



新機能ガイド

---

# Replication Server<sup>®</sup> 15.7.1

ドキュメント ID：DC37519-01-1571-01

改訂：2012年4月

Copyright © 2012 by Sybase, Inc. All rights reserved.

このマニュアルは Sybase ソフトウェアの付属マニュアルであり、新しいマニュアルまたはテクニカル・ノートで特に示されないかぎり、後続のリリースにも付属します。このマニュアルの内容は予告なしに変更されることがあります。このマニュアルに記載されているソフトウェアはライセンス契約に基づいて提供されるものであり、無断で使用することはできません。

アップグレードは、ソフトウェア・リリースの所定の日時に定期的に提供されます。このマニュアルの内容を弊社の書面による事前許可を得ずに、電子的、機械的、手作業、光学的、またはその他のいかなる手段によっても、複製、転載、翻訳することを禁じます。

Sybase の商標は、Sybase の商標リスト (<http://www.sybase.com/detail?id=1011207>) で確認できます。Sybase およびこのリストに掲載されている商標は、米国法人 Sybase, Inc. の商標です。® は、米国における登録商標であることを示します。

このマニュアルに記載されている SAP、その他の SAP 製品、サービス、および関連するロゴは、ドイツおよびその他の国における SAP AG の商標または登録商標です。

Java および Java 関連のすべての商標は、米国またはその他の国での Oracle およびその関連会社の商標または登録商標です。

Unicode と Unicode のロゴは、Unicode, Inc. の登録商標です。

このマニュアルに記載されている上記以外の社名および製品名は、当該各社の商標または登録商標の場合があります。

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

# 目次

表記の規則 .....	1
<b>Replication Server 15.7.1 の新機能 .....</b>	<b>5</b>
Adaptive Server レプリケーション機能サポートの強化 .....	5
Adaptive Server データ圧縮 .....	5
ロー内/ロー外の LOB .....	6
マスタ・キーと rs パスワード .....	6
master データベース・レプリケーションのパスワード有効期間 .....	6
Adaptive Server コマンドとシステム・プロセスのサポート .....	7
マルチパス・レプリケーション .....	7
異機種間マルチパス・レプリケーション .....	8
コネクション別分散 .....	9
High-Volume Adaptive Replication と Real-Time Loading .....	11
セキュリティの向上 .....	12
入力時のパスワードの非表示 .....	12
パスワード・ポリシー管理 .....	12
パスワードの暗号化 .....	12
Replication Server 設定のデフォルト・パスワードの削除 .....	13
sa ユーザ・パスワードのリセット .....	14
コマンド監査 .....	14
パスワード・セキュリティを実現するためのシステム・テーブル・サポート .....	14
セキュリティの推奨事項 .....	14
パフォーマンスの強化 .....	16

非同期パーサ、ASCII パッキング、および直接 コマンド・レプリケーション .....	16
利便性の向上 .....	17
ウォーム・スタンバイ環境と MSA 環境におけ るカスタム・ファンクション文字列の複製 定義の削減 .....	17
アップグレードの簡素化 .....	18
システム管理ツール .....	19
<b>Replication Server 15.7 の新機能 .....</b>	<b>21</b>
Replication Server ライセンス .....	21
マルチパス・レプリケーション .....	21
パフォーマンスの強化 .....	22
SQM コマンド・キャッシュ .....	22
エグゼキュータ・コマンド・キャッシュ .....	23
sqm_cache_size の上限値 .....	24
セグメントを削除するための専用デーモン .....	24
利便性とプロセスの向上 .....	25
複製定義の使用の削減 .....	25
rs_functions に加えられた変更 .....	26
メモリ消費の制御 .....	27
Unicode の機能強化 .....	29
SySAM ライセンス情報の要求 .....	30
サブスクリプション名の拡張 .....	30
後続ゼロの削除 .....	30
Sybase Control Center for Replication および Data Assurance .....	31
Adaptive Server レプリケーション機能サポートの強 化 .....	34
RepAgent の自動起動 .....	34
Real-Time Loading と High-Volume Adaptive Replication .....	35
メモリ消費の制御 .....	36

Sybase IQ データベース・オプションの設定 .....	37
スキーマ変換とデータ型変換 .....	37
パラメータのデフォルト値の変更 .....	38
Replication Server Data Assurance オプション .....	39
<b>Replication Server バージョン 15.6 ESD #1 の新機能 .....</b>	<b>41</b>
Replication Server と Sybase IQ InfoPrimer の統合 .....	41
ライセンス .....	42
Replication Server と Sybase IQ InfoPrimer の 統合の使用 .....	42
パラメータ .....	48
Replication Server のコンポーネント .....	49
デフォルトのデータ型変換 .....	53
サポートされない機能 .....	53
<b>Replication Server 15.6 の新機能 .....</b>	<b>55</b>
Replication Server ライセンス .....	55
サブキャパシティ・ライセンス .....	55
Replication Server 15.6 の製品エディションと ライセンス .....	55
Real-Time Loading を使用した Oracle から Sybase IQ への複写 .....	57
Real-Time Loading ソリューション .....	57
Sybase IQ のレプリケート・データ・サーバ .....	62
Sybase IQ での複写の干渉と影響 .....	62
Sybase IQ に関するレプリケート・データベ ースのコネクティビティ .....	64
Sybase IQ レプリケート・データベースのパー ミッション .....	64
Sybase IQ レプリケート・データベースの設定 に関する問題 .....	66
Sybase IQ への複写シナリオ .....	70
参照制約のあるテーブル .....	74
RTL 情報の表示 .....	76

最終的な変更のデータベース .....	77
混合バージョンのサポートと下位互換性 .....	77
ステージング・ソリューションから RTL にマ イグレートする .....	78
パフォーマンスの強化 .....	80
HVAR と RTL のリトライ・メカニズムの強化 .....	80
キュー・ブロック・サイズの増加の機能強化 .....	82
利便性とプロセスの向上 .....	88
Adaptive Server のレプリケート・データベー スの再同期 .....	88
範囲による例外の削除 .....	102
ロー・カウントの検証を制御する .....	106
ロー・カウントの検証エラー・メッセージに テーブル名を表示する .....	107
シームレス・アップグレード .....	109
Adaptive Server 複写機能サポートの強化 .....	109
インメモリ・データベースおよびリラックス 持続性データベース .....	109
image と Java データ型のバルク・コピー・イ ン .....	111
<b>Replication Server バージョン 15.5 の新機能 .....</b>	<b>113</b>
Replication Server 15.5 の製品エディションとライセ ンス .....	113
Real-Time Loading を使用した Sybase IQ の複写 .....	114
異機種間複写サポートの機能強化 .....	115
異機種環境での並列 DSI サポート .....	115
Oracle に対する異機種ウォーム・スタンバ イ・サポート .....	117
Oracle レプリケート・データベースでのトリ ガ制御 .....	118
パフォーマンスの強化 .....	118

Replication Server – Advanced Services Option .....	118
動的 SQL の機能強化 .....	121
ファンクション文字列の効率の向上 .....	123
利便性とプロセスの強化 .....	124
複写定義の変更要求プロセスの強化 .....	124
複写タスクのスケジューリング .....	126
複写の遅延 .....	126
レプリケート・データベースの再同期 .....	127
ロー・カウント検証での変更 .....	128
alter error class の機能強化 .....	129
リファレンス複写環境の実装に使用するツールセット .....	129
admin who コマンドの機能強化 .....	131
データベース世代番号のリセット .....	132
インバウンド・キューに rs_ticket マーカを挿入する .....	133
デフォルト設定と予約語での変更 .....	133
パラメータのデフォルト値の変更 .....	133
RSSD ロック・スキーマでの変更 .....	134
予約語 .....	135
Adaptive Server 複写機能サポートの強化 .....	135
bigdatetime と bigtime の複写 .....	135
遅延名前解決 .....	136
SQL 文の複写スレッシュホールドの設定 .....	137
増分データ転送 .....	137
インメモリ・データベースおよびリラックス持続性データベース .....	138
混合バージョン環境 .....	140
新しくサポートされたオペレーティング・システム .....	140

64 ビット・コンピューティング・プラットフォーム のサポート .....	140
Replication Server 設定パラメータの変更 .....	141
memory_limit 設定パラメータの変更 .....	142
<b>Replication Manager 15.5 の新機能 .....</b>	<b>143</b>
bigdatetime と bigtime の複写の有効化 .....	143
<b>Replication Server バージョン 15.2 の新機能 .....</b>	<b>145</b>
DSI でのバルク・コピー・インのサポート .....	145
サブスクリプションのマテリアライゼーショ ンの強化 .....	145
新しいコネクション・パラメータ .....	146
バルク・コピー・イン用の新しいカウンタ .....	147
制限事項 .....	148
非ブロッキング・コミット .....	149
Adaptive Server の遅延コミット機能 .....	149
dsi_non_blocking_commit 設定パラメータ .....	150
rs_non_blocking_commit システム・ファンク ション .....	150
rs_non_blocking_commit_flush システム・ファ ンクション .....	151
サポートされる Adaptive Server 以外のデー タベース .....	152
引用符付き識別子 .....	152
引用符付き識別子のサポートを有効にする設 定パラメータ .....	153
引用符付き識別子としてマークを付けるコマ ンド .....	153
rs_set_quoted_identifier ファンクション文字列 .....	155
rs_helprep の変更 .....	155
Replication Server ゲートウェイ .....	158
カスケード・コネクション .....	159



Replication Server ゲートウェイを有効にする コマンド .....	159
コネクションを追跡するコマンド .....	160
コネクションを削除するコマンド .....	161
非 SQL 文の複写ロー・カウン트의檢証 .....	161
Replication Server エラー・クラスを作成する コマンド .....	162
エラー・アクションを割り当てるコマンド .....	163
Replication Server エラー・クラスを表示する ストアド・プロシージャ .....	165
Replication Server システム・データベースの 変更 .....	165
SQL 文の複写 .....	165
SQL 文の複写の有効化 .....	166
システム設定の変更 .....	166
SQL 文の複写設定 .....	168
SQL 文の複写に対するロー・カウン트의檢証 ..	171
SQL の複写でのウォーム・スタンバイ・デー タベースの設定 .....	173
SQL の複写でのウォーム・スタンバイ・デー タベースの設定 .....	174
Replication Server システム・データベースの 変更 .....	174
Adaptive Server 以外のためのエラー・クラスのサ ポート .....	175
デフォルトの ASE 以外のためのエラー・クラ ス .....	175
create error class コマンドの変更 .....	176
alter error class コマンドの変更 .....	176
Adaptive Server 以外での複写サポートの機能強化 ..	177
単純化されたインストールと設定 .....	177
接続プロファイル .....	178

<b>Replication Server バージョン 15.1 の新機能</b> .....	<b>183</b>
動的 SQL の機能強化 .....	183
ファンクション複製の機能強化 .....	184
Adaptive Server 共有ディスク・クラスタのサポート .....	185
モニタリングとカウンタの機能強化 .....	186
新しいアクティブ・オブジェクト・カウンタ ...	186
新しいプロシージャ・インタフェース .....	187
ステابل・キューの管理の改善 .....	187
sysadmin dump_queue の変更 .....	188
sysadmin sqt_dump_queue の変更 .....	188
変更された resume connection コマンド .....	189
変更された sysadmin log_first_tran コマンド ...	189
新しい sysadmin sqm_zap_tran コマンド .....	189
新しい sysadmin sqm_unzap_tran コマンド ....	190
新しい sysadmin dump_tran コマンド .....	190
locales ディレクトリに対する変更 .....	190
拡張パスワード暗号化のサポート .....	191
rs_ticket ストアド・プロシージャ・バージョン 2 .....	191
Replication Server の新しいカウンタ .....	193
レンジ・オブジェクト・データ型の拡張サポート ....	194
レンジ・オブジェクト・データ型の部分更新 .....	195
timestamp の拡張サポート .....	195
新しい opaque データ型 .....	196
ダンプ・トランザクションの機能強化 .....	196
dump サブコマンドの新しいパラメータ .....	197
rs_dumptran の変更 .....	197
ディストリビュータ・ステータスの記録 .....	198
テキスト更新の強化 .....	198
Adaptive Server における整数 ID のサポート .....	199
ステابل・キュー・マネージャのパフォーマンス の強化 .....	199

ステابل・キューのキャッシュ .....	199
セグメントの事前割り付け .....	201
ダイレクト I/O によるファイル・アクセスのサ ポート .....	202
<b>Replication Manager 15.1 の新機能 .....</b>	<b>203</b>
動的 SQL のサポートの強化 .....	203
ファンクション複写定義のサポートの強化 .....	203
レンジ・オブジェクト・データ型のサポート .....	204
Sybase Central 6.0 .....	204
opaque データ型のサポート .....	204
timestamp データ型のサポート .....	205
<b>追加の説明や情報の入手 .....</b>	<b>207</b>
サポート・センタ .....	207
Sybase EBF と Maintenance レポートのダウンロー ド .....	207
Sybase 製品およびコンポーネントの動作確認 .....	208
MySybase プロファイルの作成 .....	208
アクセシビリティ機能 .....	209
<b>索引 .....</b>	<b>211</b>

# 目次

# 表記の規則

ここでは、Sybase® マニュアルで使用しているスタイルおよび構文の表記規則について説明します。

## 表記の規則

構文要素	定義
mono-spaced (fixed-width)	<ul style="list-style-type: none"> <li>SQL およびプログラム・コード</li> <li>表示されたとおりに入力する必要のあるコマンド</li> <li>ファイル名</li> <li>ディレクトリ名</li> </ul>
<i>italic mono-spaced</i>	SQL またはプログラム・コードのスニペット内では、ユーザ指定の値のプレースホルダ (以下の例を参照)
<i>italic</i>	<ul style="list-style-type: none"> <li>ファイルおよび変数の名前</li> <li>他のトピックまたはマニュアルとの相互参照</li> <li>本文中では、ユーザ指定の値のプレースホルダ (以下の例を参照)</li> <li>用語解説に含まれているテキスト内の用語</li> </ul>
<b>bold san serif</b>	<ul style="list-style-type: none"> <li>コマンド、関数、ストアド・プロシージャ、ユーティリティ、クラス、メソッドの名前</li> <li>用語解説のエントリ (用語解説内)</li> <li>メニュー・オプションのパス</li> <li>番号付きの作業または手順内では、クリックの対象となるボタン、チェック・ボックス、アイコンなどのユーザ・インタフェース (UI) 要素</li> </ul>

