業 (WRKRDBDIRE) コマンドを使用して、この名前を検出することができます。</p>
RMVREG	<p>このコマンドが、サブスクリプション・セット内のすべてのサブスクリプション・セット・メンバーのターゲット表に関連付けられた登録を除去するかどうかを指定します。このパラメーターは、RMVMBRS パラメーターが *YES に設定されている場合にのみ使用してください。</p> <p>*NO (デフォルト) 登録を除去しません。</p> <p>*YES 登録を除去します。</p>
DLTTGTTBL	<p>このコマンドが、サブスクリプション・セットのドロップ後に、サブスクリプション・セット・メンバーのターゲット表をドロップするかどうかを指定します。このパラメーターは、RMVMBRS パラメーターが *YES に設定されている場合にのみ使用してください。</p> <p>*NO (デフォルト) ターゲット表はドロップされません。</p> <p>*YES ターゲット表はドロップされます。</p>
RMVMBRS	<p>このコマンドが、サブスクリプション・セットと、そのサブスクリプション・セット内のすべてのメンバーを除去するかどうかを指定します。</p> <p>*NO (デフォルト) サブスクリプション・セット内に既存のメンバーがある場合には、サブスクリプション・セットは除去されません。</p> <p>*YES サブスクリプション・セットと、そのサブスクリプション・セット・メンバーがすべて除去されます。</p>

RMVDPRSUB の例

次の例は、**RMVDPRSUB** コマンドの使用方法を示しています。

例 1

サブスクリプション・セット・メンバーを含まない、**SETHR** という名前のサブスクリプション・セットを除去するには、次のようにします。

```
RMVDPRSUB APYQUAL(AQHR) SETNAME(SETHR)
```

例 2

SETHR という名前のサブスクリプション・セットと、そのサブスクリプション・セット・メンバーをすべて除去するには、次のようにします。

```
RMVDPRSUB APYQUAL(AQHR) SETNAME(SETHR) RMVMBRS(*YES)
```

例 3

SETHR という名前のサブスクリプション・セットと、そのサブスクリプション・セット・メンバーのすべて、および関連する登録を除去するには、次のようにします。

```
RMVDPRSUB APYQUAL(AQHR) SETNAME(SETHR) RMVREG(*YES) RMVMBRS(*YES)
```

関連タスク:

- 209 ページの『第 13 章 SQL レプリケーション環境の変更』
- 69 ページの『第 4 章 SQL レプリケーションのソースのサブスクライブ』

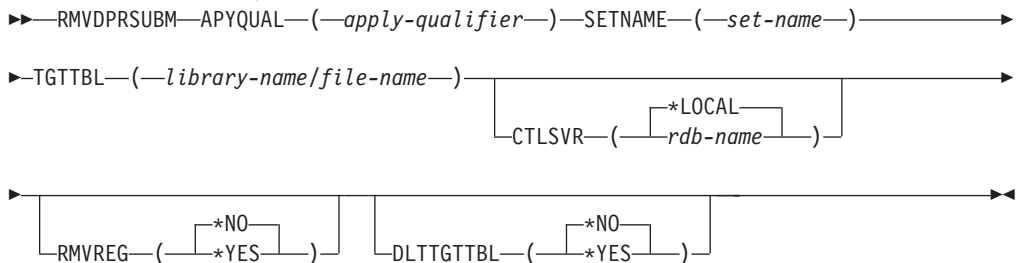
RMVDPRSUBM: DPR サブスクリプション・セット・メンバーの除去 (OS/400)

サブスクリプション・セットから 1 つのサブスクリプション・セット・メンバーを除去するには、DPR サブスクリプション・セット・メンバーの除去 (**RMVDPRSUBM**) コマンドを使用します。

コマンド行にコマンド名を入力してから、F4 キーを押してコマンド構文を表示することができます。

このコマンド、およびこのコマンドのすべてのパラメーターの詳しい記述を表示するには、画面の一番上のコマンドにカーソルを移動し、F1 キーを押します。特定のパラメーターの記述を表示するには、そのパラメーター上にカーソルを移動し、F1 キーを押します。

RMVDPRSUBM コマンドを使用してサブスクリプション・セットから 1 つのサブスクリプション・セット・メンバーを除去する方法



457 ページの表 57 では、呼び出しパラメーターをリストしています。

表 57. RMVDPRSUBM コマンド・パラメーター定義 (OS/400 版)

パラメーター	定義およびプロンプト
APYQUAL	<p>アプライ・プログラムでサブスクリプション・セットの識別に使用されるアプライ修飾子を指定します。このパラメーターは必須です。</p> <p><i>apply-qualifier</i> アプライ修飾子の名前。</p>
SETNAME	<p>サブスクリプション・セットの名前を指定します。このパラメーターは必須です。</p> <p><i>set-name</i> サブスクリプション・セットの名前。指定されたアプライ修飾子に対して存在していないサブスクリプション・セット名が指定されると、エラー・メッセージが戻されます。</p>
TGTTBL	<p>サブスクリプション・セット・メンバーに関して登録されているターゲット表を指定します。このパラメーターは必須です。</p> <p><i>library-name/file-name</i> ターゲット表の修飾名。</p>
CTLSVR	<p>アプライ・コントロール表を含むシステムのリレーショナル・データベース名を指定します。</p> <p>*LOCAL (デフォルト) アプライ・コントロール表はローカルに (RMVDPRSUBM コマンドが実行されたマシンに) 存在します。</p> <p><i>rdb-name</i> アプライ・コントロール表が置かれているリレーショナル・データベースの名前。RDB ディレクトリー項目の作業 (WRKRDBDIRE) コマンドを使用して、この名前を検出することができます。</p>
RMVREG	<p>このコマンドが、サブスクリプション・セット・メンバーのターゲット表に関連付けられた登録を除去するかどうかを指定します。</p> <p>*NO (デフォルト) 登録を除去しません。</p> <p>*YES 登録を除去します。</p>
DLTTGTTBL	<p>このコマンドが、サブスクリプション・セット・メンバーのドロップ後に、サブスクリプション・セット・メンバーのターゲット表をドロップするかどうかを指定します。</p> <p>*NO (デフォルト) ターゲット表はドロップされません。</p> <p>*YES ターゲット表はドロップされます。</p>

RMVDPRSUBM の例

次の例は、RMVDPRSUBM コマンドの使用方法を示しています。

例 1

EMP という名前のターゲット表を使用するサブスクリプション・セット・メンバーを、RMTRDB1 という名前のリレーショナル・データベース上の SETEMP サブスクリプション・セットから除去するには、次のようにします。

```
RMVDPRSUBM APYQUAL(AQHR) SETNAME(SETEMP) TGTTBL(TGTEMP/EMP) CTLSVR(RMTRDB1)
```

例 2

SETHR サブスクリプション・セットからサブスクリプション・セット・メンバーをドロップし、登録をドロップしてから表をドロップするには、次のようにします。

```
RMVDPRSUBM APYQUAL(AQHR) SETNAME(SETHR) TGTTBL(TGTHR/YTD TAX) RMVREG(*YES)
DLTTGTTBL(*YES)
```

関連タスク:

- 69 ページの『第 4 章 SQL レプリケーションのソースのサブスクリプト』

RVKDPRAUT: 権限の取り消し (OS/400)

DPR 権限の取り消し (**RVKDPRAUT**) コマンドは、ユーザーがレプリケーション・ソースおよびサブスクリプション・セットの定義または変更を行うことができないようにするため、レプリケーション・コントロール表に対する権限を取り消します。

コマンド行にコマンド名を入力してから、F4 キーを押してコマンド構文を表示することができます。

このコマンド、およびこのコマンドのすべてのパラメーターの詳しい記述を表示するには、画面の一番上のコマンドにカーソルを移動し、F1 キーを押します。特定のパラメーターの記述を表示するには、そのパラメーター上にカーソルを移動し、F1 キーを押します。

RVKDPRAUT コマンドを使用してレプリケーション・コントロール表の権限を取り消す方法

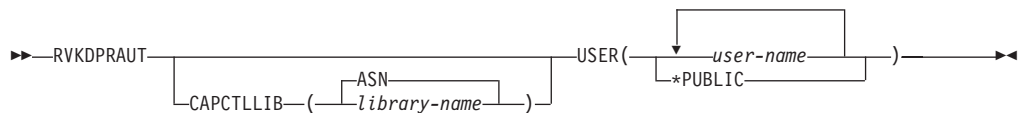


表 58 では、呼び出しパラメーターをリストしています。

表 58. RVKDPRAUT コマンド・パラメーター定義 (OS/400 版)

パラメーター	定義およびプロンプト
CAPCTLLIB	ユーザー権限が取り消されるライブラリーの名前である、キャプチャー・スキーマを指定します。
ASN (デフォルト)	キャプチャー・コントロール表は ASN ライブラリー内にあります。
<i>library-name</i>	レプリケーション・コントロール表を含むライブラリーの名前。

表 58. RVKDPRAUT コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
USER	<p>権限が取り消されるユーザーを指定します。このパラメーターは必須です。</p> <p><i>user-name</i></p> <p>権限が取り消されるユーザーの名前を最大 50 まで指定します。</p> <p>*PUBLIC</p> <p>特定の許可を所有していないすべてのユーザー、許可リストにないすべてのユーザー、およびグループ・プロファイルが許可を所有していないすべてのユーザーから、許可を取り消すように指定します。</p>

使用上の注意

以下のいずれかの状態が生じると、このコマンドはエラー・メッセージを戻します。

- 指定したユーザーが存在しない。
- コマンドを実行しているユーザーが、指定したユーザー・プロファイルへの権限を所有していない。
- このコマンドを実行しているユーザーには、DB2 DataPropagator for iSeries のコントロール表に対する権限を取り消す許可がない。
- DB2 DataPropagator for iSeries のコントロール表が存在しない。
- キャプチャー・プログラムまたはアプライ・プログラムが実行中。

RVKDPRAUT の例

次の例は、RVKDPRAUT コマンドの使用方法を示しています。

例 1

ASN ライブラリー下のコントロール表に対する権限を、HJONES という名前のユーザーから取り消すには、次のようにします。

```
RVKDPRAUT CAPCTLLIB(ASN) USER(HJONES)
```

例 2

GRTDPRAUT コマンドに指定されていないすべてのユーザーの権限を取り消し、ASN ライブラリー内のコントロール表にアクセスできないようにするには、以下のようになります。

```
RVKDPRAUT CAPCTLLIB(ASN) USER(*PUBLIC)
```

関連タスク:

- 17 ページの『第 2 章 SQL レプリケーション用のサーバーの構成』

関連資料:

- 436 ページの『GRTDPRAUT: ユーザーの許可 (OS/400)』

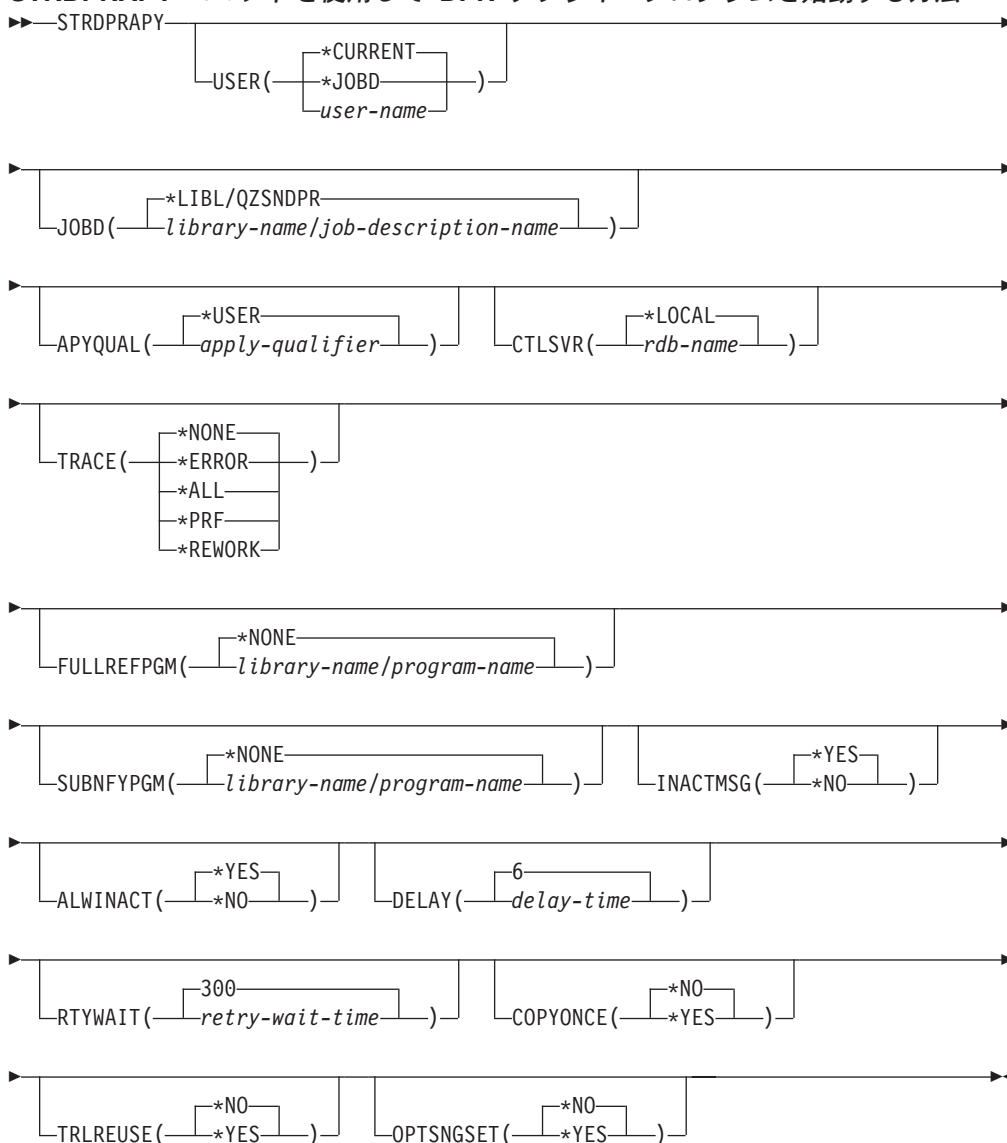
STRDPRAPY: アプライ・プログラムの始動 (OS/400)

DPR アプライ・プログラムの始動 (**STRDPRAPY**) コマンドは、ローカル・システム上でアプライ・プログラムを始動するために使用されます。アプライ・プログラムの実行は、ユーザーがプログラムを停止するまで、またはリカバリー不能エラーが検出されるまで続けられます。

コマンド行にコマンド名を入力してから、F4 キーを押してコマンド構文を表示することができます。

このコマンド、およびこのコマンドのすべてのパラメーターの詳しい記述を表示するには、画面の一番上のコマンドにカーソルを移動し、F1 キーを押します。特定のパラメーターの記述を表示するには、そのパラメーター上にカーソルを移動し、F1 キーを押します。

STRDPRAPY コマンドを使用して DPR アプライ・プログラムを始動する方法



461 ページの表 59 では、呼び出しパラメーターをリストしています。

表 59. STRDPRAPY コマンド・パラメーター定義 (OS/400 版)

パラメーター	定義およびプロンプト
USER	<p>アプライ・プログラムが開始するユーザー ID の名前を指定します。このコマンドを実行するには、指定されたユーザー・プロファイルに対する許可が必要です (*USE 権限を持つ)。アプライ・プログラムは、この指定されたユーザー・プロファイルの下で実行されます。</p> <p>コントロール表は、CTLSVR パラメーターで指定されたりレーショナル・データベース上にあります。同じコントロール表は、USER パラメーターに指定した値を無視して使われます。</p> <p>*CURRENT (デフォルト) 現行ジョブに関連付けられたユーザー ID は、このアプライ・プログラムと関連付けられたユーザー ID と同じです。</p> <p>*JOB このアプライ・プログラムに関連付けられたジョブ記述の中で指定されたユーザー ID。ジョブ記述は USER(*RQD) を指定することはできません。</p> <p><i>user-name</i> このアプライ・プログラムに関連付けられたユーザー ID。以下の IBM 提供のオブジェクトは、このパラメーターに対しては有効ではありません：QDBSHR、QDFTOWN、QDOC、QLPAUTO、QLPINSTALL、QRJE、QSECOFR、QSPL、QSYS、または QTSTRQS。</p> <p>STRDPRAPY コマンドでプロンプトを出すと、F4 キーを押して、サブスクリプション・セットを定義したユーザーのリストを表示することができます。</p>
JOB	<p>アプライ・プログラムをサブミットするときに使用するジョブ記述の名前を指定します。</p> <p>*LIBL/QZSNDPR (デフォルト) DB2 DataPropagator for iSeries から提供されるデフォルトのジョブ記述。</p> <p><i>library-name/job-description-name</i> アプライ・プログラムで使用するジョブ記述の名前。</p>

表 59. STRDPRAPY コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
APYQUAL	<p>アプライ・プログラムによって使われるアプライ修飾子を指定します。このアプライ修飾子によりグループ化されるすべてのサブスクリプション・セットが、アプライ・プログラムで実行されます。</p> <p>*USER (デフォルト) ユーザーが入力する USER パラメーターの値が、アプライ修飾子の名前として使用されます。</p> <p><i>apply-qualifier</i> このアプライ・プログラムによって実行されるサブスクリプション・セットをグループ化するために使用される名前。アプライ修飾子名として最大 18 文字を指定できます。この名前は、リレーショナル・データベース名と同じ命名規則に準拠します。</p> <p>STRDPRAPY コマンドでプロンプトを出すと、F4 キーを押して、既存のサブスクリプション・セットを使用したアプライ修飾子のリストを表示することができます。</p>
CTLSVR	<p>アプライ・コントロール表を含むシステムのリレーショナル・データベース名を指定します。</p> <p>*LOCAL (デフォルト) アプライ・コントロール表はローカルに (STRDPRAPY コマンドが実行されたマシンに) 存在します。</p> <p><i>rdb-name</i> アプライ・コントロール表が置かれているリレーショナル・データベースの名前。RDB ディレクトリー項目の作業 (WRKRDBDIRE) コマンドを使用して、この名前を検出することができます。</p> <p>STRDPRAPY コマンドでプロンプトを出すと、F4 キーを押して、使用可能な RDB 名を表示することができます。</p>
TRACE	<p>アプライ・プログラムがトレースを生成するかどうかを指定します。アプライ・プログラムは、QPZSNATRC と呼ばれるスプール・ファイルにトレース・データを書き込みます。</p> <p>*NONE (デフォルト) トレースが生成されません。</p> <p>*ERROR トレースはエラー情報だけを保持します。</p> <p>*ALL トレースはエラーおよび実行フロー情報を保持します。</p> <p>*PRF トレースは、アプライ・プログラムの実行のさまざまな段階でパフォーマンスの分析に使用できる情報を保持します。</p> <p>*REWORK トレースは、アプライ・プログラムによって再処理された行についての情報を保持します。</p>

表 59. STRDPRAPY コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
FULLREFPGM	<p>アプライ・プログラムが、出力ルーチン呼び出し、ターゲット表を初期化するかどうかを指定します。アプライ・プログラムはターゲット表のフル・リフレッシュの実行準備が整うと、自分でフル・リフレッシュを実行せずに、指定された出力ルーチン呼び出しします。</p> <p>フル・リフレッシュ出力ルーチンがアプライ・プログラムで使用される場合、アプライ・トレール (IBMSNAP_APPLYTRAIL) 表にある ASNLOAD 列の値は Y です。</p> <p>詳しい情報と使用例については、164 ページの『ASNLOAD 出力ルーチンを使ったターゲット表のリフレッシュ』を参照してください。</p> <p>*NONE (デフォルト) フル・リフレッシュ出力ルーチンが使用されません。</p> <p><i>library-name/program-name</i> ターゲット表のフル・リフレッシュを実行するアプライ・プログラムから呼び出されるプログラムの修飾名。たとえば、ライブラリー DATAPROP にあるプログラム ASNLOAD を呼び出すための修飾名は DATAPROP/ASNLOAD です。</p>
SUBNFYPGM	<p>サブスクリプション・セットの処理が終了したときに、アプライ・プログラムから出力ルーチン呼び出しかどうかを指定します。出力ルーチンへの入力には、サブスクリプション・セット名、アプライ修飾子、完了状況、およびリジェクト回数を含む統計が含まれます。</p> <p>通知プログラムを使用すると、作業単位 (UOW) 表を検査して、リジェクトされたトランザクションを判別したり、メッセージの発行やイベントの生成などのアクションをさらに続けるかを判別できます。</p> <p>詳細については、162 ページの『ASNDONE 出力ルーチンの変更 (OS/400)』を参照してください。</p> <p>*NONE (デフォルト) 出力ルーチンが使用されません。</p> <p><i>library-name/program-name</i> サブスクリプション・セットの処理時に、アプライ・プログラムから呼び出される出力ルーチン・プログラムの修飾名。たとえば、ライブラリー DATAPROP にあるプログラム APPLYDONE を呼び出すための修飾名は DATAPROP/APPLYDONE です。</p>
INACTMSG	<p>アプライ・プログラムが作業を完了し、一定の期間非アクティブになったときに、メッセージを生成するかどうかを指定します。</p> <p>*YES (デフォルト) アプライ・プログラムがアクティブではなくなる前に、メッセージ ASN1044 を生成します。メッセージ ASN1044 は、アプライ・プログラムが非アクティブであった時間の長さを示します。</p> <p>*NO メッセージが生成されません。</p>

表 59. STRDPRAPY コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
ALWINACT	<p>アプライ・プログラムを非アクティブ (スリープ) 状態で実行できるかどうかを指定します。</p> <p>*YES (デフォルト) 処理するものが何もない場合、アプライ・プログラムはスリープします。</p> <p>*NO アプライ・プログラムで処理するものが何もない場合、アプライ・プログラムをサブミットして開始したジョブは終了します。</p>
DELAY	<p>連続レプリケーションが使用される場合に、それぞれのアプライ・プログラムのサイクルの終わりにおける遅延時間 (秒単位) を指定します。</p> <p>6 (デフォルト) 遅延時間は 6 秒です。</p> <p><i>delay-time</i> 0 から 6 までの数値で入力された遅延時間。</p>
RTYWAIT	<p>エラーが生じた場合に、失敗した操作を再試行するまでアプライ・プログラムが待機する時間 (秒単位) を指定します。</p> <p>300 (デフォルト) 再試行待ち時間は 300 秒 (5 分) です。</p> <p><i>retry-wait-time</i> 0 から 35000000 の数値で入力された待ち時間。この時間が過ぎると、アプライ・プログラムは失敗した操作を再試行します。</p>

表 59. STRDPRAPY コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
COPYONCE	<p>アプライ・プログラムが呼びだされた時点で適格と見なされたサブスクリプション・セットごとに、アプライ・プログラムがコピー・サイクルを 1 回実行するかどうかを指定します。その後、アプライ・プログラムは終了します。適格と見なされるサブスクリプション・セットとは、以下の基準を満たすものです。</p> <ul style="list-style-type: none"> サブスクリプション・セット (IBMSNAP_SUBS_SET) 表の中で (ACTIVATE > 0)。ACTIVATE 列の値がゼロより大きい場合、そのサブスクリプション・セットは無期限にアクティブであるか、または 1 回のみサブスクリプション処理に使用されています。 (REFRESH_TYPE = R または B) または (REFRESH_TYPE = E であり、指定されたイベントが発生)。REFRESH_TYPE 列の値は IBMSNAP_SUBS_SET 表に保管されます。 <p>IBMSNAP_SUBS_SET 表で MAX_SYNCH_MINUTES 限度と、サブスクリプション・イベント (IBMSNAP_SUBS_EVENT) 表で END_OF_PERIOD タイム・スタンプが指定されている場合、これらの設定が使用されます。</p> <p>*NO (デフォルト) アプライ・プログラムは、適格なサブスクリプション・セットごとにコピー・サイクルを 1 回実行しません。</p> <p>*YES アプライ・プログラムは、適格なサブスクリプション・セットごとにコピー・サイクルを 1 回実行した後、終了します。</p>
TRLREUSE	<p>アプライ・プログラムの始動時に、アプライ・プログラムがアプライ・トレール (IBMSNAP_APPLYTRAIL) 表を空にするかどうかを指定します。</p> <p>*NO (デフォルト) アプライ・プログラムはプログラム始動時に IBMSNAP_APPLYTRAIL 表を空にしません。</p> <p>*YES アプライ・プログラムはプログラム始動時に IBMSNAP_APPLYTRAIL 表を空にします。</p>

表 59. STRDPRAPY コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
OPTSNGSET	<p>アプライ・プログラムのパフォーマンスを、処理するサブスクリプション・セットが 1 つだけである状況に対して最適化するかどうかを指定します。このパラメーターは、レプリカ・ターゲット表には影響しません。</p> <p>このパラメーターが *YES に設定された場合、アプライ・プログラムはサブスクリプション・セットのメンバーおよび列を一度しかフェッチせず、複数の連続する処理サイクルで同じサブスクリプション・セットを処理する場合は、このフェッチした情報を再利用します。</p> <p>*NO (デフォルト) 処理するサブスクリプション・セットが 1 つだけである場合、アプライ・プログラムのパフォーマンスは最適化されません。</p> <p>*YES 処理するサブスクリプション・セットが 1 つだけである場合、アプライ・プログラムのパフォーマンスは最適化されます。アプライ・プログラムはその後の処理サイクルでサブスクリプション・セット情報を再利用するため、CPU リソースの所要量が減少し、スループット率が向上します。</p>

使用上の注意

システムの QSTRUPPGM 値で参照されるコマンドを追加することにより、サブシステムを自動的に開始するようシステムを設定することができます。

QDP4/QZSNDPR サブシステムを使用する場合、それは **STRDPRAPY** コマンド処理の一部として開始されます。

CTLSVR パラメーターによって指定されたリレーショナル・データベース (RDB) が DB2 Universal Database for iSeries データベースである場合、サーバーの表は ASN ライブラリーにあります。RDB が DB2 Universal Database for iSeries データベースでない場合、ASN を修飾子として使って、表にアクセスすることができます。

アプライ・プログラムを始動するときのエラー条件

以下のいずれかの状態が生じると、**STRDPRAPY** コマンドはエラー・メッセージを戻します。

- ユーザーが存在していない場合。
- コマンドを実行しているユーザーが、そのコマンドまたはジョブ記述で指定した、ユーザー・プロファイルへの権限を所有していない場合。
- アプライ・プログラムのインスタンスが、このアプライ修飾子とコントロール・サーバーの組み合わせのローカル・システムですでにアクティブの場合。
- **CTLSVR** パラメーターによって指定された RDB 名がリレーショナル・データベース・ディレクトリーの中に入らない場合。
- **CTLSVR** パラメーターによって指定された RDB 上にコントロール表が存在しない場合。
- **APYQUAL** パラメーターによって指定されたアプライ修飾子に対して定義されたサブスクリプション・セットがない場合。

すべてのサブスクリプション・セット (IBMSNAP_SUBS_SET) 表の中のユニークなアプライ修飾子ごとに、1つのアプライ・プログラムを始動する必要があります。**STRDPRAPY** コマンドを発行するたびに異なるアプライ修飾子を指定することにより、複数のアプライ・プログラムを始動できます。これらのアプライ・プログラムは、同じユーザー・プロファイルの下で実行します。

アプライ・プログラムのジョブの識別

各アプライ・プログラムは、アプライ修飾子とコントロール・サーバー名の両方を使って識別されます。実行時には、アプライ・プログラムに応じて始まるジョブに十分な外部属性がないため、特定のアプライ修飾子とコントロール・サーバーとの組み合わせに関連付けられているアプライ・プログラムを正確に識別することはできません。そのため、以下に示す方法でジョブが識別されます。

- **USER** パラメーターと関連付けられたユーザー・プロファイルの下で、ジョブが開始します。
- アプライ修飾子の最初の 10 文字は切り捨てられて、ジョブ名になります。
- DB2 DataPropagator for iSeries は、ローカル・システム上の ASN ライブラリー内に名前のあるアプライ・ジョブ (IBMSNAP_APPLY_JOB) 表を保持しています。この表が、アプライ修飾子およびコントロール・サーバーの値を、正しいアプライ・プログラムのジョブにマップします。
- ジョブ・ログを表示することができます。アプライ修飾子とコントロール・サーバー名が、アプライ・プログラムに対する呼び出しで使用されています。

通常、以下に示す 2 つの条件が当てはまれば、QZSNDPR サブシステムで実行中のジョブのリストを表示することにより、正しいアプライ・プログラムを識別することができます。

- アプライ修飾子の最初の 10 文字がユニークである。
- アプライ・プログラムが、ローカル・コントロール・サーバーでのみ開始される。

STRDPRAPY の例

次の例は、**STRDPRAPY** コマンドの使用方法を示しています。

例 1

AQHR アプライ修飾子、およびローカルに常駐しているコントロール表を使用して、エラーおよび実行フローの情報を含むトレース・ファイルを生成するアプライ・プログラムを始動するには、次のようにします。

```
STRDPRAPY APYQUAL(AQHR) CTLSVR(*LOCAL) TRACE(*ALL)
```

例 2

ローカルに常駐するアプライ・コントロール表を使用してアプライ・プログラムを始動し、アプライ・プログラムで処理するものが何もない場合は、このアプライ・プログラムを始動したジョブが自動的に終了するように指定するには、次のようにします。

```
STRDPRAPY APYQUAL(AQHR) CTLSVR(*LOCAL) ALWINACT(*NO)
```

例 3

プログラム始動時に IBMSNAP_APPLYTRAIL 表を空にするアプライ・プログラムを始動するには、次のようにします。

```
STRDPRAPY APYQUAL(AQHR) CTLSVR(*LOCAL) TRLREUSE(*YES)
```

例 4

すべてのデフォルト値を使用してアプライ・プログラムを始動するには、次のようにします。

```
STRDPRAPY
```

関連タスク:

- 147 ページの『第 10 章 SQL レプリケーションに関するアプライ・プログラムの操作』

関連資料:

- 334 ページの『asnapply: アプライの始動』

STRDPRCAP: キャプチャー・プログラムの始動 (OS/400)

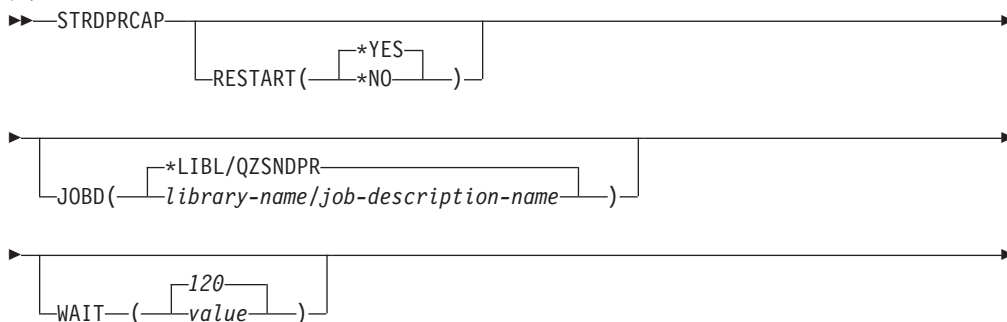
iSeries サーバー上の OS/400 データベース表への変更のキャプチャーを開始するには、DPR キャプチャー・プログラムの始動 (**STRDPRCAP**) コマンドを使用します。このコマンドは登録 (IBMSNAP_REGISTER) 表内のすべてのレプリケーション・ソースを処理するため、適切な権限に基づいてこのコマンドを実行していることを確認してください。

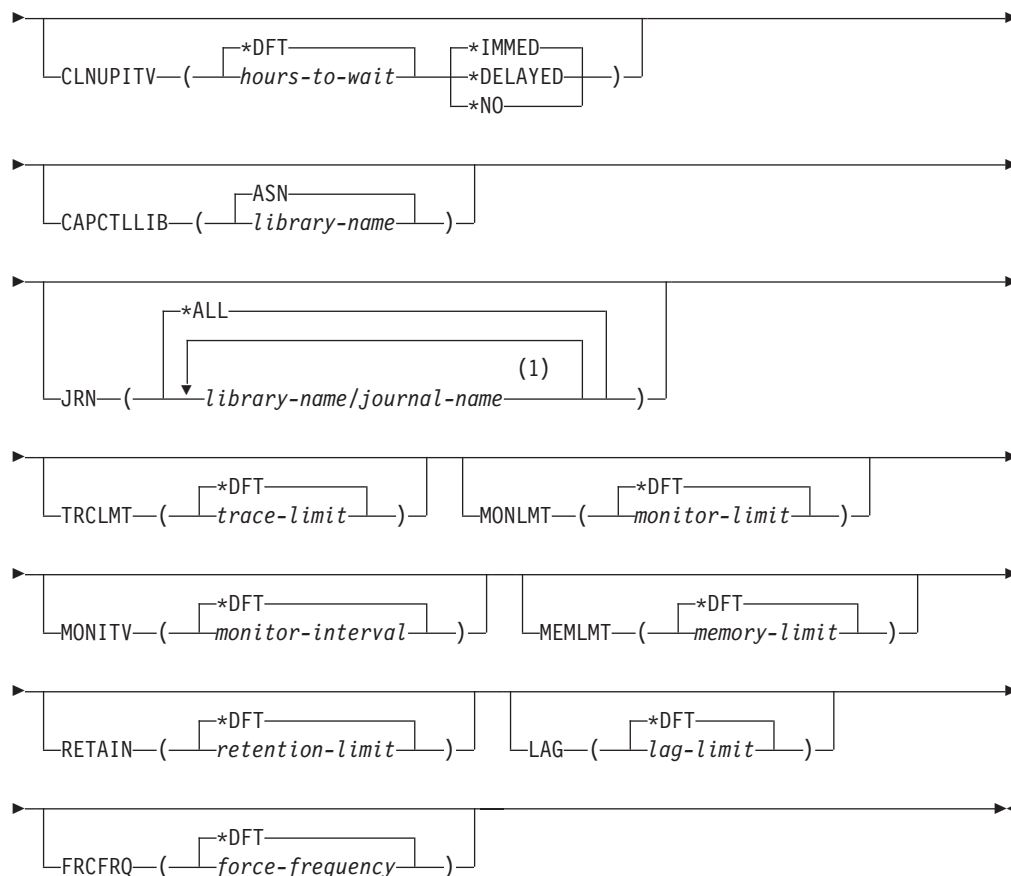
キャプチャー・プログラムを始動すると、停止されるかまたはリカバリー不能エラーが検出されるまで実行を続けます。

コマンド行にコマンド名を入力してから、F4 キーを押してコマンド構文を表示することができます。

このコマンド、およびこのコマンドのすべてのパラメーターの詳しい記述を表示するには、画面の一番上のコマンドにカーソルを移動し、F1 キーを押します。特定のパラメーターの記述を表示するには、そのパラメーター上にカーソルを移動し、F1 キーを押します。

STRDPRCAP コマンドを使用して DPR キャプチャー・プログラムを始動する方法





注:

1 最大 50 のジャーナルを指定できます。

表 60 では、呼び出しパラメーターをリストしています。

表 60. STRDPRCAP コマンド・パラメーター定義 (OS/400 版)

パラメーター	定義およびプロンプト
RESTART	<p>キャプチャー・プログラムがウォーム・スタートとコールド・スタートを処理する方法を指定します。</p> <p>*YES (デフォルト) キャプチャー・プログラムは、以前に終了したときに停止したポイントから変更の処理を継続します。これは、ウォーム・スタートとしても知られており、操作の通常モードです。</p> <p>*NO キャプチャー・プログラムは、変更データ (CD) 表からすべての情報を消去します。JRN(*ALL) を指定すると、キャプチャー・プログラムは作業単位 (UOW) 表からもすべての情報を消去します。</p> <p>変更の取り込みが再開される前に、影響を受けたソース表のすべてのサブスクリプションはフル・リフレッシュされます。この処理は、コールド・スタート としても知られています。</p> <p>RESTART(*NO) および JRN(library-name/journal-name) を指定すると、指定したジャーナルのキャプチャー・プログラムのコールド・スタートを行うことができます。</p>

表 60. STRDPRCAP コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
JOBID	<p>キャプチャー・プログラムをサブミットするときに使用するジョブ記述の名前を指定します。</p> <p>*LIBL/QZSNDPR (デフォルト) DB2 DataPropagator for iSeries から提供されるデフォルトのジョブ記述を指定します。</p> <p><i>library-name/job-description-name</i> キャプチャー・プログラムで使用するジョブ記述の名前。</p>
WAIT	<p>キャプチャー・プログラムが状況をチェックするまで待つ最大時間を秒単位 (60 から 6,000) で指定します。この値を使用して、キャプチャー・プログラムの応答時間を調整することができます。</p> <p>この値を低くすると、キャプチャー・プログラムが終了または初期化するまでにかかる時間は減りますが、システム・パフォーマンスによくない影響を及ぼす場合があります。この値を高くすると、キャプチャー・プログラムが終了または初期化するまでにかかる時間は増えますが、システム・パフォーマンスを向上させることができます。この値をあまり高くしすぎると、キャプチャー・プログラムは定期的に処理を行っていても、応答時間が悪くなることがあります。応答時間がどの程度下がるかは、ソース表に加える変更の量やシステムで発生する他の作業の量によって異なります。</p> <p>120 (デフォルト) キャプチャー・プログラムは 120 秒待機します。</p> <p><i>value</i> キャプチャー・プログラムが待機する最大時間数 (秒)。</p>

表 60. STRDPRCAP コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
CLNUPITV	<p>キャプチャー・プログラムが変更データ (CD) 表、作業単位 (UOW) 表、シグナル (IBMSNAP_SIGNAL) 表、キャプチャー・モニター (IBMSNAP_CAPMON) 表、キャプチャー・トレース (IBMSNAP_CAPTRACE) 表、およびアプライ修飾子相互参照 (IBMSNAP_AUTHTKN) 表から古いレコードを除去するまでの最大時間 (時間単位) を指定します。</p> <p>このパラメーターは、RETAIN パラメーターと組み合わせて、CD、UOW、IBMSNAP_SIGNAL、および IBMSNAP_AUTHTKN 表の整理、MONLMT パラメーターと組み合わせて、IBMSNAP_CAPMON 表の整理、TRCLMT パラメーターと組み合わせて、IBMSNAP_CAPTRACE 表の整理をコントロールします。</p> <p>(キャプチャー・プログラムの RETAIN、MONLMT、および TRCLMT パラメーターを設定するには、STRDPRCAP コマンドを使用します。これらのパラメーターの設定値を変更するには、CHGDPRCAPA または OVRDPRCAPA コマンドを使用します。)</p> <p>CLNUPITV パラメーターには 2 つの部分があります。</p> <p>*DFT (デフォルト) キャプチャー・プログラムは、キャプチャー・パラメーター (IBMSNAP_CAPPARMS) 表の PRUNE_INTERVAL 列の値を使用します。</p> <p><i>hours-to-wait</i> 時間数 (1 から 100) で指定された、整理インターバル。</p> <p>*IMMED (デフォルト) キャプチャー・プログラムは、指定されたインターバルの開始時に (つまり即時に)、そしてその後は各インターバルで、古いレコードを除去します。</p> <p>*DELAYED キャプチャー・プログラムは、指定されたインターバルの終了時に、そしてその後は各インターバルで、古いレコードを除去します。</p> <p>*NO キャプチャー・プログラムはレコードを除去しません。</p>
CAPCTLLIB	<p>キャプチャー・スキーマ (キャプチャー・コントロール表が存在するライブラリーの名前) を指定します。</p> <p>ASN (デフォルト) キャプチャー・コントロール表が常駐するデフォルト・ライブラリーです。</p> <p><i>library-name</i> キャプチャー・コントロール表が常駐するライブラリーの名前。</p>

表 60. STRDPRCAP コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
JRN	<p>最大 50 のジャーナルのサブセットをキャプチャー・プログラムが使用して作動するように指定します。キャプチャー・プログラムは、現在のジャーナルに記録されているすべてのソース表の処理を開始します。</p> <p>*ALL (デフォルト) キャプチャー・プログラムは、ソース表が記録されているジャーナルをすべて使用して作業を開始します。</p> <p><i>library-name/journal-name</i> キャプチャー・プログラムが使用して作動するジャーナルの修飾名。複数のジャーナルを入力するときには、スペースを使用してジャーナルを区切ってください。</p>
TRCLMT	<p>トレース限度を指定します (分単位)。キャプチャー・プログラムは、トレース限度よりも古いキャプチャー・トレース (IBMSNAP_CAPTRACE) 表の行をすべて除去します。デフォルトは 10,080 分 (トレース項目が 7 日間) です。</p> <p>*DFT (デフォルト) キャプチャー・プログラムは、キャプチャー・パラメーター (IBMSNAP_CAPPARMS) 表の TRACE_LIMIT 列の値を使用します。</p> <p><i>trace-limit</i> 整理後に、トレース・データが IBMSNAP_CAPTRACE 表に保持される分数。</p>
MONLMT	<p>モニター限度を指定します (分単位)。キャプチャー・プログラムは、モニター限度よりも古いキャプチャー・モニター (IBMSNAP_CAPMON) 表の行をすべて除去します。デフォルトは 10,080 分 (モニター項目が 7 日間) です。</p> <p>*DFT (デフォルト) キャプチャー・プログラムは、キャプチャー・パラメーター (IBMSNAP_CAPPARMS) 表の MONITOR_LIMIT 列の値を使用します。</p> <p><i>monitor-limit</i> 整理後に、モニター・データが IBMSNAP_CAPMON 表に保持される分数。</p>

表 60. STRDPRCAP コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
MONITV	<p>キャプチャー・プログラムが、キャプチャー・モニター (IBMSNAP_CAPMON) 表に行を挿入する頻度 (秒単位) を指定します。デフォルトは 300 秒 (5 分) です。</p> <p>*DFT (デフォルト) キャプチャー・プログラムは、キャプチャー・パラメーター (IBMSNAP_CAPPARMS) 表の MONITOR_INTERVAL 列の値を使用します。</p> <p><i>monitor-interval</i> IBMSNAP_CAPMON 表への行挿入の間隔の秒数。モニター・インターバルは、少なくとも 120 秒 (2 分) あける必要があります。120 よりも小さい数値がユーザーから入力された場合、パラメーター値は 120 に設定されます。</p>
MEMLMT	<p>キャプチャー・ジャーナル・ジョブが使用できるメモリの最大サイズ (MB 単位) を指定します。デフォルトは 32 MB です。</p> <p>*DFT (デフォルト) キャプチャー・プログラムは、キャプチャー・パラメーター (IBMSNAP_CAPPARMS) 表の MEMORY_LIMIT 列の値を使用します。</p> <p><i>memory-limit</i> メモリの最大値の MB 数。</p>
RETAIN	<p>データが除去されずに、変更データ (CD) 表、作業単位 (UOW) 表、シグナル (IBMSNAP_SIGNAL) 表、およびアプライ修飾子相互参照 (IBMSNAP_AUTHTKN) 表に保存される分数である、新規の保持制限を指定します。この値は、CLNUPITV パラメーターの値と共同で処理を行います。CLNUPITV の値に達すると、CD、UOW、IBMSNAP_SIGNAL、および IBMSNAP_AUTHTKN データのうちで、保持制限よりも古いデータが削除されます。</p> <p>表内のデータに矛盾が生じないように、データがこの RETAIN パラメーター値に達する前に変更情報がコピーされるようにアプライ・インターバルを設定してください。データの矛盾が発生した場合、アプライ・プログラムはフル・リフレッシュを実行します。</p> <p>デフォルトは 10,080 分 (7 日) です。最大値は、35000000 分です。</p> <p>*DFT (デフォルト) キャプチャー・プログラムは、キャプチャー・パラメーター (IBMSNAP_CAPPARMS) 表の RETENTION_LIMIT 列の値を使用します。</p> <p><i>retention-limit</i> CD、UOW、IBMSNAP_SIGNAL、および IBMSNAP_AUTHTKN データが保存される分数。</p>

表 60. STRDPRCAP コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義およびプロンプト
LAG	<p>キャプチャー・プログラムの処理が遅れても、その間は再始動が行われない分数である、新規の遅延限度を指定します。</p> <p>遅延限度に達すると (つまり、ジャーナル項目のタイム・スタンプが現在の時間から遅延限度を引いたものより古くなると)、キャプチャー・プログラムは、そのジャーナルで処理中の表のコールド・スタートを開始します。アプライ・プログラムはその後フル・リフレッシュを実行し、キャプチャー・プログラムに新しい開始点を提供します。</p> <p>デフォルトは 10,080 分 (7 日) です。最大値は、35000000 分です。</p> <p>*DFT (デフォルト) キャプチャー・プログラムは、キャプチャー・パラメーター (IBMSNAP_CAPPARMS) 表の LAG_LIMIT 列の値を使用します。</p> <p><i>lag-limit</i> キャプチャー・プログラムの遅れとして許容される分数。</p>
FRCFRQ	<p>キャプチャー・プログラムが変更データ (CD) 表および作業単位 (UOW) 表に変更を書き込む頻度 (30 秒から 600 秒) を指定します。キャプチャー・プログラムは、バッファーがフルになるか、FRCFRQ 時間制限が満了するか、いずれか先に発生した時点で、これらの変更をアプライ・プログラムから使用可能にします。</p> <p>このパラメーターは、ソース表の変更率が少ないサーバー上で、アプライ・プログラムが変更をより早期に使用できるようにするために使用してください。FRCFRQ パラメーター値はグローバル値であり、すべての定義済みソース表で使用されます。FRCFRQ 値を低い数値に設定すると、システム・パフォーマンスに影響を受ける可能性があります。</p> <p>デフォルトは 30 秒です。</p> <p>*DFT (デフォルト) キャプチャー・プログラムは、キャプチャー・パラメーター (IBMSNAP_CAPPARMS) 表の COMMIT_INTERVAL 列の値を使用します。</p> <p><i>force-frequency</i> キャプチャー・プログラムが、アプライ・プログラムで変更を使用できるようにする前に、CD 表および UOW 表の変更をバッファー・スペース内に保持する秒数。</p>

使用上の注意

STRDPRCAP コマンドの **CLNUPITV** パラメーターは、キャプチャー・プログラムが変更データ (CD) 表、作業単位 (UOW) 表、シグナル (IBMSNAP_SIGNAL) 表、キャプチャー・モニター (IBMSNAP_CAPMON) 表、キャプチャー・トレース (IBMSNAP_CAPTRACE) 表、およびアプライ修飾子相互参照 (IBMSNAP_AUTHTKN) 表から古いレコードを除去するまでに待つ最大時間数を指定します。

STRDPRCAP コマンドは手動で実行することができます。また、このコマンドは、初期プログラム・ロード (IPL 始動プログラム) の一部として、自動的に実行することができます。

JOB パラメーターで指定されたジョブ記述がジョブ・キュー QDP4/QZSNDPR を使用する場合に、DB2 DataPropagator for iSeries サブシステムがアクティブでない場合は、**STRDPRCAP** コマンドはサブシステムを始動します。ジョブ記述が別のジョブ・キューおよびサブシステムを使用するよう定義されている場合には、**STRDPRCAP** コマンドの実行前または後のいずれかに、サブシステムの始動 (**STRSBS**) コマンドを使用して、このサブシステムを手動で始動する必要があります。

STRSBS QDP4/QZSNDPR

システムの QSTRUPPGM システム値で参照されるプログラムに **STRSBS** コマンドを追加することにより、サブシステムを自動的に開始するようシステムを設定することができます。

ウォーム・スタートまたはコールド・スタートを使用したキャプチャー・プログラムの再始動

STRDPRCAP コマンドに対する **RESTART** パラメーターの値が、キャプチャー・プログラムがウォーム・スタートとコールド・スタートを処理する方法をコントロールします。