必要に応じて、プレースホルダ (システムまたは設定固有の値) の説明が本文中に追加されます。次に例を示します。

次のコマンドを実行します。

```
installation directory¥start.bat
```

*installation directory* はアプリケーションがインストールされた場所です。

構文の表記規則

構文要素	定義
{ }	中カッコで囲まれたオプションの中から必ず1つ以上を選択する。コマンドには中カッコは入力しない。
[ ]	角カッコは、オプションを選択しても省略してもよいことを意味する。コマンドには角カッコは入力しない。
( )	このカッコはコマンドの一部として入力する。
	縦線はオプションのうち1つのみを選択できることを意味する。
,	カンマは、表示されているオプションを必要な数だけ選択でき、選択したものをコマンドの一部として入力するときにカンマで区切ることを意味する。
...	省略記号(...)は、直前の要素を必要な回数だけ繰り返し指定できることを意味する。省略記号はコマンドには入力しない。

大文字と小文字の区別

- すべてのコマンド構文およびコマンドの例は、小文字で表記しています。ただし、複写コマンド名では、大文字と小文字が区別されません。たとえば、**RA\_CONFIG**、**Ra\_Config**、**ra\_config** は、すべて同じです。
- 設定パラメータの名前では、大文字と小文字が区別されます。たとえば、**Scan\_Sleep\_Max** は、**scan\_sleep\_max** とは異なり、パラメータ名としては無効になります。
- データベース・オブジェクト名は、複写コマンド内では、大文字と小文字が区別されません。ただし、複写コマンドで大文字と小文字が混在したオブジェクト名を使用する場合(プライマリ・データベースの大文字と小文字が混在したオブジェクト名と一致させる場合)、引用符でオブジェクト名を区切ります。次に例を示します。 **pdb\_get\_tables "TableName"**
- 識別子および文字データでは、使用しているソート順によっては大文字と小文字が区別されます。
  - “binary” などの大文字と小文字を区別するソート順を使用する場合には、識別子や文字データは、大文字と小文字を正しく入力してください。
  - “nocase” などの大文字と小文字を区別しないソート順を使用する場合には、識別子や文字データは、大文字と小文字をどのような組み合わせでも入力できます。

*用語*

Replication Agent™ は、Adaptive Server® Enterprise、Oracle、IBM DB2 UDB、Microsoft SQL Server 用の Replication Agent を表現するために使用される一般的な用語です。具体的な名前は、次のとおりです。

- RepAgent — Adaptive Server Enterprise 用の Replication Agent スレッド
- Replication Agent for Oracle
- Replication Agent for Microsoft SQL Server
- Replication Agent for UDB — Linux、Unix、Windows 用の IBM DB2





# Replication Server 15.7.1 の新機能

Replication Server® 15.7.1 では、パフォーマンス、利便性、セキュリティ、データベース・サポートが強化されました。

## Adaptive Server レプリケーション機能サポートの強化

Replication Server 15.7.1 は Adaptive Server レプリケーションをサポートしています。

### Adaptive Server データ圧縮

Replication Server では、Adaptive Server データ圧縮機能がサポートされています。

Adaptive Server バージョン 15.7 では、データ圧縮機能を使用できます。データ圧縮を行うと、同じ容量のデータをより小さい記憶領域に格納して、キャッシュ・メモリの消費量を削減し、I/O 要求の緩和によってパフォーマンスを向上させることができます。Adaptive Server は、通常のデータとラージ・オブジェクト (LOB) のデータ型を圧縮できます。たとえば、text、image、unitext などです。

『Adaptive Server Enterprise 圧縮ユーザズ・ガイド』を参照してください。

Adaptive Server では、データをロー内またはロー外に格納します。ロー内データは、ロー・メタデータに物理的に連続している場所に格納されます。LOB データは、データ・サイズが大きいためロー外の他の場所に格納されます。ロー外データの実際の場所を示すポインタがロー内にあります。

Replication Server は、圧縮解除を行わず、プライマリ Adaptive Server データベースの圧縮された LOB カラムを圧縮されたままの形式でテキスト値の圧縮解除を行わずにレプリケートします。Adaptive Server データベース間の圧縮データ・レプリケーションのサポートについては、『Replication Server 管理ガイド 第 1 巻』の「RepAgent の管理と Adaptive Server のサポート」の「Adaptive Server データ圧縮」を参照してください。

#### バージョンのサポート

- Adaptive Server – プライマリ・データベースとレプリケート・データベースの両方について、バージョン 15.7 ESD #1 以降。Replication Server 15.7.1 と互換性のある Adaptive Server のバージョンについては、『Replication Server リリース・ノート』の「製品の互換性」の「Replication Server の相互運用性」を参照してください。

- Replication Server – プライマリおよびレプリケート Replication Server について、バージョン 15.7.1 以降。

## ロー内/ロー外の LOB

Replication Server では、Adaptive Server 15.7 以降でのロー内/ロー外の LOB サポートの変更がサポートされています。

Adaptive Server Enterprise の「新機能の概要」の「Adaptive Server バージョン 15.7 の新機能」の「ラージ・オブジェクトの変更」の「ロー内/ロー外の LOB」を参照してください。

## マスタ・キーと rs パスワード

レプリケーションを続行するには、マスタ・キー・パスワードと **rs password** 属性を設定します。

Adaptive Server では、マスタ・キーとともに `syb_extpasswdkey` サービス・キーを作成しても、マスタ・キー・パスワードをメモリに手動でも自動的にも設定していない場合、Adaptive Server RepAgent が起動時にブロックされ、マスタ・キー・パスワードを設定するまでは **sp\_who** によって "MASTER KEY SLEEP" と表示されます。レプリケーション・パスごとに、RepAgent が Replication Server へのログインに使う **rs password** 属性が 1 つ存在します。`syb_extpasswdkey` サービス・キーが削除されると、Adaptive Server は既存の RepAgent **rs password** 属性をすべてリセットします。「**sp\_encryption helpextpasswd**」と入力すると、「必要とするリセット」と表示されます。レプリケーションを続行するには、**rs password** 属性をすべてリセットします。

Adaptive Server Enterprise の『暗号化カラム・ユーザズ・ガイド』の「外部パスワードと隠しテキストのセキュリティ保護」の「サービス・キー」を参照してください。

## master データベース・レプリケーションのパスワード有効期間

ウォーム・スタンバイ環境で Adaptive Server master データベース・レプリケーションを設定する場合、スタンバイ master データベースにはアクティブ master データベースより長いパスワード有効期間を設定することをお勧めします。これにより、アクティブ master データベースはパスワードの変更を管理し、パスワード変更のレプリケーションを続行できます。

『Replication Server 管理ガイド 第 2 巻』の「ウォーム・スタンバイ・アプリケーションの管理」の「ASE のウォーム・スタンバイ環境での master データベースの複写」を参照してください。

## Adaptive Server コマンドとシステム・プロシージャのサポート

---

Adaptive Server のいくつかのコマンドとシステム・プロシージャのレプリケーションについて、サポートが拡張されました。

レプリケーションでサポートされている DDL コマンドとシステム・プロシージャは次のとおりです。

- **alter login**
- **alter login profile**
- **alter...modify owner** – Replication Server は、所有者の異なるテーブルを別のテーブルとして扱います。**alter...modify owner** を使用して Adaptive Server でレプリケートされたテーブルの所有者を変更するには、該当するテーブル複写定義も変更する必要があります。『Replication Server 管理ガイド 第 1 巻』の「複写テーブルの管理」の「複写定義の修正」の「複写定義の変更」の「複写定義に可能な変更」の「テーブルの所有者の変更」を参照してください。
- **create login**
- **create login profile**
- **drop login**
- **drop login profile**
- **sp\_hidetext**

master データベース・レプリケーションでサポートされているシステム・プロシージャは次のとおりです。

- **sp\_addexternlogin**
- **sp\_dropexternlogin**
- **sp\_maplogin**
- **sp\_addremotelogin**
- **sp\_dropremotelogin**
- **sp\_addserver**
- **sp\_dropserver**

Replication Server の『リファレンス・マニュアル』の「Adaptive Server コマンドとシステム・プロシージャ」の「**sp\_reptostandby**」の「サポートされている DDL コマンドとシステム・プロシージャ」を参照してください。

## マルチパス・レプリケーション

---

Replication Server 15.7.1 では、マルチパス・レプリケーションのサポートが強化されました。

## 異機種間マルチパス・レプリケーション

Replication Server 15.7.1 では、マルチパス・レプリケーションのサポートが異機種データベースを搭載したレプリケーション・システムにまで拡張されました。

Replication Server 15.7 以降、トランザクションを Replication Server 間の専用ルートを介してプライマリ Adaptive Server データベースから Replication Server への複数のレプリケーション・パスと、レプリケート Adaptive Server データベースへの複数のコネクションに分散できます。『Replication Server 管理ガイド 第2巻』の「パフォーマンス・チューニング」の「マルチパス・レプリケーション」を参照してください。

バージョン 15.7.1 では、Adaptive Server と Sybase IQ、Adaptive Server と Oracle など、データベース間に複数のレプリケーション・パスと専用ルートを作成できません。

Sybase IQ データベースへのマルチパス・レプリケーションを設定するには、『Replication Server 異機種間複写ガイド』の「レプリケート・データ・サーバとしての Sybase IQ」の「Sybase IQ へのマルチパス・レプリケーション」を参照してください。異機種間マルチパス・レプリケーションのシナリオについては、『Replication Server 異機種間複写ガイド』の「異機種間におけるマルチパス・レプリケーション」を参照してください。

サポートされている異機種データベース・マルチパス・レプリケーション・システム

表 1: 異機種間マルチパス・レプリケーション・システムでサポートされているプライマリ・データベースとレプリケート・データベースのペア

プライマリ・データベース	レプリケート・データベース
Adaptive Server	Sybase IQ
Oracle	Sybase IQ
Adaptive Server	Oracle
Oracle	Adaptive Server
Oracle	Oracle

**注意:** Oracle プライマリ・データベースからのレプリケーションでは、Replication Server Options に搭載されている Replication Agent for Oracle が必要です。

表 2: マルチパス・レプリケーションでサポートされているデータベース・バージョン

データベース	サポートされているバージョン
Adaptive Server	15.7 以降
Oracle	Oracle 10g と 11g。Replication Server Options の『Replication Server Options リリース・ノート』の「製品の概要」の「製品の互換性」を参照してください。
Sybase IQ	15.1 以降。『Replication Server リリース・ノート』の「製品の互換性」の「Replication Server の相互運用性」を参照してください。

### ライセンス

マルチパス・レプリケーションは、Advanced Services Option の一部としてライセンスされます。RTL を使用した Sybase IQ へのレプリケーションは、Real-Time Loading Edition (RTLLE) で実行可能です。『Replication Server インストール・ガイド』の「インストールの計画」の「ライセンスの取得」を参照してください。

Oracle からのレプリケーションには、Replication Agent for Oracle が必要です。Replication Server Options の『Replication Agent インストール・ガイド』の「インストールの計画」の「ライセンス」を参照してください。

## コネクション別分散

マルチパス・レプリケーション環境では、さまざまな分散モードを使用して並列レプリケーションとレプリケーション・パフォーマンスの向上を達成できます。具体的には、プライマリ・データベースから始まる使用可能なプライマリ・レプリケーション・パスを介してプライマリ・データベースのレプリケーション負荷を分散します。

Replication Server 15.7 と Adaptive Server 15.7 では、複数のオブジェクト (テーブルやストアド・プロシージャなど) を特定のレプリケーション・パスにバインドすることで、これらオブジェクトの並列レプリケーションが可能になります。『Replication Server 管理ガイド 第2巻』の「パフォーマンス・チューニング」の「マルチパス・レプリケーション」の「複数のプライマリ・レプリケーション・パス」の「レプリケーション・パスへのオブジェクトのバインド」を参照してください。

Replication Server 15.7.1 と Adaptive Server 15.7 ESD #1 では、レプリケーション負荷をコネクション別に分散できます。デフォルト・モードは、オブジェクト・バインド別分散です。Adaptive Server RepAgent では、一度に複数の分散モードは使用できません。

コネクション別分散では、Adaptive Server RepAgent がさまざまなクライアント・プロセスから発生したトランザクションを使用可能なレプリケーション・パスに割り当てます。時間の経過とともに、使用可能なパス全体でデータ分散のバランスが取れていきます。使用可能な RepAgent パスが多数あり、クライアント・プロセスの数が多い場合、パフォーマンスが向上し、レプリケーション負荷分散はより均一化します。

『Replication Server 管理ガイド 第2巻』の「パフォーマンス・チューニング」の「マルチパス・レプリケーション」の「並列トランザクション・ストリーム」の「マルチパス・レプリケーションの分散モード」の「コネクション別分散」を参照してください。

#### データベースのサポート

Replication Server では、プライマリとレプリケートの Adaptive Server データベース間のマルチパス・レプリケーションについて、コネクション別分散をサポートしています。

- プライマリ・データベース – Adaptive Server 15.7 ESD #1 以降
- レプリケート・データベース – Adaptive Server 15.7 以降

『Replication Server リリース・ノート』の「製品の互換性」の「Replication Server の相互運用性」を参照してください。

#### ライセンス

マルチパス・レプリケーションは、Advanced Services Option の一部としてライセンスされます。『Replication Server インストール・ガイド』の「インストールの計画」の「ライセンスの取得」を参照してください。

### **Adaptive Server のモニタリング・テーブルの変更箇所**

レプリケーション負荷をコネクション別に分散する場合、monRepSenders Adaptive Server モニタリング・テーブル内のフィールドを使用して、データ分散の統計的スナップショットを撮影し、Adaptive Server パフォーマンスを分析します。

**表 3 : monRepSenders**

カラム名	説明
NumberOfCommandsProcessed	各 RepAgent スレッドが LTL を生成するために処理するコマンド ( <b>insert</b> 、 <b>delete</b> 、 <b>begin trans</b> 、 <b>commit trans</b> など) の数。
AvgBytesPerCmd	NumberOfBytesSent と NumberOfCommandsProcessed の比率。

『Replication Server システム管理ガイド 第2巻』の「パフォーマンス・チューニング」の「マルチパス・レプリケーション」の「複数のレプリケーション・パス用の Adaptive Server モニタリング・テーブル」を参照してください。

Adaptive Server Enterprise の『パフォーマンス&チューニング・シリーズ：モニタリング・テーブル』の「モニタリング・テーブルの概要」の「Adaptive Server のモニタリング・テーブル」を参照してください。

## High-Volume Adaptive Replication と Real-Time Loading

Replication Server 15.7.1 では、High-Volume Adaptive Replication (HVAR) と Real-Time Loading (RTL) でのメモリ使用率と大規模トランザクションのサポートが向上しています。

次のような改善点があります。

- SQT メモリ消費制御 – HVAR と RTL でトランザクション・プロファイリング中に DSISQT キャッシュでパックされていないコマンドが消費する最大メモリ量を制御できます。
  - HVAR – 『Replication Server 管理ガイド 第2巻』の「HVAR の SQT メモリ消費の制御」
  - RTL – 『Replication Server 異機種間複写ガイド』の「RTL の SQT メモリ消費の制御」

を参照してください。

- 最終的な変更を保管するデータベースのサイズ予測とトランザクション・プロファイリング – トランザクションが DSI SQT キャッシュ・サイズより大きい場合でも、Replication Server はそのトランザクションにコンパイル不可のマークを付けません。

『Replication Server 管理ガイド 第2巻』の「最終的な変更を保管するデータベースのサイズ予測とトランザクション・プロファイリング」を参照してください。

- フル・インクリメンタル・コンパイル – Replication Server は、連続レプリケーション・モードより効率的な HVAR モードを使用して、大規模トランザクションのコンパイルとレプリケートを行います。

『Replication Server 管理ガイド 第2巻』の「HVAR のフル・インクリメンタル・コンパイル」を参照してください。

## セキュリティの向上

---

Replication Server 15.7.1 では、パスワード・セキュリティ管理およびパスワード暗号化が向上し、設定変更に関するコマンドを監査できるようになりました。

### 入力時のパスワードの非表示

**isql** オプションを指定すると、**alter user** または **create user** を使用するとき、パスワードの入力時にパスワードを非表示にできます。

『Replication Server 管理ガイド 第 1 巻』の「パスワード入力の非表示」を参照してください。

### パスワード・ポリシー管理

最小の長さ、必須文字型、有効期限などの要件をユーザ・パスワードに強制することができます。

個別ユーザに対しては Replication Server ユーザの作成時、全ユーザに対してはサーバ・レベルでパスワード要件を設定します。

**rs\_dictionary** RSSD システム・テーブルを使用して、**simple\_passwords\_allowed** パラメータを **true** に設定した場合にパスワードに使用できない文字の組み合わせを格納します。

『Replication Server 管理ガイド 第 1 巻』の「全ユーザを対象としたパスワード設定オプション」を参照してください。

### パスワードの暗号化

Replication Server 15.7.1 では、パスワードの暗号化アルゴリズムが変更されており、既存のパスワードが暗号化されている場合でも、このアルゴリズムに従って既存のすべてのパスワードの復号化と暗号化を行います。Replication Server 15.7.1 以降にアップグレードする場合、暗号化に関するこの変更は、サイト・バージョンを 1571 以降に設定した後で反映されます。

Replication Server では、新しい Replication Server のインストールに対してすべてのパスワードを保存するときに、クリア・テキストではなくパスワードの暗号化を使用します。Replication Server では、この新しいアルゴリズムを使用して **rs\_users** と **rs\_maintusers** RSSD システム・テーブルおよび Replication Server 設定ファイルに格納されたパスワードをすべて暗号化します。Replication Server 15.7.1 には、**rs\_encryptionkeys** RSSD システム・テーブルと設定ファイルの **RS\_random** 属性が導入されており、これらを使用してパスワード暗号化が実行さ



れます。アップグレードした Replication Server を起動する場合、および Replication Server によってシステム・テーブルまたは設定ファイル内にインストールに固有のランダム値が見つからない場合、Replication Server はシステム・テーブルの `rs_password_key` ローおよび `RS_random` 属性のために、それらの値を自動的に生成します。

`alter encryption key rs_password_key regenerate` コマンドで、システム・テーブルと設定ファイルにパスワード暗号化キーのランダム値を再生成できます。

パスワード・セキュリティ要件が変更されたため、`password_encryption` パラメータは廃止されます。

『Replication Server 管理ガイド 第1巻』の「パスワードの暗号化」を参照してください。

## Replication Server 設定のデフォルト・パスワードの削除

Replication Server 15.7.1 では、`rs_init` はデフォルト・パスワードの提示、提案、構成、および使用を行いません。

Replication Server リソース・ファイルにいくつかの属性とともに指定するパスワードに、`USE_DEFAULT` と `UNCHANGED` は入力できません。

パスワード属性	ユーザ ID
<code>rs.rs_rs_sa_pass</code>	Replication Server の sa ログイン
<code>rs.rs_idserver_pass</code>	ID サーバ・ユーザ
<code>rs.rs_rssd_prim_pass</code>	RSSD プライマリ・ユーザ
<code>rs.rs_rssd_maint_pass</code>	RSSD メンテナンス・ユーザ
<code>rs.rs_rs_pass</code>	Replication Server ログイン名
<code>rs.rs_ltm_rs_pass</code>	ログイン転送マネージャ ID
<code>rs.rs_db_maint_password</code>	データベース・メンテナンス・ユーザ

管理者が指定したパスワードのセキュリティ要件に適合するパスワードを入力してください。リソース・ファイルのリストと改訂版サンプル・リソース・ファイルについては、『Replication Server 設定ガイド』の「rs\_init による Replication Server の設定とデータベースの追加」の「リソース・ファイルを使用した rs\_init」の「rs\_init でのリソース・ファイルの使用」の「リソース・ファイルの構文とパラメータ」を参照してください。

## sa ユーザ・パスワードのリセット

sa ユーザのパスワードを紛失したか忘れた場合、リセットできます。

『Replication Server 管理ガイド 第1巻』の「Replication Server のセキュリティ管理」の「Replication Server ユーザのセキュリティ管理」の「Replication Server ログイン名とパスワードの管理」の「sa ユーザ・パスワードを紛失した場合または忘れた場合のリセット」を参照してください。

## コマンド監査

Replication Server のコマンド監査機能を有効にして、ユーザおよびユーザが Replication Server で入力したコマンドに関する情報を記録します。

『Replication Server 管理ガイド 第1巻』の「Replication Server のセキュリティ管理」の「コマンド監査」を参照してください。

## パスワード・セキュリティを実現するためのシステム・テーブル・サポート

パスワード・セキュリティをサポートするため、Replication Server では rs\_passwords、rs\_dictionary、および rs\_encryptionkeys システム・テーブルを使用し、rs\_users に新しいカラムを追加します。

『Replication Server リファレンス・マニュアル』の「Replication Server システム・テーブル」を参照してください。

## セキュリティの推奨事項

管理タスクの実行、SSL、暗号化、パーミッションとロール、設定ファイルなど、Replication Server のセキュリティ問題に関する推奨事項です。

- ベスト・プラクティスとして、管理タスクは Replication Server のローカル・ホストに対してだけ実行してください。  
デフォルトでは、Replication Server のホスト名とポート番号を知っている管理者はだれでも、Replication Server にリモートでアクセスして管理できます。
- master データベース・トランザクションに依存するユーザ・データベース・トランザクション (テーブルの作成など) を実行する場合は、master データベース・トランザクション (ユーザの新規作成やパスワードの変更など) がすべてのレプリケート Adaptive Servers に正常にレプリケートされるまで待ってから行ってください。

Replication Server では、単一の Adaptive Server データベース内で実行されるトランザクションのトランザクション・コミット順が維持されます。しかし、複数の Adaptive Server データベースにまたがって実行されるトランザクションの

コミット順は維持されません。たとえば、プライマリ Adaptive Server で次のようにします。

- master データベース・トランザクション (mylogin ユーザの作成など) を作成するには、sa ユーザを使用して次のコマンドを入力します。

```
sp_addlogin 'mylogin', 'password'
go
use mydb
go
sp_adduser
'mylogin'
go
```

- ユーザ・データベース・トランザクション (mylogin ユーザ ID を使用した mytab テーブルの作成など) を作成するには、次のコマンドを入力します。

```
use mydb
go
create table mytab (mycol int)
go
```

Replication Server が、**sp\_addlogin** プロシージャの前に **create table** コマンドをレプリケートすることは可能ですが、この場合、レプリケート Adaptive Server での **create table** は失敗します。これは、mylogin ユーザがレプリケート・データベースにまだ存在していないからです。

- Replication Server は、Secure Sockets Layer (SSL) を使用してセッションベースのセキュリティを提供できます。SSL は、認証局 (CA) によって発行された証明書を使用して、身元の確認と検証を行います。SSL 証明書の機密性が損なわれた場合、新しい Replication Server 名と証明書番号で CA に新しい証明書を要求する必要があります。
- 管理者は、Replication Server ログでパーミッションを管理して、監査人に読み込み専用アクセス権を付与します。デフォルトでは、Replication Server で作成されたユーザは、ロールを付与されていない状態で、サポート・ロールに十分な RSSD テーブルへの読み込み専用アクセス権を持っています。
- ステアブル・キューでの機密データについては、ディスクレベルの暗号化を検討してください。  
プライマリ・データベースおよびレプリケート・データベースと Replication Server との間の SSL ベースの接続を使用した場合でも、Replication Server はデータをステアブル・キューに一時的に保持する必要があるため、この保持データは暗号化されません。
- 機密データを送信するコネクションまたはルートには、SSL を使用することをお勧めします。Replication Server の SSL (Secure Sockets Layer) Advanced Security オプションは、セッションベースのセキュリティを提供します。
- Replication Server は、ホスト名、ポート、ユーザ名、パスワードなどの初期設定プロパティを、**rs\_init** ユーティリティが使用する **.res** サフィックスのファイルに格納します。**.res** ファイルについて、UNIX では **umask** パーミッシヨ

ン、Windows ではディレクトリ・パーミッションを適切に設定します。このファイルが必要ない場合は、削除してください。

初期設定以降は、**rs\_init** は **.res** ファイルを必要としませんが、Replication Server はこのファイルをオペレーティング・システム・パーミッションによってのみ保護されるオペレーティング・システムのファイル・システムに格納します。

## パフォーマンスの強化

---

Replication Server 15.7.1 では、レプリケーション・パフォーマンスを強化するためにいくつかの変更が加えられました。

### 非同期パーサ、ASCII パッキング、および直接コマンド・レプリケーション

非同期パーサ、ASCII パッキング、およびインバウンドとアウトバウンドの直接コマンド・レプリケーションの各機能を同時に活用することで、データの変換と転送時にレプリケーション・プロセス全体が改善されます。

Replication Server 15.7 では、インバウンド・コマンドの直接レプリケーションを使用して、Replication Server EXEC モジュールと DIST モジュールとの間のインバウンド・レプリケーション・パスにおけるコマンド変換と I/O を低減できます。

Replication Server 15.7.1 では、非同期パーサ機能とアウトバウンド直接コマンド・レプリケーション機能により、Replication Agent とエグゼキュータ・スレッドとの間および DIST モジュールと DSI モジュールとの間のレプリケーション・パフォーマンスが向上し、その一方で、ASCII パッキングにより、ステابل・キューのストレージ消費量が削減されます。

次のいずれかの方法を使用してください。

- 非同期パーサ — エグゼキュータが追加のエグゼキュータ・スレッドを設定して Replication Agent からのコマンドをパースするのを Replication Agent が待機する時間を短縮します。
- ASCII パッキング — 非同期パーサと併用することで、インバウンド・キューにパックされたコマンドが消費するステابل・キューの記憶領域を低減します。
- インバウンド・コマンドの直接レプリケーション — Replication Server EXEC モジュールと DIST モジュールとの間のインバウンド・レプリケーション・パスにおけるコマンド変換と I/O を低減します。

- アウトバウンド・コマンドの直接レプリケーション – Replication Server DIST モジュールと DSI モジュールとの間のアウトバウンド・レプリケーション・パスにおけるコマンド変換と I/O を低減します。

これらの機能をすべて同時に使用することで、パフォーマンスが最大になりキューのストレージ消費量を低減できます。これらの機能を個別に設定するのではなく、**async\_parser** と **alter connection** を使用してこれらの機能をデフォルト値で同時に設定します。また、**async\_parser** を on に設定してから個々のパラメータを個別に設定して、パフォーマンスとリソース消費量を調整しバランスを取ることもできます。

『Replication Server 管理ガイド 第 2 巻』の「パフォーマンス・チューニング」の「チューニング・パラメータの使用についての注意事項」の「非同期パーサ、ASCII パッキング、および直接コマンド・レプリケーション」を参照してください。

## 利便性の向上

---

Replication Server 15.7.1 では、利便性を向上させるためにいくつかの変更が加えられました。

### ウォーム・スタンバイ環境と MSA 環境におけるカスタム・ファンクション文字列の複写定義の削減

Adaptive Server データベースだけがあるレプリケーション・システムでは、複写定義の目的がレプリケート・テーブルまたはストアド・プロシージャのカスタム・ファンクション文字列を指定することだけである場合、ウォーム・スタンバイ環境または Multi-Site Availability (MSA) 環境におけるプライマリ・テーブルまたはストアド・プロシージャの複写定義を作成する必要はありません。

Replication Server 15.7 では、複写定義の目的がプライマリキー・カラムあるいは引用符付きテーブル名またはカラム名を指定することだけである場合、ウォーム・スタンバイ環境または Multi-Site Availability (MSA) 環境におけるプライマリ・テーブルの複写定義を作成する必要はありません。『Replication Server 管理ガイド 第 1 巻』の「プライマリ・キー・カラムと引用符付きのテーブル名またはカラム名」を参照してください。

Replication Server 15.7.1 では、レプリケート・テーブルまたはスタンバイ・テーブルあるいはストアド・プロシージャに対してカスタム・ファンクション文字列を直接作成できます。これらのテーブルまたはストアド・プロシージャの複写定義を定義する必要はありません。このようなファンクション文字列は、「ターゲットスコープ・ファンクション文字列」と呼ばれます。これにより、ウォーム・スタンバイ環境または MSA 環境で複写定義を作成する必要性がさらに低下します。