ウォーム・スタート処理: ほとんどの場合、ウォーム・スタート情報は保管されます。時々、ウォーム・スタート情報が保管されないことがあります。そのような場合、キャプチャー・プログラムは、CD 表、UOW 表、または整理コントロール (IBMSNAP_PRUNCNTL) 表を使用して、それが停止した時刻に再同期します。

自動コールド・スタート: 場合によっては、ウォーム・スタートが指定されていても、キャプチャー・プログラムは自動的にコールド・スタートに切り替えます。OS/400 システムでは、コールド・スタートはジャーナルによるジャーナル方式で動作します。たとえば、ジャーナルがラグの制限を超える場合、そのジャーナルを使用するすべてのレプリケーション・ソースは、別のジャーナルを使用するレプリケーション・ソースがコールド・スタートでなくても、コールド・スタートで始動します。

キャプチャー・プログラムが様々なタイプのジャーナル項目を処理する方法について詳しくは、601 ページの表 114 を参照してください。

STRDPRCAP の例

次の例は、**STRDPRCAP** コマンドの使用方法を示しています。

例 1

2 つの異なるジャーナルに関して、キャプチャー・プログラムのウォーム・スタートを開始するには、次のようにします。

```
STRDPRCAP RESTART(*YES) JRN(HR/QSQJRN ACCTS/QSQJRN)
```

例 2

1 つの指定されたジャーナルに関してキャプチャー・プログラムを始動するには、次のようにします。

```
STRDPRCAP CAPCTLLIB(BSN) JRN(MARKETING/QSQJRN)
```

キャプチャー・コントロール表は BSN という名前のライブラリーにあります。

例 3

2 つのジャーナルの整理を行わないでキャプチャー・プログラムを始動するには、次のようにします。

```
STRDPRCAP RESTART(*YES) CLNUPITV(*DFT *NO) JRN(HR/QSQJRN ACCTS/QSQJRN)
```

例 4

デフォルトのキャプチャー・コントロール・ライブラリーにある、1 つの指定されたジャーナルについてキャプチャー・プログラムを始動し、トレース限度整理、モニター限度整理、IBMSNAP_CAPMON 表挿入、およびメモリー限度のデフォルト・パラメーターを変更するには、次のようにします。

```
STRDPRCAP CAPCTLLIB(ASN) JRN(SALES/QSQJRN) TRCLMT(1440) MONLMT(1440)  
MONITV(3600) MEMLMT(64)
```

例 5

キャプチャー・プログラムのコールド・スタートを開始するには、次のようにします。

```
STRDPRCAP RESTART(*NO)
```

例 6

すべてのデフォルト値を使用してキャプチャー・プログラムを始動するには、次のようにします。

```
STRDPRCAP
```

関連タスク:

- 125 ページの『第 9 章 SQL レプリケーションに関するキャプチャー・プログラムの操作』

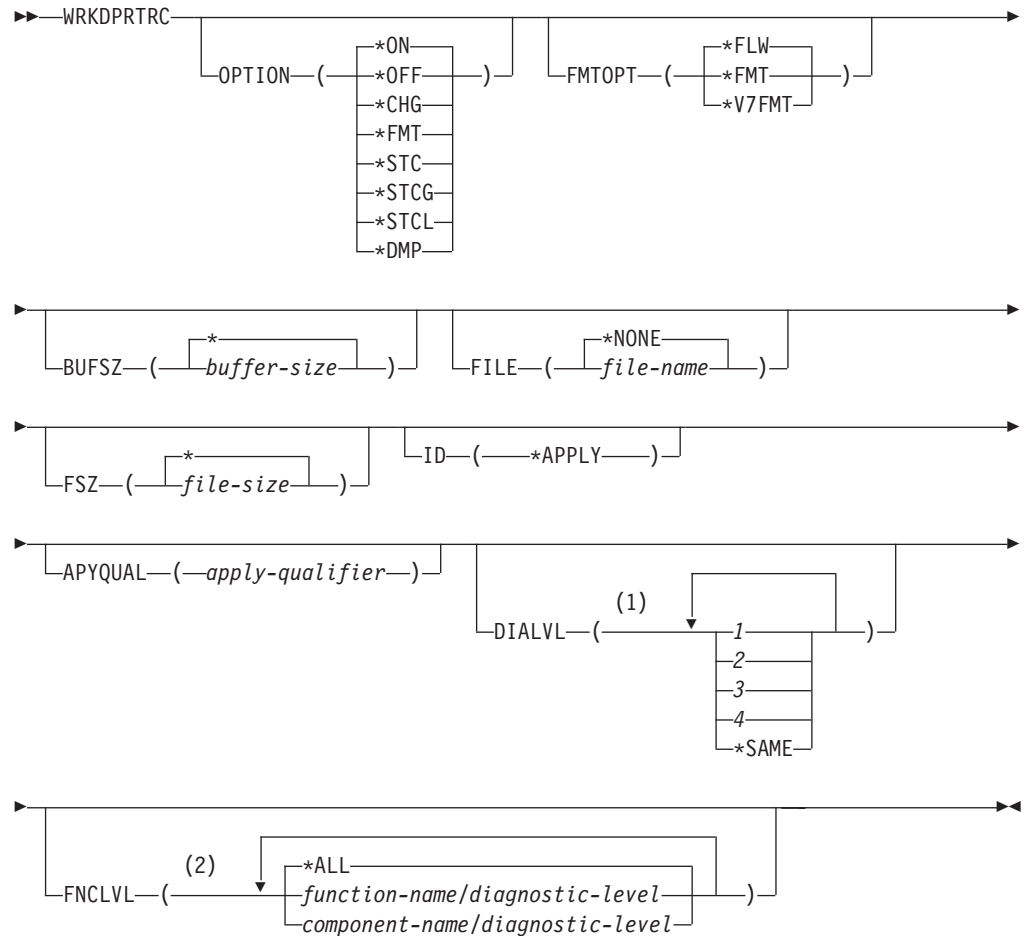
WRKDPRTTC: DPR トレース機能の使用法 (OS/400)

トレース機能を実行するには、DPR トレース (**WRKDPRTTC**) コマンドを実行します。トレース機能は、指定されたアプライ・プログラムのプログラム・フロー情報を記録します。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。

コマンド行にコマンド名を入力してから、F4 キーを押してコマンド構文を表示することができます。

このコマンド、およびこのコマンドのすべてのパラメーターの詳しい記述を表示するには、画面の一番上のコマンドにカーソルを移動し、F1 キーを押します。特定のパラメーターの記述を表示するには、そのパラメーター上にカーソルを移動し、F1 キーを押します。

WRKDPRTTC コマンドを使用して DPR トレース機能を実行する方法



注:

- 1 複数の値を指定できます。
 - 2 最大 20 の関数またはコンポーネントを指定できます。
- 478 ページの表 61 では、呼び出しパラメーターをリストしています。

表 61. WRKDPRTTC コマンド・パラメーター定義 (OS/400 版)

パラメーター	定義
OPTION	<p>トレース関数を 1 つ指定します。</p> <p>*ON (デフォルト) トレース機能をオンにします。このオプションは、トレース用の共有メモリー・セグメントを自動的に作成します。</p> <p>*OFF トレース機能をオフにします。</p> <p>*CHG トレース機能パラメーターの値を変更します。</p> <p>*FMT 共有メモリーからのトレース機能の出力をフォーマットします。</p> <p>*STC トレース機能の状況を表示します。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、バッファー・サイズ、使用中のバッファー量、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。</p> <p>このパラメーター・オプションは、UNIX、Windows および z/OS オペレーティング・システムで使用される asnlrc コマンドの stat オプションと同等です。</p> <p>*STCG レプリケーション・センターで読み取れるフォーマットでトレース機能の状況を表示します。</p> <p>*STCL 追加のバージョン・レベル情報を含めてトレース機能の状況を表示します。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せん。</p> <p>このパラメーター・オプションは、UNIX、Windows および z/OS オペレーティング・システムで使用される asnlrc コマンドの statlong オプションと同等です。</p> <p>*DMP トレース・バッファーの現在の内容をファイルに書き込みます。</p> <p>WRKDPRTTC コマンドのプロンプトから F4 キーを押すと、トレース・オプションのリストを表示できます。</p>

表 61. WRKDPRTTC コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義
FMTOPT	<p>フォーマット ID のオプションを指定します。 OPTION(*FMT) パラメーターと一緒に使用されます。</p> <p>*FLW (デフォルト) 関数呼び出しのフローを表示します。</p> <p>*FMT トレース・バッファまたはトレース・ファイルのフォーマットを表示します。すべての詳細データが表示されます。</p> <p>*V7FMT トレース・バッファまたはトレース・ファイル情報をバージョン 7 の形式にフォーマットします。</p> <p>WRKDPRTTC コマンドのプロンプトから F4 キーを押すと、フォーマット・オプションのリストを表示できます。</p>
BUFSZ	<p>トレース・バッファのサイズを指定します (バイト単位)。数字の後ろに M、K、または G を入力すれば、メガバイト、キロバイト、またはギガバイトをそれぞれ指定できます。</p> <p>デフォルトは 2 メガバイトです。</p> <p>* (デフォルト) 2 MB のデフォルト・サイズが使用されます。</p> <p><i>buffer-size</i> バッファ・サイズ (バイト単位)。</p>
FILE	<p>トレース出力をファイルに書き込むかどうかを指定します。</p> <p>*NONE (デフォルト) トレース出力は共有メモリーにのみ入れられます。</p> <p><i>file-name</i> 出力ファイルの名前。 OPTION(*DMP) パラメーターが使用された場合、このファイル名はダンプ・ファイルの名前を表します。</p>
FSZ	<p>トレース・データが保管されるファイルのサイズを指定します (バイト単位)。数字の後ろに M、K、または G を入力すれば、メガバイト、キロバイト、またはギガバイトをそれぞれ指定できます。</p> <p>デフォルトは 2 GB です。</p> <p>* (デフォルト) 2 GB のデフォルト・サイズが使用されます。</p> <p><i>file-size</i> ファイル・サイズ (バイト単位)。</p>

表 61. WRKDPRTTC コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義
ID	<p>トレース対象のプログラムのタイプを指定します。</p> <p>*APPLY (デフォルト) アプライ・プログラムのトレース。</p>
APYQUAL	<p>トレースするアプライ・プログラムの名前を指定します。</p> <p><i>apply-qualifier</i> アプライ修飾子の名前。</p>
DIALVL	<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。関数の入口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>*SAME このコマンドは、直前のトレース機能で使用された診断レベル設定値を使用します。</p> <p>1 つ、または複数の診断マスク番号を入力できます。番号は昇順に入力する必要があります。数字間にスペースを入れないでください。</p> <p>重要: 番号レベルは包括的ではありません。それぞれが独立したレベルを表しています。</p> <p>トレース機能の開始時のデフォルトは DIALVL(1234) です。その後にトレース機能呼び出した場合のデフォルトは *SAME です。</p> <p>WRKDPRTTC コマンドのプロンプトから F4 キーを押すと、使用可能な診断レベルのリストを表示できます。</p>

表 61. WRKDPRTTC コマンド・パラメーター定義 (OS/400 版) (続き)

パラメーター	定義
FNCLVL	<p>特定の関数またはコンポーネント ID をトレースするかどうかを指定します。</p> <p>*ALL (デフォルト) すべての関数およびコンポーネントがトレース関数に含まれます。</p> <p><i>function-name/ diagnostic-level</i> トレースする関数の名前と、対応する診断マスク番号。</p> <p><i>component-name/ diagnostic-level</i> トレースするコンポーネントの名前と、対応する診断マスク番号。</p> <p>最大 20 の関数またはコンポーネント名を入力できません。</p>

WRKDPRTTC の例

次の例は、WRKDPRTTC コマンドの使用方法を示しています。

例 1

すべての関数およびコンポーネントについて、アプライ修飾子 AQ1 のアプライ・トレースを開始し、出力を TRCFILE という名前のファイルに書き込むには、次のようにします。

```
WRKDPRTTC OPTION(*ON) FILE(TRCFILE) ID(*APPLY) APYQUAL(AQ1)
```

例 2

アプライ修飾子 AQ1 のアプライ・トレースを終了するには、次のようにします。

```
WRKDPRTTC OPTION(*OFF) ID(*APPLY) APYQUAL(AQ1)
```

例 3

すべての関数およびコンポーネントについて、アプライ修飾子 AQ1 のアプライ・トレースを診断レベル 3 および 4 (詳細データおよびパフォーマンス・データ) に変更するには、次のようにします。

```
WRKDPRTTC OPTION(*CHG) ID(*APPLY) APYQUAL(AQ1) DIALVL(34)
```

例 4

アプライ修飾子 AQ1 のアプライ・トレースの状況を表示するには、次のようにします。

```
WRKDPRTTC OPTION(*STC) ID(*APPLY) APYQUAL(AQ1)
```

例 5

アプライ修飾子 AQ1 の関数呼び出しを診断レベル 3 および 4 で表示するには、次のようにします。

```
WRKDPRTTC OPTION(*FMT) FMTOPT(*FLW) ID(*APPLY) APYQUAL(AQ1) DIALVL (34)
```

例 6

アプライ修飾子 AQ1 のアプライ・トレース情報を DMPFILE という名前のダンプ・ファイルに書き込むには、次のようにします。

```
WRKDPRTTC OPTION(*DMP) FILE(DMPFILE) ID(*APPLY) APYQUAL(AQ1)
```

関連資料:

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

第 20 章 SQL レプリケーション・プログラムの操作 (z/OS)

この章は、以下のセクションで構成されています。

- 『レプリケーション・プログラムの操作に JCL またはシステム開始タスクを使用する方法 (z/OS)』
- 486 ページの『レプリケーション・プログラムを自動的に再始動するために MVS 自動リスタート・マネージャー (ARM) を使用する方法 (z/OS)』
- 487 ページの『データ共有モードへのレプリケーション環境の移行 (z/OS)』

レプリケーション・プログラムの操作に JCL またはシステム開始タスクを使用する方法 (z/OS)

z/OS では、JCL を使用するか、システム開始タスクとして、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・モニターを操作できます。

JCL を使用したレプリケーション・プログラムの操作

このセクションでは、JCL を使用して、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・モニターを操作する方法を説明します。

DB2 DataPropagator V8 サンプル・ライブラリーには、サンプル JCL およびスクリプトが入っています。

推奨: 変更を加える前に、SASNSAMP ライブラリーから別のライブラリーにジョブをコピーしてください。SASNSAMP ライブラリーにあるサンプル・ジョブの完全なリストに関しては、プログラム・ディレクトリーを参照してください。

JCL を使用して z/OS でキャプチャー・プログラムを始動する方法

1. 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 を準備します。レプリケーション・プログラムを実行しているユーザーのホーム・ディレクトリーにある、システム規模の /etc/profile ファイルまたは .profile ファイルで TZ 環境変数を設定しなかった場合、JCL で TZ および環境変数を設定することが必要です。TZ 変数の設定に関する詳細は、「z/OS UNIX システム・サービス・ユーザーズ・ガイド」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=ASNCAP, PARM='ENVAR('TZ=PST8PDT','LANG=en_US')/  
DSN6 cold capture_schema=JFA autostop'
```

2. TSO または MVS コンソールから JCL をサブミットします。

JCL を使用して z/OS 上でアプライ・プログラムを始動する方法

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。

z/OS オペレーティング・システムの場合の、呼び出し JCL でのこの行の例を次に示します。

```
//apyasn EXEC PGM=ASNAPPLY,PARM='control_server=CTLDB1
                                DB2_SUBSYSTEM=DSN
                                apply_qual=myqual spillfile=disk'
```

UNIX および Windows オペレーティング・システムの場合の、呼び出し JCL でのこの行の例を次に示します。

```
//apyasn EXEC PGM=ASNAPPLY,PARM='control_server=CTLDB1
                                apply_qual=myqual spillfile=disk'
```

JCL を使用して z/OS 上でレプリケーション・アラート・モニターを開始する方法

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。

呼び出し JCL でのこの行の例を次に示します。

```
//monasn EXEC PGM=ASNMON,PARM='monitor_server=DSN
                                monitor_qual=monqual'
```

この DSN はサブシステム名で、monqual はモニター修飾子です。

z/OS 上で JCL を使ってバッチ・モードでプログラムを実行する方法は以下のとおりです。

該当するプログラムのライブラリー SASNSAMP で JCL をカスタマイズします。表 62 は、指定された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 62.

サンプル	プログラム
ASNSTRA	アプライ
ASNSTRC	キャプチャー
ASNSTRM	アラート・モニター

DPROPR ジョブ (キャプチャー、アプライ、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 を準備します。TSO または MVS コンソールから JCL をサブミットします。

JCL を使用して z/OS 上の開始済みプログラムを変更する方法

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・プログラムを開始した後で、MODIFY コマンドを使用して、プ

プログラムを停止したり、関連するタスクを実行することができます。MODIFY コマンドは MVS コンソールから実行する必要があります。以下の構文例に示すように、F という省略形を使用できます。

(1)

```
►►F—jobname—,—| パラメーター |—————►►
```

注:

- 1 パラメーターの記述は、329 ページの『第 18 章 SQL レプリケーション用のシステム・コマンド (Linux、UNIX、Windows、z/OS)』を参照してください。

基本的に、F *jobname* , は、 **asnacmd**、**asnccmd**、または **asnmcmd** など、実際のコマンド名を置き換えます。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 の情報については、「z/OS MVS システム・コマンド (SA88-8593)」を参照してください。

システム開始タスク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

このセクションでは、システム開始タスクを使用して、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・モニターを操作する方法を説明します。

Capture for z/OS プログラムをシステム開始タスクとして開始するためのセットアップ

1. PROCLIB にプロシージャー (*procname*) を作成します。
2. *procname* について、RACF STARTED クラス内に項目を作成します。この項目は、*procname* を、キャプチャー・プログラムを始動するために使用される RACF ユーザー ID と関連付けます。キャプチャー・プログラムを始動する前に、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. MVS システム・コンソールから、**start** *procname* を実行します。

以下のサンプル・プロシージャーはキャプチャー・プログラム用です。

```
//CAPJAYC PROC
//ASNCAP EXEC PGM=ASNCAP,REGION=M,
//PARM='V71A autostop LOGSTDOUT startmode=COLD
//capture_schema=JAY logreuse'
//STEPLIB DD DISP=SHR,DSN=DPROPR.ASN81 .SASNLOAD
//DD DISP=SHR,DSN=SYS1.SCEERUN
//DD DISP=SHR,DSN=DSN7.SDSNLOAD
//CEEDUMP DD SYSOUT=
//SYSPRINT DD SYSOUT=
//SYSTEM DD DUMMY
//
```

レプリケーション・プログラムを自動的に再始動するために MVS 自動リスタート・マネージャー (ARM) を使用する方法 (z/OS)

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・モニターと一緒に、MVS 自動リスタート・マネージャー (ARM) を使用できます。ARM は、特定のバッチ・ジョブまたは開始タスクの可用性を改善するための MVS リカバリー関数です。ジョブまたはタスクが失敗するか、ジョブやタスクを実行しているシステムに障害が発生した場合、ARM はオペレーターの介入なしに、ジョブまたはタスクを再始動できます。ARM は、処理対象の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 が使用可能な各アプリケーションは、自分自身についてユニークな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 の設定についての詳細は、「z/OS MVS プログラミング: シスプレックス・サービス・ガイド (SA88-8583)」を参照してください。

前提条件:

ARM がインストール済みであり、レプリケーション・プログラムが正しく設定されていることを確認してください。レプリケーション・プログラムと一緒に ARM を使用する場合は、レプリケーション・プログラムが APF 許可であることを確認してください。たとえば、アプライ・プログラムまたはレプリケーション・アラート・モニターで ARM を使用する場合は、適切なロード・モジュールを APF 許可ライブラリーにコピーする必要があります。(キャプチャー・プログラムは、ARM を使用するかしないかに関係なく、必ず APF 許可にする必要があります。)

ARM を構成するときには、レプリケーション・プログラムに以下のエレメント名を使用してください。

キャプチャー・プログラム

ASNTCxxxxyyyy

アプライ・プログラム

ASNTAxxxxyyyy

レプリケーション・アラート・モニター

ASNAMxxxxyyyy

この xxxx は DB2 サブシステム名であり、yyyy はデータ共有メンバー名です (後者はデータ共有構成の場合にのみ必要)。エレメント名の長さは常に 16 文字であり、ブランクが埋め込まれます。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、イベント出口は必要ありません。) 登録済みプログラムが異常終了した場合 (たとえばセグメント違反の発生)、ARM はユーザーの代わりに登録済みプログラムを再始

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（たとえば、STOP コマンドによる終了）、または無効な登録を検出した場合は、登録解除されます。

ヒント: パラメーター NOTERM=Y を指定してキャプチャー・プログラムまたはアプライ・プログラムを始動すると、DB2 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 が再始動されるまでは、データをキャプチャーしません。

データ共有モードへのレプリケーション環境の移行 (z/OS)

キャプチャー・プログラムが非データ共有モードで実行されているときに、システムをデータ共有モードに移行するときには、**ASNPLXFY** ユーティリティを一度実行して、システムが Sysplex で実行されるように、準備する必要があります。キャプチャー・プログラムが正しい LRSN から開始するように、キャプチャー・プログラムをウォーム・スタートする前に、このユーティリティをデータ共有構成で実行してください。このユーティリティは、再始動 (IBMSNAP_RESTART) 表の中のデータを移行します。非データ共有のログ・シーケンス番号 (RBA) は、データ共有環境での同等のシーケンス番号 (LRSN) に変換されます。

前提条件:

キャプチャー・プログラムを実行するために使用したものと同一ユーザー ID を使用するか、同じ特権を持つものを使用します。**ASNPLXFY** ユーティリティが APF 許可であることを確認してください。**ASNPLXFY** プランは、サブシステムにバインドする必要があります。また、サブシステムはデータ共有モードで実行されている必要があります。このユーティリティのバインディングの詳細は、プログラム・ディレクトリーを参照してください。

手順:

USS データ共有環境で **ASNPLXFY** ユーティリティを実行するには、以下のようになります。

1. キャプチャー・プログラムを停止してください。
2. コマンド行から以下のコマンドを入力します。

```
ASNPLXFY yoursystem captureschema
```

サブシステムの名前が必要です。キャプチャー・スキーマはオプションです。デフォルトのキャプチャー・スキーマは ASN です。

3. キャプチャー・プログラムをウォーム・スタートします。

第 21 章 Windows Service Control Manager を使用して SQL レプリケーションのシステム・コマンドを発行する方法 (Windows)

このセクションでは、Windows オペレーティング・システムの場合に、レプリケーション・プログラムを始動するサービスを作成する方法を説明します。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、およびモニター・コントロール・サーバーごとにサービスを作成できます。サービスは他の DB2 サービスとグループ化されます。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。

- 『レプリケーション・サービスの作成』
- 490 ページの『レプリケーション・サービスの操作』
- 491 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

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。

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、366 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。

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ログ・ファイルを調べてください。デフォルトでは、ログ・ファイルは **DB2PATH** 環境変数で指定されたディレクトリ内にあります。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。

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。

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、各データベースがアプライ・コントロール・サーバー、キャプチャー・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。

ユーザーが作成したレプリケーション・サービスは、自動モードで SCM に追加され、サービスが開始されます。Windows はユニーク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。

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ます。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
```

ここで、

- *instance* は DB2 インスタンスの名前です。
- *alias* 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、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です。
- *program* は以下の値のいずれかです。CAP (キャプチャー・プログラム)、APP (アプライ・プログラム)、または MON (レプリケーション・アラート・モニター・プログラム)
- *qualifier_or_schema* は、アプライ修飾子、モニター修飾子、またはキャプチャー・スキーマという ID のいずれかです。

インスタンス: 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 を処理しているキャプチャー・プログラムのものです。

```
DB2.INST1.DB1.CAP.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。以下に例を示します。

```
DB2 - INST1 DB1 CAPTURE 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、Service Control Manager (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。

レプリケーション・サービスの操作

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。

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。

- SCM
- **net stop** コマンド

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 (**asnacmd**、**asnccmd**、または **asnmcmd**) を使用してプログラムを停止すると、プログラムを始動するために使用されたサービスは、ユーザーが明示的に停止するまで、実行を続けます。

レプリケーション・コマンドのサービスを開始するには、以下の方法のいずれかを使用します。

- SCM
- **net start** コマンド

重要: サービスからレプリケーション・プログラムを始動した場合、同じスキーマまたは修飾子を使用してプログラムを始動しようとすると、エラーが発生します。

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だけの場合は、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。また、サービスによって開始されるプログラムの始動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、新しい始動パラメーターを使用する新しいサービスを作成する必要があります。

レプリケーション・コマンドのサービスをドロップするには、**asnsdrop** コマンドを使用します。

関連資料:

- 366 ページの『**asnsrct**: 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』
- 370 ページの『**asnsdrop**: DB2 レプリケーション・サービスのドロップ』

第 22 章 各種オペレーティング・システムでの SQL レプリケーション・プログラムのスケジューリング

オペレーティング・システム (OS) のコマンドを使用してあらかじめ指定しておいた時刻にキャプチャー・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・プログラムが始動するようにスケジューリングできます。

UNIX および Linux 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

プログラムを特定の時刻に開始するには **at** コマンドを使用します。たとえば、次のコマンドを使用すると、金曜日の午後 3 時にプログラムが開始されます。

キャプチャー・プログラムのスケジューリング:

```
at 3pm Friday asncap autoprune=n
```

アプライ・プログラムのスケジューリング:

```
at 3pm Friday asnapply applyqual=myqual
```

レプリケーション・アラート・モニター・プログラムのスケジューリング:

```
at 3pm Friday asnmon monitor_server=db2srv1 monitor_qualifier=mymon
```

Windows 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

Windows Service Control Manager を使用していない場合は、**AT** コマンドを使用してこのプログラムを特定の時刻に開始できます。**AT** コマンドを入力する前に、Windows スケジュール・サービスを始動してください。

以下の例では、各プログラムは 15 時 (午後 3 時) に始動されます。

キャプチャー・プログラムのスケジューリング:

```
c:>%AT 15:00 /interactive "c:%SQLLIB%BIN%db2cmd.exe c:%CAPTURE%asncap.exe"
```

アプライ・プログラムのスケジューリング:

```
c:>%AT 15:00 /interactive "c:%SQLLIB%BIN%db2cmd.exe  
c:%SQLLIB%BIN%asnapply.exe control_server=cnt1db apply_qual=qualid1"
```

レプリケーション・アラート・モニター・プログラムのスケジューリング:

```
c:>%AT 15:00 /interactive "c:%SQLLIB%BIN%db2cmd.exe  
c:%CAPTURE%asnmon.exe monitor_server=db2srv1 monitor_qualifier=mymon"
```

z/OS 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

指定した時点で Capture for z/OS を始動するには、**\$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。

z/OS 上でプログラムをスケジューリングする場合

1. z/OS 用のプログラムを呼び出すプロシージャーを PROCLIB に作成する。
2. ICHRIN03 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、プロシージャーをユーザー 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。

\$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。

OS/400 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

特定の時刻にアプライ・プログラムを始動するには、**ADDJOBSCDE** コマンドを使用します。

SBMJOB コマンドを使用して、OS/400 でのキャプチャー・プログラムの始動をスケジューリングします。

```
SBMJOB CMD('STRDPRCAP...')SCDDATE(...).SCDTIME(...)
```

第 23 章 SQL レプリケーション・コンポーネントの通信方法

レプリケーション・コンポーネントはお互いに独立して実行され、各コンポーネントがコントロール表に保管する情報を頼りにお互いに通信します。DB2® レプリケーションには次の 4 つのコンポーネントがあります。

- レプリケーション・センター
- キャプチャー・プログラムまたはトリガー
- 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・センターは、登録されたソース、サブスクリプション・セット、およびアラート条件についての初期情報をコントロール表に保管します。キャプチャー・プログラム、アプライ・プログラム、およびキャプチャー・トリガーは、コントロール表を更新することにより、レプリケーションの進行状況を示し、変更の処理を調整します。レプリケーション・アラート・モニターは、キャプチャー・プログラム、アプライ・プログラム、およびキャプチャー・トリガーにより更新されたコントロール表を読み取り、サーバーでの問題や進行状況を理解します。

レプリケーション・センター、キャプチャー・プログラムまたはトリガー、およびアプライ・プログラム

レプリケーション・ソースとして、表、ビュー、またはニックネームを登録すると、レプリケーション・センターは SQL スクリプトを作成します。この SQL スクリプトは、このソースの情報を、すべての登録情報を含む登録表 (IBMSNAP_REGISTER) が入ったレプリケーション・コントロール表に保管します。レプリケーション・センターにより生成される SQL スクリプトは、登録されたソース用の CD 表も作成します。

IBMSNAP_REGISTER 表には、登録されたソースごとに 1 つの行があり、また、登録されたビュー内の基本表ごとに 1 つの行があります。この表は、登録されたそれぞれのソースについて、次の種類の情報を含んでいます。

- ソース表のスキーマ名および名前
- 登録されたそれぞれのソース表の構造タイプ
- CD 表のスキーマ名および名前
- 登録されたビューの場合、このビュー内の基本表の CD 表の名前 (基本表が登録されている場合)
- 内部 CCD 表がある場合は、そのスキーマ名および名前
- Update-anywhere ソースの競合検出レベル

キャプチャー・プログラムとアプライ・プログラムは、IBMSNAP_REGISTER 表の情報を使用して、自分の該当する状況をお互いに知らせます。この表は関連する情報用にいくつかの列を持っています。この表についての詳細は、532 ページの『*schema.IBMSNAP_REGISTER*』を参照してください。

OS/400® ソースの場合、リモート側でジャーナルに記録された表を含め、IBMSNAP_REGISTER 表への拡張があり、IBMSNAP_REG_EXT には、iSeries™ システムにユニークな追加の情報 (ジャーナル・ライブラリーやジャーナル名など) が含まれます。

サブスクリプション・セットを作成し、これにメンバーを追加すると、レプリケーション・センターは SQL スクリプトを作成します。このスクリプトは、このサブスクリプション・セット用の情報を複数のレプリケーション・コントロール表に保管します。これらの表にはすべてのサブスクリプション・セット情報が含まれ、その内訳は、サブスクリプション・セット表 (IBMSNAP_SUBS_SET)、サブスクリプション・セット・メンバー表 (IBMSNAP_SUBS_MEMBR)、サブスクリプション・セット列の表 (IBMSNAP_SUBS_COLS)、およびサブスクリプション・セット・ステートメント表 (IBMSNAP_SUBS_STMTS) です。レプリケーション・センターにより生成される SQL スクリプトは、ターゲット表も作成します (まだ存在しない場合)。

メインのサブスクリプション・セット表である IBMSNAP_SUBS_SET には、それぞれのサブスクリプション・セットごとに 1 つの行があります。この表は、それぞれのサブスクリプション・セットについて、次の種類の情報を含んでいます。

- アプライ修飾子
- サブスクリプション・セットの名前
- サブスクリプション・セットのタイプ: 読み取り専用か、または読み取り/書き込み (Update-anywhere)
- ソースおよびターゲット・データベースの名前と別名
- サブスクリプション・セットを処理するタイミング
- サブスクリプション・セットの現在の状況

この表は関連する情報用にいくつかの列を持っています。この表についての詳細は、568 ページの『ASN.IBMSNAP_SUBS_SET』を参照してください。

その他のサブスクリプション・セット表には、サブスクリプション・セットのメンバー、列、およびセットを使用して処理される SQL ステートメント (またはストアド・プロシージャ) についての情報が含まれます。

キャプチャー・プログラムおよびアプライ・プログラム

キャプチャー・プログラムは、レプリケーション・コントロール表のいくつかを使用して、ソース・データベースにどのような変更がなされたかを示し、アプライ・プログラムはそれらのコントロール表の値を使用して、ターゲット・データベースに何をコピーする必要があるかを検出します。キャプチャー・プログラムは、アプライ・プログラムから指示を受けない限り何も情報をキャプチャーせず、また、アプライ・プログラムは、レプリケーション・ソースおよびそれに関連するサブスクリプション・セットが定義されるまで、キャプチャー・プログラムに変更のキャプチャーを開始する指示を出しません。

以下に示す処理は、データ保全性を保つために、通常のレプリケーション・シナリオでは、アプライ・プログラムとキャプチャー・プログラムがどのように通信するかを説明しています。

ソース・データベースからデータを取り込む

1. キャプチャー・プログラムは始動時に `IBMSNAP_REGISTER` 表を読み取り、登録されたレプリケーション・ソースの中のどれについて変更をキャプチャーする必要があるかを判断し、登録情報をメモリーに保留します。
2. キャプチャー・プログラムは `DB2` ログまたはジャーナルを継続して読み取り、登録されたソース表またはビューの変更レコード (`INSERT`、`UPDATE`、および `DELETE`) を検出します。また、アプライ・プログラムまたはユーザーによって開始されたシグナル・アクションを拾い出すため、シグナル表 (`IBMSNAP_SIGNAL`) への挿入も検出します。アプライ・プログラムが `CAPSTART` シグナルを `IBMSNAP_SIGNAL` 表に挿入し、コミットされたシグナルをキャプチャー・プログラムが検出すると、キャプチャー・プログラムは登録を開始し、関連するソースの変更のキャプチャーを開始します。
3. 登録されたソースの変更のキャプチャーをキャプチャー・プログラムが開始すると、キャプチャー・プログラムは、`DB2` ログまたはジャーナルで検出したコミット済みの変更ごとに、`CD` 表に 1 行を書き込みます (更新 `ID` を `DELETE` または `INSERT` ステートメントとして保管するように指定した場合は 2 行)。キャプチャー・プログラムは、コミットされていない変更については、変更がコミットされるかまたは打ち切られるまで、メモリーに保持します。外部 `CCD` 表でない、登録済みのレプリケーション・ソースは、それぞれ 1 つの関連する `CD` 表を持ちます。
4. コミット・インターバルごとに、キャプチャー・プログラムは `CD` および `UOW` 表に書き込んだデータをコミットし、さらに `IBMSNAP_REGISTER` 表を更新して、どの `CD` 表に新しくコミットされた変更があるかを示すフラグを付けます。

データをターゲット・データベースに適用する

5. 新しく定義されたサブスクリプション・セットについてはすべて、アプライ・プログラムは最初に、キャプチャー・プログラムに変更をキャプチャーするようにシグナルを出します。その後、セットのそれぞれのメンバーについてフル・リフレッシュを実行します (不完全なターゲット表は除く)。
6. レプリケーションの対象として適格なサブスクリプション・セットがある場合、アプライ・プログラムは `IBMSNAP_REGISTER` 表を調べて、複製しなければならない変更があるかどうかを判別します。
7. アプライ・プログラムは `CD` 表からターゲット表に変更をコピーします。
8. アプライ・プログラムは `IBMSNAP_SUBS_SET` 表を更新し、アプライ・プログラムがそれぞれのサブスクリプション・セット用にコピーしたデータの量を記録します。
9. アプライ・プログラムは、`CD` 表から変更を読み取った時点を示す値で、整理セット表 (`IBMSNAP_PRUNE_SET`) を更新します。

CD 表の整理

10. キャプチャー・プログラムが `CD` 表の整理を行う場合、`IBMSNAP_PRUNE_SET` 表にある情報を使用して、どの変更が適用されたかを判別し、これらの既にレプリケーション済みの変更を `CD` 表から削除します。

キャプチャー・トリガーおよびアプライ・プログラム

キャプチャー・トリガーは、レプリケーション・コントロール表のいくつかを使用して、ソース・データベースにどのような変更がなされたかを示し、アプライ・プログラムはそれらのコントロール表の値を使用して、ターゲット・データベースに何をコピーする必要があるかを検出します。

キャプチャー・トリガーは情報のキャプチャーを即時に開始します。キャプチャー・プログラムと異なり、アプライ・プログラムからのシグナルを待つことはしません。

以下に示す処理は、データ保全性を保つために、通常のレプリケーション・シナリオでは、キャプチャー・トリガーとアプライ・プログラムがどのように通信するかを説明しています。

ソースからデータを取り込む

1. 登録されたレプリケーション・ソース表で DELETE、UPDATE、または INSERT 操作が行われた場合はいつでも、キャプチャー・トリガーはその変更を、そのソース表用の CCD 表に記録します。

データをターゲットに適用する

2. 新しく定義されたサブスクリプション・セットについてはすべて、アプライ・プログラムは最初に、キャプチャー・トリガーにシグナルを出し、どの時点から変更データのフェッチを開始するかを示す、有効な開始点を CCD 表に記録するように指示します。その後、セットのそれぞれのメンバーについてフル・リフレッシュを実行します (不完全なターゲット表は除く)。
3. アプライ・プログラムは、DB2 以外のリレーショナル・ソース用のサブスクリプション・セットを処理する時には、登録同期化表 (IBMSNAP_REG_SYNCH) を更新します。この更新により、この表の UPDATE トリガーが着火されます。このトリガーは、IBMSNAP_REGISTER 表の SYNCHPOINT 値を更新し、ターゲットにコピーした CCD 表内の最高の SYNCHPOINT 値を記録します。次のサイクルで、アプライ・プログラムは、この SYNCHPOINT より小さいか等しい SYNCHPOINT 値を持つ、CCD 表内の新しいデータを処理します。
IBMSNAP_REG_SYNCH 表は DB2 データベースにはないので、アプライ・プログラムはレプリケーション・センターが作成したニックネームを使用して表に書き込みます。
4. アプライ・プログラムは IBMSNAP_REGISTER 表を調べて、複製しなければならない変更があるかどうかを判別します。
5. アプライ・プログラムは CCD 表から得られる変更をターゲット表にコピーします。
6. アプライ・プログラムはサブスクリプション・セット表 (IBMSNAP_SUBS_SET) を更新し、アプライ・プログラムがそれぞれのサブスクリプション・セット用にコピーしたデータの量を記録します。
7. アプライ・プログラムは、CCD 表から変更を読み取った時点を示す値で、登録されたそれぞれのソースの整理コントロール表 (IBMSNAP_PRUNCNTL) を更新します。

CCD 表の整理

8. IBMSNAP_PRUNCNTL 表の UPDATE トリガーは、ソース・データベースにあるすべての CCD 表を調べて、すでに複製された変更を CCD 表から削除します。

レプリケーション・センターおよびレプリケーション・アラート・モニター

アラート条件を定義する際に、この条件が起こった時に誰に通知するかを定義します。レプリケーション・センターは、このアラート条件とその連絡先情報をレプリケーション・コントロール表に保管する SQL スクリプトを作成します。これらのコントロール表には、すべてのアラート条件および通知情報が含まれ、その内訳は、モニター条件表 (IBMSNAP_CONDITIONS)、モニター連絡先表 (IBMSNAP_CONTACTS)、モニター・グループ表 (IBMSNAP_GROUPS)、およびモニター・グループ連絡先表 (IBMSNAP_CONTACTGRP) です。

メインのモニター・アラート表である、モニター条件表には、モニターの対象となるそれぞれの条件ごとに 1 行登録されています。この表は、それぞれのアラート条件について、次の種類の情報を保持します。

- モニター修飾子
- モニターしたいキャプチャー・サーバーまたはアプライ・サーバーの名前および別名
- モニターしたいコンポーネント (キャプチャー・プログラムまたはアプライ・プログラム)
- キャプチャー・スキーマまたはアプライ修飾子
- サブスクリプション・セットの名前 (セットをモニターする場合)
- モニターしたいアラート条件
- 条件が起こった時に知らせる連絡先

この表は関連する情報用にいくつかの列を持っています。この表についての詳細は、578 ページの『IBMSNAP_CONDITIONS 表』を参照してください。

レプリケーション・アラート・モニターのその他の表には、アラート条件が起こった場合に誰に知らせるか (個別の連絡先または連絡先のグループ)、その連絡先にどのような方法で知らせるか (E メールやページャー)、およびその条件が起こり続ける場合、どの程度の頻度で連絡先に知らせるか、といった情報が含まれます。

レプリケーション・アラート・モニター、キャプチャー・プログラム、およびアプライ・プログラム

レプリケーション・アラート・モニターは、キャプチャー・コントロール表のいくつかを使用してキャプチャー・プログラムをモニターし、アプライ・コントロール表のいくつかを使用してアプライ・プログラムをモニターします。モニターは、それぞれのキャプチャー・コントロール・サーバーまたはアプライ・コントロール・サーバーで、何をモニターするかによって、異なるレプリケーション・コントロール表から読み取ります。レプリケーション・アラート・モニターは、キャプチャー・プログラムまたはアプライ・プログラムへの介入や通信は行いません。

次の処理は、レプリケーション・アラート・モニターがキャプチャー・プログラムやアプライ・プログラムの条件をどのようにモニターし、アラート条件が起こった場合にどのように連絡先に知らせるかを説明しています。

1. レプリケーション・アラート・モニターは、モニター条件表 (IBMSNAP_CONDITIONS) 内のそれぞれの条件 (モニター修飾子について) のアラート条件および連絡先を読み取ります。
2. アラート条件が定義されているキャプチャー・コントロール・サーバーまたはアプライ・コントロール・サーバーごとに、レプリケーション・アラート・モニターは次のタスクを行います。
 - a. レプリケーション・アラート・モニターはサーバーに接続し、そのサーバーの各アラート条件に関連付けられたレプリケーション・コントロール表を読み、条件のいずれかを満たすかどうかを調べます。
 - b. いずれかの条件を満たした場合、レプリケーション・アラート・モニターはその条件に関するデータをメモリーに保管し、そのサーバーの残りのアラート条件の処理を続けます。
 - c. そのサーバーのすべてのアラート条件の処理が終了したら、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・サーバーまたはアプライ・コントロール・サーバーを切断し、アラートをモニター・アラート表 (IBMSNAP_ALERTS) に挿入し、その条件について連絡先に知らせます。

関連概念:

- 265 ページの『第 15 章 SQL レプリケーションの場合のレプリケーション・センターの使用法』

関連資料:

- 511 ページの『アプライ・コントロール・サーバーで使用される表のリスト』
- 508 ページの『キャプチャー・コントロール・サーバーで使用される表のリスト』
- 512 ページの『モニター・コントロール・サーバーのコントロール表のリスト』

第 24 章 SQL レプリケーションの表構造

この章では、各サーバー (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、モニター・コントロール・サーバー、およびターゲット・サーバー) のレプリケーションに使用される、リレーショナル・データベース表について説明します。また、以下のような、表を参照する 3 つの方法を紹介します。

- 『表の概観』 のセクションには早見表があります。これには、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、およびモニター・コントロール・サーバー用の表のリスト、各表の列、そして各表の索引が含まれます。
- 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。
 - 508 ページの『キャプチャー・コントロール・サーバーで使用される表のリスト』
 - 511 ページの『アプライ・コントロール・サーバーで使用される表のリスト』
 - 512 ページの『モニター・コントロール・サーバーのコントロール表のリスト』
 - 514 ページの『ターゲット・サーバーで使用される表のリスト』
- 各サーバー上の表に関する詳細な記述、および各表の列の記述は、以下を参照してください。
 - 514 ページの『キャプチャー・コントロール・サーバー上の表、およびその列に関する記述』
 - 548 ページの『アプライ・コントロール・サーバー上の表、およびその列に関する記述』
 - 576 ページの『モニター・コントロール・サーバー上の表、およびその列に関する記述』
 - 592 ページの『ターゲット・サーバー上の表、およびその列に関する記述』

各セクションでは、コントロール表は実表名 (たとえば、IBMSNAP_APPLYTRACE) のアルファベット順にリストされ、ターゲット表は、英語の表名 (たとえば、replica table) のアルファベット順にリストされています。各表の列は、表内に現れる順にリストされています。

コントロール表の中には、更新に SQL を使用しない ことを必要とするものがあります (詳細については、特定の表の記述を参照のこと)。コントロール表を正しく変更しないと、予期しない結果や、データの損失、レプリケーション・パフォーマンスの低下などの問題を引き起こす可能性があります。

表の概観

502 ページの図 15、503 ページの図 16、および 504 ページの図 17 は、キャプチャー・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。506 ページの図 19 および 505 ページの図 18 は、アプライ・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。507 ページの図 20

および 508 ページの図 21 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。

キャプチャー・コントロール・サーバーで使用されるコントロール表 (1/3)

<p>schema.IBMSNAP_AUTHTKN</p> <p>OS/400 のみ (JRN_LIB, JRN_NAME)</p> <p>APPLY_QUAL CHAR(18) NOT NULL IBMSNAP_AUTHTKN CHAR(26) NOT NULL JRN_LIB CHAR(10) NOT NULL JRN_NAME CHAR(10) NOT NULL IBMSNAP_LOGMARKER TIMESTAMP NOT NULL</p>	<p>schema.IBMSNAP_CAPPARMS</p> <p>(索引なし)</p> <p>RETENTION_LIMIT INT LAG_LIMIT INT COMMIT_INTERVAL INT PRUNE_INTERVAL INT TRACE_LIMIT INT MONITOR_LIMIT INT MONITOR_INTERVAL INT MEMORY_LIMIT SMALLINT REMOTE_SRC_SERVER CHAR(18) AUTOPRUNE CHAR(1) TERM CHAR(1) AUTOSTOP CHAR(1) LOGREUSE CHAR(1) LOGSTDOUT CHAR(1) SLEEP_INTERVAL SMALLINT CAPTURE_PATH VARCHAR(1040) STARTMODE VARCHAR(10)</p>
<p>schema.IBMSNAP_CAPENQ</p> <p>UNIX、Windows、および z/OS のみ (索引なし)</p> <p>LOCK_NAME CHAR(9)</p>	<p>schema.IBMSNAP_CAPTRACE</p> <p>UNIX、Windows、および z/OS のみ (TRACE_TIME)</p> <p>OPERATION CHAR(8) NOT NULL TRACE_TIME TIMESTAMP NOT NULL DESCRIPTION VARCHAR(1024) NOT NULL</p>
<p>schema.IBMSNAP_CAPMON</p> <p>(MONITOR_TIME)</p> <p>MONITOR_TIME TIMESTAMP NOT NULL RESTART_TIME TIMESTAMP NOT NULL CURRENT_MEMORY INT NOT NULL CD_ROWS_INSERTED INT NOT NULL RECAP_ROWS_SKIPPED INT NOT NULL TRIGR_ROWS_SKIPPED INT NOT NULL CHG_ROWS_SKIPPED INT NOT NULL TRANS_PROCESSED INT NOT NULL TRANS_SPILLED INT NOT NULL MAX_TRANS_SIZE INT NOT NULL LOCKING_RETRIES INT NOT NULL JRN_LIB CHAR(10) JRN_NAME CHAR(10) LOGREADLIMIT INT NOT NULL CAPTURE_IDLE INT NOT NULL SYNCHTIME TIMESTAMP NOT NULL</p>	<p>schema.IBMSNAP_CAPTRACE</p> <p>OS/400 (TRACE_TIME)</p> <p>OPERATION CHAR(8) NOT NULL TRACE_TIME TIMESTAMP NOT NULL JOB_NAME CHAR(26) NOT NULL JOB_STR_TIME TIMESTAMP NOT NULL DESCRIPTION VARCHAR(298) NOT NULL</p>
<p>ASN.IBMSNAP_CAPSCHEMAS</p> <p>UNIX、Windows、および z/OS のみ (CAP_SCHEMA_NAME)</p> <p>CAP_SCHEMA_NAME VARCHAR(30)1</p>	<p>schema.IBMSNAP_PARTITIONINFO</p> <p>(PARTITIONID, USAGE)</p> <p>PARTITIONID INT NOT NULL USAGE CHAR(1) SEQUENCE CHAR(10) FOR BIT DATA NOT NULL STATUS CHAR(1) LAST_UPDATE TIMESTAMP</p>
<p>schema.IBMSNAP_CAPSCHEMAS</p> <p>OS/400 のみ (CAP_SCHEMA_NAME)</p> <p>CAP_SCHEMA_NAME VARCHAR(30) STATUS CHAR(1)</p>	<p>schema.IBMSNAP_PRUNE_LOCK</p> <p>(索引なし)</p> <p>DUMMY CHAR(1)</p>