『Replication Server 管理ガイド 第2巻』の「ファンクション文字列の作成」を参照してください。

### ストアド・プロシージャのサポート

**rs\_helpobjstring** ストアド・プロシージャを使用して、ターゲットスコープ・ファンクション文字列に関する情報を表示します。『Replication Server リファレンス・マニュアル』の「RSSD ストアド・プロシージャ」の「**rs\_helpobjstring**」を参照してください。

### システム・テーブルのサポート

Replication Server には、ターゲットのテーブルまたはストアド・プロシージャに関する情報を格納するための `rs_targetobjs` システム・テーブルが用意されています。Replication Server では、`rs_targetobjs` の値を他の Replication Server の RSSD にレプリケートしません。`rs_targetobjs` は STS プライマリ・キャッシュ・キーとして STS キャッシュ (`objname`、`objowner`、`dbid`、`objtype`) 内にあります。**sts\_full\_cache\_rs\_targetobjs** を使用して、テーブルのフル・キャッシュの有効または無効を切り替えます。

```
configure replication server set sts_full_cache_rs_targetobjs to {on|off}
```

**sts\_full\_cache\_rs\_targetobjs** のデフォルト値は off です。

Replication Server は、`rs_funcstrings` テーブルの `attributes` カラムのデータ型を `smallint` から `int` に変更します。

『Replication Server リファレンス・マニュアル』の「Replication Server システム・テーブル」の「`rs_targetobjs`」を参照してください。

## アップグレードの簡素化

Replication Server では、ユーザ・データベースと RSSD のアップグレード・プロセスが簡素化されています。

Replication Server 15.6 では、Sybase Central™ の Replication Manager プラグインではなく **sysadmin upgrade, "route"** を使用して、ルートをシームレスにアップグレードできます。

Replication Server 15.7.1 では、さらに簡素化されたアップグレード・プロセスを使用して、Embedded Replication Server システム・データベース (ERSSD) または Replication Server システム・データベース (RSSD) を自動的にアップグレードできます。Replication Server は、メンテナンス・ユーザのアクセス権限を持つ各ユーザ・データベースにも接続し、データベースにアップグレード・スクリプトを自動的に適用します。

『Replication Server 設定ガイド』の「Replication Server のアップグレードまたはダウングレード」の「Replication Server のアップグレード」の「RSSD または ERSSD およびユーザ・データベースへのアップグレード」を参照してください。

## システム管理ツール

---

Replication Server 15.7.1 では、Replication Server インストール・メディアに Replication Manager と Replication Monitoring Services (RMS) が含まれていません。

Replication Server 15.7.1 には、レプリケーション・システムの管理に使用できる Sybase Control Center が搭載されています。『Sybase Control Center 3.2.6 for Replication』を参照してください。

Replication Manager と Replication Monitoring Services を継続して使用するには、Sybase Web サイトの [Replication Server] からこれらのツールをダウンロードしてインストールします。

Replication Manager プラグインから Sybase Central へのルート・アップグレードは、廃止されました。代わりに、**sysadmin upgrade "route"** Replication Server コマンドを使用します。『Replication Server 設定ガイド』の「ルートのアップグレード」を参照してください。

### 参照：

- アップグレードの簡素化 (18 ページ)
- Sybase EBF と Maintenance レポートのダウンロード (207 ページ)





# Replication Server 15.7 の新機能

Replication Server® 15.7 では、パフォーマンス、利便性、プロセス、データベース・サポートが強化されました。

## Replication Server ライセンス

---

Replication Server 15.7 は、Enterprise Edition としてリリースされます。

Replication Server 15.7 では、Sybase® IQ への Real-Time Loading (RTL) レプリケーションが強化されました。Replication Server Real-Time Loading Edition を使用している場合、Replication Server 15.7 にアップグレードすることで強化された RTL 機能を使用できます。

『Replication Server インストール・ガイド』の「インストールの計画」の「ライセンスの取得」を参照してください。

参照：

- Real-Time Loading と High-Volume Adaptive Replication (35 ページ)

## マルチパス・レプリケーション

---

複数のレプリケーション・パスを使用して、レプリケーションのスループットとパフォーマンスを向上させ、競合を低減します。

マルチパス・レプリケーションでは、さまざまなストリームを介したデータのレプリケーションをサポートすると同時に、パス内でのデータ整合性を維持しますが、さまざまなパス間でのコミット順には従いません。

レプリケーション・パスは、Replication Server とプライマリ・データベースまたはレプリケート・データベースの間のコンポーネントとモジュールをすべて含んでいます。マルチパス・レプリケーションでは、プライマリ・データベースから 1 つまたは複数の Replication Server への複数の Replication Agent コネクションのために、複数のプライマリ・レプリケーション・パスを作成できます。また、1 つまたは複数の Replication Server からレプリケート・データベースへのコネクションのために、複数のレプリケート・パスを作成できます。マルチパス・レプリケーションは、ウォーム・スタンバイ環境と Multi-Site Availability (MSA) 環境で設定できます。トランザクションを Replication Server 間の専用ルートで伝達して、共有ルート上での輻輳を回避できます。また、プライマリ・データベースから Replication Server を経由してレプリケート・データベースに至るエンドツーエンド

のレプリケーション・パスをオブジェクト (テーブルやストアド・プロシージャなど) 専用にすることができます。

『Replication Server 管理ガイド 第2巻』の「パフォーマンス・チューニング」の「マルチパス・レプリケーション」を参照してください。

### ライセンス

マルチパス・レプリケーションは、Advanced Services Option の一部としてライセンスされます。『Replication Server インストール・ガイド』の「インストールの計画」の「ライセンスの取得」を参照してください。

### システムの稼働条件

Replication Server では、プライマリ・データ・サーバが Adaptive Server 15.7 以降の Adaptive Server データベース間でのマルチパス・レプリケーションをサポートしません。

## パフォーマンスの強化

---

Replication Server 15.7 では、レプリケーション・パフォーマンスを強化するためのいくつかの変更が加えられました。

### SQM コマンド・キャッシュ

---

SQM コマンド・キャッシュを使用して、ディストリビュータ・スレッドが直接取得できるエグゼキュータ・スレッドからの解析済みデータを格納することで、レプリケーション・パフォーマンスが向上します。

エグゼキュータ・スレッドは、LTL コマンドを Replication Agent から Replication Server に転送します。また、LTL コマンドを解析しそれらを内部解析形式で格納します。解析済みデータはバイナリ形式でパックされます。エグゼキュータ・スレッドは、このバイナリ・データを SQM スレッドに送信して、エグゼキュータ・スレッドが Replication Agent から新しいデータを受信できるようにします。SQM スレッドは、バイナリ・データがインバウンド・ステープル・キューに書き込まれるまで、データを SQM キャッシュに格納します。ディストリビュータ・スレッドは、バイナリ・データを取得し、データを元の形式にリストアし、データの送信先を決定します。

`cmd_direct_replicate` を on に設定すると、エグゼキュータ・スレッドはバイナリ・データとともに内部解析データを送信します。Replication Server は、解析済みデータを独立した SQM コマンド・キャッシュに格納します。SQM コマンド・キャッシュ内の解析済みデータは、SQM キャッシュに格納されたバイナリ・データにマッピングされます。ディストリビュータ・モジュールは、必要に応じて解析済

みデータからデータを直接取得して処理できるので、バイナリ・データの解析に費やされる時間を節約できます。

**sqm\_cmd\_cache\_size** パラメータと **sqm\_max\_cmd\_in\_block** パラメータを使用して、SQM コマンド・キャッシュ・メモリ設定を設定します。

『Replication Server 管理ガイド 第2巻』の「パフォーマンス・チューニング」の「チューニング・パラメータの使用についての注意事項」の「SQM コマンド・キャッシュ」を参照してください。

## エグゼキュータ・コマンド・キャッシュ

Sybase RepAgent が最初にプライマリ Adaptive Server データベース・テーブルの **insert**、**delete**、または **update** LTL コマンドを送信するとき、エグゼキュータ・コマンド・キャッシュを使用して、そのテーブルのカラム名とデータ型をキャッシュします。

カラム名やデータ型などのメタデータは、RepAgent が **insert**、**delete**、または **update** コマンドに関連付けられたデータとともに送信するテーブル・スキーマに含まれています。ただし、キャッシュでは、以下のようになります。

- RepAgent が **insert**、**update**、**delete** コマンドに関連付けられたメタデータおよびデータを送信するのは、RepAgent が起動してから、または Replication Server とのコネクションが再開してから、RepAgent がその特定のテーブルに対するオペレーションを初めて処理するときだけです。RepAgent は、それ以降にそのテーブルのトランザクションを処理するとき、テーブル・メタデータを送信しません。
- スキーマ定義をすべて維持するのに十分なメモリが RepAgent がない場合、RepAgent はメタデータとデータを再送信できます。
- RepAgent は、Adaptive Server の **alter table** オペレーションの後など、テーブル・スキーマの変更後に特定のテーブルでの変更を処理するとき、テーブルのメタデータとデータを送信します。

同じテーブルに対するそれ以降のオペレーションをレプリケートするため、RepAgent はカラム・データのみを送信します。これは、Replication Server エグゼキュータ・コマンド・キャッシュがメタデータを格納しているからです。RepAgent のメタデータの低減と Replication Server エグゼキュータ・コマンド・キャッシュの使用を組み合わせることで、レプリケーション・パフォーマンスが向上します。その要因は、キャッシュに以下の性質があるためです。

- RepAgent がメタデータをログ転送言語 (LTL) パケットにパックするために費やす時間を短縮します。
- 各パケットで送信されるデータの量を増やすことで、ネットワーク・トラフィックを低減します。

- RepAgent は節約した時間をメタデータのパックではなくプライマリ・データベース・ログのスキャンに充てることができます。
- Replication Server エグゼキュータは多くのカラムがあるテーブルを効率よく処理できます。

---

**注意：** キャッシュには、**insert**、**update**、または **delete** オペレーションによって変更されたテーブルのメタデータのみが格納されます。

---

『Replication Server 管理ガイド 第2巻』の「パフォーマンス・チューニング」の「チューニング・パラメータの使用についての注意事項」の「エグゼキュータ・コマンド・キャッシュ」を参照してください。

### システムの稼働条件

テーブル・メタデータを低減するには、LTL バージョン 740 以降および Adaptive Server 15.7 以降が必要です。

## sqm\_cache\_size の上限値

**sqm\_cache\_size** の上限値が、以前の 512 から 4096 に引き上げられました。

**sqm\_cache\_size** の上限値を引き上げることで、Replication Server がさらに多くのトランザクションをキャッシュに保持できるようになりました。これは、キャッシュ検索の全体的なパフォーマンスに影響を及ぼします。

- 『Replication Server リファレンス・マニュアル』の「Replication Server コマンド」の「**configure replication server**」
- 『Replication Server 管理ガイド 第2巻』の「パフォーマンス・チューニング」の「パフォーマンスに影響する設定パラメータ」の「パフォーマンスに影響する Replication Server パラメータ」

を参照してください。

## セグメントを削除するための専用デーモン

セグメントを削除する専用デーモンを有効にし、インバウンドとアウトバウンドのキュー処理のパフォーマンスを向上させるには、**sqm\_async\_seg\_delete** を on に設定します。

**configure replication server** を使用してサーバ・レベルで **sqm\_async\_seg\_delete** を設定する必要があります。

デフォルト値は on

このパラメータ設定の変更を有効にするには、Replication Server を再起動する必要があります。

`sqm_async_seg_delete` はデフォルトで `on` なので、バージョン 15.7 以降にアップグレードする場合は Replication Server に大規模なパーティションが必要になる可能性があります。

- 『Replication Server 設定ガイド』の「Replication Server のインストールと設定の準備」の「複写システムのプラン作成」の「各 Replication Server の最初のディスク・パーティション」
- 『Replication Server 管理ガイド 第 1 巻』の「Replication Server の技術的概要」の「Replication Server でのトランザクション処理」の「ステープル・キュー」の「ステープル・キューのパーティション」
- 『Replication Server リファレンス・マニュアル』の「Replication Server コマンド」の「`alter partition`」

を参照してください。

## 利便性とプロセスの向上

---

Replication Server 15.7 では、利便性とプロセスを向上させるためにいくつかの変更が加えられました。

### 複写定義の使用の削減

Adaptive Server データベースだけがあるレプリケーション・システムでは、ウォーム・スタンバイ環境または Multi-Site Availability (MSA) 環境におけるテーブルの複写定義の必要性を削減できます。これは、Adaptive Server 用 RepAgent でログ転送言語 (LTL) を使用して、引用符が付く可能性のあるテーブル名またはカラム名と、テーブル・カラムがテーブルのプライマリ・キーの一部であるかどうかを指定するためです。

RepAgent がプライマリ・キーと引用符付き識別子情報を Replication Server に送信するので、複写定義の目的がプライマリ・キーと引用符付き識別子情報を指定することのみである場合、複写定義は必要ありません。複写定義の必要性が低下することにより、多くのテーブルがあるデータベース、多くのカラムがあるテーブル、またはスキーマを頻繁に変更するテーブルが関係する複写環境の管理が容易になります。現在、複写定義がないテーブルのレプリケーション・パフォーマンスは向上します。これは、RepAgent が Replication Server にテーブルのプライマリ・キー情報を直接提供し、結果として Replication Server が `update`、`delete`、および `select` コマンドの `where` 句にプライマリ・キー・カラムだけをパックするからです。

複写定義の目的が以下の一部またはすべてを指定することのみである場合は、プライマリ・テーブルの複写定義を作成する必要はありません。

- プライマリキー・カラム
- 引用符が付く可能性があるテーブル名またはカラム名があるとき。

複写定義を低減するようレプリケーション・システムを設定する方法の詳細については、『Replication Server 管理ガイド 第1巻』の「MSAを使用した複写オブジェクトの管理」の「複写定義およびサブスクリプションの使用の削減」を参照してください。

### システムの稼働条件

RepAgent は、LTN バージョン 740 以降でのみプライマリ・キー情報および引用符付き識別子情報を送信します。これは、Adaptive Server 15.7 以降および Replication Server 15.7 以降でサポートされています。