1DB2 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 15. キャプチャー・コントロール・サーバーで使用される表。: これらの表は、キャプチャー・コントロール・サーバーで、キャプチャー・プログラム、アプライ・プログラム、およびキャプチャー・トリガーによって使用されます。各表の主索引を構成する列は、表名の下の括弧内に示されています。

キャプチャー・コントロール・サーバーで使用されるコントロール表 (2/3)

schema.IBMSNAP_PRUNE_SET

(TARGET_SERVER, APPLY_QUAL, SET_NAME)	
TARGET_SERVER	CHAR(18) NOT NULL
APPLY_QUAL	CHAR(18) NOT NULL
SET_NAME	CHAR(18) NOT NULL
SYNCHTIME	TIMESTAMP
SYNCHPOINT	CHAR(10) FOR BIT DATA NOT NULL

schema.IBMSNAP_PRUNCNTL

(SOURCE_OWNER, SOURCE_TABLE, SOURCE_VIEW_QUAL, APPLY_QUAL, SET_NAME, TARGET_SERVER, TARGET_TABLE, TARGET_OWNER)	
TARGET_SERVER	CHAR(18) NOT NULL
TARGET_OWNER	VARCHAR(30)1 NOT NULL
TARGET_TABLE	VARCHAR(128)2 NOT NULL
SYNCHTIME	TIMESTAMP
SYNCHPOINT	CHAR(10) FOR BIT DATA
SOURCE_OWNER	VARCHAR(30)1 NOT NULL
SOURCE_TABLE	VARCHAR(128)2 NOT NULL
SOURCE_VIEW_QUAL	SMALLINT NOT NULL
APPLY_QUAL	CHAR(18) NOT NULL
SET_NAME	CHAR(18) NOT NULL
CNTL_SERVER	CHAR(18) NOT NULL
TARGET_STRUCTURE	SMALLINT NOT NULL
CNTL_ALIAS	CHAR(8)
PHYS_CHANGE_OWNER	VARCHAR(30)1
PHYS_CHANGE_TABLE	VARCHAR(128)2
MAP_ID	VARCHAR(10) NOT NULL

schema.IBMSNAP_REG_EXT

OS/400 のみ (VERSION, SOURCE_OWNER, SOURCE_TABLE, SOURCE_VIEW_QUAL)	
VERSION	INT NOT NULL
SOURCE_OWNER	VARCHAR(30) NOT NULL
SOURCE_TABLE	VARCHAR(128) NOT NULL
SOURCE_NAME	CHAR(10)
SOURCE_MBR	CHAR(10)
SOURCE_TABLE_RDB	CHAR(18)
JRN_LIB	CHAR(10)
JRN_NAME	CHAR(10)
FR_START_TIME	TIMESTAMP
SOURCE_VIEW_QUAL	SMALLINT NOT NULL
CMT_BEHAVIOR_CASE	SMALLINT NOT NULL WITH DEFAULT
MAX_ROWS_BTWN_CMTS	SMALLINT NOT NULL WITH DEFAULT

schema.IBMSNAP_REGISTER

(SOURCE_OWNER, SOURCE_TABLE, SOURCE_VIEW_QUAL)	
SOURCE_OWNER	VARCHAR(30)1 NOT NULL
SOURCE_TABLE	VARCHAR(128)2 NOT NULL
SOURCE_VIEW_QUAL	SMALLINT NOT NULL
GLOBAL_RECORD	CHAR(1) NOT NULL
SOURCE_STRUCTURE	SMALLINT NOT NULL
SOURCE_CONDENSED	CHAR(1) NOT NULL
SOURCE_COMPLETE	CHAR(1) NOT NULL
CD_OWNER	VARCHAR(30)1
CD_TABLE	VARCHAR(128)2
PHYS_CHANGE_OWNER	VARCHAR(30)1
PHYS_CHANGE_TABLE	VARCHAR(128)2
CD_OLD_SYNCHPOINT	CHAR(10) FOR BIT DATA
CD_NEW_SYNCHPOINT	CHAR(10) FOR BIT DATA
DISABLE_REFRESH	SMALLINT NOT NULL
CCD_OWNER	VARCHAR(30)1
CCD_TABLE	VARCHAR(128)2
CCD_OLD_SYNCHPOINT	CHAR(10) FOR BIT DATA
SYNCHPOINT	CHAR(10) FOR BIT DATA
SYNCHTIME	TIMESTAMP
CCD_CONDENSED	CHAR(1)
CCD_COMPLETE	CHAR(1)
ARCH_LEVEL	CHAR(4) NOT NULL
DESCRIPTION	CHAR(254)
BEFORE_IMG_PREFIX	VARCHAR(4)
CONFLICT_LEVEL	CHAR(1)
CHG_UPD_TO_DEL_INS	CHAR(1)
CHGONLY	CHAR(1)
RECAPTURE	CHAR(1)
OPTION_FLAGS	CHAR(4) NOT NULL
STOP_ON_ERROR	CHAR(1) WITH DEFAULT
STATE	CHAR(1) WITH DEFAULT
STATE_INFO	CHAR(8)

schema.IBMSNAP_REG_SYNCH

非 DB2 リレーショナルのみ (TRIGGER_ME)	
TRIGGER_ME	CHAR(1) NOT NULL

1DB2 for z/OS V8 互換モードがそれ以前の場合 VARCHAR(30)、DB2 for z/OS V8 新規関数モードの場合 VARCHAR(128)。

DB2 for z/OS V8 互換モードがそれ以前の場合 VARCHAR(18)、DB2 for z/OS V8 新規関数モードの場合 VARCHAR(128)。

図 16. キャプチャー・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、キャプチャー・コントロール・サーバーで、キャプチャー・プログラム、アプライ・プログラム、およびキャプチャー・トリガーによって使用されます。各表の主索引を構成する列は、表名の下の括弧内に示されています。

キャプチャー・コントロール・サーバーで使用されるコントロール表 (3/3)

schema.IBMSNAP_RESTART		schema.IBMSNAP_SIGNAL	
UNIX、Windows、および z/OS のみ (索引なし)		(SIGNAL_TIME)	
MAX_COMMITSEQ	CHAR(10) FOR BIT DATA	SIGNAL_TIME	TIMESTAMP NOT NULL
MAX_COMMIT_TIME	NOT NULL	SIGNAL_TYPE	WITH DEFAULT
MIN_INFLIGHTSEQ	TIMESTAMP NOT NULL	SIGNAL_SUBTYPE	VARCHAR(30) NOT NULL
	CHAR(10) FOR BIT DATA	SIGNAL_INPUT_IN	VARCHAR(30)
CURR_COMMIT_TIME	NOT NULL	SIGNAL_STATE	VARCHAR(500)
CAPTURE_FIRST_SEQ	TIMESTAMP NOT NULL	SIGNAL_LSN	CHAR(1) NOT NULL
	CHAR(10) FOR BIT DATA		CHAR(10) FOR BIT DATA
	NOT NULL		
OS/400 のみ (JRN_LIB, JRN_NAME)		schema.IBMSNAP_UOW	
		(IBMSNAP_COMMITSEQ, IBMSNAP_LOGMARKER)	
MAX_COMMITSEQ	CHAR(10) FOR BIT DATA	IBMSNAP_UOWID	CHAR(10) FOR BIT DATA
MAX_COMMIT_TIME	NOT NULL	IBMSNAP_COMMITSEQ	NOT NULL
MIN_INFLIGHTSEQ	TIMESTAMP NOT NULL	IBMSNAP_LOGMARKER	CHAR(10) FOR BIT DATA
	CHAR(10) FOR BIT DATA	IBMSNAP_AUTHTKN	NOT NULL
CURR_COMMIT_TIME	NOT NULL	IBMSNAP_AUTHID	TIMESTAMP NOT NULL
CAPTURE_FIRST_SEQ	TIMESTAMP NOT NULL	IBMSNAP_REJ_CODE	VARCHAR(30) NOT NULL
UID	CHAR(10) FOR BIT DATA		VARCHAR(30)1 NOT NULL
SEQNBR	NOT NULL	IBMSNAP_APPLY_QUAL	CHAR(1) NOT NULL
JRN_LIB	INTEGER NOT NULL		WITH DEFAULT
JRN_NAME	BIGINT NOT NULL		CHAR(18) NOT NULL
STATUS	CHAR(10) NOT NULL		WITH DEFAULT
	CHAR(10) NOT NULL		
	CHAR(1)		
schema.IBMSNAP_SEQTABLE		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)。	
Informix のみ (SEQ)		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)。	
SEQ	INTEGER NOT NULL		

図 17. キャプチャー・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、キャプチャー・コントロール・サーバーで、キャプチャー・プログラム、アプライ・プログラム、およびキャプチャー・トリガーによって使用されます。各表の主索引を構成する列は、表名の下の括弧内に示されています。

アプライ・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_APPLYTRAIL

(LASTRUN, APPLY_QUAL)	
APPLY_QUAL	CHAR(18) NOT NULL
SET_NAME	CHAR(18) NOT NULL
SET_TYPE	CHAR(1) NOT NULL
WHOS_ON_FIRST	CHAR(1) NOT NULL
ASNLOAD	CHAR(1)
FULL_REFRESH	CHAR(1)
EFFECTIVE_MEMBERS	INT
SET_INSERTED	INT NOT NULL
SET_DELETED	INT NOT NULL
SET_UPDATED	INT NOT NULL
SET_REWORKED	INT NOT NULL
SET_REJECTED_TRXS	INT NOT NULL
STATUS	SMALLINT NOT NULL
LASTRUN	TIMESTAMP NOT NULL
LASTSUCCESS	TIMESTAMP
SYNCHPOINT	CHAR(10) FOR BIT DATA
SYNCHTIME	TIMESTAMP
SOURCE_SERVER	CHAR (18) NOT NULL
SOURCE_ALIAS	CHAR(8)
SOURCE_OWNER	VARCHAR(30)1
SOURCE_TABLE	VARCHAR(128)2
SOURCE_VIEW_QUAL	SMALLINT
TARGET_SERVER	CHAR(18) NOT NULL
TARGET_ALIAS	CHAR(8)
TARGET_OWNER	VARCHAR(30)1 NOT NULL
TARGET_TABLE	VARCHAR(128)2 NOT NULL
CAPTURE_SCHEMA	VARCHAR(30)1 NOT NULL
TGT_CAPTURE_SCHEMA	VARCHAR(30)1
FEDERATED_SRC_SRVR	VARCHAR(18)
FEDERATED_TGT_SRVR	VARCHAR(18)
JRN_LIB	CHAR(10)
JRN_NAME	CHAR(10)
COMMIT_COUNT	SMALLINT
OPTION_FLAGS	CHAR(4) NOT NULL
EVENT_NAME	CHAR(18)
ENDTIME	TIMESTAMP NOT NULL WITH DEFAULT
SOURCE_CONN_TIME	TIMESTAMP
SQLSTATE	CHAR(5)
SQLCODE	INT
SQLERRP	CHAR(8)
SQLERRM	VARCHAR(70)
APPERRM	VARCHAR(760)

ASN.IBMSNAP_APPENQ

(APPLY_QUAL)	
APPLY_QUAL	CHAR(18)

ASN.IBMSNAP_APPLY_JOB

OS/400 のみ (索引なし)	
APPLY_QUAL	CHAR(18) NOT NULL
CONTROL_SERVER	CHAR(18) NOT NULL
JOB_NAME	CHAR(10) NOT NULL
USER_NAME	CHAR(10) NOT NULL
JOB_NUMBER	CHAR(6) NOT NULL

ASN.IBMSNAP_APPLYTRACE

(APPLY_QUAL, TRACE_TIME)	
APPLY_QUAL	CHAR(18) NOT NULL
TRACE_TIME	TIMESTAMP NOT NULL
OPERATION	CHAR(8) NOT NULL
DESCRIPTION	VARCHAR(1024) NOT NULL

ASN.IBMSNAP_APPPARMS

(APPLY_QUAL)	
APPLY_QUAL	CHAR(18) NOT NULL
APPLY_PATH	VARCHAR(1040)
COPYONCE	CHAR(1) WITH DEFAULT
DELAY	INT WITH DEFAULT
ERRWAIT	INT WITH DEFAULT
INAMSG	CHAR(1) WITH DEFAULT
LOADXIT	CHAR(1) WITH DEFAULT
LOGREUSE	CHAR(1) WITH DEFAULT
LOGSTDOUT	CHAR(1) WITH DEFAULT
NOTIFY	CHAR(1) WITH DEFAULT
OPT4ONE	CHAR(1) WITH DEFAULT
SLEEP	CHAR(1) WITH DEFAULT
SQLERRCONTINUE	CHAR(1) WITH DEFAULT
SPIILLFILE	VARCHAR(10) WITH DEFAULT
TERM	CHAR(1) WITH DEFAULT
TRLREUSE	CHAR(1) 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 18. アプライ・コントロール・サーバーで使用される表： これらの表は、アプライ・コントロール・サーバーでアプライ・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。

アプライ・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_SUBS_COLS

(APPLY_QUAL, SET_NAME, WHOS_ON_FIRST, TARGET_OWNER, TARGET_TABLE, TARGET_NAME)	
APPLY_QUAL	CHAR(18) NOT NULL
SET_NAME	CHAR(18) NOT NULL
WHOS_ON_FIRST	CHAR(1) NOT NULL
TARGET_OWNER	VARCHAR(30)1 NOT NULL
TARGET_TABLE	VARCHAR(128)2 NOT NULL
COL_TYPE	CHAR(1) NOT NULL
TARGET_NAME	VARCHAR(30) NOT NULL
IS_KEY	CHAR(1) NOT NULL
COLNO	SMALLINT NOT NULL
EXPRESSION	VARCHAR(254) NOT NULL

ASN.IBMSNAP_SUBS_EVENT

(EVENT_NAME, EVENT_TIME)	
EVENT_NAME	CHAR(18) NOT NULL
EVENT_TIME	TIMESTAMP NOT NULL
END_SYNCHPOINT	CHAR(10) FOR BIT DATA
END_OF_PERIOD	TIMESTAMP

ASN.IBMSNAP_SUBS_MEMBR

(APPLY_QUAL, SET_NAME, WHOS_ON_FIRST, SOURCE_OWNER, SOURCE_TABLE, SOURCE_VIEW_QUAL, TARGET_OWNER, TARGET_TABLE)	
APPLY_QUAL	CHAR(18) NOT NULL
SET_NAME	CHAR(18) NOT NULL
WHOS_ON_FIRST	CHAR(1) NOT NULL
SOURCE_OWNER	VARCHAR(30)1 NOT NULL
SOURCE_TABLE	VARCHAR(128)2 NOT NULL
SOURCE_VIEW_QUAL	SMALLINT NOT NULL
TARGET_OWNER	VARCHAR(30)1 NOT NULL
TARGET_TABLE	VARCHAR(128)2 NOT NULL
TARGET_CONDENSED	CHAR(1) NOT NULL
TARGET_COMPLETE	CHAR(1) NOT NULL
TARGET_STRUCTURE	SMALLINT NOT NULL
PREDICATES	VARCHAR(1024)
MEMBER_STATE	CHAR(1)
TARGET_KEY_CHG	CHAR(1) NOT NULL
UOW_CD_PREDICATES	VARCHAR(1024)
JOIN_UOW_CD	CHAR(1)
LOADX_TYPE	SMALLINT
LOADX_SRC_N_OWNER	VARCHAR(30)1
LOADX_SRC_N_TABLE	VARCHAR(128)2

ASN.IBMSNAP_SUBS_SET

(APPLY_QUAL, SET_NAME, WHOS_ON_FIRST)	
APPLY_QUAL	CHAR(18) NOT NULL
SET_NAME	CHAR(18) NOT NULL
SET_TYPE	CHAR(1) NOT NULL
WHOS_ON_FIRST	CHAR(1) NOT NULL
ACTIVATE	SMALLINT NOT NULL
SOURCE_SERVER	CHAR(18) NOT NULL
SOURCE_ALIAS	CHAR(8)
TARGET_SERVER	CHAR(18) NOT NULL
TARGET_ALIAS	CHAR(8)
STATUS	SMALLINT NOT NULL
LASTRUN	TIMESTAMP NOT NULL
REFRESH_TYPE	CHAR(1) NOT NULL
SLEEP_MINUTES	INT
EVENT_NAME	CHAR(18)
LASTSUCCESS	TIMESTAMP
SYNCHPOINT	CHAR(10) FOR BIT DATA
SYNCHTIME	TIMESTAMP
CAPTURE_SCHEMA	VARCHAR(30)1 NOT NULL
TGT_CAPTURE_SCHEMA	VARCHAR(30)1
FEDERATED_SRC_SRVR	VARCHAR(18)
FEDERATED_TGT_SRVR	VARCHAR(18)
JRN_LIB	CHAR(10)
JRN_NAME	CHAR(10)
OPTION_FLAGS	CHAR(4) NOT NULL
COMMIT_COUNT	SMALLINT
MAX_SYNCH_MINUTES	SMALLINT
AUX_STMTS	SMALLINT NOT NULL
ARCH_LEVEL	CHAR(4) NOT NULL

ASN.IBMSNAP_SUBS_STMTS

(APPLY_QUAL, SET_NAME, WHOS_ON_FIRST, BEFORE_OR_AFTER, STMT_NUMBER)	
APPLY_QUAL	CHAR(18) NOT NULL
SET_NAME	CHAR(18) NOT NULL
WHOS_ON_FIRST	CHAR(1) NOT NULL
BEFORE_OR_AFTER	CHAR(1) NOT NULL
STMT_NUMBER	SMALLINT NOT NULL
EI_OR_CALL	CHAR(1) NOT NULL
SQL_STMT	VARCHAR(1024)
ACCEPT_SQLSTATES	VARCHAR(50)

1DB2 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 19. アプライ・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、アプライ・コントロール・サーバーでアプライ・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 (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

1DB2 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 20. 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
LASTSUCCESS	TIMESTAMP		WITH DEFAULT
STATUS	SMALLINT NOT NULL	LAST_MONITOR_TIME	TIMESTAMP 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 21. 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、表名の下の括弧内に示されています。

キャプチャー・コントロール・サーバーで使用される表のリスト

キャプチャー・コントロール・サーバーに保管される表には、ユーザーの登録済みソースに関する情報と、キャプチャー・プログラムまたはトリガーがソースを処理する方法に関する情報が入っています。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。OS/400 の場合は、DataPropagator for iSeries のインストール時に、ASN ライブラリーの中にこれらのコントロール表が自動的に作成されます。OS/400 上でレプリケーションのシステム・コマンドを使用して、代替のキャプチャー・スキーマ内にキャプチャー・コントロール表を作成できます。

表 63. キャプチャー・コントロール・サーバーで使用される表のクイック・リファレンス

表名	説明	参照 ページ
IBMSNAP_CAPSCHEMAS	キャプチャー・スキーマ表 すべてのキャプチャー・スキーマの名前を保持します。	515
schema.IBMSNAP_AUTHTKN (OS/400)	アプライ修飾子相互参照表 Update-anywhere をサポートするための情報が入っています。	515

表 63. キャプチャー・コントロール・サーバーで使用される表のクイック・リファレンス (続き)

表名	説明	参照 ページ
<i>schema</i> .IBMSNAP_CAPENQ (Linux、UNIX、Windows、z/OS)	<p>キャプチャー・エンキュー表</p> <p>この表は、各キャプチャー・スキーマについて、次のことを保証するために使用されます。</p> <ul style="list-style-type: none"> • DB2 for Linux、DB2 for UNIX および DB2 for Windows の場合は、1 つのデータベースに対して 1 つのキャプチャー・プログラムのみが実行される。 • データを共有しない DB2 for z/OS の場合は、1 つのサブシステムに対して 1 つのキャプチャー・プログラムのみが実行される。 • データを共有する DB2 for z/OS の場合は、1 つのデータ共有グループに対して 1 つのキャプチャー・プログラムのみが実行される。 	516
<i>schema</i> .CD_table	<p>変更データ (CD) 表</p> <p>ソースに発生する変更に関する情報を保持します。この表は、レプリケーション・ソースが登録されるまでは作成されません。</p>	525
<i>schema</i> .CCD_table	<p>整合変更データ (CCD) 表</p> <p>ソースに発生する変更に関する情報と、これらの変更の順序を識別するための追加の列を保持します。</p>	523
<i>schema</i> .IBMSNAP_CAPMON	<p>キャプチャー・モニター表</p> <p>キャプチャー・プログラムの進行状況をモニターする助けとなる操作統計を保持します。</p>	517
<i>schema</i> .IBMSNAP_CAPPARMS	<p>キャプチャー・パラメーター表</p> <p>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します。</p>	519
<i>schema</i> .IBMSNAP_CAPTRACE	<p>キャプチャー・トレース表</p> <p>キャプチャー・プログラムからの重要なメッセージを保持します。</p>	522
<i>schema</i> .IBMSNAP_PARTITIONINFO	<p>区画情報表</p> <p>必要とされる中で一番古いログ・シーケンス番号からキャプチャー・プログラムを再始動できるようにするための情報が含まれます。</p>	526
<i>schema</i> .IBMSNAP_PRUNE_LOCK	<p>整理ロック表</p> <p>コールド・スタート、または保存限度整理 (保存限度に達したか、超えたときの整理) 時に、キャプチャー・プログラムの CD 表へのアクセスをシリアルライズするために使用されます。</p>	529
<i>schema</i> .IBMSNAP_PRUNE_SET	<p>整理セット表</p> <p>CD 表の整理を調整します。</p>	529

表 63. キャプチャー・コントロール・サーバーで使用される表のクイック・リファレンス (続き)

表名	説明	参照 ページ
<i>schema</i> .IBMSNAP_PRUNCNTL	整理コントロール表 キャプチャー・プログラムとアプライ・プログラムの間で同期点更新を調整します。	527
<i>schema</i> .IBMSNAP_REG_EXT (OS/400)	登録拡張表 登録表を拡張したものです。ジャーナル名やリモート・ソース表のデータベース項目名など、レプリケーション・ソースに関する追加情報が含まれます。	530
<i>schema</i> .IBMSNAP_REGISTER	登録表 レプリケーション・ソース表の名前、その属性、および対応する CD 表および CCD 表の名前など、レプリケーション・ソースに関する情報が入ります。	532
<i>schema</i> .IBMSNAP_REG_SYNCH (DB2 以外のリレーショナル)	登録同期表 DB2 以外のリレーショナル・データ・ソースから複製するとき使用されます。この表の更新トリガーは、アプライ・プログラムが登録表から情報を読み込む前に、登録表のすべての行で SYNCHPOINT 値の更新を始めることにより、キャプチャー・プログラムをシミュレートします。	539
<i>schema</i> .IBMSNAP_RESTART	再始動表 キャプチャー・プログラムがログまたはジャーナル内の正しい時点からキャプチャーを再開できるようにするための情報が入っています。OS/400 環境では、この表は RCVJRNE (ジャーナル項目の受信) コマンドの開始時刻を判別するためにも使用されません。	540
<i>schema</i> .IBMSNAP_SEQTABLE (Informix)	順序付け表 Informix 表のログ・シーケンス番号と同等のものとして DB2 レプリケーションが使用される一連のユニーク番号を保持します。	542
<i>schema</i> .IBMSNAP_SIGNAL	シグナル表 キャプチャー・プログラムにプロンプトを出すために使用されるすべてのシグナルを保持します。これらのシグナルは手動で、またはアプライ・プログラムから送信できます。	543
<i>schema</i> .IBMSNAP_UOW	作業単位 (UOW) 表 ソース表にコミットされたトランザクションに関する追加情報を提供します。	546

関連資料:

- 515 ページの『ASN.IBMSNAP_CAPSCHEMAS』
- 515 ページの『*schema*.IBMSNAP_AUTHTKN (OS/400)』
- 516 ページの『*schema*.IBMSNAP_CAPENQ (UNIX、Windows、z/OS)』
- 517 ページの『*schema*.IBMSNAP_CAPMON』

- 519 ページの『*schema.IBMSNAP_CAPPARMS*』
- 522 ページの『*schema.IBMSNAP_CAPTRACE* (DB2 のみ)』
- 523 ページの『*schema.CCD_table* (DB2 以外)』
- 525 ページの『*schema.CD_table*』
- 527 ページの『*schema.IBMSNAP_PRUNCNTL*』
- 529 ページの『*schema.IBMSNAP_PRUNE_LOCK*』
- 529 ページの『*schema.IBMSNAP_PRUNE_SET*』
- 530 ページの『*schema.IBMSNAP_REG_EXT* (OS/400)』
- 532 ページの『*schema.IBMSNAP_REGISTER*』
- 539 ページの『*schema.IBMSNAP_REG_SYNCH* (DB2 以外のリレーショナル)』
- 540 ページの『*schema.IBMSNAP_RESTART*』
- 542 ページの『*schema.IBMSNAP_SEQTABLE* (Informix)』
- 543 ページの『*schema.IBMSNAP_SIGNAL*』
- 546 ページの『*schema.IBMSNAP_UOW*』
- 526 ページの『*schema.IBMSNAP_PARTITIONINFO*』

アプライ・コントロール・サーバーで使用される表のリスト

アプライ・コントロール・サーバーに保管される表には、ユーザーのサブスクリプション定義に関する情報が入っています。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。OS/400 の場合は、DataPropagator for iSeries のインストール時に、これらのコントロール表が自動的に作成されます。

表 64. アプライ・コントロール・サーバーで使用される表のクイック・リファレンス

表名	説明	参照 ページ
IBMSNAP_APPENQ	アプライ・エンキュー表 1 つのアプライ修飾子に対して 1 つのアプライ・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ます。	548
IBMSNAP_APPLY_JOB (OS/400)	アプライ・ジョブ表 アプライ・コントロール・サーバーで実行中のアプライ・プログラムのインスタンスごとにユニークなアプライ修飾子が確実に存在するようにするために使用されます。	549
IBMSNAP_APPLYTRACE	アプライ・トレース表 アプライ・プログラムからの重要なメッセージを保持します。	554
IBMSNAP_APPLYTRAIL	アプライ・トレール表 アプライ・プログラムに関する監査証跡情報を保持します。	554
IBMSNAP_APPPARMS	アプライ・パラメーター表 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が含まれています。	550

表 64. アプライ・コントロール・サーバーで使用される表のクイック・リファレンス (続き)

表名	説明	参照 ページ
IBMSNAP_SUBS_COLS	サブスクリプション列表 ターゲット表またはビュー内の列を、ソース表またはビュー内の対応する列にマップします。	560
IBMSNAP_SUBS_EVENT	サブスクリプション・イベント表 アプライ・プログラムによるサブスクリプション・セットの処理をコントロールするためにユーザーが定義するイベントを保持します。	562
IBMSNAP_SUBS_MEMBR	サブスクリプション・メンバー表 ソースとターゲット表の対を識別し、その対の処理情報を指定します。	563
IBMSNAP_SUBS_SET	サブスクリプション・セット表 アプライ・プログラムによりグループとして処理されるサブスクリプション・セット・メンバーの各セットの処理情報を保持します。	568
IBMSNAP_SUBS_STMTS	サブスクリプション・ステートメント表 ユーザーがサブスクリプション・セットに対して定義する、SQL ステートメント、またはストアド・プロシージャー呼び出しを保持します。これらは、アプライ・プログラムによるセットの処理前、または処理後に呼び出されます。	574

関連資料:

- 548 ページの『ASN.IBMSNAP_APPENQ』
- 549 ページの『ASN.IBMSNAP_APPLY_JOB (OS/400)』
- 554 ページの『ASN.IBMSNAP_APPLYTRACE』
- 554 ページの『ASN.IBMSNAP_APPLYTRAIL』
- 560 ページの『ASN.IBMSNAP_SUBS_COLS』
- 562 ページの『ASN.IBMSNAP_SUBS_EVENT』
- 563 ページの『ASN.IBMSNAP_SUBS_MEMBR』
- 568 ページの『ASN.IBMSNAP_SUBS_SET』
- 574 ページの『ASN.IBMSNAP_SUBS_STMTS』
- 550 ページの『ASN.IBMSNAP_APPPARMS』

モニター・コントロール・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・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 65. 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
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します。
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ます。
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ます。
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ます。
IBMSNAP_GROUPS	各連絡先グループの名前と記述が入っています。
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ます。
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ます。
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します。
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します。
IBMSNAP_MONTRAIL	各モニター・サイクルに関する重要な情報を保持します。

関連資料:

- 576 ページの『IBMSNAP_ALERTS 表』
- 578 ページの『IBMSNAP_CONDITIONS 表』
- 584 ページの『IBMSNAP_CONTACTGRP 表』
- 591 ページの『IBMSNAP_MONTRAIL 表』
- 585 ページの『IBMSNAP_CONTACTS 表』
- 586 ページの『IBMSNAP_GROUPS 表』
- 586 ページの『IBMSNAP_MONENQ 表』
- 588 ページの『IBMSNAP_MONSERVERS 表』
- 590 ページの『IBMSNAP_MONTRACE 表』
- 586 ページの『IBMSNAP_MONPARMS 表』

ターゲット・サーバーで使用される表のリスト

ターゲット・サーバーには、さまざまなタイプのターゲット表が保管されます。ユーザーがターゲット表として既存の表を使用しない場合、レプリケーション・センターは、ユーザーがサブスクリプション・セット・メンバーをどのように定義しているかに従って、ユーザーの指定に合わせてターゲット表を構築します。

表 66. ターゲット表のクイック・リファレンス

表名	記述	参照 ページ
<i>schema.base_aggregate</i>	基礎集約表 ソース表から集約されたデータを保持します。	592
<i>schema.change_aggregate</i>	変更集約表 CD 表から集約されたデータを保持します。	593
<i>schema.CCD</i>	整合変更データ (CCD) 表 ソースに発生する変更に関する情報と、これらの変更の順序を識別するための追加の列を保持します。	594
<i>schema.point_in_time</i>	ポイント・イン・タイム表 ソース・ログの中でデータがコミットされた特定の時刻を記録する追加の列を持つ、ソース・データのコピー。	596
<i>schema.replica</i>	レプリカ表 Update-anywhere レプリケーションで使用されるターゲット表のタイプ。	597
<i>schema.user_copy</i>	ユーザー・コピー表 ソース表のコピー。	597

関連資料:

- 592 ページの『基礎集約表』
- 593 ページの『変更集約表』
- 594 ページの『整合変更データ (CCD) 表』
- 596 ページの『ポイント・イン・タイム表』
- 597 ページの『レプリカ表』
- 597 ページの『ユーザー・コピー表』

キャプチャー・コントロール・サーバー上の表、およびその列に関する記述

このセクションでは、キャプチャー・コントロール・サーバー上に保管される各表について、かなり詳しく説明します。また、各表の列のリストやその簡単な説明もあります。コントロール表はアルファベット順にリストされ、列は各表内に現れる順に左から右へリストされています。

ASN.IBMSNAP_CAPSCHEMAS

サーバー: キャプチャー・コントロール・サーバー

索引: CAP_SCHEMA_NAME

重要: SQL を使用してこの表を更新するときには、注意してください。この表を不適切に変更すると、管理ツールの使用中に予期しない結果が生じることがあります。

キャプチャー・スキーマ表は、すべてのキャプチャー・スキーマの名前を保留します。レプリケーション・センターやその他のユーティリティーはこの表を使用して、特定のキャプチャー・コントロール・サーバーのすべての表を迅速に検出できます。ユーザーが新しいキャプチャー・スキーマを作成するたびに、行が 1 つ自動的に挿入されます。

次の 2 つの表は、オペレーティング・システム別のキャプチャー・スキーマ表のレイアウトを示しています。

表 67. OS/400 以外のオペレーティング・システムの場合のキャプチャー・スキーマ表の列

列名	説明
CAP_SCHEMA_NAME	<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>キャプチャー・スキーマの名前。キャプチャー・スキーマごとに 1 つの行があります。</p>

表 68. OS/400 の場合のキャプチャー・スキーマ表の列

列名	説明
CAP_SCHEMA_NAME	<p>データ・タイプ: VARCHAR(30)。 NULL 可能: はい。</p> <p>キャプチャー・スキーマの名前。キャプチャー・スキーマごとに 1 つの行があります。</p>
STATUS	<p>データ・タイプ: CHAR(1)。 NULL 可能: はい。</p> <p>このキャプチャー・スキーマによって識別されるキャプチャー・プログラムが実行中かどうかを示すフラグ。</p> <p>Y キャプチャー・プログラムは実行中です。</p> <p>N キャプチャー・プログラムは実行されていません。</p>

schema.IBMSNAP_AUTHTKN (OS/400)

サーバー: キャプチャー・コントロール・サーバー

デフォルト・スキーマ: ASN

索引: JRN_LIB, JRN_NAME

IBMSNAP_AUTHTKN (OS/400)

重要: SQL を使用してこの表を更新するときには、注意してください。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。

アプライ修飾子相互参照表は、OS/400 環境でのみ使用されます。この表は、Update-anywhere レプリケーション時に、アプライ修飾子により識別される、特定のアプライ・プログラムによって実行されたトランザクションをトラッキングするために使用されます。キャプチャー・プログラムは、ユーザーが設定した保存限度に基づいて、この表の整理を行います。

表 69 では、アプライ修飾子相互参照表の列の要旨を示します。

表 69. アプライ修飾子相互参照表の列

列名	説明
APPLY_QUAL	<p>データ・タイプ: CHAR(18)。 NULL 可能: いいえ。</p> <p>トランザクションを処理したアプライ・プログラムを識別するためのアプライ修飾子。この修飾子は、Update-anywhere レプリケーションにおいて、アプライ・プログラムが同じ変更を繰り返し複製しないようにするために使用されます。</p>
IBMSNAP_AUTHTKN	<p>データ・タイプ: CHAR(26)。 NULL 可能: いいえ。</p> <p>トランザクションに関連付けられたジョブ名。Capture for iSeries はこの列の名前と、トランザクションを発行したジョブの名前を突き合わせて、トランザクションがアプライ・プログラムから発行されたものか、ユーザー・アプリケーションから発行されたものかを判別します。ジョブ名が一致すると、Capture for iSeries は、この表の APPLY_QUAL 列にあるアプライ修飾子を、UOW 表の対応する行の APPLY_QUAL 列にコピーします。名前が一致しない場合、Capture for iSeries は UOW 行の APPLY_QUAL 列を NULL に設定します。この列は自動的に他の表にコピーされません。ユーザー・データ列として選択してコピーする必要があります。</p>
JRN_LIB	<p>データ・タイプ: CHAR(10)。 NULL 可能: いいえ。</p> <p>トランザクションの発行元のジャーナルのライブラリー名。</p>
JRN_NAME	<p>データ・タイプ: CHAR(10)。 NULL 可能: いいえ。</p> <p>トランザクションの発行元のジャーナルの名前。</p>
IBMSNAP_LOGMARKER	<p>データ・タイプ: TIMESTAMP。 NULL 可能: いいえ。</p> <p>トランザクションがキャプチャー・コントロール・サーバーでコミットされたおおよその時間。</p>

schema.IBMSNAP_CAPENQ (UNIX、Windows、z/OS)

サーバー: キャプチャー・コントロール・サーバー

デフォルト・スキーマ: ASN

索引: なし

重要: SQL を使用してこの表を更新するときには、注意してください。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。

キャプチャー・エンキュー表は、DB2 以外のリレーショナル・サーバー、または OS/400 サーバーでは使用されません。

1 つのキャプチャー・スキーマについて、キャプチャー・エンキュー表は以下のことを保証します。

- DB2 for Linux、DB2 for UNIX および DB2 for Windows の場合は、1 つのデータベースに対して 1 つのキャプチャー・プログラムのみが実行される。
- データを共有しない DB2 for z/OS の場合は、1 つのサブシステムに対して 1 つのキャプチャー・プログラムのみが実行される。
- データを共有する DB2 for z/OS の場合は、1 つのデータ共有グループに対して 1 つのキャプチャー・プログラムのみが実行される。

キャプチャー・プログラムは実行中、この表を排他的にロックします。

表 70 では、キャプチャー・エンキュー表の列の要旨を示します。

表 70. キャプチャー・エンキュー表の列

列名	説明
LOCKNAME	データ・タイプ: CHAR(9)。NULL 可能: はい。 この列にはデータは含まれません。

schema.IBMSNAP_CAPMON

サーバー: キャプチャー・コントロール・サーバー

デフォルト・スキーマ: ASN

索引: MONITOR_TIME

キャプチャー・プログラムは、インターバルが終了するたびにキャプチャー・モニター表に行を挿入して、操作統計を提供します。レプリケーション・センターはこの表 (およびその他の表) の情報を使用するため、ユーザーはキャプチャー・プログラムの状況をモニターできます。キャプチャー・パラメーター (IBMSNAP_CAPPARMS) 表で、ユーザーが MONITOR_INTERVAL に対して指定した値は、キャプチャー・プログラムがキャプチャー・モニター表に挿入を行う頻度を示し、ユーザーが MONITOR_LIMIT に対して指定した値は、表の行が整理の対象となるまでに、表内に留まる分数を示します。

表 71 では、キャプチャー・モニター表の列の要旨を示します。

表 71. キャプチャー・モニター表の列

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タイム・スタンプ (キャプチャー・コントロール・サーバーの)。
RESTART_TIME	データ・タイプ: TIMESTAMP。NULL 可能: いいえ。 キャプチャー・プログラムの現在の呼び出しが再始動されたときのタイム・スタンプ。

IBMSNAP_CAPMON

表 71. キャプチャー・モニター表の列 (続き)

列名	説明
CURRENT_MEMORY	<p>データ・タイプ: INT。 NULL 可能: いいえ。</p> <p>キャプチャー・プログラムが使用したメモリーの量 (バイト単位)。</p>
CD_ROWS_INSERTED	<p>データ・タイプ: INT。 NULL 可能: いいえ。</p> <p>すべてのソース表について、キャプチャー・プログラムが CD 表に挿入した行数。</p>
RECAP_ROWS_SKIPPED	<p>データ・タイプ: INT。 NULL 可能: いいえ。</p> <p>Update-anywhere レプリケーションの場合、キャプチャー・プログラムが処理したが、CD 表に挿入していない行数。キャプチャー・プログラムの登録では、この表に複製された変更でも、このソース・サーバーから発生したものではない変更は、再キャプチャーしないように定義されているため、これらの行はスキップされます。</p>
TRIGR_ROWS_SKIPPED	<p>データ・タイプ: INT。 NULL 可能: いいえ。</p> <p>キャプチャー・プログラムが処理したが、CD 表に挿入していない行数。キャプチャー・プログラムの登録で、特定の行を抑制するようにトリガーが定義されているため、これらの行はスキップされます。</p>
CHG_ROWS_SKIPPED	<p>データ・タイプ: INT。 NULL 可能: いいえ。</p> <p>キャプチャー・プログラムが処理したが、CD 表に挿入していない行数。キャプチャー・プログラムの登録では、登録済みの列で発生した変更のみをキャプチャーするように定義されているため、これらの行はスキップされます。</p>
TRANS_PROCESSED	<p>データ・タイプ: INT。 NULL 可能: いいえ。</p> <p>キャプチャー・プログラムが処理した、ソース・システム上のトランザクションの数。</p>
TRANS_SPILLED	<p>データ・タイプ: INT。 NULL 可能: いいえ。</p> <p>メモリー制限のためにキャプチャー・プログラムがディスクに書き出した、ソース・システム上のトランザクションの数。</p>
MAX_TRAN_SIZE	<p>データ・タイプ: INT。 NULL 可能: いいえ。</p> <p>ソース・システムで発生した最大のトランザクション。トランザクション・サイズを知ることによって、メモリー・パラメーターの変更を検討することになります。</p>
LOCKING_RETRIES	<p>データ・タイプ: INT。 NULL 可能: いいえ。</p> <p>デッドロックにより再処理が必要となった回数。</p>
JRN_LIB (OS/400)	<p>データ・タイプ: CHAR(10)。 NULL 可能: はい。</p> <p>キャプチャー・プログラムが処理していたジャーナルのライブラリー名。</p>
JRN_NAME (OS/400)	<p>データ・タイプ: CHAR(10)。 NULL 可能: はい。</p> <p>キャプチャー・プログラムが処理していたジャーナルの名前。</p>

表 71. キャプチャー・モニター表の列 (続き)

列名	説明
LOGREADLIMIT	<p>データ・タイプ: INT。 NULL 可能: いいえ。</p> <p>1000 のレコードが読み取られたが、それら 1000 のレコードの中に完了済みのトランザクションが見つからなかったために、キャプチャー・プログラムがログ・レコード読み取りを一時停止した回数。</p>
CAPTURE_IDLE	<p>データ・タイプ: INT。 NULL 可能: いいえ。</p> <p>処理するものがないため、キャプチャー・プログラムがスリープした回数。</p>
SYNCHTIME	<p>データ・タイプ: TIMESTAMP。 NULL 可能: いいえ。</p> <p>この表にモニター・レコードが挿入されたときに、登録表のグローバル行から読み取られた SYNCHTIME の現行値。</p>

schema.IBMSNAP_CAPPARMS

サーバー: キャプチャー・コントロール・サーバー

デフォルト・スキーマ: ASN

索引: なし

この表の情報は、SQL を使って更新できます。

キャプチャー・パラメーター表は、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、キャプチャー・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、ログ・レコードの処理においてキャプチャー・プログラムに許される遅延時間などの値を設定できます。ユーザーがこの表のパラメーターを変更しても、キャプチャー・プログラムは始動時にしか変更を読み取りません。

表 72 では、キャプチャー・パラメーター表の列の要旨を示します。

表 72. キャプチャー・パラメーター表の列

列名	説明
RETENTION_LIMIT	<p>データ・タイプ: INT。 NULL 可能: はい。</p> <p>通常基準に基づいて整理が行われていないときに、CD 表、UOW 表、およびシグナル表の中の行が整理の対象となるまで、表の中に留まる時間の長さ。一般的に CD 行および UOW 行は、すべてのターゲットに適用された後で除去され、シグナル行は、サイクルが完了 (SIGNAL_STATE = C) したときに除去されません。</p>
LAG_LIMIT	<p>データ・タイプ: INT。 NULL 可能: はい。</p> <p>ログ・レコードの処理時にキャプチャー・プログラムがシャットダウンせずに処理を遅らせることができる分数。更新頻度の高い期間では、更新よりもフル・リフレッシュの方が経済的です。</p>

IBMSNAP_CAPPARMS

表 72. キャプチャー・パラメーター表の列 (続き)

列名	説明
COMMIT_INTERVAL	<p>データ・タイプ: INT。 NULL 可能: はい。</p> <p>UOW 表および CD 表を含めて、キャプチャー・プログラムがキャプチャー・コントロール表にデータをコミットする頻度 (秒単位)。キャプチャー・スレッドと整理スレッドの競合を避けるために、この値は DB2 ロックアウトの値よりも小さい値にする必要があります。</p>
PRUNE_INTERVAL	<p>データ・タイプ: INT。 NULL 可能: はい。</p> <p>キャプチャー・プログラムが、不必要になった CD 表、UOW 表、シグナル表、トレース表、およびキャプチャー・モニター表の行を自動的に除去する (AUTOPRUNE = Y) 頻度 (秒単位)。除去するインターバルを短くす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。除去するインターバルを長くすると CD 表および UOW 表のスペースはより多く必要になりますが、処理コストは減少します。</p>
TRACE_LIMIT	<p>データ・タイプ: INT。 NULL 可能: はい。</p> <p>整理の対象となるまでに、行がキャプチャー・トレース (IBMSNAP_CAPTRACE) 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・スタンプからキャプチャー・トレース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、キャプチャー・トレース表の中の行を削除します。</p>
MONITOR_LIMIT	<p>データ・タイプ: INT。 NULL 可能: はい。</p> <p>整理の対象となるまでに、行がキャプチャー・モニター (IBMSNAP_CAPMON) 表の中に留まる分数。整理プロセス時に、分数の値 (現在のタイム・スタンプから MONITOR_TIME を引いたもの) が MONITOR_LIMIT の値を超えると、キャプチャー・モニター表の中の行を削除します。</p>
MONITOR_INTERVAL	<p>データ・タイプ: INT。 NULL 可能: はい。</p> <p>モニター・スレッドがキャプチャー・モニター (IBMSNAP_CAPMON) 表に行を追加する頻度 (秒単位)。Capture for iSeries の場合は、120 よりも長いインターバルを入力してください。</p>
MEMORY_LIMIT	<p>データ・タイプ: SMALLINT。 NULL 可能: はい。</p> <p>キャプチャー・プログラムが使用することを許されているメモリーの量 (MB 単位)。この割り振りがすべて使用されてしまうと、メモリー・トランザクションはファイルに書き出されます。</p>
REMOTE_SRC_SERVER	<p>データ・タイプ: CHAR(18)。 NULL 可能: はい。</p> <p>DB2 レプリケーションの将来のオプション用に予約済み。この列には現在、デフォルト値の NULL が入っています。</p>
AUTOPRUNE	<p>データ・タイプ: CHAR(1)。 NULL 可能: はい。</p> <p>キャプチャー・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、およびキャプチャー・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 自動整理はオン。 N 自動整理はオフ。</p>

表 72. キャプチャー・パラメーター表の列 (続き)

列名	説明
TERM	<p>データ・タイプ: CHAR(1)。 NULL 可能: はい。</p> <p>DB2 が終了したときにキャプチャー・プログラムが終了するかどうかを示すフラグ。</p> <p>Y DB2 が終了するとキャプチャー・プログラムは終了します。</p> <p>N キャプチャー・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>
AUTOSTOP	<p>データ・タイプ: CHAR(1)。 NULL 可能: はい。</p> <p>アクティブ・ログの最後に達したときにキャプチャー・プログラムが変更のキャプチャーを終了するかどうかを示すフラグ。</p> <p>Y アクティブ・ログの最後に達するとキャプチャー・プログラムは変更のキャプチャーをすぐに終了します。</p> <p>N アクティブ・ログの最後に達してもキャプチャー・プログラムは実行を続けます。</p>
LOGREUSE	<p>データ・タイプ: CHAR(1)。 NULL 可能: はい。</p> <p>キャプチャー・プログラムがキャプチャー・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 キャプチャー・プログラムは再始動時に、最初にログ・ファイルを削除してから再作成してログ・ファイルを再利用します。</p> <p>N キャプチャー・プログラムは新しい情報をキャプチャー・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: はい。</p> <p>キャプチャー・プログラムがログ・ファイル・メッセージを誘導するかどうかを示すフラグ。</p> <p>Y キャプチャー・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・メッセージを誘導します。</p> <p>N 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>
SLEEP_INTERVAL (Linux、UNIX、Windows、z/OS)	<p>データ・タイプ: SMALLINT。 NULL 可能: はい。</p> <p>アクティブ・ログの最後に達したとき (Linux、UNIX および Windows、または z/OS のデータを共有しない環境の場合)、または戻されたデータの量が不十分なとき (z/OS のデータ共有環境の場合) にキャプチャー・プログラムがスリープする秒数。</p>
CAPTURE_PATH	<p>データ・タイプ: VARCHAR(1040)。 NULL 可能: はい。</p> <p>キャプチャー・プログラムからの出力が送信されるパス。</p>

表 72. キャプチャー・パラメーター表の列 (続き)