## rs\_functions に加えられた変更

rs\_functions の代わりに、Replication Server 15.7 に新しいシステム・テーブルがいくつか追加されました。

Replication Server 15.7 では、以下のシステム・テーブルが追加されています。

- rs\_clsfunctions - クラス全体の関数を格納します。  
『リファレンス・マニュアル』の「Replication Server システム・ファンクション」の「rs\_clsfunctions」を参照してください。
- rs\_objfunctions - オブジェクト全体の関数を格納します。  
『リファレンス・マニュアル』の「Replication Server システム・ファンクション」の「rs\_objfunctions」を参照してください。
- rs\_asyncfuncs - 複写定義に対するユーザ定義関数についての情報を格納します。同じローが rs\_objfunctions にも格納されます。  
『リファレンス・マニュアル』の「Replication Server システム・ファンクション」の「rs\_asyncfuncs」を参照してください。

15.7 より前のバージョンの Replication Server では、rs\_functions がクラス全体とオブジェクト全体の関数に関する情報を格納していました。これらの関数は、*funcname* でキャッシュされます。多くの複写定義が同じ関数名 (**rs\_insert**、**rs\_update**、**rs\_delete** など) を共有しているので、オブジェクト全体の関数のローを *funcname* でキャッシュしないでください。バージョン 15.7 以降では、rs\_functions のローは、2つのカテゴリに分類され、rs\_clsfunctions と rs\_objfunctions に格納されます。

バージョン 15.7 以降では、rs\_functions はテーブルではなくなりました。Replication Server の下位互換性をサポートするため、rs\_functions は rs\_clsfunctions と rs\_objfunctions の union のビューとして維持されています。

『Replication Server リファレンス・マニュアル』の「Replication Server システム・テーブル」の「rs\_asyncfuncs」、「rs\_clsfunctions」、および「rs\_objfunctions」を参照してください。

- 『リファレンス・マニュアル』の「Replication Server システム・テーブル」の「rs\_asyncfuncs」
- 『リファレンス・マニュアル』の「Replication Server システム・テーブル」の「rs\_clsfunctions」
- 『リファレンス・マニュアル』の「Replication Server システム・テーブル」の「rs\_objfunctions」

## メモリ消費の制御

メモリ消費を制御し、使用できるメモリの指定値を超えても Replication Server が自動的に停止しないように、設定パラメータ **memory\_limit** が強化されました。Replication Server のスレッドのメモリ制御動作は、新しい設定パラメータ **memory\_control** によって管理されます。

バージョン 15.7 では、メモリ消費量が使用できる合計メモリの指定スレッシュールド・パーセンテージを超えると警告メッセージを表示するように Replication Server を設定できます。この強化されたメモリ管理機能は、2つの新しい設定可能なパラメータによりサポートされています。

- **mem\_warning\_thr1** – この値を超えると最初の警告メッセージが生成される、合計メモリのスレッシュールド・パーセンテージを指定します。  
デフォルト値は **memory\_limit** 値の 80%。  
範囲：1 – 100
- **mem\_warning\_thr2** – この値を超えると 2 番目の警告メッセージが生成される、合計メモリのスレッシュールド・パーセンテージを指定します。  
デフォルト値は **memory\_limit** 値の 90%。  
範囲：1 – 100

さらに、Replication Server 15.7 では、**memory\_limit** で指定された使用できるメモリを超過したときの Replication Server の自動停止の問題にも対応しています。Replication Server でメモリを大量に必要とするスレッドは、次のとおりです。

- DSI
- EXEC
- SQT

バージョン 15.7 では、これらのスレッドはメモリ使用量チェックを実行してから新しいデータを受信または処理することで、メモリ制御を実行します。メモリ制

御時にメモリ使用量が多いことが判明すると、次の動作によりスレッド機能が調整されます。

- スレッドによる新しいデータのグループ化を停止し、既存データのクリーニングと処理を行います。または、
- 空きメモリが確保されるまで新しいデータを受信しないよう、スレッドをスリープ・モードにします。

EXEC、DST、および SQT の各スレッドでフロー制御を行うための新しいサーバレベルの設定パラメータが3つあります。

- **mem\_thr\_dsi** – DSI スレッドによる SQT キャッシュの入力を停止する合計メモリのパーセンテージを指定します。  
デフォルト値は **memory\_limit** 値の 80%。
- **mem\_thr\_exec** – EXEC スレッドによる RepAgent からのコマンドの受信を停止する合計メモリのパーセンテージを指定します。  
デフォルト値は **memory\_limit** 値の 90%。
- **mem\_thr\_sqt** – SQT スレッドでキャッシュからの最大トランザクションをフラッシュする合計メモリのパーセンテージを指定します。  
デフォルト値は **memory\_limit** 値の 85%。

スレッドのメモリ制御動作は、新しいサーバレベルの設定パラメータ **memory\_control** によって管理されます。**memory\_control** の有効な値は、enable (デフォルト値) または disable です。これにより、Replication Server はメモリ消費を制御し、メモリの問題で停止することはなくなりました。

これらの新しい設定パラメータのデフォルト値を変更するには、**configure replication server** を使用します。デフォルト値または既存の値を表示するには、**admin config** を使用します。

- 『Replication Server リファレンス・マニュアル』の「Replication Server コマンド」の「**configure replication server**」
- 『Replication Server 管理ガイド 第2巻』の「パフォーマンス・チューニング」の「パフォーマンスに影響する設定パラメータ」の「パフォーマンスに影響する Replication Server パラメータ」

を参照してください。

スレッド情報をモニタする

スレッドのメモリ制御動作に関する情報を表示するには、**admin who** を使用します。

状態	説明
Controlling Mem	スレッドはメモリ制御を実行しています。



状態	説明
Sleeping For Mem	空きメモリが確保されるまで、スレッドはスリープします。

『Replication Server リファレンス・マニュアル』の「Replication Server コマンド」の「**admin who**」を参照してください。

#### メモリ管理統計

**admin stats** を使用して、メモリ管理統計を表示します。メモリ・カウンタは、rsh モジュールで有効です。メモリ・カウンタをレポートするには、次のコマンドを使用します。

```
admin stats,rsh display_name instance_id
```

構文の説明は次のとおりです。

- *display\_name* – カウンタ名。有効な表示名を確認するには、**rs\_helpcounter** を使用します。*display\_name* は必ず *module\_name* と組み合わせて使用します。
- *instance\_id* – SQT や SQM などのモジュールの特定のインスタンスを識別します。インスタンス ID を確認するには、**admin who** を実行し、*Info* カラムを表示します。rsh モジュールの場合、*SPID* を使用してください。*SPID* を確認するには、**admin who** を実行し、*Spid* カラムを表示します。

『Replication Server リファレンス・マニュアル』の「Replication Server コマンド」の「**admin stats**」を参照してください。

## Unicode の機能強化

新しいデータ・サーバ・インタフェース (DSI) 設定パラメータである

**unicode\_format** が追加され、Unicode データを U&" 形式で送信できるようになりました。

以前のバージョンの Replication Server では、文字セットを UTF-8 に設定しないと、unichar、univarchar、unitext などの Unicode データ型をすべてレプリケートできませんでした。Replication Server 15.7 では、この制限がなくなり、Unicode データを次のいずれかの形式で送信できるようになりました。

- 文字列
- U&"

Adaptive Server Enterprise ではさらに、次の Unicode データ形式もサポートしていません。

Replication Server を設定するときは、**unicode\_format** を次の値のいずれかに設定します。

## Replication Server 15.7 の新機能

- `string-Unicode` 文字を文字列形式に変換します。たとえば、文字列 "hello" は "hello" として送信されます。
- `ase-Unicode` 文字を `U&''` 形式で送信します。たとえば、文字列 "hello" は "U&¥0068¥0065¥006c¥006c¥006f" として送信されます。2 バイト Unicode 値は、Adaptive Server Enterprise が要求するネットワーク順序で送信されます。

`unicode_format` は、完全な動的パラメータではありません。変更を反映するためにコネクションを再開するか、Replication Server を再起動します。

『Replication Server リファレンス・マニュアル』の「Replication Server コマンド」の「**configure replication server**」を参照してください。

## SySAM ライセンス情報の要求

Replication Server 15.7 では、新しいコマンド `sysadmin lmconfig` が導入されました。これは、ライセンス管理に関連する情報の設定と表示を行うためのコマンドです。

『Replication Server リファレンス・マニュアル』の「Replication Server コマンド」の「`sysadmin lmconfig`」を参照してください。

## サブスクリプション名の拡張

Replication Server 15.7 以降では、`rs_subscriptions` でのサブスクリプションの文字数の制限が以前の 30 文字から 255 文字に引き上げられています。

`rs_subscriptions` システム・テーブルの `subname` カラムが、`varchar(30)` から `varchar(255)` に変更されました。

『Replication Server リファレンス・マニュアル』の「Replication Server システム・テーブル」の「`rs_subscriptions`」を参照してください。

### *混合バージョンの複写環境*

混合バージョンの複写環境では、レプリケート Replication Server とプライマリ Replication Server のバージョンがどちらも 15.7 以降である必要があります。

## 後続ゼロの削除

`varbinary_strip_trailing_zeros` を off に設定すると、`varbinary` 値の後続ゼロのレプリケーションが有効になります。

デフォルト値の on に設定すると、後続ゼロが `varbinary` 値から削除されます。Replication Server 15.7 より前のすべてのバージョンの動作はこのデフォルト設定でした。

サーバ・レベルで `configure replication server` を使用して `varbinary_strip_trailing_zeros` を設定する必要があります。パラメータの変更を有効

にするために、Replication Server を再起動する必要も接続をサスペンドしてから再開する必要もありません。

## Sybase Control Center for Replication および Data Assurance

---

Sybase Control Center は、大規模な Sybase エンタープライズ・サーバのリアルタイム・パフォーマンス、ステータス、および可用性モニタリングのための、単一の包括的な Web 管理コンソールです。履歴モニタリング、スレッショルドベースのアラートおよび通知、アラートベースのスクリプト実行、およびパフォーマンスおよび使用の傾向を識別するためのインテリジェント・ツールが含まれます。

Sybase Control Center for Replication では、特定のサーバの可用性とステータスを表示するためにサーバ・モニタとヒート・チャートを使用し、ステータス情報を簡単に確認できます。サーバ・モニタには、サーバのバージョンやプラットフォームなど、高レベルの情報が表示されます。また、レプリケーション・パフォーマンスのトラブルシューティングに役立つ重要なパフォーマンス・カウンタも表示されます。

Sybase Control Center for Data Assurance (DA) では、プライマリ・データベースから 1 つまたは複数のレプリケート・データベースへのデータ比較をサポートしています。比較タスクをスケジュールすることもできます。Sybase Control Center for Data Assurance では、独自の SQL Anywhere® データベースを使用し、展開します。これには、システム設定とその他のさまざまな設定、タスク、およびタスク履歴が格納されます。自動調整オプションにより、DA ジョブの作成時に、ローの欠落、孤立、および矛盾が示されます。

データ・フローの制御とレプリケーション・パラメータの設定を支援してサーバ・パフォーマンスを向上させるため、Sybase Control Center for Replication にはどのレプリケーション・モニタからも簡単にアクセスできるクイック管理ツールが搭載されています。

Sybase Control Center for Replication には、モニタ以外にトポロジ・ビューも用意されており、サーバ、サーバ間接続、環境内でのデータ・フロー、およびレプリケーション・パスのソースとターゲットがグラフィカルに表示されます。パフォーマンス・カウンタをモニタリングするためにグラフとチャートも使用できます。

Sybase Control Center 3.2.4 の「Sybase Control Center for Replication」と「Sybase Control Center for Data Assurance」を参照してください。

### ライセンス

Sybase Control Center (Replication Server、Replication Server Data Assurance オプションなど) で管理される製品の有料ライセンスをお持ちの場合、Sybase Control Center のライセンスは無料で提供されます。評価ライセンスも入手可能です。『Sybase Control Center インストール・ガイド』の「インストールの計画」の「ライセンスの取得」を参照してください。

### Sybase Control Center for Replication と Sybase Central for Replication の機能の比較

レプリケーション・システム管理について、Sybase Control Center for Replication バージョン 3.2.3 (SCC Replication) と、Sybase Central and Replication Monitoring Services (RMS) の Replication Manager Plug-in (RMP) バージョン 15.0 以降のサポートを比較します。

機能	SCC for Repli- cation 3.2.3	RMP 15.x	RMS 15.x
Replication Server と Replication Agent のステータスのモニタリング	X	X	X
Replication Server と Replication Agent の可用性のモニタリング	X	X	X
レプリケーション・パフォーマンスのモニタリング	X		
エンドツーエンド遅延のモニタリング ( <b>rs_ticket</b> ハートビート)	X		X
ASE のプライマリとレプリケートの各データベースを含むパスのモニタリング	X		
Sybase IQ レプリケート・データベースを含むパスのモニタリング	X		
Sybase 以外のプライマリとレプリケートの各データベース (Oracle、Microsoft SQL Server、および Linux、UNIX、Windows 向け IBM DB2) を含むパスのモニタリング	X		
Replication Server 統計のモニタリング	X	X	X
パフォーマンス統計履歴情報の保存	X		

機能	SCC for Repli- cation 3.2.3	RMP 15.x	RMS 15.x
パフォーマンス統計履歴情報の図示	X		
レプリケーション・トポロジの表示	X		
レプリケーション・パスのトレース	X		
ヒート・チャートの表示	X		
アラート通知の設定	X		X
Replication Server ユーザ・セキュリティとロールの設定	X		
Replication Server と Replication Agent のバージョン 15.7 以降のライセンス情報の表示	X		
Replication Server と Replication Agent の設定	X	X	X
Replication Server と Replication Agent のサスペンドと再開	X	X	X
Replication Server と Replication Agent の停止		X	X
ユーザが選択した方法でのサーバのグループ化(例:地域別、機能別など)	X		X
RCL と SQL エディタを使用した Replication Server でのコマンドの実行		X	
コマンドライン・インタフェース (isql など) を使用した API コマンドの実行			X
ウィザードを使用した Adaptive Server 複写環境の設定の簡略化		X	
Adaptive Server RepAgent スレッドの設定		X	
Adaptive Server 自動マテリアライゼーションの設定		X	
Replication Server コネクションの作成と削除		X	
Replication Server 論理コネクションの作成と削除		X	
Replication Server ルートの作成と削除		X	

機能	SCC for Repli- cation 3.2.3	RMP 15.x	RMS 15.x
Multi-Site Availability (MSA) の複写定義とサブスクリプションの作成と削除		X	
Replication Server の複写定義とサブスクリプションの作成、変更、削除		X	
Replication Server ユーザの作成、変更、削除		X	
Replication Server ルートのアップグレード		X	
Replication Server キュー・データの表示		X	