列名	説明
STARTMODE	<p>データ・タイプ: VARCHAR(10)。NULL 可能: はい。</p> <p>キャプチャー・プログラムの始動時に使用される処理プロシージャ。</p> <p>cold キャプチャー・プログラムは、初期化時に CD 表と UOW 表の中のすべての行を削除します。これらのレプリケーション・ソースに対するすべてのサブスクリプションは、次のアプライ処理サイクル時にフル・リフレッシュされます (つまり、すべてのデータがソース表からターゲット表にコピーされます)。キャプチャー・プログラムがコールド・スタートを試行したときに、フル・リフレッシュが使用不可な場合には、キャプチャー・プログラムは開始されますが、アプライ・プログラムは失敗し、エラー・メッセージが発行されます。</p> <p>warmsi キャプチャー・プログラムはウォーム・スタートします。ただし、キャプチャー・プログラムを初めて開始する場合には、コールド・スタートに切り替えられます。warmsi 開始モードは、キャプチャー・プログラムを初めて開始する場合にのみコールド・スタートが行われるようにします。</p> <p>warmns キャプチャー・プログラムはウォーム・スタートします。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 開始モードは、予期せずにコールド・スタートが発生することを防止します。これは、修復が必要な、ウォーム・スタートの進行を阻害している問題 (データベースまたは表スペースが使用できないなど) が発生した場合に便利です。ウォーム・スタートした場合、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。キャプチャー・プログラムの始動後にエラーが発生した場合は、キャプチャー・プログラムは終了し、すべての表はそのまま残されます。</p> <p>warmsa ウォーム・スタート情報が使用可能な場合、キャプチャー・プログラムは前回の実行で終了したところから処理を再開します。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、すべてのターゲット表をリフレッシュするコールド・スタートに切り替わります。</p>

schema.IBMSNAP_CAPTRACE (DB2 のみ)

サーバー: キャプチャー・コントロール・サーバー

デフォルト・スキーマ: ASN

索引: TRACE_TIME

キャプチャー・トレース表は、キャプチャー・プログラムから重要なメッセージを保持します。

以下の 2 つの表は、オペレーティング・システム別のキャプチャー・トレース表のレイアウトを示しています。

表 73. Linux、UNIX、Windows、および z/OS のキャプチャー・トレース表の列

列名	説明
OPERATION	<p>データ・タイプ: CHAR(8)。NULL 可能: いいえ。</p> <p>キャプチャー・プログラムの操作のタイプ (たとえば、初期化、キャプチャー、またはエラー条件)。</p>
TRACE_TIME	<p>データ・タイプ: TIMESTAMP。NULL 可能: いいえ。</p> <p>キャプチャー・トレース表に行が挿入されたときの、キャプチャー・コントロール・サーバーにおける時刻。</p>
説明	<p>データ・タイプ: VARCHAR(1024)。NULL 可能: いいえ。</p> <p>メッセージ ID とメッセージ・テキスト。エラー・メッセージ、警告メッセージ、または情報メッセージです。この列に入れられるテキストは英語のみです。</p>

表 74. OS/400 のキャプチャー・トレース表の列

列名	説明								
OPERATION	<p>データ・タイプ: CHAR(8)。NULL 可能: いいえ。</p> <p>初期化、キャプチャー、またはエラー条件など、キャプチャー・プログラムで実行された操作のタイプ。</p>								
TRACE_TIME	<p>データ・タイプ: TIMESTAMP。NULL 可能: いいえ。</p> <p>キャプチャー・トレース表に行が挿入された時刻。トレース限度整理の対象となる TRACE_TIME 行は、キャプチャー・プログラムが CD 表および UOW 表の整理を行うときに削除されます。</p>								
JOB_NAME	<p>データ・タイプ: CHAR(26)。NULL 可能: いいえ。</p> <p>このトレース項目を書き込んだジョブの完全修飾名。</p> <table border="0"> <thead> <tr> <th>位置</th> <th>説明</th> </tr> </thead> <tbody> <tr> <td>1-10</td> <td>キャプチャー・スキーマ名またはジャーナル・ジョブ名</td> </tr> <tr> <td>11-20</td> <td>キャプチャー・プログラムを始動したユーザーの ID</td> </tr> <tr> <td>21-26</td> <td>ジョブ番号</td> </tr> </tbody> </table>	位置	説明	1-10	キャプチャー・スキーマ名またはジャーナル・ジョブ名	11-20	キャプチャー・プログラムを始動したユーザーの ID	21-26	ジョブ番号
位置	説明								
1-10	キャプチャー・スキーマ名またはジャーナル・ジョブ名								
11-20	キャプチャー・プログラムを始動したユーザーの ID								
21-26	ジョブ番号								
JOB_STR_TIME	<p>データ・タイプ: TIMESTAMP。NULL 可能: いいえ。</p> <p>JOB_NAME 列で指定されたジョブの開始時刻。</p>								
DESCRIPTION	<p>データ・タイプ: VARCHAR(298)。NULL 可能: いいえ。</p> <p>メッセージ ID とメッセージ・テキスト。メッセージ ID は DESCRIPTION 列の最初の 7 文字です。メッセージ・テキストは、DESCRIPTION 列の位置 9 から始まります。</p>								

schema.CCD_table (DB2 以外)

サーバー: キャプチャー・コントロール・サーバー

重要: SQL を使用してこの表を更新するときには、注意してください。この表を不適切に変更すると、データの損失が生じることがあります。

CCD 表

キャプチャー・コントロール・サーバー上の整合変更データ (CCD) 表は、DB2 以外のソースで発生した変更に関する情報と、これらの変更の順序を識別するための追加の列を含む表です。キャプチャー・コントロール・サーバー上の CCD 表は、アプライ・プログラム以外のプログラムによってデータがキャプチャーされる表です。次のいずれかになります。

- DB2 以外のリレーショナル・ソースの内部 CCD 表。

変更キャプチャー・レプリケーションの場合、DB2 以外のリレーショナル・ソースで更新が行われると、キャプチャー・プログラムはこの表で挿入変更を起動します。このタイプの CCD 表の名前は、登録 (IBMSNAP_REGISTER) 表の中で、変更元のレプリケーション・ソースと同じ行に保管されます。この表は、DB2 以外のリレーショナル・ソースを登録したときに作成される整理トリガーによって自動的に整理が行われます。

- 非リレーショナルおよびマルチベンダーのデータ用の外部 CCD 表。

外部プログラムは、DB2 レプリケーションにより、レプリケーション・ソースとして使用される CCD 表を作成できます。これらの外部プログラムは、IMS データのコピーをリレーショナル・データベース内に再作成できるように、CCD 表の中の IMS 変更をキャプチャーします。外部プログラムは、コントロール列の正しい値を初期化、保守および提供する必要があります。IMS DataPropagator または DataRefresher などのプログラムによって保守されていない、外部で取り込まれる CCD 表がある場合には、アプライ・プログラムがソースとして CCD 表を読み取り、正しく機能することができるように、ユーザー自身でこれらの表を保守する必要があります。外部で取り込まれる CCD の保守の詳細は、66 ページの『CCD 表をソースとして保守する (IMS)』を参照してください。

サブスクリプション・セット・メンバー内のターゲットとしての CCD 表の詳細は、594 ページの『整合変更データ (CCD) 表』を参照してください。

表 75 では、CCD 表の列の要旨を示します。

表 75. CCD 表の列

列名	説明
IBMSNAP_INTENTSEQ	変更を一意的に識別するシーケンス番号。この値はグローバルに昇順です。
IBMSNAP_OPERATION	レコードの操作のタイプを示すフラグ。 I 挿入 U 更新 D 削除
IBMSNAP_COMMITSEQ	トランザクションの順序を指定するシーケンス番号。
IBMSNAP_LOGMARKER	データがコミットされた時刻。
<i>user key columns</i>	CCD 表が圧縮されている場合、この列には、ターゲット・キーを構成する列が含まれます。
<i>user non-key columns</i>	ソース表からの非キー・データ列。ソース表の中の列名とこれらの列名が一致している必要はありませんが、データ・タイプは互換性がなければなりません。
<i>user computed columns</i>	SQL 式から派生したユーザー定義の列。ソース・データ・タイプを別のターゲット・データ・タイプに変換するために、SQL 関数で算出列を使用することができます。

関連資料:

- 594 ページの『整合変更データ (CCD) 表』

schema.CD_table

サーバー: キャプチャー・コントロール・サーバー

重要: SQL を使用してこの表を更新するときには、注意してください。この表を不適切に変更すると、データの損失が生じることがあります。

変更データ (CD) 表は、レプリケーション・ソースに対して行われたすべてのコミット済み変更を記録します。CD 表の整理は、整理セット (IBMSNAP_PRUNE_SET) 表によって調整されます。(CD 表の整理についての詳細は、529 ページの

『*schema.IBMSNAP_PRUNE_SET*』を参照してください。) CD 表はキャプチャー・コントロール表とは異なり、ユーザーがレプリケーション・ソースを定義したときに作成されます。キャプチャー・コントロール・サーバー用のコントロール表を作成するときに自動的に作成されるものではありません。

表 76 では、CD 表の列の要旨を示します。

表 76. CD 表の列

列名	説明
IBMSNAP_COMMITSEQ	キャプチャーされたコミット・ステートメントのログ・シーケンス番号。UOW 表にもあるこの列は、アプライ・プログラムが、UOW 表と CD 表を結合する必要なしに、ユーザー・コピー・ターゲット表を処理できるようにするために、CD 表に含められているものです。CD 表と UOW 表の結合が必要な場合は、IBMSNAP_COMMITSEQ 列を使用して結合が行われます。
IBMSNAP_INTENTSEQ	変更 (挿入、更新、または削除) のログ・レコードのログ・シーケンス番号。この値はグローバルに昇順です。更新を削除と挿入のペアとして処理するようにユーザーが選択した場合には、削除行の IBMSNAP_INTENTSEQ 値は、挿入行の対応する値よりも、わずかに小さくなるように作成されます。
IBMSNAP_OPERATION	レコードの操作のタイプを示すフラグ。 I 挿入 U 更新 D 削除
<i>user column after-image</i>	多くの場合、変更後イメージ列には、変更発生後のソース列にある値が含まれます。この列は、ソース列と同じ名前、データ・タイプ、および NULL 属性になります。更新の場合、この列は、更新されたデータの新しい値を反映します。削除の場合、この列は、削除されたデータの値を反映します。挿入の場合、この列は、挿入されたデータの値を反映します。
<i>user column before-image</i>	この列は、ソースが変更前イメージ列の値を含むものとして登録されている場合にのみ CD 表の中に存在します。多くの場合、変更前イメージ列には、変更発生前のソース列にあった値が含まれます。この列は、ソース列と同じ名前になり、登録 (IBMSNAP_REGISTER) 表の中の BEFORE_IMG_PREFIX 列にある値が接頭部として使用されます。また、データ・タイプもソース列と同じになります。しかし、ソース列が NULL 属性であるかどうかに関係なく、挿入操作ではいつでも NULL 値が許されます。更新の場合、この列は、更新されたデータを反映します。削除の場合、この列は、削除されたデータを反映します。挿入の場合、この列は NULL になります。

schema.IBMSNAP_PARTITIONINFO

サーバー: キャプチャー・コントロール・サーバー

デフォルト・スキーマ: ASN

索引: PARTITIONID、USAGE

重要: SQL を使用してこの表を更新するときには、注意してください。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。ユーザーがこの表から行を削除した場合、キャプチャー・プログラムはコールド・スタートせざるをえません。

パーティション情報表は、複数のパーティション化された環境において再始動 (IBMSNAP_RESTART) 表を補強し、各パーティションのログ・ファイルのセットのうち、必要とされる最も古いログ・シーケンスからキャプチャー・プログラムを再始動するための情報を含んでいます。複数パーティション環境では、パーティション情報表と再始動表が、バージョン 7 以前の DB2 レプリケーションの warm_start 表の代わりに使用されます。パーティションが追加されるごとに、この表に行が挿入されます。キャプチャー・プログラムは、最初のデータベース CONNECT の発行後に DB2 が使用した最初のログ・シーケンス番号から、新規パーティションのログ・ファイルの読み取りを開始します。

キャプチャー・プログラムをまだ開始したことがない場合は、この表は空であるため、キャプチャー・プログラムはコールド・スタートを実行する必要があります。

表 77 では、パーティション情報表の列の要旨を示します。

表 77. パーティション情報表の列

列名	説明
PARTITIONID	データ・タイプ: INT。 NULL 可能: いいえ。 各有効なパーティションのパーティション ID。
USAGE	データ・タイプ: CHAR(1)。 NULL 可能: いいえ。 ログ・シーケンス番号 (LSN) の使用状況。この列の "R" は、LSN が始動していることを示します。
SEQUENCE	データ・タイプ: ビット・データの場合 CHAR(10)。 NULL 可能: いいえ。 パーティション ID を持つノードの再始動 LSN。
STATUS	データ・タイプ: CHAR(1)。 NULL 可能: はい。 パーティションの状況。この列の "A" は、パーティションがアクティブであることを示します。この列は将来の利用のために予約されています。
LAST_UPDATE	データ・タイプ: TIMESTAMP。 NULL 可能: はい。 パーティション ID を持つノードの再始動 LSN が最後に更新されたときのタイム・スタンプ。

schema.IBMSNAP_PRUNCNTL

サーバー: キャプチャー・コントロール・サーバー

デフォルト・スキーマ: ASN

索引:

SOURCE_OWNER、SOURCE_TABLE、SOURCE_VIEW_QUAL、APPLY_QUAL、SET_NAME、TARGET_SERVER、TARGET_TABLE、TARGET_OWNER

重要: SQL を使用してこの表を更新するときには、注意してください。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。

整理コントロール表には、このキャプチャー・スキーマに対して定義されている、すべてのサブスクリプション・セット・メンバーに関する詳細情報が入っています。この表は、整理時に整理セット (IBMSNAP_PRUNE_SET) 表と一緒に使用されます。また、アプライ・プログラムとキャプチャー・プログラムの間で初期化ハンドシェイク・プロセス時にも使用されます。

DB2 ソースの場合は、**prune** コマンドを発行して整理を呼び出すこともできますし、自動的に起動することもできます。AUTOPRUNE を設定するためにキャプチャー・パラメーターを使用する方法についての詳細は、519 ページの

『*schema.IBMSNAP_CAPPARMS*』を参照してください。DB2 以外のリレーショナル・ソースの場合の整理は、ユーザーがソースを登録したときに作成された整理トリガーを使用して行われます。

表 78 では、整理コントロール表の列の要旨を示します。

表 78. 整理コントロール表の列

列名	説明
TARGET_SERVER	データ・タイプ: CHAR(18)。NULL 可能: いいえ。 このメンバーのターゲット表またはビューが置かれているサーバー名。
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。 このメンバーのターゲット表またはビューの上位修飾子。
TARGET_TABLE	データ・タイプ: VARCHAR(128)、DB2 UDB for z/OS バージョン 8 互換モード・サブシステムかそれ以前の場合 VARCHAR(18)。NULL 可能: いいえ。 このメンバーのターゲット表またはビューの名前。
SYNCHTIME	データ・タイプ: TIMESTAMP。NULL 可能: はい。 キャプチャー・プログラムは、アプライ・プログラムとの初期化ハンドシェイク・プロセス時にこのタイム・スタンプを設定します。この値は、CAPSTART シグナル挿入のトランザクションと関連付けられたコミット・ログ・レコードのタイム・スタンプによって決まります。この値は、その後初期化処理が行われなにかぎり、再度更新されることはありません。

IBMSNAP_PRUNCNTL

表 78. 整理コントロール表の列 (続き)

列名	説明														
SYNCHPOINT	<p>データ・タイプ: ビット・データの場合 CHAR(10)。 NULL 可能: はい。</p> <p>キャプチャー・プログラムは、アプライ・プログラムとの初期化ハンドシェイク・プロセス時にこの値を設定します。この値は、CAPSTART シグナル挿入のトランザクションと関連付けられたコミット・ログ・レコードのログ・シーケンス番号によって決まります。この値は、その後に初期化処理が行われないかぎり、再度更新されることはありません。</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>このメンバーのソース表またはビューの上位修飾子。</p>														
SOURCE_TABLE	<p>データ・タイプ: VARCHAR(128)、 DB2 UDB for z/OS バージョン 8 互換モード・サブシステムかそれ以前の場合 VARCHAR(18)。 NULL 可能: いいえ。</p> <p>このメンバーのソース表またはビューの名前。</p>														
SOURCE_VIEW_QUAL	<p>データ・タイプ: SMALLINT。 NULL 可能: いいえ。</p> <p>この列は、SOURCE_OWNER および SOURCE_TABLE 列に同じ値を持つ、複数の異なるソース・ビューがある場合の複数登録をサポートするために使用されます。この値は、ソースとして定義されている物理表の場合は 0 に、ソースとして定義されているビューの場合は 0 より大きい値に設定されます。</p>														
APPLY_QUAL	<p>データ・タイプ: CHAR(18)。 NULL 可能: いいえ。</p> <p>このメンバーを処理しているアプライ・プログラムを識別するためのアプライ修飾子。</p>														
SET_NAME	<p>データ・タイプ: CHAR(18)。 NULL 可能: いいえ。</p> <p>このサブスクリプション・セット・メンバーが所属するサブスクリプション・セットの名前。</p>														
CNTL_SERVER	<p>データ・タイプ: CHAR(18)。 NULL 可能: いいえ。</p> <p>APPLY_QUAL により識別される、このアプライ・プログラムのアプライ・コントロール表が置かれているサーバーの名前。</p>														
TARGET_STRUCTURE	<p>データ・タイプ: SMALLINT。 NULL 可能: いいえ。</p> <p>ターゲット表またはビューのタイプを示す値。</p> <table border="0"> <tr> <td>1</td> <td>ソース表</td> </tr> <tr> <td>3</td> <td>CCD 表</td> </tr> <tr> <td>4</td> <td>ポイント・イン・タイム表</td> </tr> <tr> <td>5</td> <td>基礎集約表</td> </tr> <tr> <td>6</td> <td>変更集約表</td> </tr> <tr> <td>7</td> <td>レプリカ表</td> </tr> <tr> <td>8</td> <td>ユーザー・コピー表</td> </tr> </table>	1	ソース表	3	CCD 表	4	ポイント・イン・タイム表	5	基礎集約表	6	変更集約表	7	レプリカ表	8	ユーザー・コピー表
1	ソース表														
3	CCD 表														
4	ポイント・イン・タイム表														
5	基礎集約表														
6	変更集約表														
7	レプリカ表														
8	ユーザー・コピー表														
CNTL_ALIAS	<p>データ・タイプ: CHAR(8)。 NULL 可能: はい。</p> <p>CNTL_SERVER 列で指定されているアプライ・コントロール・サーバーに対応する DB2 Universal Database 別名。</p>														

表 78. 整理コントロール表の列 (続き)

列名	説明
PHYS_CHANGE_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>この特定のサブスクリプション・セット・メンバーのソースに関連付けられた登録 (IBMSNAP_REGISTER) 表の PHYS_CHANGE_OWNER 列の中の値。</p>
PHYS_CHANGE_TABLE	<p>データ・タイプ: VARCHAR(128)、 DB2 UDB for z/OS バージョン 8 互換モード・サブシステムかそれ以前の場合 VARCHAR(18)。 NULL 可能: はい。</p> <p>この特定のサブスクリプション・セット・メンバーのソースに関連付けられた登録 (IBMSNAP_REGISTER) 表の PHYS_CHANGE_TABLE 列の中の値。</p>
MAP_ID	<p>データ・タイプ: VARCHAR(10)。 NULL 可能: いいえ。</p> <p>この表で、より短い、簡単に使用できる索引を提供する一意性の因子。シグナル表への CAPSTART 挿入を、整理コントロール表の中の適切な行に関連付けるためにも使用されます。</p>

schema.IBMSNAP_PRUNE_LOCK

サーバー: キャプチャー・コントロール・サーバー

デフォルト・スキーマ: ASN

索引: なし

重要: SQL を使用してこの表を更新するときには、注意してください。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。

整理ロック表は、コールド・スタート、または保存限度整理時に、CD 表へのアクセスを直列化するために使用されます。この表は、これらの重要なフェーズでアプライ・プログラムが CD 表にアクセスしないようにするものです。この表には行がありません。

schema.IBMSNAP_PRUNE_SET

サーバー: キャプチャー・コントロール・サーバー

デフォルト・スキーマ: ASN

索引: TARGET_SERVER、APPLY_QUAL、SET_NAME

重要: SQL を使用してこの表を更新するときには、注意してください。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。

整理セット表は、各サブスクリプション・セットについてキャプチャー・プログラムおよびアプライ・プログラムの進行をトラッキングすることにより、CD 表および UOW 表の整理の調整を行います。1 つのソースからターゲットへのマッピング

IBMSNAP_PRUNE_SET

に対して 1 つの行がある整理コントロール (IBMSNAP_PRUNCNTL) 表とは異なり、整理セット表には、1 つのサブスクリプション・セットに対して 1 つの行があります。

表 79 では、整理セット表の列の要旨を示します。

表 79. 整理セット表の列

列名	説明
TARGET_SERVER	データ・タイプ: CHAR(18)。 NULL 可能: いいえ。 このセットのターゲット表またはビューが置かれているサーバー名。
APPLY_QUAL	データ・タイプ: CHAR(18)。 NULL 可能: いいえ。 このセットを処理しているアプライ・プログラムを識別するためのアプライ修飾子。
SET_NAME	データ・タイプ: CHAR(18)。 NULL 可能: いいえ。 サブスクリプション・セットの名前。
SYNCHTIME	データ・タイプ: TIMESTAMP。 NULL 可能: はい。 アプライ・プログラムはこの列を使用して進行状況を記録します。サブスクリプション・セットのデータの処理が、このタイム・スタンプまで終了していることを示します。
SYNCHPOINT	データ・タイプ: ビット・データの場合 CHAR(10)。 NULL 可能: いいえ。 アプライ・プログラムはこの列を使用して進行状況を記録します。サブスクリプション・セットのデータの処理が、この同期点の値まで終了していることを示します。

schema.IBMSNAP_REG_EXT (OS/400)

サーバー: キャプチャー・コントロール・サーバー

デフォルト・スキーマ: ASN

索引: VERSION、SOURCE_OWNER、SOURCE_TABLE、SOURCE_VIEW_QUAL

重要: SQL を使用してこの表を更新するときには、注意してください。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。

登録拡張表は、登録 (IBMSNAP_REGISTER) 表の補足情報を提供する OS/400 固有の表です。登録拡張表には、登録表のすべての行に対して、OS/400 固有の列をいくつか追加で含む、対応する行があります。

この表は、登録表のトリガー・プログラム (ライブラリー QDP4 の QZSNJLV8) によって保守されます。トリガーは、登録表が作成されるときに定義されます。

この表の情報は、OS/400 サーバー上でレプリケーション・ソースがどこでどのように定義されているかをトラッキングするために使用されます。

531 ページの表 80 では、登録拡張表の列の要旨を示します。

表 80. 登録拡張表の列

列名	説明
VERSION	<p>データ・タイプ: INT。 NULL 可能: いいえ。</p> <p>ソースの登録に使用された DB2 DataPropagator for iSeries のバージョン。</p>
SOURCE_OWNER	<p>データ・タイプ: VARCHAR(30)。 NULL 可能: いいえ。</p> <p>ユーザーにより登録されたソース表またはビューの上位修飾子。</p>
SOURCE_TABLE	<p>データ・タイプ: VARCHAR(128)。 NULL 可能: いいえ。</p> <p>ユーザーにより登録されたソース表またはビューの名前。</p>
SOURCE_NAME	<p>データ・タイプ: CHAR(10)。 NULL 可能: はい。</p> <p>コマンドの発行に使用されたソース表またはビューの 10 文字のシステム名。</p>
SOURCE_MBR	<p>データ・タイプ: CHAR(10)。 NULL 可能: はい。</p> <p>ジャーナル項目の受信 (RCVJRNE) コマンドの発行および ALIAS サポートに使用されるソース表メンバーの名前。</p>
SOURCE_TABLE_RDB	<p>データ・タイプ: CHAR(18)。 NULL 可能: はい。</p> <p>リモート・ジャーナルを使用する場合、この列には、ソース表が実際に置かれているシステムのデータベース名が含まれます。ローカル・ジャーナルの場合、この列は NULL になります。</p>
JRN_LIB	<p>データ・タイプ: CHAR(10)。 NULL 可能: はい。</p> <p>ソース表が使用するジャーナルのライブラリー名。</p>
JRN_NAME	<p>データ・タイプ: CHAR(10)。 NULL 可能: はい。</p> <p>ソース表が使用するジャーナルの名前。この列で、アスタリスクの後ろに 9 つの空白が続くときには、ソース表が現在ジャーナルの中にあることを意味します。この場合キャプチャー・プログラムは、このソースのデータをキャプチャーすることはできません。</p>
FR_START_TIME	<p>データ・タイプ: TIMESTAMP。 NULL 可能: はい。</p> <p>アプライ・プログラムがフル・リフレッシュの実行を開始した時刻。</p>
SOURCE_VIEW_QUAL	<p>データ・タイプ: SMALLINT。 NULL 可能: いいえ。</p> <p>登録表内の類似した列を突き合わせることによって、サブスクリプションのビューをサポートします。この値は、ソースとして定義されている物理表の場合は 0 に、ソースとして定義されているビューの場合は 0 より大きい値に設定されます。SOURCE_OWNER および SOURCE_TABLE 列の値が同じである、複数の異なるソース・ビューについて複数のサブスクリプションをサポートするには、この列が必要です。</p>

IBMSNAP_REG_EXT

表 80. 登録拡張表の列 (続き)

列名	説明
CMT_BEHAVIOR_CASE	<p>データ・タイプ: SMALLINT。 NULL 可能: デフォルトでは不可。デフォルト: 0。</p> <p>ソース表を更新するアプリケーション・プログラムがコミットメント・コントロールを使用する方法を表す整数。キャプチャー・プログラムはこの値を使用して、構成済みであるが、CD 表にまだ書き込んでいない CD 行に使用するメモリーを管理します。</p> <p>-1 アプリケーションのコミットメント・コントロール・パターンはまだ設定されていません。これは、この列の初期値です。</p> <p>0 ソースを更新するアプリケーションではコミットメント・コントロールが使用されていません。</p> <p>1 ソースを更新するアプリケーションはすべて、コミットメント・コントロールを使用します。このため、コミットメント・コントロール下の同じソース表が、2 つの異なるアプリケーションから同時に更新されることはありません。</p> <p>2 ソースを更新する同時アプリケーションの中にはコミットメント・コントロールを使用するものも使用しないものもあります。2 つのアプリケーションが同時にコミットメント・コントロールを使用してソース表を更新する可能性があります。</p>
MAX_ROWS_BTWN_CMTS	<p>データ・タイプ: SMALLINT。 NULL 可能: デフォルトでは不可。デフォルト: 0。</p> <p>キャプチャー・プログラムがデータを CD 表にコミットする前に処理できる行の最大数。</p>

schema.IBMSNAP_REGISTER

重要: SQL を使用してこの表を更新するときには、注意してください。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。

サーバー: キャプチャー・コントロール・サーバー

デフォルト・スキーマ: ASN

索引: SOURCE_OWNER、SOURCE_TABLE、SOURCE_VIEW_QUAL

登録表は、レプリケーション・ソース表の名前、属性、および、これらに関連付けられた CD 表および CCD 表の名前など、レプリケーション・ソースに関する情報を保持します。キャプチャー・プログラムで処理されるように、新しいレプリケーション・ソース表またはビューが定義されるたびに、この表に行が 1 つ自動的に挿入されます。

レプリケーション・ソースの定義を調べる必要があるときには、登録表を使用します。

533 ページの表 81 では、登録表の列の要旨を示します。

表 81. 登録表の列

列名	説明														
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・サブシステムの場合 VARCHAR(128)。 NULL 可能: いいえ。</p> <p>ユーザーにより登録されたソース表またはビューの上位修飾子。</p>														
SOURCE_TABLE	<p>データ・タイプ: VARCHAR(128)、 DB2 UDB for z/OS バージョン 8 互換モード・サブシステムかそれ以前の場合 VARCHAR(18)。 NULL 可能: いいえ。</p> <p>ユーザーにより登録されたソース表またはビューの名前。</p>														
SOURCE_VIEW_QUAL	<p>データ・タイプ: SMALLINT。 NULL 可能: いいえ。</p> <p>この列は、SOURCE_OWNER および SOURCE_TABLE 列に同じ値を持つ、複数の異なるソース・ビューがある場合の複数登録をサポートするために使用されます。この値は、ソースとして定義されている物理表の場合は 0 に、ソースとして定義されているビューの場合は 0 より大きい値に設定されます。</p>														
GLOBAL_RECORD	<p>データ・タイプ: CHAR(1)。 NULL 可能: いいえ。</p>														
SOURCE_STRUCTURE	<p>データ・タイプ: SMALLINT。 NULL 可能: いいえ。</p> <p>ソース表またはビューの構造を示す値</p> <table border="0"> <tr> <td>1</td> <td>ユーザー表</td> </tr> <tr> <td>3</td> <td>CCD 表</td> </tr> <tr> <td>4</td> <td>ポイント・イン・タイム表</td> </tr> <tr> <td>5</td> <td>基礎集約表</td> </tr> <tr> <td>6</td> <td>変更集約表</td> </tr> <tr> <td>7</td> <td>レプリカ表</td> </tr> <tr> <td>8</td> <td>ユーザー・コピー表</td> </tr> </table>	1	ユーザー表	3	CCD 表	4	ポイント・イン・タイム表	5	基礎集約表	6	変更集約表	7	レプリカ表	8	ユーザー・コピー表
1	ユーザー表														
3	CCD 表														
4	ポイント・イン・タイム表														
5	基礎集約表														
6	変更集約表														
7	レプリカ表														
8	ユーザー・コピー表														
SOURCE_CONDENSED	<p>データ・タイプ: CHAR(1)。 NULL 可能: いいえ。</p> <p>ソース表がコンデンス表であるかどうかを示すフラグ。圧縮されている場合は、同じキーを持つすべての行が、1 つの行に圧縮されます。</p> <table border="0"> <tr> <td>Y</td> <td>ソースは圧縮されています。</td> </tr> <tr> <td>N</td> <td>ソースは圧縮されていません。</td> </tr> <tr> <td>A</td> <td>ソースは基礎集約表または変更集約表です。</td> </tr> </table>	Y	ソースは圧縮されています。	N	ソースは圧縮されていません。	A	ソースは基礎集約表または変更集約表です。								
Y	ソースは圧縮されています。														
N	ソースは圧縮されていません。														
A	ソースは基礎集約表または変更集約表です。														
SOURCE_COMPLETE	<p>データ・タイプ: CHAR(1)。 NULL 可能: いいえ。</p> <p>主キー値を持つ行がソース表にどのように保管されるかを示すフラグ。</p> <table border="0"> <tr> <td>Y</td> <td>ソース表は、関係するそれぞれの主キー値につき 1 行を保持します。</td> </tr> <tr> <td>N</td> <td>ソース表は、主キー値の行のサブセットを保持します。</td> </tr> </table>	Y	ソース表は、関係するそれぞれの主キー値につき 1 行を保持します。	N	ソース表は、主キー値の行のサブセットを保持します。										
Y	ソース表は、関係するそれぞれの主キー値につき 1 行を保持します。														
N	ソース表は、主キー値の行のサブセットを保持します。														

IBMSNAP_REGISTER

表 81. 登録表の列 (続き)

列名	説明
CD_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・サブシステムの場合 VARCHAR(128)。 NULL 可能: はい。</p> <p>ソースの CD 表の上位修飾子。</p> <p>ソースが表の場合 外部 CCD 表ではないすべての登録済みソース表の場合、この列は、このソース表に関連付けられた CD 表の上位修飾子を保持します。</p> <p>ソースがビューの場合 この列は、CD ビューの上位修飾子を保持します。</p> <p>ソースが外部 CCD 表の場合 この列は NULL です。</p>
CD_TABLE	<p>データ・タイプ: VARCHAR(128)、 DB2 UDB for z/OS バージョン 8 互換モード・サブシステムかそれ以前の場合 VARCHAR(18)。 NULL 可能: はい。</p> <p>ソースの CD 表の名前。</p> <p>ソースが表の場合 外部 CCD 表ではないすべての登録済みソース表の場合、この列には、このソース表のキャプチャーされた更新を保留する CD 表の名前が入ります。</p> <p>ソースがビューの場合 この列は、CD ビューの名前を保持します。</p> <p>ソースが外部 CCD 表の場合 この列は NULL です。</p>
PHYS_CHANGE_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・サブシステムの場合 VARCHAR(128)。 NULL 可能: はい。</p> <p>アプライ・プログラムが変更キャプチャー・レプリケーションに使用する、表またはビューの上位修飾子。</p> <p>ソースが表の場合 外部 CCD 表ではないすべての登録済みソース表の場合、この列は、このソース表に関連付けられた物理 CD 表の上位修飾子を保持します。</p> <p>ソースがビューの場合 この列は、このソース・ビューに関連付けられた物理 CD 表の上位修飾子を保持します。</p> <p>ソースが外部 CCD 表の場合 この列は、外部 CCD 表の上位修飾子を保持します。</p>

表 8I. 登録表の列 (続き)

列名	説明
PHYS_CHANGE_TABLE	<p>データ・タイプ: VARCHAR(128)、 DB2 UDB for z/OS バージョン 8 互換モード・サブシステムかそれ以前の場合 VARCHAR(18)。 NULL 可能: はい。</p> <p>アプライ・プログラムが変更キャプチャー・レプリケーションに使用する、表またはビューの名前。</p> <p>ソースが表の場合 外部 CCD 表ではないすべての登録済みソース表の場合、この列は、このソース表に関連付けられた物理 CD 表の名前を保持します。</p> <p>ソースがビューの場合 この列は、このソース・ビューに関連付けられた物理 CD 表の名前を保持します。</p> <p>ソースが外部 CCD の場合 この列は、外部 CCD 表の名前を保持します。</p>
CD_OLD_SYNCHPOINT	<p>データ・タイプ: ビット・データの場合 CHAR(10)。 NULL 可能: はい。</p> <p>この列は、アプライ・プログラムとキャプチャー・プログラムの間の初期ハンドシェイクに使用されます。キャプチャー・プログラムはその後、ソース・ログのこのログ・シーケンス番号からデータのキャプチャーを開始します。この列は、CD 表で保存限度整理が行われたことを示すためにも使用されます。この値が NULL の場合、登録は非アクティブです。</p>
CD_NEW_SYNCHPOINT	<p>データ・タイプ: ビット・データの場合 CHAR(10)。 NULL 可能: はい。</p> <p>この列は、キャプチャー・プログラムが CD 表に新しい行を挿入するにつれて進められます。アプライ・プログラムはこの列を使用して、複製する新しい変更があるかどうかを確認します。</p>
DISABLE_REFRESH	<p>データ・タイプ: SMALLINT。 NULL 可能: はい。</p> <p>フル・リフレッシュが使用可能かどうかを示すフラグ。</p> <p>0 フル・リフレッシュを使用可能にします。</p> <p>1 フル・リフレッシュを使用不可にします。</p>
CCD_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・サブシステムの場合 VARCHAR(128)。 NULL 可能: はい。</p> <p>ソースに内部 CCD 表が関連付けられている場合、この列は、内部 CCD の上位修飾子を保持します。外部 CCD 表の場合、この列は NULL になります。</p>
CCD_TABLE	<p>データ・タイプ: VARCHAR(128)、 DB2 UDB for z/OS バージョン 8 互換モード・サブシステムかそれ以前の場合 VARCHAR(18)。 NULL 可能: はい。</p> <p>ソースに内部 CCD 表が関連付けられている場合、この列は、内部 CCD の名前を保持します。外部 CCD 表の場合、この列は NULL になります。</p>

IBMSNAP_REGISTER

表 81. 登録表の列 (続き)

列名	説明
CCD_OLD_SYNCHPOINT	<p>データ・タイプ: ビット・データの場合 CHAR(10)。 NULL 可能: はい。</p> <p>CCD 表が再初期化されたときのログ・シーケンス番号。この列は、CCD のフル・リフレッシュ処理に関連しています。この列の値を変更する必要があるのは、CCD 表が初期に、またはその後に、フル・リフレッシュされた場合だけです。この値は、CCD 表に残っているどの行よりも古いものである可能性があります。この列が保守されないと、CCD 表をレプリケーション・ソースとして使用するアプライ・プログラムは、CCD 表が再初期化されたかどうか分からないので、CCD ソースの完全なコピーを再初期化できません。</p>
SYNCHPOINT	<p>データ・タイプ: ビット・データの場合 CHAR(10)。 NULL 可能: はい。</p> <p>グローバル行の場合 (GLOBAL_RECORD = Y)、同期点は、キャプチャー・プログラムによって処理された最後のログまたはジャーナル・レコードのログ・シーケンス番号を表します。CCD 表 (内部または外部) に関する登録情報を含む IBMSNAP_REGISTER 表の中の行の場合、CCD 表の中に使用可能な新しいデータがあることを示すために、同期点の値は、CCD 表を保守するプログラムによって進められます。</p>
SYNCHTIME	<p>データ・タイプ: TIMESTAMP。 NULL 可能: はい。</p> <p>グローバル行の場合 (GLOBAL_RECORD = Y)、同期時刻は、キャプチャー・プログラムによって処理された最後のログまたはジャーナル・レコードのタイム・スタンプを表します。キャプチャー・プログラムが DB2 ログの最後に達すると、同期時刻は現在の DB2 タイム・スタンプまで進められます。CCD 表 (内部または外部) に関する登録情報を含む IBMSNAP_REGISTER 表の中の行の場合、CCD 表の中の使用可能データの現行性 (currency) を示すために、同期時刻の値は、CCD 表を保守するプログラムによって進められます。</p>
CCD_CONDENSED	<p>データ・タイプ: CHAR(1)。 NULL 可能: はい。</p> <p>このソースに関連付けられた内部 CCD が圧縮されているかどうかを示すフラグ。圧縮されている場合は、同じキーを持つすべての行が、1 つの行に圧縮されます。</p> <p>Y 内部 CCD は圧縮されています。</p> <p>N 内部 CCD は圧縮されていません。</p> <p>NULL このソースに対して内部 CCD 表が定義されていません。</p>
CCD_COMPLETE	<p>データ・タイプ: CHAR(1)。 NULL 可能: はい。</p> <p>このソースに関連付けられた内部 CCD 表が完成しているかどうかを示すフラグ。完成しているとは、ソース表からのすべての行が初期に含まれていることを意味します。</p> <p>N 内部 CCD は完成していません。</p> <p>NULL このソースに対して内部 CCD 表が定義されていません。</p>

表 81. 登録表の列 (続き)

列名	説明
ARCH_LEVEL	<p>データ・タイプ: CHAR(4)。NULL 可能: いいえ。</p> <p>レプリケーション・コントロール表の構造レベルは以下のとおりです。</p> <p>0801 バージョン 8 SQL Replication</p> <p>0803 バージョン 8 SQL Replication (Oracle ソースの拡張サポート付き)</p> <p>0805 バージョン 8 SQL Replication (DB2 for z/OS 新機能のサポート付き)</p>
DESCRIPTION	<p>データ・タイプ: CHAR(254)。NULL 可能: はい。</p> <p>レプリケーション・ソースの記述。</p>
BEFORE_IMG_PREFIX	<p>データ・タイプ: VARCHAR(4)。NULL 可能: はい。</p> <p>CD 表の中の変更前イメージ列の名前を識別する 1 文字の接頭部。変更前イメージの接頭部と CD 列名の組み合わせは判別可能なものにする必要があります。つまり、接頭部の付いた CD 列名は、現在の列名、そして想定される変更後イメージ列の名前と同じではありません。BEFORE_IMG_PREFIX の長さ (バイト単位) は次のとおりです。</p> <ol style="list-style-type: none"> 1 ASCII または EBCDIC の 1 バイトの接頭文字の場合。 2 ASCII の 2 バイトの接頭文字の場合。 4 EBCDIC DBCS 接頭文字の場合。この長さは、シフトインおよびシフトアウト文字を見込んでいます。
CONFLICT_LEVEL	<p>データ・タイプ: CHAR(1)。NULL 可能: はい。</p> <p>このソースの競合検出のレベルを示すフラグ。</p> <ol style="list-style-type: none"> 0 アプライ・プログラムは競合をチェックしません。更新の競合を防止するために、アプリケーション側でデータ整合性を適用する必要があります。 1 カスケード・トランザクション・リジェクトの標準検出。アプライ・プログラムは、これまでにキャプチャーされた変更に基づいて競合をチェックします。アプライ・プログラムは、レプリカに競合するトランザクションがあればそれを元に戻し、競合するトランザクションに対して従属関係を持つトランザクションがあれば、それも元に戻します。アプライ・プログラムが競合検出を開始した後でキャプチャーされた変更は、このアプライ・サイクルではチェックされません。 2 カスケード・トランザクション・リジェクトの拡張検出。アプライ・プログラムは、キャプチャー・プログラムがログまたはジャーナルからすべての変更をキャプチャーするまで待ち (SYNCHTIME 列の記述を参照)、1 に設定された場合と同じように、標準競合検出を行います。アプライ・プログラムはエンキュー中、競合検出プロセス中に変更が行われないように、ソース表にロックをかけます。

IBMSNAP_REGISTER

表 81. 登録表の列 (続き)

列名	説明
CHG_UPD_TO_DEL_INS	<p>データ・タイプ: CHAR(1)。 NULL 可能: はい。</p> <p>キャプチャー・プログラムが CD 表に更新を保管する方法を示すフラグ。</p> <p>Y キャプチャー・プログラムは、1 つは削除用、1 つは挿入用の 2 つの行を CD 表で使用して更新を保管します。アプライ・プログラムは最初に削除を処理し、次に挿入を処理します。このフラグを Y に設定すると、レプリケーション・ソースに対するすべての更新は、2 つの行を使用して CD 表に保管されます。このフラグは、パーティション化された列、またはサブスクリプション・セット述部から参照される列の更新が正しく処理されるようにします。</p> <p>N ソース表に対する更新はそれぞれ、CD 表の中の 1 つの行に保管されません。</p>
CHGONLY	<p>データ・タイプ: CHAR(1)。 NULL 可能: はい。</p> <p>キャプチャー・プログラムがソースで発生したすべての変更をキャプチャーするか、登録済み列で発生した変更のみをキャプチャーするかを示すフラグ。一般的には、キャプチャー・プログラムによって CD 表に挿入される行数を最小化するために、このオプションは Y に設定されます。しかし、ソース表の中のどの行が更新されたかを正確にトラッキングするために、このオプションを N に設定することも考えられます。たとえば、ソース表の中のどの行が更新されたかを監査するために、主キー列の値だけをキャプチャーできます。</p> <p>Y キャプチャー・プログラムは、ソース表の登録済み列で発生した変更のみをキャプチャーします。</p> <p>N キャプチャー・プログラムは、ソース表すべての列で発生した変更をキャプチャーします。</p>
RECAPTURE	<p>データ・タイプ: CHAR(1)。 NULL 可能: はい。</p> <p>この列は Update-anywhere レプリケーションに使用されます。表またはビューで行われた変更の再キャプチャーを行い、他の表またはビューに転送するかどうかを示すフラグを保持します。</p> <p>マスター側の表の場合:</p> <p>N レプリカから適用されたマスターに対する更新の再キャプチャーは行われず、他のレプリカに複製されません。</p> <p>Y レプリカから適用されたマスターに対する更新は他のレプリカに複製されます。</p> <p>レプリカ側の表の場合:</p> <p>Y マスターから適用されたレプリカに対する更新の再キャプチャーが行われ、このレプリカをソースとして使用する他の表に複製するために使用できます。</p> <p>N マスターから適用されたレプリカに対する更新の再キャプチャーは行われません。</p>
OPTION_FLAGS	<p>データ・タイプ: CHAR(4)。 NULL 可能: いいえ。</p> <p>DB2 レプリケーションの将来のオプション用に予約済み。この列には現在、デフォルト値の NNNN が入っています。</p>

表 81. 登録表の列 (続き)

列名	説明
STOP_ON_ERROR	<p>データ・タイプ: CHAR(1)。 NULL 可能: はい、デフォルトあり。デフォルト: Y。</p> <p>キャプチャー・プログラムが、始動、開始、再開、または CD 表への行の挿入時にエラーを検出したときに、終了するか、登録の処理を停止するだけかを示すフラグ。</p> <p>Y キャプチャー・プログラムは、始動、開始、再開、または CD 表への行の挿入時にエラーを検出すると終了します。</p> <p>N キャプチャー・プログラムは、始動、開始、再開、または CD 表への行の挿入時にエラーを検出すると、登録を停止しますが、終了はしません。プログラムはその他の登録処理を続けます。</p>
STATE	<p>データ・タイプ: CHAR(1)。 NULL 可能: はい、デフォルトあり。デフォルト: I。</p> <p>登録の状態を示すフラグ。</p> <p>S キャプチャー・プログラムはこの登録の処理を停止しました。アプライ・プログラムは、登録が修復され、I (非アクティブ) 状態になるまで、この登録を処理しません。</p> <p>A 登録はアクティブです。</p> <p>I 登録は非アクティブです。</p>
STATE_INFO	<p>データ・タイプ: CHAR(8)。 NULL 可能: はい。</p> <p>キャプチャー・プログラムが登録の処理を停止した場合、この列は、この障害に関して発行されたエラー・メッセージを保持します。</p>

schema.IBMSNAP_REG_SYNCH (DB2 以外のリレーショナル)

サーバー: キャプチャー・コントロール・サーバー

デフォルト・スキーマ: ASN

索引: TRIGGER_ME

重要: SQL を使用してこの表を更新するときには、注意してください。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。

登録同期表は、アプライ・プログラムが DB2 以外のリレーショナル・データ・ソースからデータをフェッチする準備をするときに、登録 (IBMSNAP_REGISTER) 表の中のすべての行の SYNCHPOINT 値の更新を開始するために、更新トリガーを使用します。

540 ページの表 82 では、登録同期表の列の要旨を示します。

IBMSNAP_REG_SYNC

表 82. 登録同期表の列

列名	説明
TRIGGER_ME	データ・タイプ: CHAR(1)。 NULL 可能: いいえ。 登録表の中のすべての行の SYNCHPOINT 値を更新するためにトリガーが開始されたかどうかを示す Y というフラグ。
TIMESTAMP	Microsoft SQL Server および Sybase のソースの場合、この列は、表のタイム・スタンプ列で更新が発生したときにシステムから生成されるユニークな番号を保持します。この値は、登録 (IBMSNAP_REGISTER) 表に記録される SYNCHPOINT 値を生成するために使用されます。

schema.IBMSNAP_RESTART

サーバー: キャプチャー・コントロール・サーバー

デフォルト・スキーマ: ASN

索引: なし

重要: SQL を使用してこの表を更新するときには、注意してください。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。ユーザーがこの表から行を削除した場合、キャプチャー・プログラムはコールド・スタートせざるをえません。

再始動表には、必要とされる中で一番古いログまたはジャーナル・レコードからキャプチャー・プログラムを再始動できるようにするための情報が含まれます。この表は、DB2 レプリケーションのバージョン 7 およびそれ以前のバージョンのウォーム・スタート表を置き換えるものです。この表は、コミット・ポイントのたびに更新される行を保持します。このため、キャプチャー・プログラムは、すでに処理済みで、CD 表および UOW 表に挿入済みの情報の再キャプチャーを行う必要がなく、常に正しい場所から正確に再始動できます。

キャプチャー・プログラムをまだ開始したことがない場合は、この表は空であるため、キャプチャー・プログラムはコールド・スタートを実行する必要があります。

次の 2 つの表は、オペレーティング・システム別の再始動表のレイアウトを示しています。

表 83. Linux、UNIX、Windows、および z/OS の再始動表の列

列名	説明
MAX_COMMITSEQ	データ・タイプ: ビット・データの場合 CHAR(10)。 NULL 可能: いいえ。 キャプチャー・プログラムが CD 表および UOW 表にコミット済みの論理ログ・シーケンス番号 (IBMSNAP_COMMITSEQ) の最大値。
MAX_COMMIT_TIME	データ・タイプ: TIMESTAMP。 NULL 可能: いいえ。 MAX_COMMITSEQ 列のログ・シーケンス番号に関連付けられたタイム・スタンプ。

表 83. Linux、UNIX、Windows、および z/OS の再始動表の列 (続き)

列名	説明
MIN_INFLIGHTSEQ	<p>データ・タイプ: ビット・データの場合 CHAR(10)。 NULL 可能: いいえ。</p> <p>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、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・シーケンス番号です。</p>
CURR_COMMIT_TIME	<p>データ・タイプ: TIMESTAMP。 NULL 可能: いいえ。</p> <p>この表がキャプチャー・プログラムによって更新されたときの、ローカルの現行タイム・スタンプ。</p>
CAPTURE_FIRST_SEQ	<p>データ・タイプ: ビット・データの場合 CHAR(10)。 NULL 可能: いいえ。</p> <p>最後にコールド・スタートを実行したときにキャプチャー・プログラムの開始時点であったリカバリー・ログに関連付けられた論理ログ・シーケンス番号。この値は、キャプチャー・プログラムでコールド・スタートを実行せざるをえなくなるようなデータベース RESTORE が発生したかどうかを検出するために使用されます。このような事態は、データベース・ログ・マネージャーが特定の RESTORE 操作時にログ・シーケンス番号を再利用することがあるために発生します。</p>

OS/400 の場合、再始動表は **RCVJRNE** (ジャーナル項目の受信) コマンドの開始時刻を判別するために使用されます。1 つのレプリケーション・ソース、またはレプリケーション・ソースのグループで使用されるジャーナルごとに、再始動表に行が 1 つ挿入されます。

索引: JRN_LIB, JRN_NAME

表 84. OS/400 の再始動表の列

列名	説明
MAX_COMMITSEQ	<p>データ・タイプ: ビット・データの場合 CHAR(10)。 NULL 可能: いいえ。</p> <p>UOW 表からの最新のコミットのジャーナル・レコード番号。</p>
MAX_COMMIT_TIME	<p>データ・タイプ: TIMESTAMP。 NULL 可能: いいえ。</p> <p>MAX_COMMITSEQ 列のジャーナル・レコード番号に関連付けられたタイム・スタンプ、または、キャプチャー・プログラムがログの処理を終了し、実行する処理がない場合は、現在のタイム・スタンプ。</p>
MIN_INFLIGHTSEQ	<p>データ・タイプ: ビット・データの場合 CHAR(10)。 NULL 可能: いいえ。</p> <p>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。</p>
CURR_COMMIT_TIME	<p>データ・タイプ: TIMESTAMP。 NULL 可能: いいえ。</p> <p>この表が更新された時点の現行タイム・スタンプ。</p>
CAPTURE_FIRST_SEQ	<p>データ・タイプ: ビット・データの場合 CHAR(10)。 NULL 可能: いいえ。</p> <p>キャプチャー・プログラムがコールド・スタート後に開始する時点のジャーナル・レコード番号。</p>

IBMSNAP_RESTART

表 84. OS/400 の再始動表の列 (続き)

列名	説明
UID	データ・タイプ: INTEGER。 NULL 可能: いいえ。 UOW 表の IBMSNAP_UOWID 列の内容で接頭部として使用されるユニーク番号。
SEQNBR	データ・タイプ: BIGINT。 NULL 可能: いいえ。 キャプチャー・プログラムが処理した最後のジャーナル項目のシーケンス番号。
JRN_LIB	データ・タイプ: CHAR(10)。 NULL 可能: いいえ。 キャプチャー・プログラムが処理しているジャーナルのライブラリー名。
JRN_NAME	データ・タイプ: CHAR(10)。 NULL 可能: いいえ。 キャプチャー・プログラムが処理しているジャーナルの名前。
STATUS	データ・タイプ: CHAR(1)。 NULL 可能: はい。 キャプチャー・プログラムが特定のジャーナル・ジョブを処理しているかどうかを示すフラグ。 Y キャプチャー・プログラムはジャーナル・ジョブを処理しています。 N キャプチャー・プログラムはジャーナル・ジョブを処理していません。

schema.IBMSNAP_SEQTABLE (Informix)

サーバー: キャプチャー・コントロール・サーバー

デフォルト・スキーマ: ASN

ユニーク索引: SEQ

重要: SQL を使用してこの表を更新するときには、注意してください。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。

順序付け表は、Informix 表のログ・シーケンス番号と同等のものとして DB2 レプリケーション使用する一連のユニーク番号を保持します。これらのユニーク ID は、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・モニターが最後のサイクル時にどこまで処理を進めたかを連絡し合えるように、登録 (IBMSNAP_REGISTER) 表で同期点値の代わりに使用されます。

表 85 では、順序付け表の列の要旨を示します。

表 85. 順序付け表の列

列名	説明
SEQ	データ・タイプ: INTEGER。 NULL 可能: いいえ。 Informix 表のログまたはジャーナル ID (同期点) として使用されるユニークな番号。

schema.IBMSNAP_SIGNAL

サーバー: キャプチャー・コントロール・サーバー

デフォルト・スキーマ: ASN

索引: SIGNAL_TIME

この表の情報は、SQL を使って更新できます。

シグナル表には、キャプチャー・プログラムに特定のアクションを実行するように促すシグナルが保管されます。シグナルは、ユーザーまたはアプライ・プログラムから入力されます。

シグナル表は DATA CAPTURE CHANGES 属性を指定して作成されます。つまり、この表に対して実行されたすべての挿入、更新、および削除操作は、DB2 リカバリー・ログから読み取られたログ・レコードのように、キャプチャー・プログラムから見るすることができます。キャプチャー・プログラムは、シグナル表の更新および削除ログ・レコードはすべて無視しますが、シグナル挿入の、有効に作成およびコミットされたログ・レコードはすべて、注意が必要な「シグナル」として認識します。シグナル挿入によるログ・レコードに対してキャプチャー・プログラムが実行するアクションは、その挿入に関してシグナル表がどのように指定されているかによって異なります。シグナル表の中の値は、取るべきアクションをキャプチャー・プログラムに指示します。

この表の中で、コンプライトを表す、SIGNAL_STATE 値が C のレコード、または保存限度整理の対象となるタイム・スタンプを持つレコードは、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。

表 86 では、シグナル表の列の要旨を示します。

表 86. シグナル表の列

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。 デフォルト: 現行タイム・スタンプ。</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。キャプチャー・プログラムはこのユニーク値を使用して、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します。タイム・スタンプ列は NOT NULL WITH DEFAULT として作成されるため、キャプチャー・シグナルは一般的に、DB2 で現行タイム・スタンプが SIGNAL_TIME 値として提供されるのと同じ方法で挿入できます。</p>

IBMSNAP_SIGNAL

表 86. シグナル表の列 (続き)

列名	説明
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 ユーザー、アプライ・プログラム、またはよく知られたシステム・コマンドまたはシグナルである、別のアプリケーションから通知されたシグナル。使用可能なシグナルのサブタイプのリストについては、この表の SIGNAL_SUBTYPE 列を参照してください。</p> <p>USER ユーザーから通知されたシグナル。キャプチャー・プログラムは、SIGNAL_LSN 列の値を、シグナルが挿入されたときのログの LSN で更新し、SIGNAL_STATE 列の値を、P (ペンディング) から R (受信) に更新します。</p>
SIGNAL_SUBTYPE	<p>データ・タイプ: VARCHAR(30)。 NULL 可能: はい。</p> <p>システム・コマンドからのシグナルが発生したときに (SIGNAL_TYPE = CMD)、キャプチャー・プログラムが実行するアクション。</p> <p>CAPSTART キャプチャー・プログラムは、SIGNAL_INPUT_IN 列の中の MAP_ID (IBMSNAP_PRUNCNTL 表から) で識別される、特定のサブスクリプション・セット・メンバーの登録済みソースでの変更のキャプチャーを開始します。たとえば、アプライ・プログラムはセット内のすべてのターゲット表に対してフル・リフレッシュを実行する前にこのシグナルを発行して、このセットで変更キャプチャー・レプリケーションを行う準備ができていることをキャプチャー・プログラムに知らせます。アプライ・プログラムがこのシグナルを通知します。</p> <p>STOP キャプチャー・プログラムは変更のキャプチャーを停止して終了します。このコマンドは、ユーザーからのみ発行できます。アプライ・プログラムからは発行できません。</p> <p>CAPSTOP キャプチャー・プログラムは、SIGNAL_INPUT_IN 列の中の <i>source_owner.source_table</i> で識別される特定の登録済みソースの変更のキャプチャーを停止します。このコマンドは、ユーザーからのみ発行できます。アプライ・プログラムからは発行できません。</p> <p>UPDANY アプライ・プログラム (SIGNAL_INPUT_IN 列の中のアプライ修飾子で識別される) は、Update-anywhere 構成で、2 つのキャプチャー・プログラムを使用していることを、キャプチャー・プログラムに知らせます。アプライ・プログラムがこのシグナルを通知します。</p> <p>シグナル・タイプが USER の場合、シグナル・サブタイプは使用されないか、キャプチャー・プログラムから認識されないため、これは必要フィールドではありません。どのような値に設定してもかまいません。</p>

表 86. シグナル表の列 (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します。 SIGNAL_TYPE = CMD の場合は、この値の意味は、このシグナルの SIGNAL_SUBTYPE によって異なります。</p> <p>CMD + CAPSTART マッピング ID。DB2 以外のリレーショナル・ソースは、キャプチャー・プログラムではなく、キャプチャー・トリガーによって処理されるため、シーケンス内の次の値で整理制御 (IBMSNAP_PRUNCNTL) 表を更新する、IBMSNAP_SIGNAL 表の更新後に起動される、SIGNAL_TRIGGER というトリガーがあります。</p> <p>CMD + UPDANY Update-anywhere 構成でアプライ・プログラムを識別するためのアプライ修飾子。</p> <p>CMD + CAPSTOP キャプチャー・プログラムによる変更キャプチャーを停止する必要のある、ソース所有者およびソース表の名前。(source_owner.source_table)</p>
SIGNAL_STATE	<p>データ・タイプ: CHAR(1)。 NULL 可能: いいえ。</p> <p>シグナルの状況を示すフラグ。</p> <p>P シグナルはペンディング。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ません。ユーザーがシグナルを通知するときには、SIGNAL_STATE を P に設定してください。</p> <p>R キャプチャー・プログラムはシグナルを受け取りました。キャプチャー・プログラムは、SIGNAL_TYPE = USER であるか、SIGNAL_TYPE = CMD および SIGNAL_SUBTYPE = STOP であるシグナルを受け取ると、SIGNAL_STATE を R に設定します (コンプリートを示す C に変更するのではなく)。</p> <p>C キャプチャー・プログラムはシグナルの処理を完了しました。キャプチャー・プログラムは、SIGNAL_TYPE = CMD の場合、STOP を除くすべての SIGNAL_SUBTYPE について、この値を C に設定します。</p>
SIGNAL_LSN	<p>データ・タイプ: ビット・データの場合 CHAR(10)。 NULL 可能: はい。</p> <p>コミット・レコードのログ・シーケンス番号。この値は、キャプチャー・プログラムからのみ設定されます。</p>

iSeries オペレーティング・システムでは、シグナル表は、ソース表で使用される各ジャーナルに関連しています。これらの表はジャーナル・シグナル表と呼ばれ、グローバル・シグナル表 schema.IBMSNAP_SIGNAL と同じ構造です。ジャーナル・シグナル表の名前は *schema.IBMSNAP_SIGNAL_xxxx_yyyy* (xxxx はジャーナル・ライブラリーで、yyyy はジャーナル名) です。この表は自動的に作成されて、ソース・サーバー上のソース・ジャーナルに記録されます。リモート・ジャーナリング用のジャーナル・シグナル表の作成方法については、238 ページの『リモート・ジャーナリング用にジャーナル・シグナル表を作成する』を参照してください。

schema.IBMSNAP_UOW

サーバー: キャプチャー・コントロール・サーバー

デフォルト・スキーマ: ASN

索引: IBMSNAP_COMMITSEQ、IBMSNAP_LOGMARKER

重要: SQL を使用してこの表を更新するときには、注意してください。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。

作業単位 (UOW) 表は、ソース表にコミットされたトランザクションに関する追加情報を提供します。ユーザー・コピー以外のすべてのターゲット表タイプの場合、アプライ・プログラムはターゲット表に変更を適用するときに、IBMSNAP_COMMITSEQ 値を突き合わせることで、UOW 表と変更データ (CD) 表を結合します。キャプチャー・プログラムをコールド・スタートすると、この表のすべての項目は削除されます。

OS/400 の場合: Capture for iSeries は、レプリケーション・ソースのサブセットのデータのキャプチャーを開始することがあるため、部分的コールド・スタートを行った場合は、UOW 表の中のすべての行が削除されることはありません。

キャプチャー・プログラムでは、キャプチャー・スキーマごとに 1 つの UOW 表が必要になります。キャプチャー・プログラムは、レプリケーション・ソースでコミットされたログまたはジャーナル・レコードごとに、この表に新しい行を 1 つ挿入します。

OS/400 の場合: 一部のユーザー・プログラムはコミットメント・コントロールを使用しません。このような場合、Capture for iSeries は、CD 表に複数の行が書き込まれた後で、任意に新しい UOW 行を挿入します。このように見せかけのコミットメント境界を設けることにより、UOW 表のサイズを削減できます。

キャプチャー・プログラムは、アプライ・プログラムが整理セット (IBMSNAP_PRUNE_SET) 表に挿入した情報に基づいて、UOW 表の整理も行います。

OS/400 の場合: UOW 表は、整理セット (IBMSNAP_PRUNE_SET) 表の情報によってではなく、保存限度に従って整理されます。

表 87 では、UOW 表の列の要旨を示します。

表 87. UOW 表の列

列名	説明
IBMSNAP_UOWID	データ・タイプ: ビット・データの場合 CHAR(10)。 NULL 可能: いいえ。 この作業単位についての、ログ・レコード・ヘッダーからの作業単位 ID。この列を、非コンプリート CCD ターゲット表の一部とすることができます。

表 87. UOW 表の列 (続き)

列名	説明
IBMSNAP_COMMITSEQ	<p>データ・タイプ: ビット・データの場合 CHAR(10)。 NULL 可能: いいえ。</p> <p>キャプチャーされたコミット・ステートメントのログ・レコード・シーケンス番号。ユーザー・コピー以外のすべてのターゲット表タイプの場合、アプライ・プログラムはターゲット表に変更を適用するときに、この列の値に基づいて、UOW 表と CD 表を結合します。</p>
IBMSNAP_LOGMARKER	<p>データ・タイプ: TIMESTAMP。 NULL 可能: いいえ。</p> <p>データがコミットされたときの時刻 (キャプチャー・コントロール・サーバーの)。</p>
IBMSNAP_AUTHTKN	<p>データ・タイプ: VARCHAR(30)。 NULL 可能: いいえ。</p> <p>トランザクションに関連付けられた許可トークン。この ID は、データベースの監査に役立ちます。DB2 Universal Database for z/OS の場合、この列は相関 ID です。DB2 Universal Database for iSeries の場合、この列は、トランザクションを発生させたジョブのジョブ名です。この列は自動的に他の表にコピーされません。ユーザー・データ列として選択してコピーする必要があります。この列を、非コンプライート CCD ターゲット表の一部とすることができます。</p>
IBMSNAP_AUTHID	<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>トランザクションに関連付けられた許可 ID。これはデータベースの監査に役立ちます。DB2 Universal Database for z/OS の場合、この列は 1 次許可 ID です。DB2 Universal Database for iSeries の場合、この列は、トランザクションを発生させたアプリケーションを実行しているユーザー・プロファイル ID の名前になります。この列にはブランクが埋め込まれた 10 文字の ID が入ります。この列は自動的に他の表にコピーされません。ユーザー・データ列として選択してコピーする必要があります。この列を、非コンプライート CCD ターゲット表の一部とすることができます。</p>

IBMSNAP_UOW

表 87. UOW 表の列 (続き)

列名	説明
IBMSNAP_REJ_CODE	<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: 0。</p> <p>リジェクトまたはロールバックされた行があるかどうかを示すフラグ。この値は、レプリケーション・ソースの定義時に競合検出が標準または拡張と指定された場合に Update-anywhere レプリケーションでのみ使用されます。この列を、非コンプリート CCD ターゲット表の一部とすることができます。</p> <p>0 トランザクションで競合の発生は報告されていません。</p> <p>1 マスターとレプリカと同じ行が更新されたため、競合が発生しています。レプリカ側のトランザクションはリジェクトされ、ロールバックされます。</p> <p>2 このトランザクションは、以前にリジェクトされたトランザクションに付属するため、リジェクトされてロールバックされました。前のトランザクションは、マスターとレプリカの中で同じ行が更新されたためにリジェクトされたものであり、レプリカ側のトランザクションが、リジェクトされてロールバックされています。</p> <p>3 参照保全制約違反が少なくとも 1 つ含まれているため、このトランザクションはリジェクトされ、ロールバックされました。このトランザクションはソース表で定義された参照制約に違反しているため、アプライ・プログラムはこのサブスクリプション・セットに失敗というマークを付けます。参照保全定義が訂正されるまで、更新はコピーできません。</p> <p>4 このトランザクションは、以前にリジェクトされたトランザクションに付属するため、リジェクトされてロールバックされました。前のトランザクションは、参照保全制約違反が少なくとも 1 つ含まれているためにリジェクトされました。</p>
IBMSNAP_APPLY_QUAL	<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: 現行ユーザー名。</p> <p>変更を適用したアプライ・プログラムを識別するためのアプライ修飾子。この列を、非コンプリート CCD ターゲット表の一部とすることができます。</p>

アプライ・コントロール・サーバー上の表、およびその列に関する記述

このセクションでは、アプライ・コントロール・サーバー上に保管される各表について、かなり詳しく説明します。また、各表の列のリストやその簡単な説明もあります。コントロール表はアルファベット順にリストされ、列は各表内に現れる順に左から右へリストされています。

ASN.IBMSNAP_APPENQ

サーバー: アプライ・コントロール・サーバー

索引: APPLY_QUAL

重要: SQL を使用してこの表を更新するときには、注意してください。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。

アプライ・エンキュー表は、1 つのアプライ修飾子に対して 1 つのアプライ・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ます。アプライ・プログラムは、アプライ・プログラムがシャットダウンするまで、この表の中の行を排他的にロックします。この表は OS/400 では使用されません。

表 88 では、アプライ・エンキュー表の列の要旨を示します。

表 88. アプライ・エンキュー表の列

列名	説明
APPLY_QUAL	<p>データ・タイプ: CHAR(18)。NULL 可能: はい。</p> <p>同じアプライ・プログラムにより処理されるサブスクリプション・セットのグループを固有に指定します。この値は大文字小文字が区別されます。サブスクリプション・セットの定義時には、この値を指定しなければなりません。</p>

ASN.IBMSNAP_APPLY_JOB (OS/400)

サーバー: アプライ・コントロール・サーバー

索引: なし

重要: SQL を使用してこの表を更新するときには、注意してください。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。

OS/400 特有のものであるアプライ・ジョブ表は、アプライ・コントロール・サーバー上で実行中のアプライ・プログラムのすべてのインスタンスの APPLY_QUAL の値がユニークなものであることを保証するために使用されます。アプライ・プログラムの 1 つのインスタンスが開始されるたびに、この表に行が追加されます。アプライ・プログラムの新しいインスタンスを始動するとき、その APPLY_QUAL 値がすでに存在していると、始動コマンドは失敗します。

表 89 では、アプライ・ジョブ表の列の要旨を示します。

表 89. アプライ・ジョブ表の列

列名	説明
APPLY_QUAL	<p>データ・タイプ: CHAR(18)。NULL 可能: いいえ。</p> <p>サブスクリプション・セットのグループのユニーク ID。この値は、サブスクリプション・セットの定義時にユーザーによって指定されます。アプライ・プログラムの各インスタンスは、APPLY_QUAL 値によって開始されます。この値は、Update-anywhere レプリケーションで、アプライ・プログラムによる変更の循環レプリケーションを避けるために使用されます。</p>
CONTROL_SERVER	<p>データ・タイプ: CHAR(18)。NULL 可能: いいえ。</p> <p>アプライ・コントロール表およびビューが定義されているデータベースの名前。</p>

IBMSNAP_APPLY_JOB (OS/400)

表 89. アプライ・ジョブ表の列 (続き)

列名	説明
JOB_NAME	<p>データ・タイプ: CHAR(10)。NULL 可能: いいえ。</p> <p>このトレース項目を書き込んだジョブの完全修飾名。</p> <p>位置 1-10 APPLY_QUAL</p> <p>位置 11-20 アプライ・プログラムを始動したユーザーの ID</p> <p>位置 21-26 ジョブ番号</p>
USER_NAME	<p>データ・タイプ: CHAR(10)。NULL 可能: いいえ。</p> <p>アプライ・プログラムの新しいインスタンスを始動したユーザーの名前。</p>
JOB_NUMBER	<p>データ・タイプ: CHAR(6)。NULL 可能: いいえ。</p> <p>特定のジャーナルに対する現行ジョブのジョブ番号。ジャーナルがアクティブでない場合は、この列には、最後に処理されたジョブのジョブ番号が入っています。</p>

ASN.IBMSNAP_APPPARMS

サーバー: アプライ・コントロール・サーバー

索引: APPLY_QUAL

この表の情報は、SQL を使って更新できます。

アプライ・パラメーター表は、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。これらのパラメーターは、サブスクリプション定義とアプライ・プログラム・コントロール表が存在する、アプライ・コントロール・サーバーの名前などの値を設定する場合に定義します。ユーザーがこの表のパラメーターを変更しても、アプライ・プログラムは始動時にしか変更を読み取りません。

表 90 では、アプライ・パラメーター表の列の要旨を示します。

表 90. アプライ・パラメーター表の列

列名	説明
APPLY_QUAL	<p>データ・タイプ: CHAR(18)。NULL 可能: いいえ。</p> <p>アプライ修飾子は、パラメーターを、それらパラメーターの適用先のアプライ・プログラムに適合させます。</p>
APPLY_PATH	<p>データ・タイプ: VARCHAR(1040)。NULL 可能: はい。</p> <p>アプライ・プログラムが使用する作業ファイルのロケーション。デフォルトは、プログラムが開始されたディレクトリーです。</p>

表 90. アプライ・パラメーター表の列 (続き)

列名	説明
COPYONCE	<p>データ・タイプ: CHAR(1)。 NULL 可能: はい、デフォルトあり。 デフォルト: N。</p> <p>アプライ・プログラムが呼び出された時点で適格と見なされたサブスクリプション・セットごとに、アプライ・プログラムがコピー・サイクルを 1 回実行するかどうかを示すフラグ。</p> <p>Y アプライ・プログラムは、適格なサブスクリプション・セットごとにコピー・サイクルを 1 回実行します。</p> <p>N アプライ・プログラムは、適格なサブスクリプション・セットごとにコピー・サイクルを 1 回実行しません。</p>
DELAY	<p>データ・タイプ: INT。 NULL 可能: はい、デフォルトあり。 デフォルト: 6。</p> <p>デフォルトでは、連続してレプリケーションを行う場合 (サブスクリプション・セットで sleep=0 分を使用する場合)、アプライ・プログラムはサブスクリプション・セットが正常に処理されてから、6 秒間待った後にサブスクリプション・セットを再試行します。複製すべきデータベース・アクティビティーがない場合は、ゼロ以外の値を使用して CPU サイクルを節約してください。待ち時間を少なくするには遅延値を小さくします。</p> <p>注: delay パラメーターは、copyonce が指定される場合は無視されます。</p>
ERRWAIT	<p>データ・タイプ: INT。 NULL 可能: はい、デフォルトあり。 デフォルト: 300。</p> <p>アプライ・プログラムがエラー状態になった後、何秒待ってから再試行するかを示す秒数 (1 ~ 300)。 copyonce が指定されている場合、このパラメーターは無視されます。</p>
INAMSG	<p>データ・タイプ: CHAR(1)。 NULL 可能: はい、デフォルトあり。 デフォルト: Y。</p> <p>アプライ・プログラムを非アクティブにしたとき、アプライ・プログラムからメッセージを出すかどうかを示すフラグ。</p> <p>Y アプライ・プログラムは非アクティブ時にメッセージを出します。</p> <p>N アプライ・プログラムは非アクティブ時にメッセージを出しません。</p>
LOADXIT	<p>データ・タイプ: CHAR(1)。 NULL 可能: はい、デフォルトあり。 デフォルト: N。</p> <p>アプライ・プログラムが、エクスポート・ユーティリティーやロード・ユーティリティーを使用してターゲット表をリフレッシュする、IBM 提供の出口ルーチン (ASNLOAD) を呼び出すかどうかを示すフラグ。</p> <p>Y アプライ・プログラムは ASNLOAD を呼び出します。</p> <p>N アプライ・プログラムは ASNLOAD を呼び出しません。</p>

IBMSNAP_APPPARMS

表 90. アプライ・パラメーター表の列 (続き)

列名	説明
LOGREUSE	<p>データ・タイプ: CHAR(1)。 NULL 可能: はい、デフォルトあり。 デフォルト: N。</p> <p>アプライ・プログラムがアプライ・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 アプライ・プログラムは、まずログ・ファイルを削除して、アプライ・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 アプライ・プログラムは新しい情報をアプライ・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: はい、デフォルトあり。 デフォルト: N。</p> <p>アプライ・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 アプライ・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 アプライ・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>
NOTIFY	<p>データ・タイプ: CHAR(1)。 NULL 可能: はい、デフォルトあり。 デフォルト: N。</p> <p>アプライ・プログラムが、サブスクリプション・セットをコピーした後に、ユーザーにコントロールを戻す出口ルーチン (ASNDONE) を呼び出すかどうかを示すフラグ。</p> <p>Y アプライ・プログラムは ASNDONE を呼び出します。</p> <p>N アプライ・プログラムは ASNDONE を呼び出しません。</p>
OPT4ONE	<p>データ・タイプ: CHAR(1)。 NULL 可能: はい、デフォルトあり。 デフォルト: N。</p> <p>アプライ・プログラムに定義されているサブスクリプション・セットが 1 つだけの場合、アプライ・プログラムのパフォーマンスを最適化するかどうかを示すフラグ。</p> <p>Y サブスクリプション・セットが 1 つの場合、アプライ・プログラムのパフォーマンスを最適化します。</p> <p>N サブスクリプション・セットが 1 つの場合、アプライ・プログラムのパフォーマンスを最適化しません。</p> <p>copyonce が指定されている場合、このパラメーターは無視されます。</p>

表 90. アプライ・パラメーター表の列 (続き)

列名	説明
SLEEP	<p>データ・タイプ: CHAR(1)。NULL 可能: はい、デフォルトあり。デフォルト: Y。</p> <p>処理の対象として適格となる新しいサブスクリプションがない場合に、アプライ・プログラムがどうするかを示すフラグ。</p> <p>Y アプライ・プログラムはスリープ状態に入ります。</p> <p>N アプライ・プログラムは停止します。</p> <p>copyonce が指定されている場合、このパラメーターは無視されます。</p>
SQLERRCONTINUE	<p>データ・タイプ: CHAR(1)。NULL 可能: はい、デフォルトあり。デフォルト: N。</p> <p>アプライ・プログラムが、SQLSTATE ファイルのエラーをチェックした後に処理を続行するかどうかを示すフラグ。</p> <p>Y アプライ・プログラムは処理中に SQLSTATE ファイルに SQL エラーがないかチェックします。エラーが見つかった場合、アプライは処理を停止します。</p> <p>N アプライ・プログラムは SQLSTATE ファイルをチェックせず、処理を続行します。</p>
SPILLFILE	<p>データ・タイプ: VARCHAR(10)。NULL 可能: はい、デフォルトあり。</p> <p>フェッチした応答セットをどこに保管するかを示すフラグ。</p> <p>UNIX および Windows の場合、有効な値は以下のとおりです。</p> <p>disk (デフォルト) ディスク・ファイル。</p> <p>z/OS の場合、有効な値は以下のとおりです。</p> <p>mem (デフォルト) メモリー・ファイル。</p> <p>disk ディスク・ファイル。</p>
TERM	<p>データ・タイプ: CHAR(1)。NULL 可能: はい、デフォルトあり。デフォルト: Y。</p> <p>DB2 が終了したときにアプライ・プログラムが終了するかどうかを示すフラグ。</p> <p>Y DB2 が終了するとアプライ・プログラムは終了します。</p> <p>N アプライ・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> <p>copyonce が指定されている場合、このパラメーターは無視されます。</p>
TRLREUSE	<p>データ・タイプ: CHAR(1)。NULL 可能: はい、デフォルトあり。デフォルト: N。</p> <p>アプライ・プログラムが、エクスポート・ユーティリティーやロード・ユーティリティーを使用してターゲット表をリフレッシュする、IBM 提供の出力ルーチン (ASNLOAD) を呼び出すかどうかを示すフラグ。</p> <p>Y アプライ・プログラムは ASNLOAD を呼び出します。</p> <p>y アプライ・プログラムは ASNLOAD を呼び出しません。</p>

ASN.IBMSNAP_APPLYTRACE

サーバー: アプライ・コントロール・サーバー

索引: APPLY_QUAL, TRACE_TIME

アプライ・トレース表は、アプライ・プログラムからの重要なメッセージを保持します。アプライ・プログラムはこの表の整理を自動的に行うことはありませんが、1つのサブスクリプション・セットの後で実行される SQL ステートメントを追加することにより、簡単に整理を自動化できます。

表 91 では、アプライ・トレース表の列の要旨を示します。

表 91. アプライ・トレース表の列

列名	説明
APPLY_QUAL	データ・タイプ: CHAR(18)。NULL 可能: いいえ。 メッセージを挿入したアプライ・プログラムを一意的に識別します。
TRACE_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アプライ・コントロール・サーバーの時刻。
OPERATION	データ・タイプ: CHAR(8)。NULL 可能: いいえ。 アプライ・プログラムの操作のタイプ、たとえば、初期化、アプライ、またはエラー条件。
DESCRIPTION	データ・タイプ: VARCHAR(1024)。NULL 可能: いいえ。 メッセージ ID とメッセージ・テキスト。メッセージ ID は DESCRIPTION 列の最初の 7 文字です。メッセージ・テキストは、DESCRIPTION 列の位置 9 から始まります。

ASN.IBMSNAP_APPLYTRAIL

サーバー: アプライ・コントロール・サーバー

索引: LASTRUN, APPLY_QUAL

アプライ・トレール表には、アプライ・プログラムによって実行される、すべてのサブスクリプション・セット・サイクルの監査証跡情報が含まれます。アプライ・トレール表には、サブスクリプションに対して実行された更新の履歴が入ります。この表は、診断およびパフォーマンス統計のリポジトリです。アプライ・プログラムで問題が生じたときにアプライ・トレール表を参照するのは非常に効果的です。アプライ・プログラムはこの表の整理を自動的に行うことはありませんが、1つのサブスクリプション・セットの後で実行される SQL ステートメントを追加することにより、簡単に整理を自動化できます。

555 ページの表 92 では、アプライ・トレール表の列の要旨を示します。

表 92. アプライ・トレール表の列

列名	説明
APPLY_QUAL	<p>データ・タイプ: CHAR(18)。 NULL 可能: いいえ。</p> <p>サブスクリプション・セットを処理したアプライ・プログラムを一意的に識別します。</p>
SET_NAME	<p>データ・タイプ: CHAR(18)。 NULL 可能: いいえ。</p> <p>アプライ・プログラムが処理したサブスクリプション・セットの名前。</p>
SET_TYPE	<p>データ・タイプ: CHAR(1)。 NULL 可能: いいえ。</p> <p>最後のアプライ・サイクルの後でサブスクリプション・セット (IBMSNAP_SUBS_SET) 表の SET_TYPE 列に表示された値。各値の意味については、568 ページの『ASN.IBMSNAP_SUBS_SET』を参照してください。</p>
WHOS_ON_FIRST	<p>データ・タイプ: CHAR(1)。 NULL 可能: いいえ。</p> <p>Update-anywhere レプリケーション・シナリオでは、処理順序をコントロールするために以下の値を使用します。</p> <p>F (first の略) ソース表がレプリカであり、ターゲット表がマスターです。レプリカ表とマスター表の間で更新の矛盾が生じた場合、レプリカの側の矛盾するトランザクションはリジェクトされます。F は読み取り専用のサブスクリプションでは使用されません。Update-anywhere で使用されるものです。</p> <p>S (second の略) ソース表はマスター表またはその他のソースであり、ターゲット表はレプリカまたはその他のコピーです。マスター表とレプリカ表の間で更新の矛盾が生じた場合、レプリカの側の矛盾するトランザクションはリジェクトされます。S は、すべての読み取り専用サブスクリプションについて使用されます。</p>
ASNLOAD	<p>データ・タイプ: CHAR(1)。 NULL 可能: はい。</p> <p>アプライ・プログラムを始動するために使用された値。</p> <p>Y パラメーター loadxit=y を指定してアプライ・プログラムを始動したため、サブスクリプション・セットのフル・リフレッシュを実行するために ASNLOAD ユーザー出口ルーチンが呼び出されることを意味します。</p> <p>N フル・リフレッシュが必要ないか、アプライ・プログラムの始動時に loadxit パラメーターが指定されていなかったため、ASNLOAD 出口ルーチンが呼び出されないことを意味します。</p> <p>NULL ASNLOAD 出口ルーチンを呼び出すかどうかをアプライ・プログラムが判断する前に、アプライ・プログラム・エラーが生じたことを示します。</p>

IBMSNAP_APPLYTRAIL

表 92. アプライ・トレール表の列 (続き)

列名	説明
FULL_REFRESH	<p>データ・タイプ: CHAR(1)。 NULL 可能: はい。</p> <p>フル・リフレッシュが発生したかどうかを示すフラグ。</p> <p>Y サブスクリプション・セットに対してフル・リフレッシュが実行されたことを示します。</p> <p>N サブスクリプション・セットに対してフル・リフレッシュが実行されなかったことを示します。</p> <p>NULL フル・リフレッシュが必要かどうかをアプライ・プログラムが判断する前に、エラーが生じたことを示します。</p>
EFFECTIVE_MEMBERS	<p>データ・タイプ: INT。 NULL 可能: はい。</p> <p>フル・リフレッシュか、挿入、更新、および削除のレプリケーションのどちらかにより、1 回のアプライ・サイクルで変更されたサブスクリプション・セット・メンバーの数。値の範囲は、0 以上、定義済みのサブスクリプション・セット・メンバーの数以下です。</p>
SET_INSERTED	<p>データ・タイプ: INT。 NULL 可能: いいえ。</p> <p>サブスクリプション・サイクルにおいてサブスクリプション・セット・メンバーに挿入された行の合計数。</p>
SET_DELETED	<p>データ・タイプ: INT。 NULL 可能: いいえ。</p> <p>サブスクリプション・サイクルにおいてサブスクリプション・セット・メンバーから削除された行の合計数。</p>
SET_UPDATED	<p>データ・タイプ: INT。 NULL 可能: いいえ。</p> <p>サブスクリプション・サイクルにおいてサブスクリプション・セット・メンバーで更新された行の合計数。</p>
SET_REWORKED	<p>データ・タイプ: INT。 NULL 可能: いいえ。</p> <p>最後のサイクルでアプライ・プログラムが再処理した合計行数。アプライ・プログラムは、以下の条件下で変更を再試行します。</p> <ul style="list-style-type: none"> 行がターゲット表にすでに存在しているため挿入が失敗した場合、アプライ・プログラムは、挿入操作を既存行の更新操作に変換します。 行がターゲット表に存在していないため更新が失敗した場合、アプライ・プログラムは、更新操作を挿入操作に変換します。
SET_REJECTED_TRXS	<p>データ・タイプ: INT。 NULL 可能: いいえ。</p> <p>Update-anywhere 競合のためにリジェクトされたトランザクションの合計数。この列は、競合検出が「標準」または「詳細」と定義されている Update-anywhere サブスクリプション・セットに対してのみ使用されます。</p>

表 92. アプライ・トレール表の列 (続き)

列名	説明
STATUS	<p>データ・タイプ: SMALLINT。NULL 可能: いいえ。</p> <p>特定のサイクル後のアプライ・プログラムの作業状況を表す値。</p> <p>-1 レプリケーションは失敗しました。アプライ・プログラムは適用済みの行のセット全体をバックアウトし、データはコミットされません。始動パラメーターが <code>SQLERRCONTINUE = Y</code> の場合、最後のサイクル中にアプライ・プログラムに戻される <code>SQLSTATE</code> は、<code>SQLERRCONTINUE (apply_qualifier.SQS)</code> の入力ファイルでユーザーが指定した許容エラーの 1 つではありません。</p> <p>0 アプライ・プログラムはサブスクリプション・セットを正常に処理しました。始動パラメーターが <code>SQLERRCONTINUE = Y</code> の場合、アプライ・プログラムは、ユーザーから <code>SQLERRCONTINUE</code> 始動パラメーターで (<code>apply_qualifier.SQS</code> で) 指定された SQL エラーを検出しておらず、行をリジェクトしていません。</p> <p>2 アプライ・プログラムはサブスクリプション・セットを複数のサイクルで処理しています。アプライ・プログラムは、<code>MAX_SYNCH_MINUTES</code> コントロール列に従って分割された 1 つの論理サブスクリプションを正常に処理しました。</p> <p>16 アプライ・プログラムはサブスクリプション・セットを正常に処理し、0 という状況に戻しました。しかしアプライ・プログラムは、ユーザーが <code>SQLERRCONTINUE</code> 始動パラメーターで (<code>apply_qualifier.SQS</code> で) 指定したいくつかの SQL エラーを検出したため、いくつかの行をリジェクトしました。失敗した行の詳細は、<code>apply_qualifier.ERR</code> ファイルで確認してください。</p> <p>例: ユーザーは <code>SQLERRCONTINUE = Y</code> と設定し、SQL の許容される状態を 23502 (SQL コード -407) と指定します。23502 エラーが発生しますが、他のエラーは発生していません。アプライ・プログラムはサブスクリプション・セットの処理を終了し、状態を 16 に設定します。次の実行時に、23502 エラーが発生した後、07006 (SQL コード -301) が発生します。アプライ・プログラムは今回は、サブスクリプション・セットの処理を停止し、適用済みの行のセット全体をバックアウトし、状況を -1 に設定します (データはコミットされていないため)。</p> <p>18 アプライ・プログラムは複数のサイクルでサブスクリプション・セットを処理し、2 という状況に戻しています。これは、<code>MAX_SYNCH_MINUTES</code> コントロール列に従って分割された 1 つの論理サブスクリプションが正常に処理されたことを意味します。しかし、ユーザーが <code>SQLERRCONTINUE</code> 始動パラメーターで (<code>apply_qualifier.SQS</code> で) 指定した SQL エラーのいくつかを検出されたため、いくつかの行がリジェクトされています。失敗した行の詳細は、<code>apply_qualifier.ERR</code> ファイルで確認してください。</p>
LASTRUN	<p>データ・タイプ: TIMESTAMP。NULL 可能: いいえ。</p> <p>最後のサブスクリプションが開始された概算の時刻。アプライ・プログラムは、サブスクリプション・セットを処理するたびに <code>LASTRUN</code> 値を設定します。これは、アプライ・プログラムがサブスクリプション・セットの処理を開始する、アプライ・コントロール・サーバーにおけるおおよその時刻です。</p>

IBMSNAP_APPLYTRAIL

表 92. アプライ・トレール表の列 (続き)

列名	説明
LASTSUCCESS	<p>データ・タイプ: TIMESTAMP。 NULL 可能: はい。</p> <p>サブスクリプション・セットが最後に正常に処理されたときの、処理開始時点のアプライ・コントロール・サーバーのタイム・スタンプ。</p>
SYNCHPOINT	<p>データ・タイプ: ビット・データの場合 CHAR(10)。 NULL 可能: はい。</p> <p>アプライ・プログラムはこの列を使用して進行状況を記録します。サブスクリプション・セットのデータの処理が、この同期点の値まで終了していることを示します。</p>
SYNCHTIME	<p>データ・タイプ: TIMESTAMP。 NULL 可能: はい。</p> <p>アプライ・プログラムはこの列を使用して進行状況を記録します。サブスクリプション・セットのデータの処理が、このタイム・スタンプまで終了していることを示します。</p>
SOURCE_SERVER	<p>データ・タイプ: CHAR(18)。 NULL 可能: いいえ。</p> <p>ソース表およびビューが定義されている DB2 Universal Database データベース名。</p>
SOURCE_ALIAS	<p>データ・タイプ: CHAR(8)。 NULL 可能: はい。</p> <p>SOURCE_SERVER 列で指定されているソース・サーバーの DB2 Universal Database 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・サブシステムの場合 VARCHAR(128)。 NULL 可能: はい。</p> <p>アプライ・プログラムが処理中であったソース表またはビューの上位修飾子。この値は、アプライ・サイクルが失敗したときにのみ設定されます。</p>
SOURCE_TABLE	<p>データ・タイプ: VARCHAR(128)、 DB2 UDB for z/OS バージョン 8 互換モード・サブシステムかそれ以前の場合 VARCHAR(18)。 NULL 可能: はい。</p> <p>アプライ・プログラムが処理中であったソース表またはビューの名前。この値は、アプライ・サイクルが失敗したときにのみ設定されます。</p>
SOURCE_VIEW_QUAL	<p>データ・タイプ: SMALLINT。 NULL 可能: はい。</p> <p>アプライ・プログラムが処理中であったソース表またはビューのソース・ビュー修飾子の値。この値は、アプライ・サイクルが失敗したときにのみ設定されます。</p>
TARGET_SERVER	<p>データ・タイプ: CHAR(18)。 NULL 可能: いいえ。</p> <p>ターゲットの表またはビューが保管されているサーバーのデータベース名。</p>
TARGET_ALIAS	<p>データ・タイプ: CHAR(8)。 NULL 可能: はい。</p> <p>TARGET_SERVER 列で指定されているターゲット・サーバーの DB2 Universal Database 別名。</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・サブシステムの場合 VARCHAR(128)。 NULL 可能: いいえ。</p> <p>アプライ・プログラムが処理中であったターゲット表の上位修飾子。この値は、アプライ・サイクルが失敗したときにのみ設定されます。</p>

表 92. アプライ・トレール表の列 (続き)

列名	説明
TARGET_TABLE	<p>データ・タイプ: VARCHAR(128)、 DB2 UDB for z/OS バージョン 8 互換モード・サブシステムかそれ以前の場合 VARCHAR(18)。 NULL 可能: いいえ。</p> <p>アプライ・プログラムが処理中であったターゲット表の名前。この値は、アプライ・サイクルが失敗したときにのみ設定されます。</p>
CAPTURE_SCHEMA	<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>このサブスクリプション・セットのキャプチャー・サーバー表のスキーマ名。</p>
TGT_CAPTURE_SCHEMA	<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: はい。</p> <p>ターゲット表が別のサブスクリプション・セットのソースでもある場合 (multi-tier 構成中の外部の CCD 表、または Update-anywhere 構成中のレプリカ表など) は、この列には、表がソースとして機能するときに使用されるキャプチャー・スキーマが含まれます。</p>
FEDERATED_SRC_SRVR	<p>データ・タイプ: VARCHAR(18)。 NULL 可能: はい。</p> <p>DB2 以外のリレーショナル・ソースの場合にのみアプライされる、サブスクリプション・セットのソースである、フェデレーテッド・リモート・サーバーの名前。</p>
FEDERATED_TGT_SRVR	<p>データ・タイプ: VARCHAR(18)。 NULL 可能: はい。</p> <p>DB2 以外のリレーショナル・ターゲット・サーバーの場合にのみアプライされる、サブスクリプション・セットのターゲットである、フェデレーテッド・リモート・サーバーの名前。</p>
JRN_LIB	<p>データ・タイプ: CHAR(10)。 NULL 可能: はい。</p> <p>OS/400 キャプチャー・サーバーにのみアプライされるこの列は、ソース表が使用するジャーナルのライブラリー名です。</p>
JRN_NAME	<p>データ・タイプ: CHAR(10)。 NULL 可能: はい。</p> <p>OS/400 キャプチャー・サーバーにのみアプライされるこの列は、ソース表が使用するジャーナルの名前です。この列で、アスタリスクの後ろに 9 つのブランクが続くときには、ソース表が現在ジャーナルの中にあることを意味します。この場合は、このソース表のデータをキャプチャーすることはできません。</p>
COMMIT_COUNT	<p>データ・タイプ: SMALLINT。 NULL 可能: はい。</p> <p>サブスクリプション・セット (IBMSNAP_SUBS_SET) 表の中に記録される、最後のアプライ・サイクルからの COMMIT_COUNT の値。</p>
OPTION_FLAGS	<p>データ・タイプ: CHAR(4)。 NULL 可能: いいえ。</p> <p>DB2 レプリケーションの将来のオプション用に予約済み。この列には現在、デフォルト値の NNNN が入っています。</p>
EVENT_NAME	<p>データ・タイプ: CHAR(18)。 NULL 可能: はい。</p> <p>セットの処理を起動したイベントを表すために使用されるユニークな文字ストリング。</p>

IBMSNAP_APPLYTRAIL

表 92. アプライ・トレール表の列 (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>アプライ・プログラムがサブスクリプション・セットの処理を終了したときの、アプライ・コントロール・サーバーにおけるタイム・スタンプ。セットの処理に要した時間を知るには、LASTRUN を ENDTIME から減算します。</p>
SOURCE_CONN_TIME	<p>データ・タイプ: TIMESTAMP。 NULL 可能: はい。</p> <p>アプライ・プログラムが初めてフェッチ・ソース・データに接続したときの、キャプチャー・コントロール・サーバーにおけるタイム・スタンプ。</p>
SQLSTATE	<p>データ・タイプ: CHAR(5)。 NULL 可能: はい。</p> <p>失敗した実行の SQL 状態コード。それ以外の場合は、NULL になります。</p>
SQLCODE	<p>データ・タイプ: INT。 NULL 可能: はい。</p> <p>失敗した実行の SQL エラー・コード。それ以外の場合は、NULL になります。</p>
SQLERRP	<p>データ・タイプ: CHAR(8)。 NULL 可能: はい。</p> <p>実行の失敗原因となった SQL エラーが生じたサーバーのデータベース製品 ID。それ以外の場合は、NULL になります。</p>
SQLERRM	<p>データ・タイプ: VARCHAR(70)。 NULL 可能: はい。</p> <p>失敗した実行の SQL エラー情報。</p>
APPERRM	<p>データ・タイプ: VARCHAR(760)。 NULL 可能: はい。</p> <p>アプライ・プログラムの実行に失敗したときのエラー・メッセージ ID およびテキスト。</p>

ASN.IBMSNAP_SUBS_COLS

サーバー: アプライ・コントロール・サーバー

索引: APPLY_QUAL、SET_NAME、WHOS_ON_FIRST、TARGET_OWNER、TARGET_TABLE、TARGET_NAME

重要: SQL を使用してこの表を更新するときには、注意してください。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。

サブスクリプション列表には、サブスクリプション・セット内にコピーされるサブスクリプション・セット・メンバーの列に関する情報が含まれます。一対のソース表とターゲット表で、1 つまたは複数の列の情報が変更されると、この表で自動的に行が挿入または削除されます。