## Adaptive Server レプリケーション機能サポートの強化

Replication Server 15.7 は Adaptive Server レプリケーションをサポートしています。

### RepAgent の自動起動

Adaptive Server 15.5 ESD #5 以降では、**auto start** パラメータと **sp\_config\_rep\_agent** を使用して、Adaptive Server が再起動しデータベースをリカバリしたら RepAgent が自動的に起動するかどうかを指定できます。

RepAgent を **sp\_start\_rep\_agent** で以前に少なくとも 1 度起動したことがあり、**sp\_stop\_rep\_agent** で停止しなかった場合、Adaptive Server が再起動すると RepAgent は自動的に起動します。Adaptive Server 15.5 ESD #5 以降では、**sp\_config\_rep\_agent** を使用して **auto start** を true に設定した場合も、RepAgent が自動的に起動します。

**sp\_stop\_rep\_agent** で RepAgent を停止した場合は、以前に **auto start** を true に設定したことがないかぎり、データベースがオンラインになっても RepAgent は自動的に起動しません。それ以外の場合、RepAgent を起動するには **sp\_start\_rep\_agent** を実行する必要があります。

構文は次のとおりです。

```
sp_config_rep_agent
[...
'auto start'[, 'true' | 'false']]
```

Adaptive Server の再起動時に RepAgent を自動的に起動するには、true に設定します。デフォルトは false です。

『Replication Server 管理ガイド 第1巻』の「RepAgent の管理と Adaptive Server のサポート」で、以下を参照してください。

- RepAgent に影響する設定パラメータ
- RepAgent の起動
- RepAgent の停止

## **Real-Time Loading と High-Volume Adaptive Replication**

Replication Server 15.7 では、Real-Time Loading (RTL) と High-Volume Adaptive Replication (HVAR) のパフォーマンスと利便性が向上しています。

### ライセンス

RTL を使用した Sybase IQ へのレプリケーションは、Real-Time Loading オプションの一部として実行可能です。

HVAR を使用した Adaptive Server へのレプリケーションは、Advanced Services Option の一部として実行可能です。

『Replication Server インストール・ガイド』の「インストールの計画」の「ライセンスの取得」を参照してください。

### データベースとプラットフォームのサポート

- Sybase IQ – Real-Time Loading を使用して Sybase IQ バージョン 12.7 ESD #3 以降に複写できます。Sybase IQ のバージョンとプラットフォームのサポートに関する最新情報については、『Replication Server リリース・ノート』の「製品の互換性」の「Replication Server の相互運用性」を参照してください。
- Adaptive Server – Replication Server は、Adaptive Server バージョン 15.0.3 またはバージョン 15.5 以降から Sybase IQ へのレプリケーションをサポートしています。
- Oracle – Replication Server は Sybase IQ から Oracle 10g および 11g へのレプリケーションをサポートしています。Replication Server Options 15.5 の『Release Bulletin Replication Agent 15.5』の「Product Summary」の「Compatible Products」を参照してください。

### 参照：

- Replication Server ライセンス (21 ページ)

## メモリ消費の制御

RTLでのメモリ消費を低減するため、Replication Serverではフル・インクリメンタル・コンパイルをサポートしており、最終的な変更を保管するデータベースのサイズを制御できます。HVARでのメモリ消費を低減するため、コンパイル可能なグループのサイズを制御することもできます。

### RTLのフル・インクリメンタル・コンパイル

フル・インクリメンタル・コンパイルを使用できるのは、RTLのみです。フル・インクリメンタル・コンパイルにより、RTLのレプリケーション・パフォーマンスが向上しますが、これは多くのコマンドを含む大規模なコンパイル可能なトランザクションの処理中のメモリ消費が低減された結果です。

フル・インクリメンタル・コンパイルでは、**insert**、**delete**、または**update**の混合オペレーションを含む大規模なトランザクションをコンパイルできます。

Replication Serverでは、フル・インクリメンタル・コンパイルを使用して大規模なコンパイル可能なトランザクションをレプリケート・データベースに適用します。その際、最終的な変更を保管する複数のインメモリ・データベース・インスタンスを使用します。

『Replication Server 異機種間複写ガイド』の「レプリケート・データ・サーバとしての Sybase IQ」の「Sybase IQ レプリケート・データベースの設定」の「メモリ消費の制御」の「フル・インクリメンタル・コンパイル」を参照してください。

### RTLの最終的な変更を保管するデータベースのサイズの制御

最終的な変更を保管するデータベースのメモリ消費を低減します。それには、最終的な変更を保管するデータベースのサイズがスレッシュールド・サイズに達したらそのデータベースをトリガして、データをレプリケート・データベースにフラッシュします。

Replication Serverが生成できる最終的な変更を保管するデータベースの最大サイズを制御できます。このサイズが設定されたスレッシュールドに達すると、Replication Serverは最終的な変更を保管するデータベースに作成中のコンパイル済みトランザクションに新しいコマンドとトランザクションをコンパイルする処理を停止し、コンパイル済みグループをレプリケート・データベースに一括して適用し、最終的な変更を保管するデータベースをクリアし、そのデータベースが消費していたメモリを解放します。

『Replication Server 異機種間複写ガイド』の「レプリケート・データ・サーバとしての Sybase IQ」の「Sybase IQ レプリケート・データベースの設定」の「メモリ消費の制御」の「最終的な変更のデータベースのサイズ」を参照してください。



### HVAR のコンパイル可能なグループのサイズの制御

コンパイル可能な大規模トランザクションのサイズのスレッシュホールドを設定することで、メモリ消費を低減し、パフォーマンスを向上させます。

スレッシュホールドに達すると、Replication Server は連続レプリケーション・モードを使用して大規模トランザクションを適用します。Replication Server は、引き続き HVAR を使用して小規模なコンパイル可能なトランザクションをグループにコンパイルし、グループ・サイズが設定されたスレッシュホールドに達するとすぐに、コンパイル済みグループをレプリケート・データベースに適用します。

『Replication Server 管理ガイド 第2巻』の「パフォーマンス・チューニング」の「Advanced Services Option」の「High Volume Adaptive Replication」の「メモリ消費の制御」を参照してください。

## Sybase IQ データベース・オプションの設定

`rs_session_setting` 関数と `create function string` コマンドを組み合わせることで、Sybase IQ レプリケート・データベースに接続している間の Sybase IQ パラメータの値を設定できます。たとえば、パフォーマンスを最適化するパラメータ値を設定できます。

『Replication Server 異機種間複写ガイド』の「レプリケート・データ・サーバとしての Sybase IQ」の「Sybase IQ レプリケート・データベースの設定」の「Replication Server インストール」の「Sybase IQ データベース・オプションの設定」を参照してください。

## スキーマ変換とデータ型変換

RTL または HVAR では、プライマリ・データベースおよびレプリケート・データベースのスキーマまたはカラムのデータ型が異なっていても、レプリケーションをサポートします。

HVAR と RTL を使用して、以下のレプリケートが可能です。

- プライマリ・テーブルのカラムのサブセットをレプリケート・テーブルにレプリケートする。
- カラムとテーブルをレプリケートする。これは、複写定義を使用することで、プライマリとレプリケートのテーブル名とカラム名が異なる場合でも可能です。
- カラムをレプリケートする。これは、プライマリとレプリケートのカラム・データ型が異なる場合でも可能です。

---

**注意：**異なるデータ型間での HVAR と RTL のレプリケーション・サポートは、Replication Server が連続モード・レプリケーションで提供している既存のカラムレベルのトランザクション・サポートと同等です。

---

- プライマリ・テーブルより多くのカラムがあるテーブルへのレプリケート。
  - Adaptive Server レプリケート・テーブル – レプリケート・テーブルで Replication Server による入力を行わないカラムの **NULL** 属性を設定します。レプリケート Adaptive Server カラムのデフォルト値を定義する場合、カラムの **NULL** 属性は設定する必要がありません。これは、レプリケート Adaptive Server がそのカラムにデフォルト値を自動的に入力するからです。
  - Sybase IQ レプリケート・テーブル – レプリケート・テーブルで Replication Server による入力を行わないカラムの **NULL** 属性を設定します。レプリケート Sybase IQ カラムのデフォルト値を定義する場合、カラムの **NULL** 属性は設定する必要がありません。これは、レプリケート Sybase IQ がそのカラムにデフォルト値を自動的に入力するからです。

---

**注意：** RTL と HVAR では、カスタマイズされたファンクション文字列を使用してデータがレプリケート・データベースにレプリケートされる方法を変更することはサポートされていません。

---

## パラメータのデフォルト値の変更

---

Replication Server 15.7 では、いくつかのパラメータのデフォルト値が変更されています。Replication Server 15.7 にアップグレードする場合、Replication Server は以前のバージョンのデフォルト値を使用します。

表 4：パラメータのデフォルト値の変更

パラメータ	古い値	新しい値	バージョン 15.7 からのダウングレード	『Replication Server リファレンス・マニュアル』の「Replication Server コマンド」で参照
<b>dsi_compile_max_cmds</b>	100,000 個のコマンド	10,000	ダウングレードでは設定されていた値は変更されない。	<b>alter connection</b>
<b>num_msg_queues</b>	178 個の Open Server™ メッセージ・キュー	300	ダウングレードでは設定されていた値は変更されない。	<b>configure replication server</b>

パラメータ	古い値	新しい値	バージョン 15.7 からのダウングレード	『Replication Server リファレンス・マニュアル』の「Replication Server コマンド」で参照
num_msgs	45,568 個の Open Server メッセージ・キュー・メッセージ	91,136	ダウングレードでは設定されていた値は変更されない。	configure replication server
num_threads	50 個の Open Server スレッド	150	ダウングレードでは設定されていた値は変更されない。	configure replication server
queue_dump_buffer_size	1,000 バイト	32,768	ダウングレードでは設定されていた値は変更されない。	configure replication server
rsi_packet_size	2,048 バイト	4096	ダウングレードでは設定されていた値は変更されない。	alter route
sts_cachesize	100 ロー	1000	ダウングレードでは設定されていた値は変更されない。	configure replication server

パラメータ、例、使用方法の情報については、『Replication Server リファレンス・マニュアル』の「Replication Server コマンド」を参照してください。

## Replication Server Data Assurance オプション

Replication Server Data Assurance (DA) オプションは 2 つ以上の Adaptive Server データベース間におけるロー・データとスキーマを比較し、不整合を報告し、必要に応じて不整合を調整します。

Replication Server Data Assurance (DA) オプションは、Replication Server の別途ライセンス製品として使用でき、Replication Server バージョン 15.1 以降をサポートしません。

## Replication Server 15.7 の新機能

Replication Server Data Assurance オプションは、SySAM ライセンス・マネージャを通じてライセンスされ、複数のプラットフォームで使用できます。詳細については、Replication Server Data Assurance オプションのマニュアルを参照してください。

# Replication Server バージョン 15.6 ESD #1 の新機能

Replication Server 15.6 ESD #1 では、Replication Server が Sybase IQ InfoPrimer に統合されています。

Sybase IQ InfoPrimer には、データを変換して Sybase IQ データベースにロードする効果的な機能が搭載されていますが、その抽出機能にはレプリケート Sybase IQ データベースのデータを最新の状態に維持するために必要な Replication Server のリアルタイム・モニタリングがありません。Replication Server Real-Time Loading (RTL) 機能では、バルク・オペレーション処理とコンパイルされたオペレーションを使用して、高パフォーマンスのレプリケーションを達成しますが、Replication Server には Sybase IQ InfoPrimer のデータ変換機能とロード機能がありません。Replication Server と Sybase IQ InfoPrimer を統合することで、ソースとは異なるスキーマを持つレプリケート Sybase IQ データベースで Adaptive Server データのほぼリアルタイムのコピーを維持できます。

## Replication Server と Sybase IQ InfoPrimer の統合

---

Replication Server と Sybase IQ InfoPrimer の統合ソリューションは、初期データ・マテリアライゼーションと進行中のデータ処理という 2 つの部分で機能します。

### マテリアライゼーション

Replication Server と Sybase IQ InfoPrimer の統合ソリューションは、Adaptive Server プライマリ・データベースからレプリケート Sybase IQ データベースへのデータのノンアトミック・バルク・マテリアライゼーションを実行します。このマテリアライゼーションは、Replication Server バルク・マテリアライゼーション・オプションに基づいており、必要に応じてオートコレクションを使用します。

Sybase IQ InfoPrimer は、レプリケート Sybase IQ データベースにステージング・テーブルを作成し、各プライマリ・データベース・テーブルに対してマテリアライゼーション処理のデータ抽出手順を実行します。これらのステージ・テーブルに対して変換ストアド・プロシージャが実行され、結果はベース・テーブルに書き込まれます。ベース・テーブルは、エンドユーザ・テーブルとも呼ばれ、ビジネス分析に使用されます。

### 進行中のデータ処理

Replication Server は、指定されたテーブルに対してマテリアライゼーション・フェーズで作成された同じステージング・テーブルと変換ストアド・プロシ

ジャを使用します。可能な場合、Replication Server はオペレーションをコンパイルしてステー징・テーブルにロードします。その後、Replication Server は変換ストアド・プロシージャを実行して、ベース・テーブルを更新します。こうして、Replication Server はレプリケート Sybase IQ データベースにほぼリアルタイムのデータ・コピーを維持します。

## ライセンス

特別なライセンスの要件は Replication Server と Sybase IQ InfoPrimer の統合に適用されます。

表 5 : Replication Server と Sybase IQ InfoPrimer の統合ライセンス

製品	機能	説明	ライセンス
Replication Server 15.6 ESD #1	Real-Time Loading (RTL)	Adaptive Server から Sybase IQ へのレプリケーションを可能にします。  <b>注意：</b> Real-Time Loading Edition を使用して Adaptive Server または Oracle に複写することはできません。	REP_RTL_IQ
Sybase IQ InfoPrimer 15.3	Sybase IQ InfoPrimer	Adaptive Server からのデータの抽出およびロードと、Sybase IQ でのデータの変換に使用されます。	SY_INFOPRIMER_SERVER

## Replication Server と Sybase IQ InfoPrimer の統合の使用

Sybase IQ InfoPrimer で Replication Server マテリアライゼーション・メソッドを使用して、データを Sybase IQ にマテリアライズし、プライマリ・データに対して行われた更新を処理するよう Replication Server を設定します。