この表は、サブスクリプション・セット・メンバーの特定の列に関する情報が必要な場合に使用してください。

561 ページの表 93 では、サブスクリプション列表の列の要旨を示します。

表 93. サブスクリプション列表の列

列名	説明
APPLY_QUAL	<p>データ・タイプ: CHAR(18)。 NULL 可能: いいえ。</p> <p>このサブスクリプション・セット・メンバーを処理するアプライ・プログラムを一意的に識別します。</p>
SET_NAME	<p>データ・タイプ: CHAR(18)。 NULL 可能: いいえ。</p> <p>このメンバーが所属するサブスクリプション・セットの名前。</p>
WHOS_ON_FIRST	<p>データ・タイプ: CHAR(1)。 NULL 可能: いいえ。</p> <p>Update-anywhere レプリケーション・シナリオでは、処理順序をコントロールするために以下の値を使用します。</p> <p>F (first の略) ソース表がレプリカであり、ターゲット表がマスターです。レプリカ表とマスター表の間で更新の矛盾が生じた場合、レプリカの側の矛盾するトランザクションはリジェクトされます。F は読み取り専用のサブスクリプションでは使用されません。Update-anywhere で使用されるものです。</p> <p>S (second の略) ソース表はマスター表またはその他のソースであり、ターゲット表はレプリカまたはその他のコピーです。マスター表とレプリカ表の間で更新の矛盾が生じた場合、レプリカの側の矛盾するトランザクションはリジェクトされます。S は、すべての読み取り専用サブスクリプションについて使用されます。</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・サブシステムの場合 VARCHAR(128)。 NULL 可能: いいえ。</p> <p>ターゲット表またはビューの上位修飾子。</p>
TARGET_TABLE	<p>データ・タイプ: VARCHAR(128)、 DB2 UDB for z/OS バージョン 8 互換モード・サブシステムかそれ以前の場合 VARCHAR(18)。 NULL 可能: いいえ。</p> <p>データが適用される表またはビュー。</p>
COL_TYPE	<p>データ・タイプ: CHAR(1)。 NULL 可能: いいえ。</p> <p>列のタイプを示すフラグ。</p> <p>A 変更後イメージ列。</p> <p>B 変更前イメージ列。</p> <p>C スカラー関数を使用する算出列または SQL 式。</p> <p>D DATALINK 列。</p> <p>F 列関数を使用する算出列。</p> <p>L LOB 標識値。</p> <p>P 変更前イメージ述部列。</p> <p>R システムから提供され、主キー列として使用される相対レコード番号列。DB2 DataPropagator for iSeries でのみ使用されます。</p>

IBMSNAP_SUBS_COLS

表 93. サブスクリプション列表の列 (続き)

列名	説明
TARGET_NAME	<p>データ・タイプ: VARCHAR(30)。 NULL 可能: いいえ。</p> <p>ターゲット表またはビューの列の名前。ソース列名と一致する必要はありません。</p> <p>内部の CCD 列名は変更できません。それらの名前はソース表の列名と一致している必要があります。</p>
IS_KEY	<p>データ・タイプ: CHAR(1)。 NULL 可能: いいえ。</p> <p>列がターゲット・キーの一部であるかどうかを示すフラグ。ターゲット・キーは、コンデンス・ターゲット表のユニーク索引、または主キーのいずれかです。</p> <p>Y 列はターゲット・キー全体であるか、ターゲット・キーの一部です。</p> <p>N 列は、ターゲット・キーの一部ではありません。</p>
COLNO	<p>データ・タイプ: SMALLINT。 NULL 可能: いいえ。</p> <p>オリジナル・ソースにおける列の数値ロケーション。これは、表示およびサブスクリプションで他のユーザー列に対して相対的に保たれます。</p>
EXPRESSION	<p>データ・タイプ: VARCHAR(254)。 NULL 可能: いいえ。</p> <p>ターゲット列の内容を作成するために使用された SQL 式、またはソース列名。</p>

ASN.IBMSNAP_SUBS_EVENT

サーバー: アプライ・コントロール・サーバー

索引: EVENT_NAME、EVENT_TIME

この表の情報は、SQL を使って更新できます。

サブスクリプション・イベント表には、サブスクリプション・セットに関連付けられたイベント・トリガーに関する情報が含まれます。また、イベント名と関連付けられた名前とタイム・スタンプも含まれます。アプライ・プログラムを始動するために新規イベントを作成するときに、この表に行を挿入してください。81 ページの『イベントに基づくスケジューリング』を参照してください。

表 94 では、サブスクリプション・イベント表の列の要旨を示します。

表 94. サブスクリプション・イベント表の列

列名	説明
EVENT_NAME	<p>データ・タイプ: CHAR(18)。 NULL 可能: いいえ。</p> <p>イベントのユニーク ID。この ID は、サブスクリプション・セットのレプリケーションを起動するために使用されます。</p>
EVENT_TIME	<p>データ・タイプ: TIMESTAMP。 NULL 可能: いいえ。</p> <p>現在または将来の通知時刻の、アプライ・コントロール・サーバーのタイム・スタンプ。レプリケーション・イベントをシグナルするユーザー・アプリケーションがこの列に値を提供します。</p>

表 94. サブスクリプション・イベント表の列 (続き)

列名	説明
END_SYNCHPOINT	<p>データ・タイプ: ビット・データの場合 CHAR(10)。 NULL 可能: はい。</p> <p>この時点までにキャプチャーされたデータのみをアプライするようにアプライ・プログラムに指示するログ・シーケンス番号。シグナル表を参照し、タイム・スタンプに関連付けられた正確なログ・シーケンス番号を検索すれば、使用する正確な END_SYNCHPOINT を知ることができます。ログのこの時点以降にコミットされたトランザクションは、その後のイベントが通知されるまでは複製されません。ユーザーが END_SYNCHPOINT および END_OF_PERIOD の値を指定した場合、アプライ・プログラムは END_SYNCHPOINT 値を使用するので、複製する最大ログ・シーケンス番号を知るためにコントロール表から計算を実行する必要はなくなります。</p>
END_OF_PERIOD	<p>データ・タイプ: TIMESTAMP。 NULL 可能: はい。</p> <p>この時点までにログに記録されたデータのみをアプライするためにアプライ・プログラムによって使用されるタイム・スタンプ。ログのこの時点以降にコミットされたトランザクションは、その後のイベントが通知されるまでは複製されません。</p>

ASN.IBMSNAP_SUBS_MEMBR

サーバー: アプライ・コントロール・サーバー

索引: APPLY_QUAL、 SET_NAME、 WHOS_ON_FIRST、 SOURCE_OWNER、 SOURCE_TARGET、 SOURCE_VIEW_QUAL、 TARGET_OWNER、 TARGET_TABLE

重要: SQL を使用してこの表を更新するときには、注意してください。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。

サブスクリプション・メンバー表には、サブスクリプション・セットに対して定義された、ソース表とターゲット表の個々のペアに関する情報が含まれています。ユーザーがサブスクリプション・セット・メンバーを追加すると、この表に 1 つの行が自動的に挿入されます。

この表は、サブスクリプション・セット内で特定のソース表とターゲット表の対を指定するために使用します。

表 95 では、サブスクリプション・メンバー表の列の要旨を示します。

表 95. サブスクリプション・メンバー表の列

列名	説明
APPLY_QUAL	<p>データ・タイプ: CHAR(18)。 NULL 可能: いいえ。</p> <p>このサブスクリプション・セット・メンバーを処理するアプライ・プログラムを一意的に識別します。</p>
SET_NAME	<p>データ・タイプ: CHAR(18)。 NULL 可能: いいえ。</p> <p>このメンバーが所属するサブスクリプション・セットの名前。</p>

IBMSNAP_SUBS_MEMBR

表 95. サブスクリプション・メンバー表の列 (続き)

列名	説明
WHOS_ON_FIRST	<p>データ・タイプ: CHAR(1)。 NULL 可能: いいえ。</p> <p>Update-anywhere レプリケーション・シナリオでは、処理順序をコントロールするために以下の値を使用します。</p> <p>F (first の略) ソース表がレプリカであり、ターゲット表がマスターです。レプリカ表とマスター表の間で更新の矛盾が生じた場合、レプリカの側の矛盾するトランザクションはリジェクトされます。F は読み取り専用のサブスクリプションでは使用されません。Update-anywhere で使用されるものです。</p> <p>S (second の略) ソース表はマスター表またはその他のソースであり、ターゲット表はレプリカまたはその他のコピーです。マスター表とレプリカ表の間で更新の矛盾が生じた場合、レプリカの側の矛盾するトランザクションはリジェクトされます。S は、すべての読み取り専用サブスクリプションについて使用されます。</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・サブシステムの場合 VARCHAR(128)。 NULL 可能: いいえ。</p> <p>このメンバーのソース表またはビューの上位修飾子。</p>
SOURCE_TABLE	<p>データ・タイプ: VARCHAR(128)、 DB2 UDB for z/OS バージョン 8 互換モード・サブシステムかそれ以前の場合 VARCHAR(18)。 NULL 可能: いいえ。</p> <p>このメンバーのソース表またはビューの名前。</p>
SOURCE_VIEW_QUAL	<p>データ・タイプ: SMALLINT。 NULL 可能: いいえ。</p> <p>登録表内の類似した列を突き合わせることによって、物理表のビューをサポートします。この値は、ソースとして定義されている物理表の場合は 0 に、ソースとして定義されているビューの場合は 0 より大きい値に設定されます。同じ SOURCE_OWNER および SOURCE_TABLE 列値を持つ別々のソース・ビューについて複数のサブスクリプションをサポートするためにこの列を使用します。</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・サブシステムの場合 VARCHAR(128)。 NULL 可能: いいえ。</p> <p>このメンバーのターゲット表またはビューの上位修飾子。</p>
TARGET_TABLE	<p>データ・タイプ: VARCHAR(128)、 DB2 UDB for z/OS バージョン 8 互換モード・サブシステムかそれ以前の場合 VARCHAR(18)。 NULL 可能: いいえ。</p> <p>このメンバーのターゲット表またはビューの名前。</p>
TARGET_CONDENSED	<p>データ・タイプ: CHAR(1)。 NULL 可能: いいえ。</p> <p>以下のことを示すフラグ。</p> <p>Y 特定の主キー値に関して、ターゲット表に表示される行は 1 つだけです。</p> <p>N 完全な更新履歴を保持して、すべての変更を残す必要があります。</p> <p>A ターゲット表は、基礎集約表または変更集約表です。</p>

表 95. サブスクリプション・メンバー表の列 (続き)

列名	説明
TARGET_COMPLETE	<p>データ・タイプ: CHAR(1)。 NULL 可能: いいえ。</p> <p>以下のことを示すフラグ。</p> <p>Y ターゲット表は、関係するそれぞれの主キー値につき 1 つの行を保持します。</p> <p>N ターゲット表は、主キー値の行のサブセットを保持します。</p>
TARGET_STRUCTURE	<p>データ・タイプ: SMALLINT。 NULL 可能: いいえ。</p> <p>ターゲット表の構造。</p> <p>1 ユーザー表</p> <p>3 CCD 表</p> <p>4 ポイント・イン・タイム表</p> <p>5 基礎集約表</p> <p>6 変更集約表</p> <p>7 レプリカ</p> <p>8 ユーザー・コピー</p>
PREDICATES	<p>データ・タイプ: VARCHAR(1024)。 NULL 可能: はい。</p> <p>TARGET_TABLE 列内の表用の WHERE 文節に入れられる述部をリストします。この WHERE 文節は、ソース表の行サブセットを作成します。述部が認識されるのは、WHOS_ON_FIRST が S に設定されている場合だけです。アプライ・プログラムは ORDER BY 文節を生成できないため、述部に ORDER BY 文節を含めることはできません。集約表は、ダミー述部とその後に続く GROUP BY 文節を必要としています。</p> <p>アプライ・プログラムはフル・リフレッシュおよび変更キャプチャー・レプリケーションの両方でこれらの述部を使用するため、この列に、CD 表または UOW 表の列に関係する述部を含めることはできません。CD 表または UOW 表の参照を含む述部は、UOW_CD_PREDICATES 列に保管されます。</p>
MEMBER_STATE	<p>データ・タイプ: CHAR(1)。 NULL 可能: はい。</p> <p>メンバーの状態を示すフラグ。</p> <p>N (New の略) メンバーはこのサブスクリプション・セットの新規メンバーです。最近使用可能になったメンバーがあれば、それらもこの状態で表示されます。</p> <p>L (Loaded の略) このサブスクリプション・セットのメンバーがロードされましたが、変更キャプチャー・サイクルはまだ発生していません。</p> <p>S (Synchronized の略) メンバーは、New (N) 状態から Loaded (L) 状態に進み、Synchronized 状態の他のすべてのサブスクリプション・セット・メンバーと同期されています。サブスクリプション・セットのすべてのメンバーが Synchronized 状態の場合は、サブスクリプション・セット・レベルで変更レプリケーションを行うことができます。</p> <p>D (Disabled の略) メンバーはこのサブスクリプション・セットでは使用不可です。</p>

IBMSNAP_SUBS_MEMBR

表 95. サブスクリプション・メンバー表の列 (続き)

列名	説明
TARGET_KEY_CHG	<p>データ・タイプ: CHAR(1)。 NULL 可能: いいえ。</p> <p>ユーザーがソース表で、ターゲット表のターゲット・キー列のソース列を変更したときに、アプライ・プログラムが更新をどのように処理するかを示すフラグ。</p> <p>Y アプライ・プログラムは、ターゲット・キー列の変更前イメージに基づいてターゲット表を更新します。つまり、アプライ・プログラムは述部を新しい値ではなく、古い値に変更します。ターゲット・キーの変更前イメージが CD 表の中に存在するように、変更前イメージの各列を登録してください。登録表の中の対応する登録項目について、CHG_UPD_TO_DEL_INS 列の値が N に設定されていることを確認してください。</p> <p>N アプライ・プログラムは、ターゲット・キーを構成する列が更新されていないという仮定に基づくロジックを使用して、更新および削除を処理します。</p>
UOW_CD_PREDICATES	<p>データ・タイプ: VARCHAR(1024)。 NULL 可能: はい。</p> <p>アプライ・プログラムが変更キャプチャー・レプリケーションにのみ必要とし、フル・リフレッシュには必要としない、CD 表または UOW 表からの列を含む述部が入っています。アプライ・プログラムは変更キャプチャー・レプリケーション時には、この列の述部と、PREDICATES 列の述部を処理します。アプライ・プログラムはフル・リフレッシュ時には、PREDICATES 列の述部のみを処理します。</p>
JOIN_UOW_CD	<p>データ・タイプ: CHAR(1)。 NULL 可能: はい。</p> <p>アプライ・プログラムがユーザー・コピー・ターゲット表を処理するときに、CD 表および UOW 表の結合を行うかどうかを示すフラグ。このフラグは、CD 表の中になく、UOW 表の列を使用する述部を持つサブスクリプション・セット・メンバーをユーザーが定義したときに必要になります。ターゲット表のタイプがユーザー・コピー以外のものである場合には、アプライ・プログラムはメンバーの処理時に CD 表および UOW 表の結合を使用するため、メンバーの処理時にはこの列は無視されます。</p> <p>Y アプライ・プログラムはメンバーの処理時に CD 表および UOW 表の結合を使用します。</p> <p>N アプライ・プログラムはメンバーの処理時に CD 表および UOW 表の結合を使用しません。CD 表からのみ変更が読み取られます。</p> <p>NULL アプライ・プログラムはメンバーの処理時にこの列を無視します。ターゲット表がユーザー・コピーであり、この列の値が NULL である場合には、アプライ・プログラムはメンバーの処理時に CD 表および UOW 表を結合しません。</p>

表 95. サブスクリプション・メンバー表の列 (続き)

列名	説明
LOADX_TYPE	<p>データ・タイプ: SMALLINT。 NULL 可能: はい。</p> <p>このメンバーのロードのタイプ。この列の値は、デフォルトのオーバーライドに使用されます。</p> <p>NULL</p> <p>z/OS の場合: クロス・ローダー機能 (DB2 Utilities Suite で使用可能) が、このメンバーに対して使用されます。</p> <p>UNIX および Windows の場合: ASNLOAD 出口が、このメンバーに最適なユーティリティを決定します。(オプション 3、4 または 5)</p> <p>1 このメンバーには ASNLOAD は使用されません。これにより、ユーザーが始動時に LOADX を指定した場合にでも、特定のサブスクリプション・セット・メンバーについて ASNLOAD オプションがオフになります。</p> <p>2 ユーザー定義の、またはユーザーが変更した ASNLOAD 終了コードが使用されます。</p> <p>3 クロス・ローダー機能 (DB2 Utilities Suite で使用可能) が、このメンバーに対して使用されます。</p> <p>4 UNIX および Windows の場合のみ: このメンバーには EXPORT/LOAD が使用されます。</p> <p>5 UNIX および Windows の場合のみ: このメンバーには EXPORT/IMPORT が使用されます。</p> <p>UNIX および Windows の場合の制約事項: LOAD ユーティリティは、範囲クラスター表の場合はサポートされません。範囲クラスター表のフル・リフレッシュを行うためには、DB2 の IMPORT ユーティリティを使用するか、アプリ・プログラムを使用して表のフル・リフレッシュを SQL を介して行います。</p>
LOADX_SRC_N_OWNER	<p>データ・タイプ: VARCHAR(30)。 NULL 可能: はい。</p> <p>ユーザーが作成したニックネームの所有者。以下のすべての条件が揃っている場合、この値は必須です。</p> <ul style="list-style-type: none"> このメンバーに対してクロス・ローダーが使用される (LOADX_TYPE は 3)。 ターゲット・サーバーは UNIX または Windows である。 ソースはニックネームではない。
LOADX_SRC_N_TABLE	<p>データ・タイプ: VARCHAR(128)。 NULL 可能: はい。</p> <p>ユーザーが作成したニックネーム表。以下のすべての条件が揃っている場合、この値は必須です。</p> <ul style="list-style-type: none"> このメンバーに対してクロス・ローダーが使用される (LOADX_TYPE は 3)。 ターゲット・サーバーは UNIX または Windows である。 ソースはニックネームではない。

ASN.IBMSNAP_SUBS_SET

サーバー: アプライ・コントロール・サーバー

索引: APPLY_QUAL、SET_NAME、WHOS_ON_FIRST

重要: SQL を使用してこの表を更新するときには、注意してください。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。

サブスクリプション・セット表には、アプライ・コントロール・サーバーで定義されたすべてのサブスクリプション・セットが記載され、これらのセットのレプリケーションの進行状況が文書化されます。行は、サブスクリプション・セット定義を作成したときに自動的にこの表に挿入されます。

表 96 では、サブスクリプション・セット表の列の要旨を示します。

表 96. サブスクリプション・セット表の列

列名	説明
APPLY_QUAL	<p>データ・タイプ: CHAR(18)。 NULL 可能: いいえ。</p> <p>このサブスクリプション・セットを処理するアプライ・プログラムを一意的に識別します。</p>
SET_NAME	<p>データ・タイプ: CHAR(18)。 NULL 可能: いいえ。</p> <p>サブスクリプション・セットの名前。</p>
SET_TYPE	<p>データ・タイプ: CHAR(1)。 NULL 可能: いいえ。</p> <p>セットが読み取り専用か読み取り/書き込みかを示すフラグ。</p> <p>R セットは読み取り専用です。</p> <p>U セットは Update-anywhere 構成であるため、読み取り/書き込みです。</p>
WHOS_ON_FIRST	<p>データ・タイプ: CHAR(1)。 NULL 可能: いいえ。</p> <p>Update-anywhere レプリケーション・シナリオでは、処理順序をコントロールするために以下の値を使用します。</p> <p>F (first の略) ソース表がレプリカであり、ターゲット表がマスターです。レプリカ表とマスター表の間で更新の矛盾が生じた場合、レプリカの側の矛盾するトランザクションはリジェクトされます。F は読み取り専用のサブスクリプションでは使用されません。Update-anywhere で使用されるものです。</p> <p>S (second の略) ソース表はマスター表またはその他のソースであり、ターゲット表はレプリカまたはその他のコピーです。マスター表とレプリカ表の間で更新の矛盾が生じた場合、レプリカの側の矛盾するトランザクションはリジェクトされます。S は、すべての読み取り専用サブスクリプションについて使用されます。</p>

表 96. サブスクリプション・セット表の列 (続き)

列名	説明
ACTIVATE	<p>データ・タイプ: SMALLINT。 NULL 可能: いいえ。</p> <p>アプライ・プログラムが次のサイクルでこのセットを処理するかどうかを示すフラグ。</p> <p>0 サブスクリプション・セットは非活動化されています。アプライ・プログラムはこのセットを処理しません。</p> <p>1 サブスクリプション・セットは無期限にアクティブです。アプライ・プログラムは、ユーザーがセットを非活動化するか、アプライ・プログラムが処理を行えなくなるまで、各アプライ・サイクルでセットを処理します。</p> <p>2 サブスクリプション・セットは、1 つのアプライ・サイクルでのみアクティブです。アプライ・プログラムはセットを一度処理した後、セットを非活動化します。</p>
SOURCE_SERVER	<p>データ・タイプ: CHAR(18)。 NULL 可能: いいえ。</p> <p>ソース表およびビューが定義されている、キャプチャー・コントロール・サーバーのデータベース名。</p>
SOURCE_ALIAS	<p>データ・タイプ: CHAR(8)。 NULL 可能: はい。</p> <p>SOURCE_SERVER 列で指定されているキャプチャー・コントロール・サーバーに対応する DB2 Universal Database 別名。</p>
TARGET_SERVER	<p>データ・タイプ: CHAR(18)。 NULL 可能: いいえ。</p> <p>ターゲットの表またはビューが保管されているサーバーのデータベース名。</p>
TARGET_ALIAS	<p>データ・タイプ: CHAR(8)。 NULL 可能: はい。</p> <p>TARGET_SERVER 列で指定されているターゲット・サーバーの DB2 Universal Database 別名。</p>

表 96. サブスクリプション・セット表の列 (続き)

列名	説明
STATUS	<p>データ・タイプ: SMALLINT。NULL 可能: いいえ。</p> <p>特定のサイクル後のアプライ・プログラムの作業状況を表す値。</p> <p>-1 レプリケーションは失敗しました。アプライ・プログラムは適用済みの行のセット全体をバックアウトし、データはコミットされません。始動パラメーターが <code>SQLERRCONTINUE = Y</code> の場合、最後のサイクル中にアプライ・プログラムに戻される <code>SQLSTATE</code> は、<code>SQLERRCONTINUE (apply_qualifier.SQS)</code> の入力ファイルでユーザーが指定した許容エラーの 1 つではありません。</p> <p>0 アプライ・プログラムはサブスクリプション・セットを正常に処理しました。始動パラメーターが <code>SQLERRCONTINUE = Y</code> の場合、アプライ・プログラムは、ユーザーから <code>SQLERRCONTINUE</code> 始動パラメーターで (<code>apply_qualifier.SQS</code> で) 指定された SQL エラーを検出しておらず、行をリジェクトしていません。</p> <p>2 アプライ・プログラムはサブスクリプション・セットを複数のサイクルで処理しています。アプライ・プログラムは、<code>MAX_SYNCH_MINUTES</code> コントロール列に従って分割された 1 つの論理サブスクリプションを正常に処理しました。</p> <p>16 アプライ・プログラムはサブスクリプション・セットを正常に処理し、0 という状況に戻しました。しかしアプライ・プログラムは、ユーザーが <code>SQLERRCONTINUE</code> 始動パラメーターで (<code>apply_qualifier.SQS</code> で) 指定したいくつかの SQL エラーを検出したため、いくつかの行をリジェクトしました。失敗した行の詳細は、<code>apply_qualifier.ERR</code> ファイルで確認してください。</p> <p>例: ユーザーは <code>SQLERRCONTINUE = Y</code> と設定し、SQL の許容される状態を 23502 (SQL コード -407) と指定します。23502 エラーが発生しますが、他のエラーは発生していません。アプライ・プログラムはサブスクリプション・セットの処理を終了し、状態を 16 に設定します。次の実行時に、23502 エラーが発生した後、07006 (SQL コード -301) が発生します。アプライ・プログラムは今回は、サブスクリプション・セットの処理を停止し、適用済みの行のセット全体をバックアウトし、状況を -1 に設定します (データはコミットされていないため)。</p> <p>18 アプライ・プログラムは複数のサイクルでサブスクリプション・セットを処理し、2 という状況に戻しています。これは、<code>MAX_SYNCH_MINUTES</code> コントロール列に従って分割された 1 つの論理サブスクリプションが正常に処理されたことを意味します。しかし、ユーザーが <code>SQLERRCONTINUE</code> 始動パラメーターで (<code>apply_qualifier.SQS</code> で) 指定した SQL エラーのいくつかを検出されたため、いくつかの行がリジェクトされています。失敗した行の詳細は、<code>apply_qualifier.ERR</code> ファイルで確認してください。</p>
LASTRUN	<p>データ・タイプ: TIMESTAMP。NULL 可能: いいえ。</p> <p>サブスクリプション・セットが最後に開始された概算の時刻。アプライ・プログラムは、サブスクリプション・セットを処理するたびに <code>LASTRUN</code> 値を設定します。これは、アプライ・プログラムがサブスクリプション・セットの処理を開始する、アプライ・コントロール・サーバーにおけるおおよその時刻です。</p>

表 96. サブスクリプション・セット表の列 (続き)

列名	説明
REFRESH_TYPE	<p>データ・タイプ: CHAR(1)。 NULL 可能: いいえ。</p> <p>アプライ・プログラムにこのサブスクリプション・セットの処理を促すために使用されるスケジューリングのタイプ。</p> <p>R アプライ・プログラムは時間ベースのスケジューリングを使用します。アプライ・プログラムは、 SLEEP_MINUTES の中の値を使用して、サブスクリプション・セットの処理を開始する時間を判断します。</p> <p>E アプライ・プログラムはイベント・ベースのスケジューリングを使用します。アプライ・プログラムはサブスクリプション・イベント (IBMSNAP_SUBS_EVENT) 表の中の時刻値を確認して、サブスクリプション・セットの処理を開始する時間を判断します。イベントが生じないと、レプリケーション (変更のキャプチャーまたはフル・リフレッシュ) を何も開始できません。</p> <p>B アプライ・プログラムは時間ベースのスケジューリングとイベント・ベースのスケジューリングの両方を使用します。このため、アプライ・プログラムは時間またはイベントの基準に基づいてサブスクリプション・セットを処理します。</p>
SLEEP_MINUTES	<p>データ・タイプ: INT。 NULL 可能: はい。</p> <p>サブスクリプション・セット処理の間の非活動時間 (分単位) を指定します。処理時間は、REFRESH_TYPE が R または B の場合にのみ使用されます。SLEEP_MINUTES の値が NULL の場合は、アプライ・プログラムが継続してセットを処理します。アプライ・プログラムは、セットを可能な限り頻繁に処理しますが、また同じアプライ修飾子で、その他すべてのアクティブなサブスクリプション・セットも処理します。</p>
EVENT_NAME	<p>データ・タイプ: CHAR(18)。 NULL 可能: はい。</p> <p>イベントの名前を表すユニークな文字ストリング。この ID は、サブスクリプション・セットに対してレプリケーションを起動したい場合にサブスクリプション・イベント表を更新するのに使用します。イベント名は、REFRESH_TYPE が E または B の場合にのみ使用されます。</p>
LASTSUCCESS	<p>データ・タイプ: TIMESTAMP。 NULL 可能: はい。</p> <p>サブスクリプション・セットが最後に正常に処理されたときの、処理開始時点のアプライ・コントロール・サーバーのタイム・スタンプ。</p>
SYNCHPOINT	<p>データ・タイプ: ビット・データの場合 CHAR(10)。 NULL 可能: はい。</p> <p>アプライ・プログラムはこの列を使用して進行状況を記録します。サブスクリプション・セットのデータの処理が、この同期点の値まで終了していることを示します。</p>
SYNCHTIME	<p>データ・タイプ: TIMESTAMP。 NULL 可能: はい。</p> <p>アプライ・プログラムはこの列を使用して進行状況を記録します。サブスクリプション・セットのデータの処理が、このタイム・スタンプまで終了していることを示します。</p>

IBMSNAP_SUBS_SET

表 96. サブスクリプション・セット表の列 (続き)

列名	説明
CAPTURE_SCHEMA	<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>このサブスクリプション・セットのソースを処理するキャプチャー・コントロール表のスキーマ名。</p>
TGT_CAPTURE_SCHEMA	<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: はい。</p> <p>ターゲット表が別のサブスクリプション・セットのソースでもある場合 (multi-tier 構成中の外部の CCD 表、または Update-anywhere 構成中のレプリカ表など) は、この列には、表がソースとして機能するときに使用されるキャプチャー・スキーマが含まれます。</p>
FEDERATED_SRC_SRVR	<p>データ・タイプ: VARCHAR(18)。 NULL 可能: はい。</p> <p>DB2 以外のリレーショナル・ソースの場合にのみアプライされる、サブスクリプション・セットのソースである、フェデレーテッド・リモート・サーバーの名前。</p>
FEDERATED_TGT_SRVR	<p>データ・タイプ: VARCHAR(18)。 NULL 可能: はい。</p> <p>DB2 以外のリレーショナル・ターゲットの場合にのみアプライされる、サブスクリプション・セットのターゲットである、フェデレーテッド・リモート・サーバーの名前。</p>
JRN_LIB	<p>データ・タイプ: CHAR(10)。 NULL 可能: はい。</p> <p>OS/400 キャプチャー・サーバーにのみアプライされるこの列は、ソース表が使用するジャーナルのライブラリー名です。</p>
JRN_NAME	<p>データ・タイプ: CHAR(10)。 NULL 可能: はい。</p> <p>OS/400 キャプチャー・サーバーにのみアプライされるこの列は、ソース表が使用するジャーナルの名前です。この列で、アスタリスクの後ろに 9 つのブランクが続くときには、ソース表が現在ジャーナルの中にあることを意味します。この場合は、このソース表のデータをキャプチャーすることはできません。</p>
OPTION_FLAGS	<p>データ・タイプ: CHAR(4)。 NULL 可能: いいえ。</p> <p>DB2 レプリケーションの将来のオプション用に予約済み。この列には現在、デフォルト値の NNNN が入っています。</p>

表 96. サブスクリプション・セット表の列 (続き)

列名	説明
COMMIT_COUNT	<p>データ・タイプ: SMALLINT。 NULL 可能: はい。</p> <p>アプライ・プログラムがサブスクリプション・セットに対して実行する処理のタイプを示すフラグ。</p> <p>NULL これは、読み取り専用のサブスクリプション・セットの場合のデフォルト設定です。アプライ・プログラムは、すべてのデータが処理されるまで、n のサブスクリプション・セット・メンバーのメンバー 1 つずつについて、フェッチした応答セットを処理していき、セット全体のデータ処理が終了すると最後に、1 つのコミットを発行します。この COMMIT_COUNT 設定値を使用する利点は、処理が早く終了する可能性があることです。</p> <p>非 NULL の整数 アプライ・プログラムはトランザクションのモードでサブスクリプション・セットを処理します。すべての応答セットがフェッチされると、各トランザクションが IBMSNAP_INTENTSEQ 値の順に並べられ、コミット・シーケンスの順番で予備ファイルの内容が適用されます。このタイプの処理では、すべての予備ファイルを同時に開いて処理できます。この列で指定された数のトランザクションの後に、コミットが発行されます。たとえば、1 は各トランザクションの後にコミットすることを意味し、2 は、トランザクション 2 つごとにコミットすることを意味します。0 という整数は、すべてのフェッチ・データがアプライされた後で 1 つのコミットを発行することを意味します。トランザクションのモードの処理を使用する利点は、この処理ではターゲットでの参照保全制約が可能であり、暫定のコミットを発行できることです。</p>
MAX_SYNC_MINUTES	<p>データ・タイプ: SMALLINT。 NULL 可能: はい。</p> <p>サブスクリプション・サイクルでフェッチおよびアプライする変更データの量を規制するための時間しきい値限度。アプライ・プログラムはキャプチャー・サーバーの UOW 表または CCD 表の中の IBMSNAP_LOGMARKER 列に基づいてサブスクリプション・セットの処理をミニサイクルに分割し、ミニサイクルが正常に終了するたびに、ターゲット・サーバーで COMMIT を発行します。設定された限度が不適切になるようなりソース制約をアプライ・プログラムが検出した場合、この限度は自動的に再計算されます。1 より小さい MAX_SYNC_MINUTES 値は、NULL の MAX_SYNC_MINUTES 値と同じものとして処理されます。</p>
AUX_STMTS	<p>データ・タイプ: SMALLINT。 NULL 可能: いいえ。</p> <p>サブスクリプション・ステートメント (IBMSNAP_SUBS_STMTS) 表で定義する SQL ステートメントの数。これらのステートメントは、アプライ・プログラムがサブスクリプション・セットを処理する前または後に実行可能です。</p>

IBMSNAP_SUBS_SET

表 96. サブスクリプション・セット表の列 (続き)

列名	説明
ARCH_LEVEL	<p>データ・タイプ: CHAR(4)。NULL 可能: いいえ。</p> <p>レプリケーション・コントロール表の構造レベル。この列は、行を作成する基礎となった規則を識別します。このレベルは、IBM で定義されていて、バージョン 8 では以下のとおりです。</p> <p>0801 バージョン 8 SQL Replication</p> <p>0803 バージョン 8 SQL Replication (Oracle ソースの拡張サポート付き)</p> <p>0805 バージョン 8 SQL Replication (DB2 for z/OS 新機能のサポート付き)</p>

ASN.IBMSNAP_SUBS_STMTS

サーバー: アプライ・コントロール・サーバー

索引: APPLY_QUAL、SET_NAME、WHOS_ON_FIRST、BEFORE_OR_AFTER、STMT_NUMBER

重要: SQL を使用してこの表を更新するときには、注意してください。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。サブスクリプションの項目数は、ASN.IBMSNAP_SUBS_SET.AUX_STMTS 列に反映されていなければなりません。あるサブスクリプション・セットの AUX_STMTS がゼロの場合、アプライ・プログラムはサブスクリプション・ステートメント表の対応する項目を無視します。

サブスクリプション・ステートメント表には、各サブスクリプション・セット処理サイクルの前または後に実行される、ユーザー定義の SQL ステートメントまたはストアード・プロシージャ呼び出しが入っています。即時実行 (EI) ステートメントまたはストアード・プロシージャは、ソースまたはターゲット・サーバーでのみ実行できます。

この表には、SQL ステートメントまたはストアード・プロシージャ呼び出しを使うサブスクリプション・セットを定義するときに、値が挿入されます。

表 97 では、サブスクリプション・ステートメント表の列の要旨を示します。

表 97. サブスクリプション・ステートメント表の列

列名	説明
APPLY_QUAL	<p>データ・タイプ: CHAR(18)。NULL 可能: いいえ。</p> <p>SQL ステートメントまたはストアード・プロシージャを処理するアプライ・プログラムを一意的に識別します。</p>
SET_NAME	<p>データ・タイプ: CHAR(18)。NULL 可能: いいえ。</p> <p>SQL ステートメントまたはストアード・プロシージャが関連付けられているサブスクリプション・セットの名前。</p>

表 97. サブスクリプション・ステートメント表の列 (続き)

列名	説明
WHOS_ON_FIRST	<p>データ・タイプ: CHAR(1)。 NULL 可能: いいえ。</p> <p>Update-anywhere レプリケーション・シナリオでは、処理順序をコントロールするために以下の値を使用します。</p> <p>F (first の略) ターゲット表はユーザー表または親レプリカです。ソース表は従属のレプリカであり、ソース表とターゲット表との間で更新の競合が生じた場合、ソース表の競合するトランザクションはリジェクトされます。F は、読み取り専用サブスクリプションについては使用されません。</p> <p>S (second の略) ソース表はユーザー表、親レプリカ、またはその他のソースです。ターゲット表は従属のレプリカまたは他のコピーであり、ソース表とターゲット表との間で更新の競合が生じた場合、ターゲット表の競合するトランザクションはリジェクトされます。S は、すべての読み取り専用サブスクリプションについて使用されます。</p>
BEFORE_OR_AFTER	<p>データ・タイプ: CHAR(1)。 NULL 可能: いいえ。</p> <p>ステートメントの発行時間と場所を示す値。</p> <p>A ステートメントは、すべての応答セット行が適用された後、ターゲット・サーバーで実行されます。</p> <p>B ステートメントは、応答セット行が適用される前に、ターゲット・サーバーで実行されます。</p> <p>S ステートメントは、応答セット・カーソルを開く前に、キャプチャー・コントロール・サーバー上で実行されます。</p> <p>G DB2 レプリケーションの使用のために予約済み。</p> <p>X DB2 レプリケーションの使用のために予約済み。</p>
STMT_NUMBER	<p>データ・タイプ: SMALLINT。 NULL 可能: いいえ。</p> <p>BEFORE_OR_AFTER 列値の有効範囲で、実行の相対順序を定義します。</p>
EI_OR_CALL	<p>データ・タイプ: CHAR(1)。 NULL 可能: いいえ。</p> <p>次のどちらであるかを示す値。</p> <p>E SQL ステートメントは、EXEC SQL EXECUTE IMMEDIATE として実行されます。</p> <p>C SQL ステートメントには、EXEC SQL CALL として実行されるストアード・プロシージャ名が入っています。</p>
SQL_STMT	<p>データ・タイプ: VARCHAR(1024)。 NULL 可能: はい。</p> <p>以下のいずれかの値が入ります。</p> <p>ステートメント EI_OR_CALL = E の場合に EXEC SQL EXECUTE IMMEDIATE ステートメントとして実行される SQL ステートメント。</p> <p>プロシージャ EI_OR_CALL が C の場合に、EXEC SQL CALL ステートメントとして実行される、パラメーター、または CALL キーワードなしの SQL ストアード・プロシージャの 8 バイトの名前。</p>

IBMSNAP_SUBS_STMTS

表 97. サブスクリプション・ステートメント表の列 (続き)

列名	説明
ACCEPT_SQLSTATES	データ・タイプ: VARCHAR(50)。 NULL 可能: はい。 サブスクリプション・セットの定義時に指定した 1~10 個の 5 バイト SQLSTATE 値。これらの非ゼロ値は、正常実行としてアプライ・プログラムで受け入れられます。それ以外の値は実行が失敗する原因となります。

モニター・コントロール・サーバー上の表、およびその列に関する記述

このセクションでは、モニター・コントロール・サーバー上に保管される各表について、かなり詳しく説明します。また、各表の列のリストやその簡単な説明もあります。コントロール表はアルファベット順にリストされ、列は各表内に現れる順に左から右へリストされています。

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 98 では、IBMSNAP_ALERTS 表の列の要旨を示します。

表 98. IBMSNAP_ALERTS 表の列

列名	説明
MONITOR_QUAL	データ・タイプ: CHAR(18)。 NULL 可能: いいえ。 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。
COMPONENT	データ・タイプ: CHAR(1)。 NULL 可能: いいえ。 モニター対象のレプリケーション・コンポーネント。 C キャプチャー・プログラム A アプライ・プログラム S Q キャプチャー・プログラム R Q アプライ・プログラム
SERVER_NAME	データ・タイプ: CHAR(18)。 NULL 可能: いいえ。 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。

表 98. IBMSNAP_ALERTS 表の列 (続き)

列名	説明
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>
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 99 では、IBMSNAP_CONDITIONS 表の列の要旨を示します。

表 99. 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>
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>

表 99. IBMSNAP_CONDITIONS 表の列 (続き)

列名	説明
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>
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越えたかどうか。</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>

IBMSNAP_CONDITIONS

表 99. IBMSNAP_CONDITIONS 表の列 (続き)

列名	説明
CONDITION_NAME (続き)	APPLY_STATUS アプライ・プログラムの状況。
	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。
	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。
	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。
	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。
	APPLY_FULLREFRESH フル・リフレッシュが発生したかどうか。
	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。
	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。
	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。
	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。

表 99. 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>

IBMSNAP_CONDITIONS

表 99. 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。

表 99. 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>

IBMSNAP_CONDITIONS

表 99. 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)。

585 ページの表 100 では、IBMSNAP_CONTACTGRP 表の列の要旨を示します。

表 100. 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 101 では、IBMSNAP_CONTACTS 表の列の要旨を示します。

表 101. 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せん。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_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 102 では、IBMSNAP_MONENQ 表の列の要旨を示します。

表 102. 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_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 103 では、IBMSNAP_GROUPS 表の列の要旨を示します。

表 103. 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_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

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 104 では、IBMSNAP_MONPARMS 表の列の要旨を示します。

表 104. 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 自動整理はオン。</p> <p>N 自動整理はオフ。</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>

IBMSNAP_MONPARMS

表 104. 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。

589 ページの表 105 では、IBMSNAP_MONSERVERS 表の列の要旨を示します。

表 105. 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>

IBMSNAP_MONSERVERS

表 105. IBMSNAP_MONSERVERS 表の列 (続き)

列名	記述
STATUS	データ・タイプ: SMALLINT。NULL 可能: いいえ。 モニター・サイクルの状況を示すフラグ。 -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。 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。 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。

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 106 では、IBMSNAP_MONTRACE 表の列の要旨を示します。

表 106. IBMSNAP_MONTRACE 表の列

列名	記述
MONITOR_QUAL	データ・タイプ: CHAR(18)。NULL 可能: いいえ。 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。
TRACE_TIME	データ・タイプ: TIMESTAMP。NULL 可能: いいえ。 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。
OPERATION	データ・タイプ: CHAR(8)。NULL 可能: いいえ。 メッセージを分類するために使用される値。 ERROR エラー・メッセージ WARNING 警告メッセージ INFO 情報メッセージ
DESCRIPTION	データ・タイプ: VARCHAR(1024)。NULL 可能: いいえ。 メッセージ・コードおよびテキスト。

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 107 では、IBMSNAP_MONTRAIL 表の列の要旨を示します。

表 107. 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>

IBMSNAP_MONTRAIL

表 107. 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>

ターゲット・サーバー上の表、およびその列に関する記述

このセクションでは、ターゲット・サーバーで使用される各表について、かなり詳しく説明します。また、各表の列のリストやその簡単な説明もあります。表名はアルファベット順にリストされ、列名は各表内に現れる順に左から右へリストされています。

基礎集約表

schema.base_aggregate

サーバー: ターゲット・サーバー

重要: SQL を使用してこの表を更新した場合は、アプライ・プログラムによってフル・リフレッシュが実行されたときにユーザーの更新情報が失われる危険性があります。

基礎集約表は、ソース表上にあるデータに対して実行された集約関数の結果を含むターゲット表です。

表 108 では、基礎集約表の列の要旨を示します。

表 108. 基礎集約表の列

列名	記述
<i>user columns</i>	ソース表から算出された集約データ。
IBMSNAP_LLOGMARKER	ソース表のデータの集約が開始されたときのソース・サーバーの現行タイム・スタンプ。
IBMSNAP_HLOGMARKER	ソース表のデータの集約が完了したときのソース・サーバーの現行タイム・スタンプ。

変更集約表

schema.change_aggregate

サーバー: ターゲット・サーバー

重要: SQL を使用してこの表を更新した場合は、アプライ・プログラムによってフル・リフレッシュが実行されたときにユーザーの更新情報が失われる危険性があります。

変更集約表は、変更データ (CD) 表にあるデータに対して実行された集約関数の結果を含むターゲット表です。この表は基礎集約表と似ていますが、CD 表で実行される関数は、特定の時間間隔で発生する変更に対してのみ使用される点が異なります。

表 109 では、変更集約表の列の要旨を示します。

表 109. 変更集約表の列

列名	記述
<i>user key columns</i>	ターゲット・キーを構成する列。
<i>user nonkey columns</i>	ソース表からの非キー・データ列。このターゲット表の中の列名はソース表の中の列名と一致している必要はありませんが、データ・タイプは一致している必要があります。
<i>user computed columns</i>	SQL 式から派生したユーザー定義の列。ソース・データ・タイプを別のターゲット・データ・タイプに変換するために、SQL 関数で算出列を使用することができます。
IBMSNAP_LLOGMARKER	集約されている (CD+UOW) または CCD 表の行内の最も古い IBMSNAP_LOGMARKER または IBMSNAP_LLOGMARKER 値。
IBMSNAP_HLOGMARKER	集約されている (CD+UOW) または CCD 表の行内の最も新しい IBMSNAP_LOGMARKER または IBMSNAP_HLOGMARKER 値。

整合変更データ (CCD) 表

schema.CCD_table

この表の情報は、SQL を使って更新できます。

サーバー: ターゲット・サーバー

重要: SQL を使用してこの表を更新した場合は、アプライ・プログラムによってフル・リフレッシュが実行されたときにユーザーの更新情報が失われる危険性があります。

整合変更データ (CCD) 表は、ソースで発生した変更に関する情報と、これらの変更の順序を識別するための追加の列を含む、サブスクリプション・セット・メンバー内のターゲットです。列の値は、CD 表と UOW 表の結合により得られます。ターゲット・サーバーである CCD 表としては次のものが考えられます。

- CD 表の代替として機能する内部 CCD 表。

変更キャプチャー・レプリケーションの場合、アプライ・プログラムはこの表からターゲットへ直接変更を適用します。このタイプの CCD 表の名前は、登録 (IBMSNAP_REGISTER) 表の中で、変更元のレプリケーション・ソースと同じ行に保管されます。

- 読み取り専用のターゲット表である外部 CCD。

このタイプの CCD は、ターゲット・サーバーのソース・データの監査証跡を保持します。

- multi-tier のレプリケーション構成の中間層である外部 CCD。

このタイプの CCD は、層 1 に対してはターゲット表であり、層 3 に対してはソース表です。このタイプの CCD 表の名前は、登録 (IBMSNAP_REGISTER) 表の中のそれ自身の行に保管されます。

CCD 表およびターゲットの使用の詳細は、85 ページの『ターゲット・タイプの選択』を参照してください。

キャプチャー・プログラムは、CCD 表にデータを挿入せず、この表の整理も行いません。アプリケーションの要件に応じて、CCD 表の履歴の保存期間を決定してください。したがって、デフォルトでは CCD 表の整理は自動ではありませんが、サブスクリプション・サイクルの後に処理される SQL ステートメントを使えば簡単に自動化することができます。

外部 CCD の場合、UOW 表から次のいくつかの列を含めるように選択できます。APPLY_QUAL、IBMSNAP_AUTHID、IBMSNAP_AUTHTKN、IBMSNAP_REJ_CODE、および IBMSNAP_UOWID。

IBMSNAP_OPERATION 列で最初に組み込まれた操作コードと、シーケンス番号 IBMSNAP_INTENTSEQ および IBMSNAP_COMMITSEQ は、CCD 表に含まれません。圧縮 CCD 表の場合、それぞれの行ごとに最新の値のみが保持されます。

キャプチャー・トリガーによりキャプチャーされる、または非リレーショナル・データを含む CCD 表の詳細は、523 ページの『*schema.CCD_table* (DB2 以外)』を参照してください。