### 1. マテリアライゼーション前:

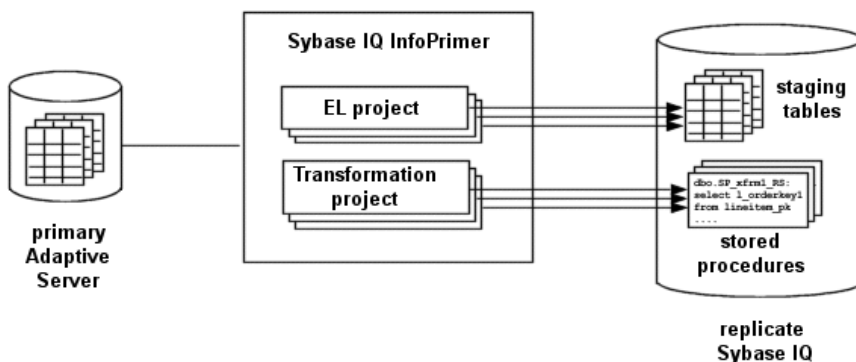
- Sybase IQ InfoPrimer で EL (抽出、ロード) プロジェクトを作成し、[Replication Server でマテリアライゼーション]を選択します。  
 EL プロジェクト・エディタの [RepServer] タブで、プライマリ Replication Server とレプリケート Replication Server (プライマリと異なる場合のみ) の接続情報も指定する必要があります。Sybase IQ InfoPrimer によって、[処理] タブにコマンドが追加されます。このコマンドは、変更も削除もしないでください。  
 各ソース・テーブルに対して、Sybase IQ InfoPrimer は必要なステー징・テーブル定義を作成します。EL プロジェクト・エディタの [テーブル] タブにある[不足している送信先テーブルを作成]アイコンを選択して、これらの

ステージング・テーブルをレプリケート Sybase IQ データベースに生成します。

**注意：**再マテリアライズを試みる場合は、rs\_status テーブルをクリアする必要があります。

- SQL 変換プロジェクトを作成し、レプリケート Sybase IQ データベースで生成されたステージング・テーブル (挿入、更新、および削除) の各セットの変換をモデル化します。この SQL 変換プロジェクトを使用して、変換の各セットをレプリケート Sybase IQ データベースでストアド・プロシージャとして展開します。

**注意：**これらの変換ストアド・プロシージャでは、オペレーションの処理が完了すると、対応するステージング・テーブルがトランケートされます。



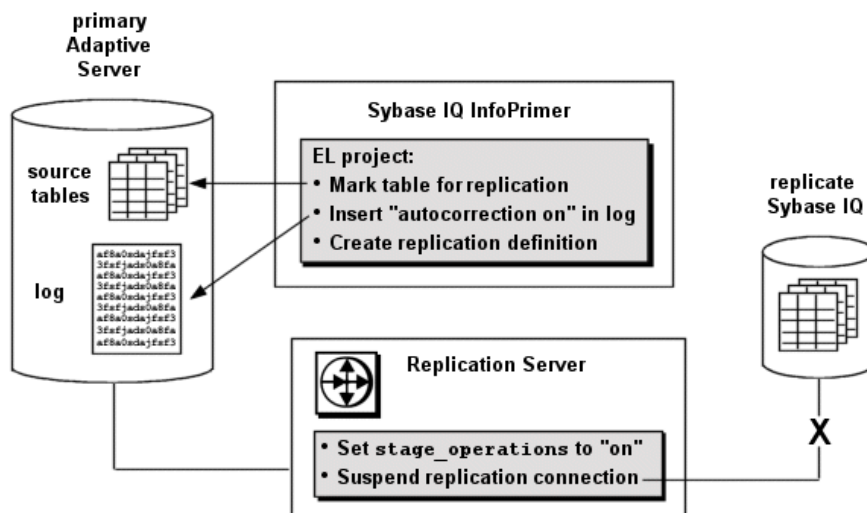
- Replication Server インスタンスでは、**stage\_operations** コネクション・パラメータを使用して、EL プロジェクトで指定されたテーブルのステージ・オペレーションへのレプリケート・データベース・コネクションを設定します。

**注意：**stage\_operations が on に設定されている場合、Replication Server は dsi\_compile\_enable の設定を無視し、そのコネクションの RTL を有効にします。オペレーションはコンパイルされ、dsi\_compile\_enable が on に設定されると、ステージングされます。

Sybase IQ InfoPrimer で、EL プロジェクトを実行します。指定された各プライマリ・テーブルについて、EL プロジェクトは以下を実行します。

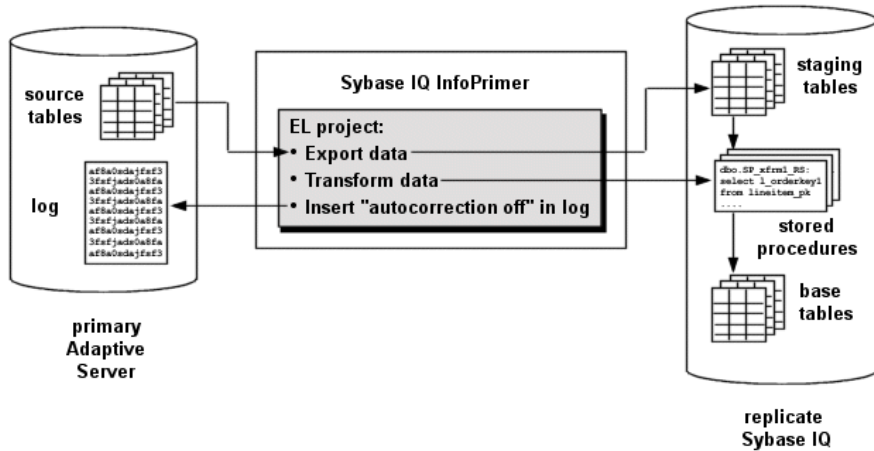
- テーブルをレプリケートするようマーク付けします。

- b) プライマリ・データベース・ログに autocorrection on レコードを挿入します。その結果、Replication Server レプリケート・データベース・コネクションがサスペンドされます。
- c) RSSD でテーブル複写定義を作成します。



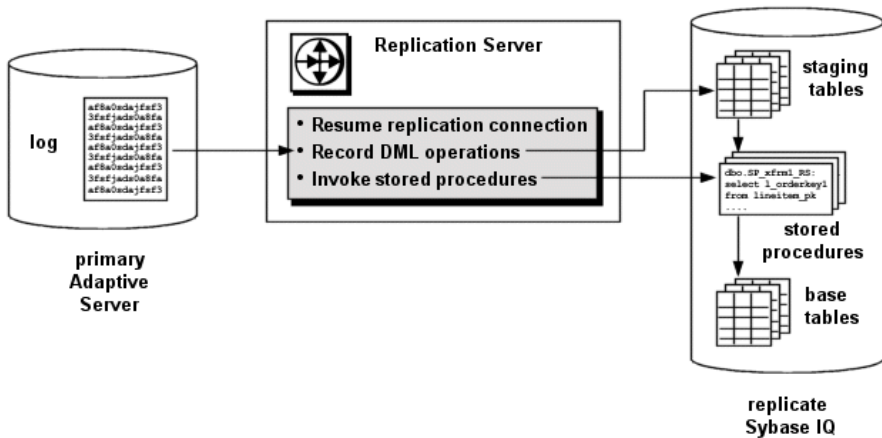
- 3. Sybase IQ InfoPrimer EL プロジェクトは、各テーブルのプライマリ・データをレプリケート Sybase IQ 上にある対応するステー징・テーブルにエクスポートし、変換ストアド・プロシーダを実行し、autocorrection off レコードをプライマリ・データベース・ログに挿入します。





- Replication Server のレプリケート・データベース・コネクションが再開され、マーク付けされたプライマリ・データベース・テーブルにさらに加えられる変更があれば、Replication Server がレプリケート Sybase IQ データベースでステーキング・テーブルと変換ストアド・プロシージャを使用して、それらの変更を処理します。

**注意：** Sybase IQ InfoPrimer は、データの移行と、ステーキング・テーブルおよび変換ストアド・プロシージャの作成にのみ使用されます。Sybase IQ InfoPrimer は、レプリケーションには関与しません。



### ベース・テーブル

ベース・テーブルには、レプリケート Sybase IQ データベースでの最終形式でデータが格納されます。

ベース・テーブル・データの発生元は、次のとおりです。

- SQL 変換 – オペレーションをステージングするよう Replication Server のレプリケート・データベース・コネクションが設定されると、ステージ・テーブルに対して実行される変換ストアド・プロシージャの結果がベース・テーブルに書き込まれます。
- レプリケーション – あるテーブルがステージングから除外されると、Replication Server はそのステージング・テーブルをバイパスしデータをベース・テーブルに直接レプリケートします。

### ステージング・テーブル

プライマリ・テーブルのためにログされるオペレーションをステージングするよう Replication Server のレプリケート・データベース・コネクションを設定すると、これらのオペレーションは可能な場合はコンパイルされ、レプリケート Sybase IQ データベースにあるステージ・テーブルに書き込まれます。

ステージングする各テーブルに対して、3つのステージング・テーブルが存在し、各ステージング・テーブルは DELETE、INSERT、および UPDATE の各オペレーションに対応します。

- *owner\_table\_name\_DELETE\_RS*
- *owner\_table\_name\_INSERT\_RS*
- *owner\_table\_name\_UPDATE\_RS*

ここで、*owner*と *table\_name*は対応するプライマリ・データベース・テーブルの所有者と名前です。これらのテーブルの名前は、EL プロジェクトによって生成され、変更できません。

---

**注意：**EL プロジェクトの [テーブル] タブには、insert ステージング・テーブルのみ表示されます。しかし、[テーブル作成] ウィンドウには、指定されたプライマリ・データベース・テーブルに対応する3つのステージング・テーブルがすべて表示されます。

---

Sybase IQ InfoPrimer の EL プロジェクトでどのプライマリ・データベース・テーブルがステージングされるのかを識別する必要があります。ステージングからレプリケート・テーブルを選択的に除外することもできます。ステージングから除外されたテーブルには、対応するステージング・テーブルを作成する必要がないため、データはレプリケート Sybase IQ データベースでプライマリ・テーブルからレプリケート・テーブルにレプリケートされます。

テーブルをステージングするようレプリケート・データベース・コネクションを設定しても、レプリケート Sybase IQ データベースにステージング・テーブルが存

在しない場合は、レプリケート・データベース・コネクションはサスペンドされます。複写定義に identity カラムとして宣言されるカラムが含まれる場合でも、これらのカラムは対応するステージング・テーブルでは identity カラムとして宣言されません。

#### テーブルのコンパイル

コンパイルできないテーブルでは、コンパイルは実行されません。テーブルがコンパイルできないと見なされるのは、RTL が無効か、ファンクション文字列が変更されているか、または最小カラム・レプリケーションが有効の場合です。コンパイルできないテーブルに対するオペレーションは、順番リストに取得され、コンパイル完了後に対応するレプリケート・テーブルに適用されます。

---

**注意：** Replication Server がステージングされたオペレーションをコミットした後、変換ストアド・プロシージャが対応するステージング・テーブルをトランケートします。したがって、Replication Server の `rs_subcmp` ユーティリティを使用してステージング・テーブルを確定化しないでください。

---

#### insert ステージング・テーブルの構造

対応する複写定義によって適用される変更とフィルタリングを除き、insert ステージング・テーブルにはプライマリ・テーブルと同じ数のカラムと同じカラム名が格納されます。

#### delete ステージング・テーブルの構造

delete ステージング・テーブルには、対応する複写定義で指定されたプライマリ・キー・カラムのみが格納されます。

複写定義でプライマリ・キーが指定されない場合、delete ステージング・テーブルには、以下を除くパブリッシュ済みカラムがすべて格納されます。

- 概数値カラム
- 暗号化カラム
- Java カラム
- LOB カラム

---

**注意：** テーブル複写定義でプライマリ・キーを指定して、処理を単純化しパフォーマンスを向上させることをお勧めします。

---

#### update ステージング・テーブルの構造

update ステージング・テーブルには、対応する複写定義で指定されたプライマリ・キー・カラムごとに2つのカラムが格納されます。1つは変更前のカラム・データ用、もう1つは変更後のカラム・データ用です。

update ステージング・テーブルには、複写定義で指定された各非プライマリ・キー・カラムのカラムも格納されます。これらの非プライマリ・キー・カラムの

データに変更が加えられたかどうかを追跡するため、update ステージング・テーブルには 1 個または複数のビットマップ・カラムが格納されます。各ビットマップ・カラムは int 型であるため、32 個の非プライマリ・キー・カラムを追跡できます。値 1 はダーティ・ビットであり、データがそのビット位置に対応するカラムで変更されたことを示します。

---

**注意：** update ステージング・テーブルの変更前カラムとビットマップ・カラムは、Sybase IQ InfoPrimer の SQL 変換プロジェクトには表示されません。

---

### 変換ストアド・プロシージャ

ステージングされるどのプライマリ・データベース・テーブルにも、対応する変換ストアド・プロシージャがレプリケート Sybase IQ データベースに存在します。Replication Server は、ステージング・テーブルに対してこれらのストアド・プロシージャを実行し、結果はベース・テーブルに書き込まれます。

Sybase IQ InfoPrimer の SQL 変換プロジェクトでこれらのストアド・プロシージャによって実行される変換を指定し、それらのストアド・プロシージャをレプリケート Sybase IQ データベースに展開する必要があります。

レプリケート Sybase IQ データベースに存在しないストアド・プロシージャを使用しようとするか、またはストアド・プロシージャの実行に失敗すると、レプリケート・データベース・コネクションがサスペンドされます。

---

**注意：** SQL 変換プロジェクトに関与するテーブルをすべて確認できるように、ストアド・プロシージャをレプリケート Sybase IQ データベースに展開する準備が整うまでは、SQL 変換プロジェクトのプロジェクト・プロパティでスキーマを選択しないでください。

---

## パラメータ

Replication Server では、**stage\_operations** パラメータと **dsi\_stage\_all\_ops** パラメータを使用して、テーブル・ステージングを制御します。

### stage\_operations

**create connection** コマンドまたは **alter connection** コマンドの **stage\_operations** パラメータを設定すると、Replication Server は指定したコネクションのステージング・テーブルにオペレーションを書き込みます。