表 110 では、CCD 表の列の要旨を示します。

表 110. CCD 表の列

列名	記述
IBMSNAP_INTENTSEQ	1 つの変更を固有に識別するログまたはジャーナル・レコードのシーケンス番号。この値はグローバルに昇順です。
IBMSNAP_OPERATION	レコードの操作のタイプを示すフラグ。 I 挿入 U 更新 D 削除
IBMSNAP_COMMITSEQ	キャプチャーされたコミット・ステートメントのログ・レコード・シーケンス番号。この値は、ソース表の元のトランザクションによる挿入、更新、および削除をグループ化します。
IBMSNAP_LOGMARKER	キャプチャー・コントロール・サーバーにおけるコミット時刻。
<i>user key columns</i>	CCD 表が圧縮されている場合、この列には、ターゲット・キーを構成する列が含まれます。
<i>user nonkey columns</i>	ソース表からの非キー・データ列。このターゲット表の中の列名はソース表の中の列名と一致している必要はありませんが、データ・タイプの互換性は必要です。
<i>user computed columns</i>	SQL 式から派生したユーザー定義の列。ソース・データ・タイプを別のターゲット・データ・タイプに変換するために、SQL 関数で算出列を使用することができます。
IBMSNAP_APPLY_QUAL (オプション)	この CCD 表を処理するアプライ・プログラムを一意的に識別します。
IBMSNAP_AUTHID (オプション)	このトランザクションに関連する許可 ID。これはデータベースの監査に役立ちます。DB2 Universal Database for z/OS の場合、この列は 1 次許可 ID です。DB2 Universal Database for iSeries の場合、この列は、トランザクションを発生させたアプリケーションを実行しているユーザー・プロファイル ID の名前になります。この列は、ブランクを埋め込んだ 10 文字の ID を保留します。この列は自動的に他の表にコピーされません。ユーザー・データ列として選択してコピーする必要があります。この列は、非コンプリート CCD ターゲット表のユーザー・データ列として選択できます。
IBMSNAP_AUTHTKN (オプション)	このトランザクションに関連する許可トークン。この ID は、データベースの監査に役立ちます。DB2 Universal Database for z/OS の場合、この列は相関 ID です。DB2 Universal Database for iSeries の場合、この列は、トランザクションを発生させたジョブのジョブ名です。この列は自動的に他の表にコピーされません。ユーザー・データ列として選択してコピーする必要があります。この列は、非コンプリート CCD ターゲット表のユーザー・データ列として選択できます。

整合変更データ表

表 110. CCD 表の列 (続き)

列名	記述
IBMSNAP_REJ_CODE (オプション)	レプリケーション・ソースの定義時に競合検出が標準または詳細と指定されている場合、この値は Update-anywhere レプリケーションでのみ設定されます。Update-anywhere 構成に参加できないため、DB2 以外のリレーショナル・ターゲットの場合は使用できません。 0 認識された競合のないトランザクション。 1 ソース表とレプリカ表の同じ行に、複製されなかった変更による競合があるトランザクション。競合が生じると、トランザクションはレプリカ表でロールバックされます。 2 最低でも 1 つの同一行競合がある、以前のトランザクションに從属するトランザクションのカスケード・リジェクト。競合が生じると、トランザクションはレプリカ表でロールバックされます。 3 最低でも 1 つの参照保全制約違反があるトランザクション。このトランザクションはソース表で定義された参照制約に違反しているため、アプライ・プログラムはこのサブスクリプション・セットに失敗というマークを付けます。参照保全定義が訂正されるまで、更新はコピーできません。 4 最低でも 1 つの制約競合がある、以前のトランザクションに從属するトランザクションのカスケード・リジェクト。
IBMSNAP_UOWID (オプション)	この作業単位についての、ログ・レコード・ヘッダーからの作業単位 ID。

関連資料:

- 523 ページの『*schema.CCD_table* (DB2 以外)』

ポイント・イン・タイム表

schema.point_in_time

サーバー: ターゲット・サーバー

重要: SQL を使用してこの表を更新した場合は、アプライ・プログラムによってフル・リフレッシュが実行されたときにユーザーの更新情報が失われる危険性があります。

ポイント・イン・タイム表は、ソース・データのコピーと、特定の行がソース・サーバーで挿入または更新されたおおよそのポイント・イン・タイムのタイム・スタンプを含む追加のシステム列 (IBMSNAP_LOGMARKER) を保持します。

表 111 では、ポイント・イン・タイム表の列の要旨を示します。

表 111. ポイント・イン・タイム表の列

列名	記述
<i>user key columns</i>	ターゲット・キーを構成する列。
<i>user nonkey columns</i>	ソース表またはビューからの非キー・データ列。このターゲット表の中の列名はソース表の中の列名と一致している必要はありませんが、データ・タイプは一致している必要があります。

表 111. ポイント・イン・タイム表の列 (続き)

列名	記述
<i>user computed columns</i>	SQL 式から派生したユーザー定義の列。ソース・データ・タイプを別のターゲット・データ・タイプに変換するために、SQL 関数で算出列を使用することができます。
IBMSNAP_LOGMARKER	キャプチャー・コントロール・サーバーにおけるおおよそのコミット時刻。この列は、フル・リフレッシュ後は NULL になります。

レプリカ表

schema.replica

サーバー: ターゲット・サーバー

この表の情報は、SQL を使って更新できます。

レプリカ表には、ソース表と同じ主キー列が必要です。この類似性のために、レプリカ表は、その後のサブスクリプション・セットでソース表として使用できます。ターゲット表からソース表への変換は、ユーザーがレプリカ・ターゲット・タイプを定義し、CHANGE DATA CAPTURE 属性を指定すると、自動的に行われます。詳細については、94 ページの『読み取り/書き込みターゲットの定義 (Update-anywhere)』を参照してください。

表 112 では、レプリカ表の列の要旨を示します。

表 112. レプリカ表の列

列名	記述
<i>user key columns</i>	ターゲット・キーを構成する列。マスター表と同じ主キーであることが必要です。
<i>user nonkey columns</i>	ソース表からの非キー・データ列。このターゲット表の中の列名はソース表の中の列名と一致している必要はありませんが、データ・タイプは一致している必要があります。

ユーザー・コピー表

schema.user_copy

サーバー: ターゲット・サーバー

重要: SQL を使用してこの表を更新した場合は、アプライ・プログラムによってフル・リフレッシュが実行されたときにユーザーの更新情報が失われる危険性があります。

ユーザー・コピー表は、ソース表の列のコピーを含むターゲット表です。このターゲット表では、ソース表の行または列のサブセットを使用できますが、追加列を含めることはできません。

サブセット化とデータ拡張の場合を除き、ユーザー・コピー表はソース表の正しい状態を反映しますが、それが最新の状態であるとはかぎりません。ユーザー・コピー表 (または他の任意のターゲット表タイプ) への参照により、ソース表への直接ア

ユーザー・コピー表

クセスが多すぎるときに生じる競合問題を減らすことができます。ローカル・ユーザー・コピー表にアクセスすると、照会ごとにネットワークを使用してリモート・ソース表へアクセスするよりも迅速です。

表 113 では、ユーザー・コピー表の列の要旨を示します。

表 113. ユーザー・コピー表の列

列名	記述
<i>user key columns</i>	ターゲット・キーを構成する列。
<i>user nonkey columns</i>	ソース表またはビューからの非キー・データ列。このターゲット表の中の列名はソース表の中の列名と一致している必要はありませんが、データ・タイプは一致している必要があります。
<i>user computed columns</i>	SQL 式から派生したユーザー定義の列。ソース・データ・タイプを別のターゲット・データ・タイプに変換するために、SQL 関数で算出列を使用することができます。

付録 A. SQL レプリケーション用の Unicode および ASCII のコード化スキーム (z/OS)

DB2 DataPropagator for OS/390 および DB2 DataPropagator for z/OS バージョン 7 以降では、Unicode と ASCII のコード化スキームをサポートします。Unicode のコード化スキームを活用するには、少なくとも DB2 for OS/390 および DB2 for z/OS バージョン 7 が必要で、以下のセクションで説明するように、DB2 DataPropagator のソース、ターゲット、およびコントロールの各表を手動で作成または変換しなければなりません。ただし、既存のレプリケーション環境は、コード化スキームが変更されていなければ、DB2 DataPropagator for OS/390 および DB2 DataPropagator for z/OS バージョン 7 以降で機能します。使用しているシステムが Unicode システムの場合は、キャプチャー、アプライ、およびレプリケーション・アラート・モニターの各プログラムの BIND PLAN コマンドと PACKAGE コマンドに、ENCODING(EBCDIC) を付け足さなければなりません。コード・ページを互換性のあるものにする、およびコード・ページ変数の設定については、13 ページの『コード・ページ・トランザクションの計画』を参照してください。

コード化スキームの選択

ソース表、CD 表、およびターゲット表で同じコード化スキームを使用していると、レプリケーション環境でのデータ変換の必要性を最小にすることができます。これらの表のコード化スキームを選択するときは、1 つの CCSID 規則に従ってください。

表スペース・データは ASCII、EBCDIC、または Unicode の CCSID を使ってエンコードされます。ある SQL ステートメントで参照されるすべての表のコード化スキームは同じでなければなりません。また、表示または結合して使用する表もすべて、同じコード化スキームを使用しなければなりません。

単一の CCSID 規則に従わないと、DB2 はバインドまたは実行中に違反を検出して SQLCODE -873 を戻します。

どの表を ASCII または Unicode にする必要があるかは、クライアント/サーバー構成によります。具体的には、表のコード化スキームを選択するとき、以下の規則に従ってください。

- DB2 for OS/390 のソースまたはターゲット表は EBCDIC、ASCII、または Unicode にする。これらの表は、サポートされているどの DBMS (DB2 ファミリー、または DataJoiner を使用した非 DB2) の同一または異なるコード化スキームを持つ表とも、相互にコピーを行うことができます。
- DB2 for OS/390 ソース・サーバーで、サブスクリプション・セット・メンバーが作成されるときにターゲット・タイプが USERCOPY で JOIN_UOW_CD が Y と等しくない場合は、同じサーバー上の CD 表と UOW 表が同じコード化スキームを使用する必要はない。そうでない場合は、CD 表と UOW 表は同じコード化スキームを使用しなければなりません。

- シグナル (IBMSNAP_SIGNAL) 表は、キャプチャー・プログラムがシグナルをその表から選択するときに EBCDIC に変換する必要がないように、EBCDIC にエンコードされている必要がある。
- 同じコントロール・サーバー上のすべてのコントロール表 (ASN.IBMSNAP_SUBS_xxxx) は、同じコード化スキームを使用しなければならない。
- その他のコントロール表は、任意のコード化スキームを使用できる。

コード化スキームの設定

表に適切なコード化スキームを指定するには、表の生成に使用される SQL を変更します。

1. 適切なコード化スキームを使用して新しいソース表とターゲット表を作成するか、あるいは既存のターゲット表とソース表のコード化スキームを変更する。既存の表のコード化スキームを変更するときは、その前にキャプチャー・プログラムとアプライ・プログラムを停止しておき、その後でキャプチャー・プログラムをコールド・スタートし、アプライ・プログラムを再始動するようお勧めします。既存の表のコード化スキーム (1 つの表スペース内では同一のコード化スキームを持っていないければなりません) のコード化スキームを変更するには、以下のようにします。
 - a. Reorg Tablespace ユーティリティを使用して、既存の表スペースをアンロードする。
 - b. 既存の表スペースをドロップする。
 - c. 新しいコード化スキームを指定する表スペースを再作成する。
 - d. Load ユーティリティを使用して、古いデータを新しい表スペースにロードする。

Load と Reorg の各ユーティリティについての詳細は、「DB2 ユニバーサル・データ・ベース (OS/390 版および z/OS 版) ユーティリティ・ガイドおよび解説書 (SC88-8773)」を参照してください。
2. レプリケーション・センターを使用して、適切なコード化スキームを持つ新しいコントロール表を作成する。
3. Reorg ユーティリティと Load ユーティリティを使用して、既存のコントロール表と CD 表のコード化スキームを変更する。
4. レプリケーション・センターを使用して新しいレプリケーション・ソースまたはサブスクリプション・セットを作成するときに、適切なコード化スキームを指定する。

CCSID についての詳細は、「DB2 Universal Database サーバー (OS/390 版) SQL 解説書 (SD88-7293)」にあります。

付録 B. SQL レプリケーションの場合にキャプチャー・プログラムがジャーナル項目タイプを処理する方法 (iSeries)

以下の表では、キャプチャー・プログラムが様々なジャーナル項目タイプを処理する方法について説明します。

表 114. ジャーナル項目によるキャプチャー・プログラムの処理

ジャーナル・コード ¹	項目タイプ	記述	処理
C	CM	コミット済みのレコード変更のセット	UOW 表にレコードを挿入します。
C	RB	ロールバック	挿入された UOW 行はありません。
F	AY	物理ファイル・メンバーに適用されたジャーナル済みの変更	ASN2004 メッセージを発行し、ファイルのフル・リフレッシュを行います。
F	CE	物理ファイルのデータの終わりの変更	ASN2004 メッセージを発行し、ファイルのフル・リフレッシュを行います。
F	CR	消去された物理ファイル・メンバー	ASN2004 メッセージを発行し、ファイルのフル・リフレッシュを行います。
F	EJ	終了した物理ファイル・メンバーのジャーナリング	ASN200A メッセージを発行し、ファイルのフル・リフレッシュを行います。フル・リフレッシュは、ユーザーまたはシステムがジャーナリングを終了させたかにかかわらず、キャプチャー・プログラムが EJ ジャーナル項目を読み取るたびに起こります。ファイルの暗黙的なジャーナル終了イベントについての情報は、OS/400 のドキュメンテーションを参照してください。
F	IZ	初期化された物理ファイル・メンバー	ASN2004 メッセージを発行し、ファイルのフル・リフレッシュを行います。
F	MD	物理ファイルから削除されたメンバー (DLTLIB、DLTF、または RMVM)	ASN200A メッセージを発行し、フル・リフレッシュを試行します。
F	MF	解放された物理ファイル・メンバーのストレージ	ASN200A メッセージを発行し、ファイルのフル・リフレッシュを行います。

表 114. ジャーナル項目によるキャプチャー・プログラムの処理 (続き)

ジャーナル・コード ¹	項目タイプ	記述	処理
F	MM	移動されたメンバーを含む物理ファイル (ライブラリーのオブジェクトの名前変更 (RNMOBJ)、ファイルのオブジェクトの移動 (MOV OBJ))	ASN200A メッセージを発行し、フル・リフレッシュを試行します。
F	MN	名前変更されたメンバー (ファイルの RNMOBJ、メンバーの名前変更 (RNMM)) を含む物理ファイル	ASN200A メッセージを発行し、フル・リフレッシュを試行します。
F	MR	リストアされた物理ファイル・メンバー	ASN2004 メッセージを発行し、ファイルのフル・リフレッシュを行います。
F	RC	物理ファイル・メンバーから削除されたジャーナル済みの変更	ASN2004 メッセージを発行し、ファイルのフル・リフレッシュを行います。
F	RG	再編成された物理ファイル・メンバー	レプリケーション・キーとしてソース表の RRN が使用されている場合には、ASN2004 メッセージを発行して、ファイルのフル・リフレッシュを行います。
J	NR	次のジャーナル・レシーバーの ID	キャプチャー・プログラムをリセットします。
J	PR	直前のジャーナル・レシーバーの ID	ユニークのシーケンス番号カウンターを増分します。
R	DL	物理ファイル・メンバーから削除されたレコード	DLT レコードを CD 表に挿入します。
R	DR	ロールバックで削除されたレコード	DLT レコードを CD 表に挿入します。
R	PT	物理ファイル・メンバーに追加されたレコード	ADD レコードを CD 表に挿入します。
R	PX	物理ファイル・メンバーに直接追加されたレコード	ADD レコードを CD 表に挿入します。
R	UB	物理ファイル・メンバーで更新されたレコードの変更前イメージ	注 2 を参照してください。
R	UP	物理ファイル・メンバーで更新されたレコードの変更後イメージ	注 2 を参照してください。
R	BR	ロールバックで更新されたレコードの変更前イメージ	注 3 を参照してください。
R	UR	ロールバックで更新されたレコードの変更後イメージ	注 3 を参照してください。

表 114. ジャーナル項目によるキャプチャー・プログラムの処理 (続き)

ジャーナル・コード	項目	記述	処理
1	タイプ		

注:

- 以下の値が、ジャーナル・コードに使用されます。
 - C** コミットメント・コントロール操作
 - F** データベース・ファイル操作
 - J** ジャーナルまたはジャーナル・レシーバー操作
 - R** 特定のレコードに対する操作
- 登録表の PARTITION_KEYS_CHG 列が N の場合、R-UP イメージおよび R-UB イメージが単一の UPD レコードを CD 表に形成します。そうでない場合には、R-UB イメージが DLT レコードを CD 表に挿入し、R-UP イメージが ADD 表を CD 表に挿入します。
- 登録表の PARTITION_KEYS_CHG 列が N の場合、R-UR イメージおよび R-BR イメージが単一の UPD レコードを CD 表に形成します。そうでない場合には、R-BR イメージが DLT レコードを CD 表に挿入し、R-UR イメージが ADD 表を CD 表に挿入します。

他のジャーナル項目タイプはすべて、キャプチャー・プログラムによって無視されます。

付録 C. アプリケーション (Linux、UNIX、Windows) 内部からの SQL レプリケーション・プログラムの始動

レプリケーションの 1 サイクルにおけるレプリケーション・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) はどれも、アプリケーションの内部から呼び出しルーチンで始動できます。API が同期実行のみをサポートしているため、これらのルーチンを使用するには、キャプチャー・プログラムの場合は AUTOSTOP オプション、アプライ・プログラムの場合は COPYONCE オプションを指定しなければなりません。

API のサンプルとそのおのおの MAKE ファイルは以下のディレクトリーにあります。

Windows 版

sqllib¥samples¥repl

Linux および UNIX の場合

sqllib/samples/repl

上記のディレクトリーには、キャプチャー・プログラムを始動するための以下のファイルが含まれています。

capture_api.c

Windows、Linux、または UNIX でキャプチャー・プログラムを始動するためのサンプル・コード。

capture_api_nt.mak

Windows でのサンプル・コードの MAKE ファイル。

capture_api_unix.mak

UNIX でのサンプル・コードの MAKE ファイル。

上記のディレクトリーには、アプライ・プログラムを始動するための以下のファイルが含まれています。

apply_api.c

Windows、Linux、または UNIX でアプライ・プログラムを始動するためのサンプル・コード。

apply_api_nt.mak

Windows でのサンプル・コードの MAKE ファイル。

apply_api_unix.mak

UNIX でのサンプル・コードの MAKE ファイル。

上記のディレクトリーには、レプリケーション・アラート・モニターを始動するための以下のファイルが含まれています。

monitor_api.c

Windows、Linux、または UNIX でレプリケーション・アラート・モニターを始動するためのサンプル・コード。

monitor_api_nt.mak

Windows でのサンプル・コードの MAKE ファイル。

monitor_api_unix.mak

UNIX でのサンプル・コードの MAKE ファイル。

用語集

用語集

A

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・プログラムと 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。

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・ログが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。

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。

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。

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。

アプライ (apply). レプリケーションでは、レプリケーション・ターゲット表を最新表示したり更新したりすること。

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・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・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。アクティブ・ログには適さなくなったレコードが保留されている。

ASP. 「補助記憶域プール (ASP) (auxiliary storage pool (ASP))」を参照。

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。

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。

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。

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。

B

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。

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。

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。

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。

BLOB. 「バイナリー・ラージ・オブジェクト (BLOB) (binary large object (BLOB))」を参照。

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。

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。

C

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・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・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 以外のリレーショナル・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。

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。

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。

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。

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。

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動するサーバー。

CLOB. 「文字ラージ・オブジェクト (CLOB) (character large object (CLOB))」を参照。

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始動するプロセス。(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。

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。

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。

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。

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。

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 発行の活動化や非活動化、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。

コントロール表 (control table). 「レプリケーション・コントロール表 (replication control table)」を参照。

D

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・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

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)。

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。

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。

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。

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。

E

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。

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。

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。

F

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。

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。

G

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に、変更データを失う可能性のある状態。

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。

H

異機種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。

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。

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。

IASP. 「独立補助記憶域プール (IASP) (Independent Auxiliary Storage Pool (IASP))」を参照。

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。

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。

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。

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。

J

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。

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・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 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・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。

K

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。

L

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。

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。

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。

LOB. 「ラージ・オブジェクト (LOB) (large object (LOB))」を参照。

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。

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。

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。

ロッキング (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。ロッキングにより、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。

M

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。

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。

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。

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。

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、対等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。

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

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。

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。

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。

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。

NULL 値 (null value). 値が指定されていないパラメーターの位置。

O

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。

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。

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). 呼び出し可能な 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。

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。

P

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。

対等レプリケーション (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。対等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。

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。

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。

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。

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。

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。

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。

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。

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。

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・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・プロシージャに渡すプログラム。

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・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 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・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 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・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・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 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。

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 発行で公開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係する Q サブスクリプションのグループ。

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トランザクションを挿入してから、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。

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。公開キュー・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。「公開キュー・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。

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。

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。

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。

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。

参照制約 (referential constraints). 外部キーの非 NULL 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。

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。

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。

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。

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。

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 アプリケーション・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リケーション・プログラムも実行できる。

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、アプリケーション・プログラム、Q キャプチャー・プログラム、Q アプリケーション・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・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・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・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・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 アプリケーション・プログラムで処理される方法に関する設定が含まれる。「公開キュー・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・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。アプリケーション・プログラムはこれらの変更内容をアプリケーションする。「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 アプリケーション・プログラムはこれらの変更内容をアプリケーションする。「ターゲット表 (target table)」も参照。

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。

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。

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。

S

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公開するのに、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。

シリアルライズ (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。

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。

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。

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。

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。

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。

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。

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。

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。

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。

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。

T

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。

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。

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、アカウンティング、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。

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・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・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。

U

UDT. 「ユーザー定義タイプ (user-defined type)」を参照。

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。

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。

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、ソース表で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。ターゲット表で加えられた変更内容は、ソース表に複製され戻されない。

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。

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。

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。「対等レプリケーション (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。

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。

V

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。

W

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。

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。

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公開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公開し、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 発行で公開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。

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]

アナライザー

OS/400 版

作成, SQL パッケージの 35

呼び出しパラメーター 422

UNIX 版, 呼び出しパラメーター 332

Windows 版, 呼び出しパラメーター

332

アナライザー・レポート

ANZDPR コマンド 422

asnanalyze コマンド 331

アプライ修飾子

アプライ・プログラムの始動時に使用

149, 159

数, 関連したサブスクリプション・セ

ットの 71

サブスクリプション・セット内の変更

232

状況のモニター 205

命名規則 327

アプライ修飾子相互参照 (AUTHTKN) 表

515

アプライ・エンキュー (APPENQ) 表

548

アプライ・コントロール表

のリスト 548

APPENQ (アプライ・エンキュー

ー) 548

APPLYTRACE (アプライ・トレー

ス) 554

APPLYTRAIL (アプライ・トレ

ール) 554

APPLY_JOB (アプライ・ジョブ) 549

APPPARMS (アプライ・パラメータ

ー) 550

使用 148

変更 160

SUBS_COLS (サブスクリプション

列) 560

SUBS_EVENT (サブスクリプション・

イベント) 562

SUBS_MEMBR (サブスクリプシ

ョン・メンバー) 563

アプライ・コントロール表 (続き)

SUBS_SET (サブスクリプション・セ

ット) 568

SUBS_STMTS (サブスクリプション・

ステートメント) 574

アプライ・コントロール・サーバー

上のコントロール表 548

レプリケーション・センターへの追加

278

アプライ・ジョブ (APPLY_JOB) 表 549

アプライ・トレース (APPLYTRACE) 表

構造 554

整理 258

アプライ・トレール (APPLYTRAIL) 表

構造 554

整理 258

アプライ・パラメーター (APPPARMS) 表

550

使用 148

変更 160

アプライ・プログラム

アラート条件 176

お互いの通信

キャプチャー・トリガー 495, 498

キャプチャー・プログラム 495,

496

レプリケーション・アラート・モニ

ター 499

レプリケーション・センター 495

許可要件 23

コネクティビティー 17

コマンド 329

スケジューリング 493

スループット分析 205

設定, デフォルトの, パラメーターの

場合 148

操作 291

データ・ブロッキング 75

トランザクション・モード処理 78

パフォーマンス・データ 201

表モード処理 78

変更, パラメーター値の 148

待ち時間分析 205

ミニサイクル 75

メッセージ 204

印刷 204

ユーザー ID 23

予備ファイル, 記憶要件 9

ランタイム処理ステートメント 120

OS/400 版

開始 159

アプライ・プログラム (続き)

OS/400 版 (続き)

作成, SQL パッケージの 35

始動 460

状況のチェック 201

セットアップ 34, 36

停止 161, 431

ALWINACT パラメーター 464

APYQUAL パラメーター 462

COPYONCE パラメーター 465

CTLSVR パラメーター 462

DELAY パラメーター 464

FULLREFPGM パラメーター 463

INACTMSG パラメーター 463

JOBID パラメーター 461

OPTSNGSET パラメーター 466

RTYWAIT パラメーター 464

SUBNFYPGM パラメーター 463

TRACE パラメーター 462

TRLREUSE パラメーター 465

USER パラメーター 461

OS/400 用

スケジューリング 494

UNIX 版

開始 149, 334

構成 32

始動 605

状況のチェック 199

セットアップ 30

操作 330

停止 161, 330

デフォルト・パラメーター 147

バインド 32

パスワード・ファイル 26

apply_path パラメーター 151, 336

apply_qual パラメーター 151, 330,

335

control_server パラメーター 152,

330, 336

copyonce パラメーター 152, 338

delay パラメーター 153, 339

errwait パラメーター 153, 339

inamsg パラメーター 154, 337

loadxit パラメーター 154, 337

logreuse パラメーター 154, 336

logstdout パラメーター 155, 336

notify パラメーター 155, 337

opt4one パラメーター 155, 339

pwdfile パラメーター 156, 336

sleep パラメーター 156, 338

spillfile パラメーター 157, 341

アプライ・プログラム (続き)

UNIX 版 (続き)

sqlerrcontinue パラメーター 157, 340
status 330
term パラメーター 158, 340
trlreuse パラメーター 158, 338

Windows 版

開始 149, 334
構成 32
始動 605
状況のチェック 199
セットアップ 30
操作 147, 330
停止 161, 330
デフォルト・パラメーター 147
バインド 32
パスワード・ファイル 26
apply_path パラメーター 151, 336
apply_qual パラメーター 151, 330, 335
control_server パラメーター 152, 330, 336
copyonce パラメーター 152, 338
delay パラメーター 153, 339
errwait パラメーター 153, 339
inamsg パラメーター 154, 337
loadxit パラメーター 154, 337
logreuse パラメーター 154, 336
logstdout パラメーター 155, 336
notify パラメーター 155, 337
opt4one パラメーター 155, 339
pwdfile パラメーター 156, 336
sleep パラメーター 156, 338
spillfile パラメーター 157, 341
sqlerrcontinue パラメーター 157, 340
status 330
term パラメーター 158, 340
trlreuse パラメーター 158, 338

z/OS 版

開始 149, 334
状況のチェック 199
セットアップ 36
操作 330
停止 161, 330
デフォルト・パラメーター 147
apply_path パラメーター 151, 336
apply_qual パラメーター 151, 330, 335
control_server パラメーター 152, 330, 336
copyonce パラメーター 152, 338
db2_subsystem パラメーター 153, 335
delay パラメーター 153, 339

アプライ・プログラム (続き)

z/OS 版 (続き)

errwait パラメーター 153, 339
inamsg パラメーター 154, 337
loadxit パラメーター 154, 337
logreuse パラメーター 154, 336
logstdout パラメーター 155, 336
notify パラメーター 155, 337
opt4one パラメーター 155, 339
pwdfile パラメーター 156, 336
sleep パラメーター 156, 338
spillfile パラメーター 157, 341
status 330
term パラメーター 158, 340
trlreuse パラメーター 158, 338

アプリケーション

始動、レプリケーション・プログラム 605

アラート条件

アプライ・プログラム 176
概要 176
キャプチャー・プログラムの場合 176
選択 187
通知基準 195
ASNMAIL 出口ルーチン 181
E メール通知 180
list 176
Q アプライ・プログラムの 176
Q キャプチャー・プログラムの 176
アラート・モニター
参照: レプリケーション・アラート・モニター
異機種のレプリケーション
制約事項
集約表 88
CCD 表 50
multi-tier レプリケーション 91
Update-anywhere 55, 94
ソースの登録 45

移行

計画 3
イベント、調整 237
イベントに基づくスケジューリング 81
イベント発行コマンド
asnslst 371
asntdiff 372
asntrep 382

印刷

アプライ・プログラム
メッセージ 204
キャプチャー・プログラム
印刷 203
モニター・プログラム
メッセージ 206
インターバル・タイミング 81

ウォーム・スタート、キャプチャー・プログラム

OS/400 版 469, 475
UNIX 版 138, 347
Windows 版 138, 347
z/OS 版 347
z/OS 用 138

エラー

アラート条件でモニター 173
操作の 196
レプリケーション
アラート条件、
APPLY_ERRORS 176
アラート条件、
CAPTURE_ERRORS 176
アラート条件、
QAPPLY_ERRORS 176
アラート条件、
QCAPTURE_ERRORS 176
monitor_errors パラメーター 190
SQL 176
「エラー発生時にキャプチャーを停止」オプション 53

延期

キャプチャー・プログラム
UNIX 版 144, 349
Windows 版 144, 349
z/OS 版 144, 349

オブジェクト

再活動化 215
属性の変更 210
登録 209
非活動化 214
変更のキャプチャーの停止 214

[力行]

カーソルからのロード関数 169

開始

アプライ・プログラム
OS/400 版 159
UNIX 版 149, 334
Windows 版 149, 334
z/OS 版 149, 334
キャプチャー・プログラム
Windows サービスの使用 489
レプリケーション・アラート・モニター
さまざまな方式 188

外部 CCD 表

multi-tier レプリケーション 91
カスタマイズ、SQL スクリプトの 123
カタログ表、登録 43
各国語サポート (NLS) 14
活動化、サブスクリプション・セットの 74, 286

- 環境変数
 - キャプチャー・プログラム 30
 - DB2CODEPAGE 14, 30
 - DB2DBDFT 30
 - DB2INSTANCE 30
 - LIBPATH 31
 - 監査
 - コールド・スタート 89
 - ソース・データ 52
 - データのギャップ 89
 - 管理
 - 許可要件 19
 - 基礎集約表
 - 構造 592
 - 使用 88
 - 定義 85
 - 既存の表、ターゲットとして 96
 - ギャップの検出 89
 - キャプチャー
 - 複数のデータベース・パーティション 36
 - 複数のデータベース・パーティションの使用 29
 - キャプチャー・エンキュー (CAPENQ) 表 516
 - キャプチャー・コントロール表のリスト 514
 - AUTHTKN (アプライ修飾子相互参照) 515
 - CAPENQ (キャプチャー・エンキュー) 516
 - CAPMON (キャプチャー・モニター) 517
 - CAPPARMS (キャプチャー・パラメーター)
 - 構造 519
 - 使用 127
 - 変更 142
 - CAPSCHEMAS (キャプチャー・スキーマ) 515
 - CAPTRACE (キャプチャー・トレース) 522
 - CCD (整合変更データ) 523
 - CD (変更データ) 525
 - PARTITIONINFO (パーティション情報) 526
 - PRUNCNTL (整理コントロール) 527
 - PRUNE_LOCK (整理ロック) 529
 - PRUNE_SET (整理セット) 529
 - REGISTER (登録) 532
 - REG_EXT (登録拡張) 530
 - REG_SYNCH (登録同期) 539
 - RESTART (再始動) 540
 - SEQTABLE (順序付け) 542
 - SIGNAL (シグナル) 543
 - UOW (作業単位) 546
- キャプチャー・コントロール・サーバー上のコントロール表 514
 - 複数のキャプチャー・スキーマ 28
 - レプリケーション・センターへの追加 278
 - キャプチャー・シグナル 237
 - キャプチャー・スキーマ
 - 使用、複数の 28
 - 変更 217
 - 命名規則 327
 - キャプチャー・スキーマ (CAPSCHEMAS) 表 515
 - キャプチャー・トリガー
 - お互いの通信
 - アプライ・プログラム 495, 498
 - レプリケーション・センター 495
 - 競合、既存のトリガーとの 13
 - 許可要件 23
 - 計画 12
 - 名前 13
 - キャプチャー・トレース (CAPTRACE) 表
 - 構造 522
 - 整理 258
 - キャプチャー・パラメーター (CAPPARMS) 表
 - 構造 519
 - 使用 127
 - 変更 142
 - キャプチャー・パラメーターの変更
 - OS/400 版 425
 - UNIX 版 349
 - Windows 版 349
 - z/OS 版 349
 - キャプチャー・プログラム
 - アラート条件 176
 - お互いの通信
 - アプライ・プログラム 495, 496
 - レプリケーション・アラート・モニター 499
 - レプリケーション・センター 495
 - 許可要件 21
 - コールド・スタートの防止 259
 - コネクティビティー 17
 - コマンド 329
 - シグナル 237
 - 実行、複数の 28
 - スキーマの変更 217
 - スケジューリング 493
 - スループット分析 203
 - 設定、環境変数 30
 - 設定、デフォルトの、パラメーターの場合 127
 - 操作 290
 - パフォーマンス・データ 201
 - 変更、パラメーター値の 127
 - キャプチャー・プログラム (続き)
 - 変更する、動作を、実行中における 141
 - 待ち時間分析 204
 - メッセージ 203
 - 印刷 203
 - メモリー、使用する 3
 - ユーザー ID 21
 - を始動する場所 133
 - OS/400 版
 - ウォーム・スタート・パラメーター 469
 - 許可要件 21
 - コールド・スタート、自動 475
 - コールド・スタート・パラメーター 469
 - 再初期化 446
 - 作成、SQL パッケージの 34, 35
 - 始動 140, 468
 - ジャーナルおよびジャーナル・レシーバーの管理 38
 - 状況のチェック 201
 - 進行 206
 - セットアップ 34, 36
 - 操作 125
 - 属性のオーバーライド 448
 - 属性の変更 425
 - 停止 143, 434
 - デフォルト・パラメーター 126, 127
 - CAPCTLLIB パラメーター 471
 - CLNUPITV パラメーター 471
 - FRCFRQ パラメーター 474
 - JOBID パラメーター 470
 - JRN パラメーター 472
 - LAG パラメーター 474
 - MEMLMT パラメーター 473
 - MONITV パラメーター 473
 - MONLMT パラメーター 472
 - RESTART パラメーター 469
 - RETAIN パラメーター 473
 - TRCLMT パラメーター 472
 - WAIT パラメーター 470
 - OS/400 用
 - ジャーナル項目タイプ 601
 - スケジューリング 494
 - UNIX 版
 - ウォーム・スタート・パラメーター 138, 347
 - 延期 144, 349
 - コールド・スタート・パラメーター 138, 347
 - 構成 31
 - 再開 144, 349
 - 再初期化 145, 349
 - 始動 129, 342, 605

キャプチャー・プログラム (続き)

UNIX 版 (続き)

状況 349
状況のチェック 199
整理 349
セットアップ 30
操作 125, 349
停止 143, 349
デフォルト・パラメーター 125
バインド 31
パラメーターの変更 349
add_partition パラメーター 131, 343
autoprunce パラメーター 131, 344, 351
autostop パラメーター 132, 344, 351
capture_path パラメーター 132, 343
capture_schema パラメーター 133, 343
capture_server パラメーター 133, 343
commit_interval パラメーター 134, 344, 351
lag_limit パラメーター 134, 344
logreusel パラメーター 134, 345, 351
logstdout パラメーター 135, 345, 352
memory_limit パラメーター 135, 345, 352
monitor_interval パラメーター 136, 345, 352
monitor_limit パラメーター 136, 345, 352
prune_interval パラメーター 136, 346, 352
pwdfile パラメーター 345
retention_limit パラメーター 137, 346, 352
sleep_interval パラメーター 138, 346, 352
startmode パラメーター 138, 347
term パラメーター 139, 348, 353
trace_limit パラメーター 139, 348, 353

Windows 版

ウォーム・スタート・パラメーター 138, 347
延期 144, 349
コールド・スタート・パラメーター 138, 347
構成 31
再開 144, 349
再初期化 145, 349

キャプチャー・プログラム (続き)

Windows 版 (続き)

始動 129, 342, 605
状況 349
状況のチェック 199
整理 349
セットアップ 30
操作 125, 349
停止 143, 349
デフォルト・パラメーター 125
バインド 31
パラメーターの変更 349
add_partition パラメーター 131, 343
autoprunce パラメーター 131, 344, 351
autostop パラメーター 132, 344, 351
capture_path パラメーター 132, 343
capture_schema パラメーター 133, 343
capture_server パラメーター 133, 343
commit_interval パラメーター 134, 344, 351
lag_limit パラメーター 134, 344
logreusel パラメーター 134, 345, 351
logstdout パラメーター 135, 345, 352
memory_limit パラメーター 135, 345, 352
monitor_interval パラメーター 136, 345, 352
monitor_limit パラメーター 136, 345, 352
prune_interval パラメーター 136, 346, 352
pwdfile パラメーター 345
retention_limit パラメーター 137, 346, 352
sleep_interval パラメーター 138, 346, 352
startmode パラメーター 138, 347
term パラメーター 139, 348, 353
trace_limit パラメーター 139, 348, 353

z/OS 版

ウォーム・スタート・パラメーター 138, 347
延期 144, 349
開始 342
コールド・スタート・パラメーター 138, 347
再開 144, 349

キャプチャー・プログラム (続き)

z/OS 版 (続き)

再初期化 145, 349
始動 129
状況 349
状況のチェック 199
整理 349
セットアップ 36
操作 125, 349
停止 143, 349
デフォルト・パラメーター 125
パラメーターの変更 349
add_partition パラメーター 131, 343
autoprunce パラメーター 131, 344, 351
autostop パラメーター 132, 344, 351
capture_path パラメーター 132, 343
capture_schema パラメーター 133, 343
capture_server パラメーター 133, 343
commit_interval パラメーター 134, 344, 351
lag_limit パラメーター 134, 344
logreusel パラメーター 134, 345, 351
logstdout パラメーター 135
memory_limit パラメーター 135, 345, 352
monitor_interval パラメーター 136, 345, 352
monitor_limit パラメーター 136, 345, 352
prune_interval パラメーター 136, 346, 352
pwdfile パラメーター 345
retention_limit パラメーター 137, 346, 352
sleep_interval パラメーター 138, 346, 352
startmode パラメーター 138, 347
term パラメーター 139, 348, 353
trace_limit パラメーター 139, 348, 353

キャプチャー・モニター (CAPMON) 表

構造 517

整理 258

キャプチャー・ログ・ファイル 134

行

サブセット化

ソースでの 49

ターゲットでの 98

使用可能、レプリケーションに 49

行 (続き)
 定義、ターゲット表での 98
 登録、ソース表内の 49
 行 (水平方向) のサブセット化
 ソースでの 49
 ターゲットでの 98
 行キャプチャー規則 49
 競合
 回避 11
 競合検出
 概要 59
 計画 11
 対等レプリケーション 11
 要件 52
 レベル 60
 Update-anywhere レプリケーション
 11
 許可
 アプライ・プログラム 23
 管理のための 19, 21
 キャプチャー・トリガーの場合 23
 キャプチャー・プログラムの場合 21
 レプリケーション・アラート・モニター
 用 25
 空間データ・タイプ 105
 区切り文字、生成済み SQL スクリプト内
 の 124
 グローバル・レコード 533
 計画
 移行 3
 競合検出 11, 59
 共存、トリガーの 13
 ストレージ要件 6
 トランザクション・スループット率
 12
 メモリー 3
 ログの影響 6, 12
 ロック、CCD 表に対する 13
 結合、ソースとしての 63
 現行レシーバーのサイズ 7, 38
 コード・ページ
 互換性のある 13
 トランザクション 13
 DB2CODEPAGE 環境変数 14
 コールド・スタート、キャプチャー・プロ
 グラム
 回避 259
 OS/400 版 469, 475
 UNIX 版 138, 347
 Windows 版 138, 347
 z/OS 版 347
 z/OS 用 138
 更新
 競合 59
 削除および挿入として 54
 更新済み主キー列 54

構成
 アプライ・プログラム
 UNIX 版 32
 Windows 版 32
 キャプチャー・プログラム
 UNIX 版 31
 Windows 版 31
 コネクティビティー 17
 レプリケーション・アラート・モニター
 UNIX 版 33
 Windows 版 33
 レプリケーション・センター 267
 構成パラメーター、DB2 用
 APPLHEAPSZ 31
 DBHEAP 31
 LOGBUFSZ 31
 LOGFILSIZ 31
 LOGPRIMARY 31
 LOGSECOND 31
 MAXAPPLS 31
 コネクティビティー
 コントロール表の障害のリカバリー
 260
 DB2 オペレーティング・システム間
 17, 18
 固有のデータ・タイプ 106
 コントロール表
 アプライ
 作成 276
 アプライ・コントロール・サーバー
 511
 アプライ・コントロール・サーバー上
 の 548
 キャプチャー
 作成 275
 キャプチャー・コントロール・サーバ
 ー上の 514
 キャプチャー・サーバー 508
 許可要件、OS/400 の場合 37
 クイック・リファレンス
 アプライ・コントロール・サーバー
 511
 概観 501
 キャプチャー・サーバー 508
 ターゲット・サーバー 514
 再バインド、パッケージおよびプラン
 253
 再編成 254
 作成
 アプライ用 276
 キャプチャー用 275
 非 DB2 リレーショナル・ソースの
 場合 28
 複数のセット 28

コントロール表 (続き)
 作成 (続き)
 複数のデータベース・オペレーティ
 ング・システム 27
 複数のデータベース・パーティショ
 ン 29
 レプリケーション・アラート・モニ
 ター 183
 レプリケーション・アラート・モニ
 ター用 277
 IASP グループ内 27
 Linux、UNIX および Windows で
 の 26
 OS/400 での 27, 430
 z/OS での 27
 ストレージ要件 8
 静的 254
 整理 255
 接続障害のリカバリー 260
 ターゲット・サーバー 514
 動的 252
 入出力エラーのリカバリー 260
 付与、権限の、OS/400 の場合 21
 プロファイル 272
 保守 252
 モニター
 作成 277
 モニター・コントロール・サーバー
 IBMSNAP_ALERTS 576
 IBMSNAP_CONDITIONS 578
 IBMSNAP_CONTACTGRP 584
 IBMSNAP_CONTACTS 585
 IBMSNAP_GROUPS 586
 IBMSNAP_MONENQ 586
 IBMSNAP_MONPARMS 586
 IBMSNAP_MONSERVERS 588
 IBMSNAP_MONTRAIL 591
 モニター・コントロール・サーバー上
 の 576
 APPENQ (アプライ・エンキュー
 ー) 548
 APPLYTRACE (アプライ・トレー
 ス) 554
 APPLYTRAIL (アプライ・トレー
 ル) 554
 APPLY_JOB (アプライ・ジョブ) 549
 APPPARMS (アプライ・パラメータ
 ー) 550
 AUTHTKN (アプライ修飾子相互参
 照) 515
 CAPENQ (キャプチャー・エンキュー
 ー) 516
 CAPMON (キャプチャー・モニター)
 構造 517
 整理 258

コントロール表 (続き)
 CAPPARMS (キャプチャー・パラメータ)
 構造 519
 CAPSCHEMAS (キャプチャー・スキーマ) 515
 CAPTRACE (キャプチャー・トレース)
 構造 522
 整理 258
 CCD (整合変更データ)
 キャプチャー・コントロール・サーバー 523
 ターゲット・サーバー 594
 CD (変更データ) 525
 OS/400 の権限の取り消し 458
 OS/400 の権限の付与 437
 PARTITIONINFO (パーティション情報) 526
 PRUNCNTL (整理コントロール) 527
 PRUNE_LOCK (整理ロック) 529
 PRUNE_SET (整理セット) 529
 REGISTER (登録) 532
 REG_EXT (登録拡張) 530
 REG_SYNCH (登録同期) 539
 RESTART (再始動) 540
 RUNSTATS ユーティリティ 253
 SEQTABLE (順序付け) 542
 SIGNAL (シグナル) 543
 SUBS_COLS (サブスクリプション列) 560
 SUBS_EVENT (サブスクリプション・イベント) 562
 SUBS_MEMBR (サブスクリプション・メンバー) 563
 SUBS_SET (サブスクリプション・セット) 568
 SUBS_STMTS (サブスクリプション・ステートメント) 574
 UOW (作業単位) 546
 コントロール・サーバー、レプリケーション・センターへの追加 278
 コンプレッション・ディクショナリー (z/OS) 251

[サ行]

サーバー
 レプリケーション・センターへの追加 278
 サービス
 Windows SCM 489
 再開
 キャプチャー・プログラム
 UNIX 版 144, 349
 Windows 版 144, 349
 z/OS 版 144, 349

再活動化
 オブジェクト 215
 登録 215
 表 215
 再キャプチャー、変更の (Update-anywhere) 55
 再始動 (RESTART) 表 540
 再初期化
 レプリケーション・アラート・モニター 189
 再初期化、キャプチャー・プログラムの UNIX 版 145
 Windows 版 145
 z/OS 版 145
 再バインド、パッケージおよびプラン 253
 再編成
 コントロール表 254
 作業単位 (UOW) 表
 構造 546
 ストレージ要件 9
 整理 257, 546
 列、CCD 表の 90
 索引
 ターゲット表 100
 作成
 コントロール表
 レプリケーション・アラート・モニター 183
 モニター 186
 作成、コントロール表の 26
 サブスクリプション列 (SUBS_COLS) 表 560
 サブスクリプション・イベント (SUBS_EVENT) 表
 構造 562
 通知、イベントの 81
 サブスクリプション・サイクル 75
 サブスクリプション・ステートメント (SUBS_STMTS) 表 574
 サブスクリプション・セット
 数、アプライ修飾子の 71
 活動化 286
 活動化レベル 74
 行 98
 削除 236, 454
 作成 72, 220, 282
 参照保全 96
 処理モード 78
 スケジューリング
 イベントに基づく 81
 時間に基づく 81
 ストアード・プロシージャ 79
 追加 395
 データ整合性 96
 非活動化 235, 286

サブスクリプション・セット (続き)
 プロモート 288
 分割 224
 変更
 アプライ修飾子 232
 属性 222
 名前 222
 マージ 229
 ミニサイクル 75
 メンバーの使用可能化 221
 メンバーの追加 82, 220
 メンバーを使用不可にする 221
 ランタイム処理ステートメント 120
 列 97
 multi-tier レプリケーション 91
 SQL ステートメント 79
 Update-anywhere レプリケーション 94
 サブスクリプション・セット (SUBS_SET) 表 568
 サブスクリプション・セットの作成 282
 サブスクリプション・セット・メンバー数、サブスクリプション・セットごとの 70
 削除 456
 使用可能化 221
 使用不可 221
 選択、ターゲット・タイプの 85
 追加 82, 220, 411
 定義、ターゲット・キーの 100
 適用、行のサブセット 98
 適用、列のサブセット 97
 マッピング、データ・タイプの 99
 マッピング、列間の 99
 multi-tier レプリケーション 91
 Update-anywhere レプリケーション 94
 サブスクリプション・メンバー (SUBS_MEMBR) 表 168, 563
 サブセット化
 行、ターゲットでの変更の 98
 高度な技法
 使用、述部の 117
 使用、ビューの 116
 使用する、トリガーを、CD 表で 116
 登録時の 115
 登録済みの行、変更を含む 49
 登録済み列の 47
 列、ターゲットでの 97
 差分リフレッシュ・レプリケーション
 参照: 変更キャプチャー・レプリケーション
 算出列 97
 作成 121
 ソース表 88

- 算出列 (続き)
 - CD 表 89
- 参照保全 96
- 時間に基づくスケジューリング 81
- シグナル
 - 分散リカバリ点の設定 241
 - CAPSTART 242
 - CAPSTOP 244
 - STOP 239, 241
 - USER 237
- シグナル (SIGNAL) 表
 - 構造 543
 - 整理 258
- システム変更ジャーナル管理 38
- システム・コマンド
 - asnslist 371
 - asntdiff 372
 - asntrep 382
- 実行、SQL スクリプトの 123
- 実行管理機能オブジェクト 39
- 始動
 - アプライ・プログラム
 - OS/400 版 460
 - UNIX 版 605
 - Windows 版 605
 - キャプチャー・プログラム
 - OS/400 版 140, 468
 - UNIX 版 129, 342, 605
 - Windows 版 129, 342, 605
 - z/OS 版 129, 342
 - レプリケーション・アラート・モニター
 - UNIX 版 605
 - Windows 版 605
- 自動整理 256
- シナリオ
 - アプライ・コントロール表の作成 302
 - アプライ・パスワード・ファイルの作成 309
 - アプライ・プログラムのアラート条件の選択 320
 - アプライ・プログラムの状況 314
 - キャプチャー・コントロール表の作成 298
 - キャプチャー・プログラムとアプライ・プログラムの停止 315
 - キャプチャー・プログラムのアラート条件の選択 318
 - キャプチャー・プログラムの状況 313
 - 計画 296
 - サブスクリプション・セットの作成 303
 - セットアップ 298
 - 前提条件 295
 - ソースの登録 299
 - ソース表の更新 312
- シナリオ (続き)
 - ソース・データベースのレプリケーションを使用可能にする 299
 - 操作 312
 - データを複製する 310
 - モニター・コントロール表の作成 316
 - レプリケーションのモニター 316
 - レプリケーション・アラート・モニターの開始 321
 - 連絡先の作成 318
- ジャーナル
 - 開始 38
 - 管理 38
 - 項目タイプ 601
 - 作成 37
 - 作成、ソース表の 37
 - 使用 36
 - 使用、リモート・ジャーナル関数 61
 - セットアップ 36
 - デフォルトのメッセージ・キュー 40
 - 登録、ソースとして 43
 - QSQRN ジャーナル 36
- ジャーナル・シグナル表 238
 - 作成 238
 - 停止 242
 - CAPSTOP 244
- ジャーナル・ジョブ
 - 状況のチェック 201
- ジャーナル・メッセージ・キュー 40
- ジャーナル・レシーバー
 - 管理 38
 - 現行、サイズ 7
 - 作成、ソース表の 37
 - しきい値 38
 - システム管理 38
 - ジャーナル・レシーバー削除出口ルーチン 40
 - 保守 248
 - 保存 251
 - ユーザー管理 39
- ジャーナル・レシーバー削除出口ルーチン
 - 削除 40
 - 説明 40
 - 登録 41
- 集約表
 - 基礎集約 88, 592
 - 変更集約 89, 593
- 終了文字、生成済み SQL スクリプト内の 124
- 主キー
 - 使用される、ターゲットとして 101
 - 相対レコード番号、OS/400 の場合 62
 - 論理パーティション化 54
- 述部
 - サブセット化 117
- 述部 (続き)
 - 定義、ターゲット表の 98
- 順序付け (SEQTABLE) 表 542
- 診断ファイル
 - ストレージ 9, 10
- 垂直方向 (列) のサブセット化
 - ソースでの 47
 - ターゲットでの 97
- 水平方向 (行) のサブセット化
 - ソースでの 49
 - ターゲットでの 98
- スキーマ
 - 変更 217
 - 命名規則 327
- スケジューリング
 - サブスクリプション・セット 80, 81
 - レプリケーション・プログラム 493
- ステー징、データの 91
- ステーjingされたレプリケーション 92
- ストアード・プロシージャ
 - 操作、データの 120
 - 定義、サブスクリプション・セットの 79
- ストレージ
 - アプライ診断ファイル 10
 - アプライ・プログラムの予備ファイル 10
 - 一時ファイル 9
 - キャプチャー診断ファイル 10
 - キャプチャー予備ファイル 10
 - コントロール表 8
 - 診断ファイル 9
 - ターゲット表 8
 - データベース・ログおよびジャーナル・データ 6
 - 要件 6
 - CD 表 9
 - UOW 表 9
- スループット
 - アプライ・プログラム 205
 - キャプチャー・プログラム 203
- スループット率
 - キャプチャー・トリガー 12
- 整合変更データ (CCD) 表
 - 外部
 - multi-tier レプリケーション 91
 - 構造
 - キャプチャー・コントロール・サーバー 523
 - ターゲット・サーバー 594
 - 使用
 - 履歴または監査 89
 - multi-tier レプリケーション 91
 - 追加、UOW 列の 90
 - 内部
 - 複数のターゲット 90

整合変更データ (CCD) 表 (続き)

非 DB2 リレーショナル・データ・ソース

使用、CCD 表の 45

非リレーショナル・データ・ソース

使用、CCD 表の 43

CCD 表の保守 66

レプリケーション・ソース 91

ロック 13

生成済み SQL スクリプト 123

静的なコントロール表 254

制約事項

異機種レプリケーション 50, 91, 94

既存のターゲット表 96

空間データ・タイプ 105

固有のデータ・タイプ 106

ストアド・プロシージャ 121

データ・タイプ 105

非 DB2 リレーショナル・データ・ソース 55, 59

ビュー 65

ユーザー定義のデータ・タイプ 106

ユニコード表 599

要約データ・タイプ 105

列名の制限 51

ASCII 表 599

CCD 表 94

DATALINK 値 59, 94

DB2 エクステンダー・ラージ・オブジェクト 106

EDITPROC 文節 105

FIELDPROC 文節 105

LOB データ・タイプ 94

LONG VARCHAR データ・タイプ 105

LONG VARGRAPHIC データ・タイプ 105

Microsoft SQL Server 50

Oracle ソース 105

Oracle 表内の LONG 列 105

Sybase 50

VALIDPROC 文節 105

WHERE 文節 98

整理

アプライ・トレース (APPLYTRACE) 表 258

アプライ・トレール (APPLYTRAIL) 表 258

キャプチャー・トレース (CAPTRACE) 表 258

キャプチャー・プログラム

UNIX 版 349

Windows 版 349

z/OS 版 349

キャプチャー・モニター (CAPMON) 表 258

整理 (続き)

コントロール表 255

シグナル (SIGNAL) 表 258

CD (変更データ) 表 257

UOW (作業単位) 表 257, 546

整理インターバル

レプリケーション・アラート・モニター 194

整理コントロール (PRUNCNTL) 表 527

整理セット (PRUNE_SET) 表 529

整理ロック (PRUNE_LOCK) 表 529

接続

iSeries サーバーへの 18

z/OS サーバーへの 18

設定、環境変数

キャプチャー・プログラム 30

セットアップ

アプライ・プログラム

OS/400 版 34

UNIX 版 30

Windows 版 30

キャプチャー・プログラム

OS/400 版 34

UNIX 版 30

Windows 版 30

ジャーナル 36

レプリケーション・アラート・モニター 33, 182

接頭部、変更前イメージ 52

ソース

サブスクライブ 72

登録

非 DB2 リレーショナル 45

ビュー 62, 65

レプリケーション・センター 281

DB2 表 43

IMS データ・ソース 43

登録、行の 49

登録、列の 47

登録オプション

エラー発生時にキャプチャーを停止 53

行 (水平方向) のサブセット化 49

競合検出 59

更新、削除および挿入として 54

使用、リモート・ジャーナル 61

相対レコード番号 62

フル・リフレッシュ・コピー 48

変更キャプチャー・レプリケーション 48

変更後イメージ列 50

変更前イメージ接頭部 52

変更前イメージ列 50

変更の再キャプチャー

(Update-anywhere) 55

列 (垂直方向) のサブセット化 47

ソース (続き)

プロファイル 273

プロモート 287

マッピング、ターゲットへの 82

CCD (整合変更データ) 表 91

CCD 表の保守 66

ソースのサブスクライブ 72

ソースの登録 281

ソース表

作成、ジャーナルの 37

脱落したデータのリトリート 260

保守 247

列の追加 211

ソース・サーバー

非 DB2 リレーショナル

ログの影響 12

DB2

ログの影響 6

ソース・システム、保守 247

ソース・ログ、保守 248

関連 ID 63

操作

アプライ・プログラム 291, 330

キャプチャー・プログラム 290, 349

レプリケーション・アラート・モニター 185, 292

操作、データの

作成、算出列の 121

サブスクリプションでの 120

登録での 119

名前変更、列の 99, 121

相対タイミング 81

相対レコード番号

サポート、OS/400 に対する 62

主キーとしての、OS/400 の場合 62

使用される、ターゲットとして 101

属性

サブスクリプション・セットの変更 222

登録済みオブジェクトの変更 210

属性のオーバーライド (OS/400)

キャプチャー・プログラム 448

[タ行]

ターゲット

フル・リフレッシュの強制 289

プロファイル 274

ターゲット索引 100

ターゲット表

新しい列 121

基礎集約

構造 592

使用 88

定義 85

ストレージ要件 8

- ターゲット表 (続き)
 - 定義、行の 98
 - 定義、ターゲット・キーの 100
 - 定義、列の 97
 - 適用、行のサブセット 98
 - 適用、列のサブセット 97
 - のリスト 592
 - 表の構造、クイック・リファレンス 514
 - フラグメント化 97, 98
 - 変更集約
 - 構造 593
 - 使用 89
 - 定義 85
 - ポイント・イン・タイム
 - 構造 596
 - 使用 88
 - 定義 85
 - 保守 261
 - マッピング、ソースへの 82
 - ユーザー定義 87, 96
 - ユーザー・コピー
 - 構造 597
 - 使用 88
 - 定義 85
 - レプリカ
 - 競合検出 11
 - 構造 597
 - 使用 94
 - 定義 85
 - CCD (整合変更データ)
 - 概要 85
 - 構造 594
 - ターゲット・キー 100
 - ターゲット・キー列
 - 更新 101
 - ターゲット・サーバー
 - 上の表 592
 - ログの影響 7
 - 対等レプリケーション
 - 競合検出 11
 - 大量のレプリケーション・ジョブ 75
 - チューニング
 - パフォーマンス 16
 - commit_interval パラメーター 4
 - memory_limit パラメーター 4
 - データ
 - 高度なサブセット化技法 115
 - サブセット化
 - 使用、述部の 117
 - 使用、ビューの 116
 - 使用する、トリガーを、CD 表で 116
 - 使用する、ビューを、述部を指定するするために 117
 - 登録時の 115
- データ (続き)
 - ソース表からのリトリート 260
 - 操作 119
 - トランスフォーム
 - 作成、算出列の 121
 - サブスクリプションでの 120
 - 登録での 119
 - 名前変更、列の 99, 121
 - 防止、二重削除の 64
 - 履歴の表示 201
 - データ整合性 96
 - データベース、変更キャプチャーの使用可能化 280
 - データ・タイプ
 - 制約事項 105
 - マッピング、列間の 99
 - レプリケーション
 - ラージ・オブジェクト (LOB) 106
 - DATALINK 値 107
 - データ・ブロッキング 75
 - データ・リンク
 - レプリケーション 107
 - 停止
 - アプライ・プログラム
 - OS/400 版 161, 431
 - UNIX 版 161, 330
 - Windows 版 161, 330
 - z/OS 版 161, 330
 - キャプチャー・プログラム
 - OS/400 版 143, 434
 - UNIX 版 143, 349
 - Windows 版 143, 349
 - z/OS 版 143, 349
 - レプリケーション・アラート・モニター 196
 - ディスク・スペース
 - 一時ファイル 9
 - 要件 6
 - 出口ルーチン
 - ジャーナル・レシーバー削除 (OS/400) 40
 - ASNDLCOPY 109
 - ASNDONE
 - 使用 161, 162
 - ASNLOAD
 - カスタマイズ 168
 - 使用 164
 - OS/400 用 170
 - UNIX 版 165
 - Windows 版 165
 - z/OS 版 167
 - デフォルト
 - アプライ・パラメーターの (Linux, UNIX, Windows, z/OS) 147, 150
- デフォルト (続き)
 - アプライ・プログラムのパラメーターの (OS/400) 160
 - キャプチャー・パラメーターの (Linux, UNIX, Windows, z/OS) 125
 - キャプチャー・パラメーターの (OS/400) 126, 127
 - キャプチャー・パラメーターの (UNIX, Windows, z/OS) 130
 - 動的なコントロール表 252
 - 登録
 - オブジェクト 209
 - オプション、ソース用の
 - エラー発生時にキャプチャーを停止 53
 - 行 (水平方向) のサブセット化 49
 - 競合検出 59
 - 更新、削除および挿入として 54
 - 使用、リモート・ジャーナル 61
 - 相対レコード番号 62
 - フル・リフレッシュ・コピー 48
 - 変更キャプチャー・レプリケーション 48
 - 変更後イメージ列 50
 - 変更前イメージ接頭部 52
 - 変更前イメージ列 50
 - 変更の再キャプチャー (Update-anywhere) 55
 - 列 (垂直方向) のサブセット化 47
 - 再活動化 215
 - 削除 216, 453
 - 属性の変更 210
 - 追加 386
 - 非 DB2 リレーショナル・データ・ソース 45
 - 非活動化 214
 - ビュー
 - 概要 62, 65
 - 手順 209
 - 表 209
 - 変更のキャプチャーの停止 214
 - 列の追加 211
 - DB2 表 43
 - IMS データ・ソース 43
 - 登録 (REGISTER) 表 532
 - 登録拡張 (REG_EXT) 表 530
 - 登録同期 (REG_SYNCH) 表 539
 - 登録変数
 - DB2CODEPAGE 14, 30
 - DB2DBDFT 30
 - DB2INSTANCE 30
 - 特殊なデータ・タイプ
 - レプリケーション
 - ラージ・オブジェクト (LOB) 106
 - DATALINK 値 107

独立補助記憶域プール (IASP) グループ
27
 トラブルシューティング・コマンド
 WRKDPTRRC 476
 トランザクション
 メモリー、使用する 3
 トランザクション・スルーット率
 キャプチャー・トリガー 12
 トランザクション・モード処理 7, 78
 トランスフォーム、データの
 作成、算出列の 121
 サブスクリプションでの 120
 登録での 119
 名前変更、列の 99, 121
 トリガー
 キャプチャー、データの 12
 マージ 13
 抑制、データのキャプチャーの 116
 CD 表の 116
 トレース機能
 OS/400 版 476

[ナ行]

内部 CCD 表
 複数のターゲット 90
 内部結合、ソースとしての 63
 名前
 アプライ修飾子の規則 327
 キャプチャー・スキーマの規則 327
 キャプチャー・トリガーの 13
 サブスクリプション・セット 222
 表示名 490
 モニター修飾子の規則 327
 レプリケーション・サービスの 490
 Windows サービス用 328
 名前変更、列の 99, 121
 二重削除 64
 ニックネーム
 カーソルからのロード関数の 169
 制約事項
 集約表 88
 CCD 表の場合 50
 multi-tier レプリケーション 91
 Update-anywhere 55, 94
 登録 45
 入出力エラーのリカバリー、コントロール
 表 260
 認証、エンド・ユーザー
 UNIX 版 17, 26
 Windows 版 17, 26
 ネットワーク・コネクティビティー 17

[ハ行]

パーティション情報 (PARTITIONINFO)
 表 526
 バイナリー・ラージ・オブジェクト
 (BLOB)
 レプリケーションの考慮事項 106
 バインド
 アプライ・プログラム
 UNIX 版 32
 Windows 版 32
 z/OS 版 36
 キャプチャー・プログラム
 UNIX 版 31
 Windows 版 31
 z/OS 版 36
 レプリケーション・アラート・モニタ
 ー
 UNIX 版 33
 Windows 版 33
 パスワード・ファイル
 保管 26
 パッケージ、再バインド 253
 バッチ・ジョブ
 メモリー、使用する 3
 パフォーマンス
 チューニング 16
 パラメーター
 レプリケーション・アラート・モニタ
 ー
 記述 190
 デフォルト値 190
 alert_prune_limit 190
 autoprune 190
 email_server 190
 max_notifications_per_alert 190
 max_notification_minutes 190
 monitor_errors 190
 monitor_limit 190
 monitor_path 190
 runonce 190
 trace_limit 190
 パラメーター、呼び出し
 アナライザー
 OS/400 版 422
 UNIX 版 332
 Windows 版 332
 アプライ・プログラム
 OS/400 版 159, 460
 UNIX 版 150, 335
 Windows 版 150, 335
 z/OS 版 150, 335
 キャプチャー・プログラム
 OS/400 版 426, 469
 UNIX 版 130, 343
 Windows 版 130, 343

パラメーター、呼び出し (続き)
 キャプチャー・プログラム (続き)
 z/OS 版 130, 343
 レプリケーション・アラート・モニタ
 ー
 UNIX 版 358
 Windows 版 358
 z/OS 版 358
 レプリケーション・コマンド
 OS/400 版 387, 397, 412, 430,
 432, 434, 437, 447, 448, 453, 454,
 456, 458, 460, 469, 477
 非 DB2 リレーショナル・データ・ソース
 使用、CCD 表の 45
 制約事項
 集約表 88
 multi-tier レプリケーション 91
 Update-anywhere 55, 59, 94
 ソース・サーバー 12
 登録 45
 ロック 13
 非アクティブなサブスクリプション・セッ
 ト 74
 非活動化
 サブスクリプション・セット 74, 235
 登録済みオブジェクト 214
 非活動化、サブスクリプション・セットの
 286
 ビュー
 使用、相関 ID の 63
 制約事項 62, 65
 属性の変更 210
 登録
 概要 62
 ソースとして 65
 手順 209
 表
 アプライ・コントロール・サーバー上
 の 548
 基礎集約 592
 キャプチャー・コントロール・サーバ
 ー上の 514
 競合検出 11
 構造 501
 コントロール表
 再編成 254
 作成 26
 静的 254
 整理 255
 接続障害のリカバリー 260
 動的 252
 入出力エラーのリカバリー 260
 保守 252
 RUNSTATS ユーティリティ
 253
 再活動化 215

表 (続き)

属性の変更 210
 ターゲット表
 保守 261
 参照: ターゲット表
 ターゲット・サーバー上の 592
 登録
 手順 209
 非 DB2 リレーショナル 45
 DB2 43
 登録の除去 216
 非活動化 214
 変更集約 593
 変更のキャプチャーの停止 214
 ポイント・イン・タイム 596
 モニター・コントロール・サーバー上
 の 576
 ユーザー・コピー 597
 列の追加 211
 レプリカ 11, 597
 APPENQ (アプライ・エンキュー
) 548
 APPLYTRACE (アプライ・トレー
 ス) 554
 APPLYTRAIL (アプライ・トレー
 ル) 554
 APPLY_JOB (アプライ・ジョブ) 549
 APPPARMS (アプライ・パラメータ
) 550
 AUTHTKN (アプライ修飾子相互参
 照) 515
 CAPENQ (キャプチャー・エンキュー
) 516
 CAPMON (キャプチャー・モニタ
) 258, 517
 CAPPARMS (キャプチャー・パラメー
 ター) 519
 CAPSCHEMAS (キャプチャー・スキマ
) 515
 CAPTRACE (キャプチャー・トレー
 ス) 258, 522
 CCD (整合変更データ)
 キャプチャー・コントロール・サー
 バー 523
 ターゲット・サーバー 594
 CCD 表の保守 66
 CD (変更データ) 525
 IBMSNAP_ALERTS 576
 IBMSNAP_CONDITIONS 578
 IBMSNAP_CONTACTGRP 584
 IBMSNAP_CONTACTS 585
 IBMSNAP_GROUPS 586
 IBMSNAP_MONENQ 586
 IBMSNAP_MONPARMS 586
 IBMSNAP_MONSERVICES 588
 IBMSNAP_MONTRAIL 591

表 (続き)

PARTITIONINFO (パーティション情
 報) 526
 PRUNCNTL (整理コントロール) 527
 PRUNE_LOCK (整理ロック) 529
 PRUNE_SET (整理セット) 529
 REGISTER (登録) 532
 REG_EXT (登録拡張) 530
 REG_SYNCH (登録同期) 539
 RESTART (再始動) 540
 SEQTABLE (順序付け) 542
 SIGNAL (シグナル) 543
 SUBS_COLS (サブスクリプション
 列) 560
 SUBS_EVENT (サブスクリプション・
 イベント) 562
 SUBS_MEMBR (サブスクリプショ
 ン・メンバー) 168, 563
 SUBS_SET (サブスクリプション・セ
 ット) 568
 SUBS_STMTS (サブスクリプション・
 ステートメント) 574
 UOW (作業単位) 546
 表構造 501
 表示名 490
 表修復ユーティリティ 382
 表相違検出ユーティリティ 372
 表モード処理 7, 78
 非リレーショナル・データ・ソース
 使用、CCD 表の 43
 CCD 表の保守 66
 ヒント
 アプライ・トレール表からの行の削除
 159
 アプライ・プログラムがセットを正常
 に処理したかのチェック 158
 検査する、変更のキャプチャーが開始
 されたことを 129
 サービスが正しくセットアップされた
 ことの検証 489
 スリープを使用するか copyonce パラ
 メーターを使用するか 152, 157
 セットの追加処理にストアード・プロ
 シージャーを使用 162
 見積もる、使用を、スペースの 6
 ASNDONE でストアード・プロシージ
 ャーを使用 163
 ファイル
 予備 9
 asndone.smp 162
 asnload.ini 169
 *.APP.log 154
 *.CAP.log 134
 *.err 157
 *.sqs 157, 158

ファイル・コピー・デーモン

ASNDLCOPYD 112
 DLFM_ASNLCOPYD 111
 複数のターゲット表 90
 複数のデータベース・パーティション
 キャプチャー 36
 フラグメント化
 垂直
 ソースでの 47
 ターゲットでの 97
 水平
 ソースでの 49
 ターゲットでの 98
 対等レプリケーション 11
 Update-anywhere レプリケーション
 11
 プラン、再バインド 253
 フル・リフレッシュ・コピー
 強制 289
 登録オプション 48
 Apply for iSeries 62, 463
 ブロッキング因数 75
 プロファイル
 記述 272
 コントロール表 272
 ソース・オブジェクト 273
 ターゲット・オブジェクト 274
 プロモート
 サブスクリプション・セット 288
 登録済みの表またはビュー 287
 レプリケーション構成 245
 分割
 サブスクリプション・セット 224
 分散リカバリー点 241
 変換、データの 14
 変更キャプチャー
 使用可能化 280
 変更キャプチャー・レプリケーション
 説明 49
 登録オプション 48
 変更後イメージ列 50
 変更集約表
 構造 593
 使用 89
 定義 85
 変更前イメージ接頭部 52
 変更前イメージ列
 制約事項 51
 登録 50
 変更集約表 98
 変更データ (CD) 表
 構造 525
 ストレージ要件 9
 整理 257
 要約、内容の 89
 変更のキャプチャーの停止 214

編集、SQL スクリプトの 123
ポイント・イン・タイム表
構造 596
使用 88

[マ行]

マージ
サブスクリプション・セット 229
トリガー 13
マスター表 (Update-anywhere)
概要 94
再キャプチャー、変更の 55
待ち時間
アプライ・プログラム 205
キャプチャー・プログラム 204
マッピング
ソースからターゲット 82
ソース列からターゲット列への 99
データ・タイプ、表間の 99
マルチ・データベース・パーティション
ログ・レコード 249
ミニサイクル 75
メッセージ 203, 204, 206
メッセージ・キュー、ジャーナル用 40
メモリー
アプライ・プログラム 5
アラート条件
APPLY_MEMORY 176
CAPTURE_MEMORY 176
QAPPLY_MEMORY 176
QCAPTURE_MEMORY 176
キャプチャー・プログラム 3
計画 3
サブスクリプション・セット 5
使用、CAPMON 表、調整のために 4
登録 4
トランザクション 3
バッチ・ジョブ 3
読み取り、ログ・レコードの 4
レプリケーション・アラート・モニター
5
文字ラージ・オブジェクト (CLOB)
レプリケーションの考慮事項 106
モニター
プログラムの状況 201
履歴の傾向 201
レプリケーション 173, 185
OS/400 版 206
参照：レプリケーション・アラート・
モニター
モニター修飾子
レプリケーション 173
モニター修飾子、命名規則 327
モニター・コントロール表
のリスト 576

モニター・コントロール・サーバー
上のコントロール表 576
レプリケーション・センターへの追加
278
IBMSNAP_ALERTS コントロール表
576
IBMSNAP_CONDITIONS コントロー
ル表 578
IBMSNAP_CONTACTGRP コントロー
ル表 584
IBMSNAP_CONTACTS コントロー
ル表 585, 586
IBMSNAP_MONENQ コントロール表
586
IBMSNAP_MONPARMS コントロール
表 586
IBMSNAP_MONSERVERS コントロー
ル表 588
IBMSNAP_MONTRAIL コントロール
表 591
モニター・プログラム
メッセージ 206
印刷 206

[ヤ行]

ユーザー ID
アプライ・プログラム 23
キャプチャー・トリガーの場合 23
キャプチャー・プログラムの場合 21
許可 22
パスワード・ファイル 26
レプリケーション・アラート・モニタ
ー用 25
ユーザー定義のデータ・タイプ 106
ユーザー定義表 87, 96
ユーザー・コピー表
構造 597
使用 88
定義 85
ユーティリティ
表修復 382
表相違検出 372
ユニコード表 599
要約データ・タイプ 105
呼び出しパラメーター
アナライザー
OS/400 版 422
UNIX 版 332
Windows 版 332
アプライ・プログラム
OS/400 版 159, 460
UNIX 版 150, 335
Windows 版 150, 335
z/OS 版 335
z/OS 用 150

呼び出しパラメーター (続き)
キャプチャー・プログラム
OS/400 版 125, 140, 426, 469
UNIX 版 130, 343
Windows 版 130, 343
z/OS 版 343
z/OS 用 130
レプリケーション・アラート・モニタ
ー
UNIX 版 358
Windows 版 358
z/OS 版 358
レプリケーション・コマンド
OS/400 版 387, 397, 412, 430,
432, 434, 437, 447, 448, 453, 454,
456, 458, 460, 469, 477
予備ファイル
ストレージ、アプライ用 10
ストレージ、キャプチャー用 10
ストレージ、診断ファイル用 9
読み取り従属関係 60

[ラ行]

ラージ・オブジェクト (LOB)
レプリケーションの考慮事項 106
ランタイム処理 79, 120
ランチパッド 269
リカバリー点、分散 241
リモート・ジャーナル、ソースとして 61
リモート・ソース表 61
履歴データ
ソース・データ 52
CCD 表 89
レシーバーのサイズ、現行 7
列
計算 97
サブセット化
ソースでの 47
ターゲットでの 97
算出 121
使用可能、レプリケーションに 47
相対レコード番号、OS/400 での 62
定義、ターゲット表での 97
登録、ソース表内の 47
登録済みソース表への追加 211
名前変更 99, 121
変更後イメージ 50
変更前イメージ 50
マッピング、ソースからターゲットへ
の 99
列 (垂直方向) のサブセット化
ソースでの 47
ターゲットでの 97
レプリカ表
構造 597

レプリカ表 (続き)
 再キャプチャー、変更の 55
 定義 85
 定義、読み取り/書き込みターゲットの 94

レプリケーション環境
 コピー 245

レプリケーション構成のコピー 245

レプリケーション・アナライザー
 OS/400 版
 作成、SQL パッケージの 35
 呼び出しパラメーター 422
 UNIX 版、呼び出しパラメーター 332
 Windows 版、呼び出しパラメーター 332

レプリケーション・アラート・モニター
 アラート 173
 アラート条件
 イベント 173
 概要 176
 しきい値 173
 選択 187
 E メール通知 180
 list 176
 status 173

お互いの通信
 アプライ・プログラム 499
 キャプチャー 499
 レプリケーション・センター 499

開始 188
 記述 173
 許可要件 25
 コントロール表
 IBMSNAP_ALERTS 576
 IBMSNAP_CONDITIONS 578
 IBMSNAP_CONTACTGRP 584
 IBMSNAP_CONTACTS 585
 IBMSNAP_GROUPS 586
 IBMSNAP_MONENQ 586
 IBMSNAP_MONPARMS 586
 IBMSNAP_MONSERVERS 588
 IBMSNAP_MONTRAIL 591

コントロール表の作成 183
 再初期化 189
 スケジューリング 493, 494
 セットアップ 182
 操作 185, 292
 停止 196
 パラメーター
 アラート条件の通知基準 195
 設定 193
 説明 190
 操作エラーの通知基準 196
 データの整理インターバル 194
 デフォルト値 190

レプリケーション・アラート・モニター (続き)
 パラメーター (続き)
 レプリケーション・アラート・モニターの実行頻度 194
 alert_prune_limit 190
 autoprune 190
 email_server 190
 max_notifications_per_alert 190
 max_notification_minutes 190
 monitor_errors 190
 monitor_interval 190
 monitor_limit 190
 monitor_path 190
 runonce 190
 trace_limit 190

メモリーの使用量 5

モニター
 再初期化 189
 作成 186

レプリケーションのモニターの概要 173
 連絡先 173
 連絡先グループ 173
 連絡先情報の定義 184

UNIX 版
 始動 605
 状況のチェック 199
 バインド 33

Windows 版
 始動 605
 状況のチェック 199
 バインド 33

z/OS 版
 状況のチェック 199

レプリケーション・イベントの調整 237

レプリケーション・コマンド
 ADDJOBSCDE 494
 asnlist 371
 asntdiff 372
 asntrep 382
 AT 493, 494
 AT NetView
 Apply for z/OS 494
 Capture for z/OS 494
 backup database (データベースのバックアップ) 31
 CRTJRNRCV 37
 db2rc 268
 DSPJRN 206

OS/400 版
 ADDDPRREG 386
 ADDDPRSUB 395
 ADDDPRSUBM 411
 ADDEXITPGM 41
 ANZDPR 422

レプリケーション・コマンド (続き)
 OS/400 版 (続き)
 ANZDPRJRN 40
 CHGDPRCAPA 425
 CHGJRN 39
 CRTDPRTBL 430
 CRTJRN 37
 ENDDPRAPY 431
 ENDDPRCAP 143, 434
 ENDJOB 435
 GRTPRAUT 35, 437
 GRTOBJAUT 35
 INZDPRCAP 446
 OVRDPRCAPA 448
 RCVJRNE 38
 RMVDPRREG 453
 RMVDPRSUB 454
 RMVDPRSUBM 456
 RMVEXITPGM 40
 RVKDPRAUT 458
 STRDPRAPY 160, 460
 STRDPRCAP 468
 STRJRNPF 38
 WRKDPRTTRC 476
 WRKJOB 201
 WRKREGINF 40
 WRKSBMJOB 201
 WRKSBSJOB 201

OS/400 用
 SBMJOB 494

UNIX 版
 asnacmd 330
 asnanalyze 331
 asnapply 334
 asncap 342
 asnccmd 349

update database configuration (データベース構成の更新) 31

Windows 版
 asnacmd 330
 asnanalyze 331
 asnapply 334
 asncap 342
 asnccmd 349

z/OS 版
 asnacmd 330
 asnapply 334
 asncap 342
 asnccmd 349

\$TA JES2
 Apply for z/OS 494
 Capture for z/OS 494

レプリケーション・サービス
 作成 489
 操作 490
 ドロップ 491

レプリケーション・サービス (続き)
 名前 490
 listing 371
 レプリケーション・センター
 アプライ・プログラムの操作 291
 お互いの通信
 アプライ・プログラム 495
 キャプチャー・トリガー 495
 キャプチャー・プログラム 495
 レプリケーション・アラート・モニター 499
 活性化、サブスクリプション・セット
 の 286
 記述 265
 キャプチャー・プログラムの操作 290
 構成 267
 コネクティビティ 17
 コントロール表 275
 コントロール表プロファイル 272
 サーバーの追加 278
 サブスクリプション・セットの作成
 282
 サブスクリプション・セットのプロモ
 ート 288
 始動 268
 ソースの登録 281
 ソース・オブジェクト・プロファイル
 273
 ターゲット・オブジェクト・プロファ
 イル 274
 データベースの変更キャプチャーの使
 用可能化 280
 定義の削除 290
 定義の除去 290
 登録済みの表またはビューのプロモー
 ト 287
 非活性化、サブスクリプション・セッ
 トの 286
 フル・リフレッシュの強制 289
 プロファイル 272
 プロモート関数 245
 ユーザー ID およびパスワード 271
 ランチパッド 269
 レプリケーション・アラート・モニタ
 ーの操作 292
 レプリケーション・センターの始動 268
 レプリケーション・センターのパスワード
 271
 レプリケーション・センターのユーザー
 ID 271
 レプリケーション・ソース
 結合 63
 サブスクライブ 72
 登録
 行 49

レプリケーション・ソース (続き)
 登録 (続き)
 非 DB2 リレーショナル・データ・
 ソース 45
 ビュー 65
 列 47
 DB2 表 43
 IMS データ・ソース 43
 マッピング、ターゲットへの 82
 CCD (整合変更データ) 表 91
 CCD 表の保守 66
 連絡先
 記述 173
 定義 184
 連絡先グループ 173
 ロールフォワード・リカバリー 31
 ログギング要件
 ターゲット・サーバー 7
 非 DB2 リレーショナル・ソース・サ
 ーバー 12
 DB2 ソース・サーバー 6
 ログ
 計画、影響の 12
 ログ・レコード
 アーカイブされた、キャプチャーの前
 に 7
 コンプレッション・ディクショナリー
 (z/OS) 251
 保守 248
 保存 248
 マルチ・データベース・パーティショ
 ン 249
 ロック
 CCD 表に対する 13
 論理パーティション化キー
 説明 54

[数字]

- 2 バイト文字ラージ・オブジェクト
 (DBCLOB)
 レプリケーションの考慮事項 106
- 3 層レプリケーション構成 92

A

ADDDPRREG コマンド 386
 ADDDPRSUB コマンド 395
 ADDDPRSUBM コマンド 411
 ADDEXITPGM コマンド 41
 ADDJOBSCDE コマンド 494
 add_partition パラメーター
 概要 131
 asncap コマンドで使用 343

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 190
 ALWINACT パラメーター 464
 ANZDPR コマンド 422
 ANZDPRJRN コマンド 40
 APPENQ (アプライ・エンキュー) 表
 548
 APPLHEAPSZ 構成パラメーター 31
 APPLYTRACE (アプライ・トレース) 表
 構造 554
 整理 258
 APPLYTRAIL (アプライ・トレール) 表
 構造 554
 整理 258
 APPLY_JOB (アプライ・ジョブ) 表 549
 apply_path パラメーター 151, 336
 apply_qual パラメーター 151, 330, 335
 APPPARMS (アプライ・パラメーター) 表
 550
 使用 148
 変更 160
 APYQUAL パラメーター 462
 ASCII 表 599
 asnacmd コマンド 330
 asnanalyze コマンド 331
 asnapply コマンド 334
 asncap コマンド 342
 asncmd コマンド 349
 ASNDLCOPY 出口ルーチン 109
 ASNDLCOPYD ファイル・コピー・デー
 モン 112
 ASNDONE 出口ルーチン
 使用 161, 162
 リジェクトされたトランザクション
 60
 asndone.smp ファイル 162
 ASNLOAD 出口ルーチン
 エラー処理 164
 カーソルからのロード関数の使用 169
 説明 164
 前提条件 164
 動作のカスタマイズ 168
 ファイル、生成される 166
 asnload.ini ファイルの使用 169
 DATALINK レプリケーションの 108
 OS/400 版 170
 UNIX 版 165
 Windows 版 165
 z/OS 版 167
 asnload.ini ファイル 169
 ASNMAIL 出口ルーチン 181
 asnslst コマンド 371
 asntdiff コマンド 372
 asntrep コマンド 382
 AT NetView コマンド
 Apply for z/OS 494

AT NetView コマンド (続き)

Capture for z/OS 494

AT コマンド

アプライ・プログラム 493, 494

キャプチャー・プログラム 493, 494

レプリケーション・アラート・モニター
— 493, 494

AUTHTKN (アプライ修飾子相互参照) 表
515

autoprune パラメーター

概要 131

asncap コマンドで使用 344

asnccmd コマンドで使用 351

autoprune パラメーター、レプリケーシ
ョン・アラート・モニター 190

autostop パラメーター 132, 344, 351

B

backup database (データベースのバックア
ップ) コマンド 31

BLOB (バイナリー・ラージ・オブジェク
ト)

レプリケーションの考慮事項 106

C

CALL プロシージャ

定義、サブスクリプション・セットの
79

レプリケーション前後のランタイム処
理 120

CAPCTLLIB パラメーター 471

CAPENQ (キャプチャー・エンキュー) 表
516

CAPMON (キャプチャー・モニター) 表
構造 517

整理 258

CAPPARMS (キャプチャー・パラメータ
ー) 表

構造 519

使用 127

変更 142

CAPSCHEMAS (キャプチャー・スキーマ)
表 515

CAPSTART シグナル 242

CAPSTOP シグナル 244

CAPTRACE (キャプチャー・トレース) 表
構造 522

整理 258

capture_path パラメーター 132, 343

capture_schema パラメーター 133, 343

capture_server パラメーター 133, 343

CCD (整合変更データ) 表

外部

multi-tier レプリケーション 91

構造

キャプチャー・コントロール・サー
バー 523

ターゲット・サーバー 594

使用

履歴または監査 89

multi-tier レプリケーション 91

追加、UOW 列の 90

内部

複数のターゲット 90

非 DB2 リレーショナル・データ・ソ
ース

使用、CCD 表の 45

非リレーショナル・データ・ソース

使用、CCD 表の 43

CCD 表の保守 66

レプリケーション・ソース 91

ロック 13

CD (変更データ) ビュー 62

CD (変更データ) 表

結合用 63

構造 525

ストレージ要件 9

整理 257

トリガー 116

ビュー用の 62

要約、内容の 89

CHGDPRCAPA コマンド 425

CHGJRN コマンド 39

CLNUPITV パラメーター 471

CLOB (文字ラージ・オブジェクト)

レプリケーションの考慮事項 106

cold 始動モード 139

commit_interval パラメーター

概要 134

チューニング 4

asncap コマンドで使用 344

asnccmd コマンドで使用 351

control_server パラメーター 152, 330,
336

COPYONCE パラメーター 465

copyonce パラメーター 152, 338

CRTDPRTBL コマンド 430

CRTJRN コマンド 37

CRTJRNRCV コマンド 37

CTLSSVR パラメーター 462

D

Data Links Manager レプリケーション・
デーモン 111

DATALINK 値

制約事項 59, 94

DATALINK 値 (続き)

保管、更新の 55

レプリケーション 107

ASNDLCOPY 出口ルーチン 109

ASNDLCOPYD ファイル・コピー・デ
ーモン 112

DLFM_ASNCOPYD ファイル・コピ
ー・デーモン 111

DB2 エクステンダー

制約事項 106

DB2 ビュー

登録 65

DB2 表

登録 43

DB2 レプリケーション

許可要件 19

DB2CODEPAGE 環境変数 14, 30

DB2DBDFT 環境変数 30

DB2INSTANCE 環境変数 30

db2rc コマンド 268

db2_subsystem パラメーター 153, 335

DBADM 21

DBCLOB (2 バイト文字ラージ・オブジェ
クト)

レプリケーションの考慮事項 106

DBHEAP 構成パラメーター 31

DELAY パラメーター 464

delay パラメーター 153, 339

DLFM_ASNCOPYD ファイル・コピー・
デーモン 111

DPR 登録 (OS/400)

削除 453

追加 386

DSPJRN コマンド 206

E

E メール通知、レプリケーション 180

EDITPROC 文節

制約事項、圧縮 105

email_server パラメーター、レプリケーシ
ョン・アラート・モニター 190

ENDDPRAPY コマンド 431

ENDDPRCAP コマンド 143, 434

ENDJOB コマンド 435

errwait パラメーター 153, 339

F

FIELDPROC 文節

制約事項、圧縮 105

FRCFRQ パラメーター 474

FULLREFPGM パラメーター 463

G

GRTDPRAUT コマンド
構文 437
SQL パッケージへの特権の付与 35
GRTOBJAUT コマンド 35

I

IASP グループ 27
IBMSNAP_ALERTS コントロール表 576
IBMSNAP_CONDITIONS コントロール表 578
IBMSNAP_CONTACTGRP コントロール表 584
IBMSNAP_CONTACTS コントロール表 585
IBMSNAP_GROUPS コントロール表 586
IBMSNAP_MONENQ コントロール表 586
IBMSNAP_MONPARMS コントロール表 586
IBMSNAP_MONSERVERS コントロール表 588
IBMSNAP_MONTRAIL コントロール表 591
IMS DataPropagator 43
IMS データ・ソース
使用、CCD 表の 43
登録 43
CCD 表の保守 66
INACTMSG パラメーター 463
inamsg パラメーター 154, 337
INZDPRCAP コマンド 446
iSeries サーバー
接続 18

J

JOBID パラメーター 461, 470
JOIN_UOW_CD 列 117
JRN パラメーター 472

L

LAG パラメーター 474
lag_limit パラメーター 134, 344
LANG 変数
設定 15
LIBPATH 31
loadxit パラメーター 154, 337
LOB (ラージ・オブジェクト)
レプリケーションの考慮事項 106
Update-anywhere の制約事項 94
LOGBUFZS 構成パラメーター 31

LOGFILSIZ 構成パラメーター 31
LOGPRIMARY 構成パラメーター 31
logreuse パラメーター (アプライ用) 154, 336
logreuse パラメーター (キャプチャー用) 134, 345, 351
LOGSECOND 構成パラメーター 31
logstdout パラメーター (アプライ用) 155, 336
logstdout パラメーター (キャプチャー用) 135, 345, 352
LONG VARCHAR データ・タイプ 105
LONG VARGRAPHIC データ・タイプ 105

M

MAXAPPLS 構成パラメーター 31
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 190
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 190
MAX_SYNC_MINUTES、データ・プロ
ッキング 75
MEMLMT パラメーター 473
memory_limit パラメーター
概要 135
チューニング 4
asncap コマンドで使用 345
asnccmd コマンドで使用 352
Microsoft SQL Server
レプリケーションの制約事項 50
monitor_errors パラメーター、レプリケー
ション・アラート・モニター 190
monitor_interval パラメーター (キャプチ
ャー用) 136, 345, 352
monitor_limit パラメーター 136, 345,
352
レプリケーション・アラート・モニタ
ー 190
monitor_path パラメーター、レプリケーシ
ョン・アラート・モニター 190
MONITV パラメーター 473
MONLMT パラメーター 472
multi-tier レプリケーション
サブスクリプション・セットの定義 91

N

NLS (各国語サポート) 14
notify パラメーター 155, 337

O

opt4one パラメーター 155, 339
OPTSNGSET パラメーター 466
OS/400 データ・ソース
リモート・ジャーナリングを使用する 61
OVRDPRCAPA コマンド 448

P

PARTITIONINFO (パーティション情報)
表 526
PREDICATES 列 117
PRUNCNTL (整理コントロール) 表 527
prune_interval パラメーター 136, 346,
352
PRUNE_LOCK (整理ロック) 表 529
PRUNE_SET (整理セット) 表 529
pwdfile パラメーター 156, 336, 345

Q

Q アプライ・プログラム
アラート条件 176
Q キャプチャー・プログラム
アラート条件 176
Q レプリケーション・コマンド
asnslist 371
asntdiff 372
asntrep 382

R

RCVJRNE コマンド 38
REGISTER (登録) 表 532
REG_EXT (登録拡張) 表 530
REG_SYNC (登録同期) 表 539
RESTART (再始動) 表 540
RESTART パラメーター 469
RETAIN パラメーター 473
retention_limit パラメーター 137, 346,
352
RMVDPREG コマンド 453
RMVDPRSUB コマンド 454
RMVDPRSUBM コマンド 456
RMVEXITPGM コマンド 40
ROWID 106
RRN 62
RTYWAIT パラメーター 464
runonce パラメーター、レプリケーシ
ョン・アラート・モニター 190
RUNSTATS ユーティリティ 253
RVKDPRAUT コマンド 458

S

SBMJOB コマンド 494
SCM (Service Control Manager)
レプリケーション・サービスの作成 489
レプリケーション・サービスの操作 490
レプリケーション・サービスのドロップ 491
レプリケーション・サービスの命名 490
SEQTABLE (順序付け) 表 542
Service Control Manager (SCM)
レプリケーション・サービスの作成 489
レプリケーション・サービスの操作 490
レプリケーション・サービスのドロップ 491
レプリケーション・サービスの命名 490
sleep パラメーター 156, 338
sleep_interval パラメーター 138, 346, 352
spillfile パラメーター 157, 341
SQL スクリプト 123
SQL ステートメント
定義、サブスクリプション・セットの 79
ランタイム処理 120
SQL パッケージ
作成、アプライ・プログラム用の 35
作成、キャプチャー・プログラム用の 34, 35
作成、レプリケーション・アナライザー用 35
SQL ファイル、編集 123
sqlerrcontinue パラメーター 157, 340
startmode パラメーター 138, 347
status
アプライ・プログラム 199, 201
キャプチャー・プログラム 199, 201
ジャーナル・ジョブ 201
レプリケーション・アラート・モニター 199
STOP シグナル 239, 241
STRDPRAPY コマンド 160, 460
STRDPRCAP コマンド 468
STRJRNPf コマンド 38
SUBNFYPGM パラメーター 463
SUBS_COLS (サブスクリプション列) 表 560
SUBS_EVENT (サブスクリプション・イベント) 表
構造 562

SUBS_EVENT (サブスクリプション・イベント) 表 (続き)
通知、イベントの 81
SUBS_MEMBR (サブスクリプション・メンバー) 表 168, 563
SUBS_SET (サブスクリプション・セット) 表 568
SUBS_STMTS (サブスクリプション・ステートメント) 表 574
Sybase
レプリケーションの制約事項 50
SYSADM 21

T

term パラメーター (アプライ用) 158, 340
term パラメーター (キャプチャー用) 139, 348, 353
TRACE パラメーター 462
trace_limit パラメーター
概要 139
レプリケーション・アラート・モニター 190
asnccap コマンドで使用 348
asnccmd コマンドで使用 353
asnmon コマンドで使用 360
TRCLMT パラメーター 472
TRLREUSE パラメーター 465
trlreuse パラメーター 158, 338

U

UOW (作業単位) 表
構造 546
ストレージ要件 9
整理 257, 546
列、CCD 表の 90
UOW_CD_PREDICATES 列 117
update database configuration (データベース構成の更新) コマンド 31
Update-anywhere レプリケーション
競合検出
概要 59
計画 11
要件 52, 59
再キャプチャー、変更の 55
サブスクリプション・セットの定義 94
フラグメント化 11
USER シグナル 237
USER パラメーター 461

V

VALIDPROC 文節 105

W

WAIT パラメーター 470
warmns 始動モード 138
warmsa 始動モード 139
warmsi 始動モード 138
WHERE 文節
行サブセット 98
PREDICATES 列の制約事項 117
Windows Service Control Manager (SCM) 489
レプリケーション・サービスのリスト 371
asnslst コマンド 371
Windows サービス名 328
WRKDPRTTRC コマンド 476
WRKJOB コマンド 201
WRKREGINF コマンド 40
WRKSBMJOB コマンド 201
WRKSBSJOB コマンド 201

Z

z/OS サーバー
接続 18

[特殊文字]

区切り文字 124
\$TA JES2 コマンド 494
*.APP.log ファイル 154
*.CAP.log ファイル 134
*.err ファイル 157
*.sqz ファイル 157, 158
; 区切り文字 124

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 World Trade Asia Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003

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が、その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

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

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
AS/400
DataPropagator
DB2iSeries
MVS
OS/390
OS/400
z/OS

以下は、それぞれ各社の商標または登録商標です。

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

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

他の会社名、製品名およびサービス名等はそれぞれ各社の商標または登録商標です。

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

SC88-9163-02



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