レプリケート・データベース・コネクションのステージングを設定できます。

例：

```
create connection to SYDNEY_IQ_RS.iq_db
using profile rs_ase_to_iq;standard
set username pubs2_maint
set password pubs2_maint_pw
set stage_operations to "on"
```

個々のテーブルについてステージングの有効または無効を指定するには、特定のレプリケート・テーブルに関して **alter connection** コマンドの **stage\_operations** パラメータを使用します。例：

```
alter connection to SYDNEY_IQ_RS.iq_db
for replicate table named lineitem_5
set stage_operations to "off"
```

この場合、Replication Server は lineitem\_5 テーブルのオペレーションをステージングしませんが、代わりにオペレーションを通常どおりレプリケートします。

**注意：** **stage\_operations** パラメータは、Sybase IQ レプリケートへの接続にのみ設定できます (ここで、**dsi\_dataserver\_make** パラメータは iq に設定されます)。Sybase IQ コネクション・プロファイルを使用して接続を作成すると、**dsi\_dataserver\_make** コネクション・パラメータが適切に設定されます。

### dsi\_compile\_enable

**stage\_operations** が on に設定されている場合、Replication Server は **dsi\_compile\_enable** の設定を無視し、その接続の RTL を有効にします。オペレーションはコンパイルされ、**dsi\_compile\_enable** が on に設定されると、ステージングされます。

### dsi\_stage\_all\_ops

**alter connection** コマンドの **dsi\_stage\_all\_ops** パラメータを使用して、指定したテーブルのオペレーション・コンパイルを回避します。

緩やかに変化する次元 (SCD) のテーブルなどのように、テーブル履歴を保存する必要がある場合、**dsi\_stage\_all\_ops** を on に設定します。例：

```
alter connection to SYDNEY_IQ_RS.iq_db
for replicate table named lineitem_5
set dsi_stage_all_ops to "on"
```

## Replication Server のコンポーネント

Replication Server では、Sybase IQ InfoPrimer との統合をサポートするために追加のコンポーネントがいくつか必要です。

### rs\_status テーブル

rs\_status テーブルは、マテリアライゼーションの進行状況に関する情報を格納します。

カラム	データ型	説明
schema	varchar (255)	マテリアライズされるテーブルの所有者

カラム	データ型	説明
tablename	varchar (255)	マテリアライズされるテーブルの名前
action	varchar (1)	<ul style="list-style-type: none"> <li>• I – 初回ロード</li> <li>• A – オートコレクション・フェーズ</li> <li>• R – レプリケーション</li> </ul>
starttime	timestamp	アクションが開始された時間
endtime	timestamp	アクションが完了した時間
status	varchar (1)	<ul style="list-style-type: none"> <li>• P – アクションが進行中</li> <li>• X – 実行が完了した</li> <li>• E – 実行エラー</li> </ul>
pid	int	予約済み

たとえば、my\_table のオートコレクションが進行中の場合、rs\_status には次のようなローが含まれます。

```

schema tablename action starttime                endtime status pid
-----
sys      my_table  A      2011-07-11 19:11:25.531                P

```

my\_table のオートコレクションが完了すると、rs\_status には次のようなローが含まれます。

```

schema tablename action starttime
-----
sys      my_table  A      2011-07-11 19:11:25.531

endtime                status pid
-----
2011-07-11 19:12:14.326 X

```

rs\_status データの自動クリーンアップはありません。テーブルの再マテリアライズを試みる前に、rs\_status から対応するローを削除する必要があります。

```
delete rs_status where tablename=tablename and schema=owner
```

### オートコレクション関数

Replication Server では、rs\_autoc\_on、rs\_autoc\_off、および rs\_autoc\_ignore 関数を使用して、rs\_status テーブルを更新します。

### rs\_autoc\_on

rs\_status テーブルを更新して、オートコレクションが on に設定されていることを示します。

データ・サーバ・インタフェース (DSI) がプライマリ・データベース・ログで **autocorrection on** レコードを検出すると、Replication Server は **rs\_autoc\_on** を呼び出します。

### 例

- **例** – **rs\_iq\_function\_class** の **rs\_autoc\_on** ファンクション文字列を作成します。

```
create function string rs_autoc_on
  for rs_iq_function_class
  output language
  'insert into rs_status (schema, tablename, action, starttime,
status) values
  (?rs_repl_objowner!sys?,
  ?rs_deliver_as_name!sys?,
  "A",
  current timestamp,
  "P");
commit'
```

### 使用法

- **rs\_autoc\_on** 関数には、ファンクション文字列クラス・スコープがあります。
- インストール時に、Replication Server は初期 **rs\_autoc\_on** ファンクション文字列を作成します。
- **rs\_autoc\_on** では、**rs\_deliver\_as\_name** システム定義変数を使用します。これは、オートコレクションの影響を受けるレプリケート・データベースのテーブルを示します。
- **rs\_autoc\_on** では、**rs\_repl\_objowner** システム定義変数を使用します。これは、オートコレクションの影響を受けるレプリケート・データベースのテーブルの所有者を示します。所有者が指定されない場合、**rs\_repl\_objowner** にはシングル・スペースが格納されます。

### rs\_autoc\_off

rs\_status テーブルを更新して、オートコレクションが off に設定されていることを示します。

プライマリ・データベース・ログで **autocorrection off** レコードを検出すると、Replication Server は **rs\_autoc\_off** を呼び出します。

## 例

- 例 – `rs_iq_function_class` の `rs_autoc_off` ファンクション文字列を作成します。

```
create function string rs_autoc_off
for rs_iq_function_class
output language
'update rs_status
  set endtime = current timestamp,
  status = "X" where schema = ?rs_repl_objowner!sys?
  and tablename = ?rs_deliver_as_name!sys?
  and action = "A" and endtime is null;
insert into rs_status (schema, tablename, action, starttime,
status) values
  (?rs_repl_objowner!sys?,
  ?rs_deliver_as_name!sys?,
  "R",
  current timestamp,
  "P");
commit'
```

## 使用法

- `rs_autoc_off` 関数には、ファンクション文字列クラス・スコープがあります。
- インストール時に、Replication Server は初期 `rs_autoc_off` ファンクション文字列を作成します。
- `rs_autoc_off` では、`rs_deliver_as_name` システム定義変数を使用します。これは、オートコレクションの影響を受けるレプリケート・データベースのテーブルを示します。
- `rs_autoc_off` では、`rs_repl_objowner` システム定義変数を使用します。これは、オートコレクションの影響を受けるレプリケート・データベースのテーブルの所有者を示します。所有者が指定されない場合、`rs_repl_objowner` にはシングル・スペースが格納されます。

### rs\_autoc\_ignore

`rs_status` テーブルを更新して、オートコレクションに失敗し、テーブルに対する DML が無視されることを示します。

オートコレクション中にプライマリ・キーが更新されると、Replication Server は `rs_autoc_ignore` を呼び出します。

## 例

- 例 – `rs_iq_function_class` の `rs_autoc_ignore` ファンクション文字列を作成します。

```
create function string rs_autoc_ignore
for rs_iq_function_class
output language
'update rs_status
  set endtime = current timestamp,
```



```
status = 'E' where schema = ?rs_repl_objowner!sys?
and tablename = ?rs_deliver_as_name!sys?
and action = 'A' and endtime is null;
commit'
```

## 使用法

- **rs\_autoc\_ignore** 関数には、ファンクション文字列クラス・スコープがあります。
- インストール時に、Replication Server は初期 **rs\_autoc\_ignore** ファンクション文字列を作成します。
- **rs\_autoc\_ignore** では、*rs\_deliver\_as\_name* システム定義変数を使用します。これは、オートコレクションの影響を受けるレプリケート・データベースのテーブルを示します。
- **rs\_autoc\_ignore** では、*rs\_repl\_objowner* システム定義変数を使用します。これは、オートコレクションの影響を受けるレプリケート・データベースのテーブルの所有者を示します。所有者が指定されない場合、**rs\_repl\_objowner** にはシングル・スペースが格納されます。

## システム変数

**rs\_autoc\_on** 関数と **rs\_autoc\_off** 関数は、*rs\_status* テーブルの更新に2つのシステム変数を使用します。

- *rs\_deliver\_as\_name* – オートコレクションの影響を受けるレプリケート・テーブルの名前を指定します。
- *rs\_repl\_objowner* – オートコレクションの影響を受けるレプリケート・テーブルの所有者を指定します。

## デフォルトのデータ型変換

Sybase IQ では、すべての Adaptive Server データ型をネイティブ形式でサポートしているため、Adaptive Server から Sybase IQ へのデータ型変換は必要ありません。

## サポートされない機能

Replication Server と Sybase IQ InfoPrimer の統合は、特定の機能とプラットフォームに限定されています。

Replication Server と Sybase IQ InfoPrimer の統合では、以下はサポートされていません。

- Sybase IQ 以外のレプリケート・データベース
- Adaptive Server 以外のプライマリ・データベース
- 複写ストアド・プロシージャ

## Replication Server バージョン 15.6 ESD #1 の新機能

- カスタム・ファンクション文字列
- RTL によって提供されたもの以外の事前ステージング・オペレーション変換
- レプリケート Sybase IQ データベースで変換ストアド・プロシージャによって実行された変換後に行われる変換

# Replication Server 15.6 の新機能

Replication Server® 15.6 では、パフォーマンス、利便性、プロセス、データベース・サポートが強化されました。

## Replication Server ライセンス

---

Replication Server 15.6 では、サブキャパシティ・ライセンスが導入され、製品エディションが変更されています。

### サブキャパシティ・ライセンス

Sybase では Replication Server のサブキャパシティ・ライセンス・オプションを提供しています。これによって Sybase 製品をコンピュータで使用可能な物理 CPU のサブセットに対してライセンスできます。

詳細については、『インストール・ガイド』の「作業を始める前に」の「インストール前の作業」で、「ライセンスの取得」の「サブキャパシティ・ライセンス」を参照してください。

### Replication Server 15.6 の製品エディションとライセンス

---

Replication Server 15.6 は、Enterprise Edition (EE) と Real-Time Loading Edition (RTLE) の 2 つの異なる製品エディションとしてリリースされました。これらは、異なるベースとオプション機能で構成されており、別々のライセンスが必要です。

#### *Replication Server 15.6 用の RTLE の変更*

Replication Manager 15.6 を使用することで、Oracle から Sybase IQ への複写が可能になります。RTLE には、Replication Server 以外に、Replication Agent for Oracle が含まれているため、Oracle プライマリ・データ・サーバに接続することができます。RTLE のマニュアルには、Replication Server 製品マニュアルに加えて Replication Server Options 製品マニュアルが含まれています。

表 6 : Enterprise Edition の機能とライセンス

機能の種類	機能	説明	ライセンス
ベース	Replication Server	Advanced Services Option、ExpressConnect for Oracle、Real-Time Loading 以外の Replication Server の機能。	REP_SERVER
オプション	Advanced Services Option	Replication Server のパフォーマンス強化機能。	REP_HVAR_ASE
	ExpressConnect for Oracle	Replication Server を Oracle に直接接続できるようにする。Replication Server Options 15.5 の製品マニュアルを参照。	REP_EC_ORA

表 7 : Real-Time Loading Edition の機能とライセンス

機能の種類	機能	説明	ライセンス
ベース	Replication Server	Advanced Services Option、ExpressConnect for Oracle、Real-Time Loading 以外の Replication Server の機能。	REP_SERVER
	Real-Time Loading (RTL)	Adaptive Server および Oracle から Sybase IQ への複写を可能にする。 <u>注意： Real-Time Loading Edition を使用して Adaptive Server または Oracle に複写することはできません。</u>	REP_RTL_IQ
	Advanced Services Option	Replication Server のパフォーマンス強化機能。	REP_HVAR_ASE
	Replication Agent for Oracle	プライマリ・データ・サーバとして Oracle に接続できる Replication Agent for Oracle を含む。	RTLE には Replication Server Options のライセンスが含まれる。
オプション	なし。		

### ライセンスの取得

Replication Server をインストールする前に、有効な SySAM ライセンスを取得します。

Sybase ソフトウェア資産管理 (SySAM: Sybase Software Asset Management) では、Sybase 製品のライセンス管理と資産管理を行います。詳細については、『インス

『ツール・ガイド』の「作業を始める前に」の「ライセンスの取得」を参照してください。

## Real-Time Loading を使用した Oracle から Sybase IQ への複写

---

Oracle から Sybase IQ への複写には、Replication Server 15.6 の Real-Time Loading Edition (RTL) に含まれている Real-Time Loading (RTL) を使用できます。

### ライセンス

RTL を使用した Sybase IQ へのレプリケーションは、Real-Time Loading Edition 製品エディションで実行可能です。

### データベースとプラットフォームのサポート

- Sybase IQ – Real-Time Loading を使用して Sybase IQ バージョン 12.7 ESD #3 以降に複写できます。Sybase IQ のバージョンとプラットフォームのサポートに関する最新情報については、『Replication Server リリース・ノート』の「製品の互換性」の「Replication Server の相互運用性」を参照してください。
- Oracle – Replication Server 15.6 は Sybase IQ から Oracle 10g および 11g への複写をサポートしています。詳細については、『Replication Agent リリース・ノート (Linux 版、Microsoft Windows 版、UNIX 版)』の「製品の互換性」を参照してください。

## Real-Time Loading ソリューション

---

RTL はできるだけ多くのコンパイル可能なトランザクションをグループ化して、グループ内のトランザクションをまとめて最終的な変更としてコンパイルしてから、レプリケート・データベースでバルク・インタフェースを使用してその変更をレプリケート・データベースに適用します。

Sybase IQ レプリケート・データベースに同一データベース・スキーマを使って複写する場合、RTL は以下を使用します。

- コンパイル – テーブルごとに複写データを **insert**、**update**、**delete** オペレーション別に整理してまとめ、オペレーションを最終的なローのオペレーションにコンパイルします。
- バルク適用 – コンパイルされた最終的な結果に対して最も効率の良いバルク・インタフェースを使用して、最終的な結果をバルク適用します。Replication Server は、メモリ内の最終的な変更が保管されるデータベースを使って最終的なロー変更を保管し、それをレプリケート・データベースに適用します。

RTL では、以下によって、連続複写モードやステージング・ソリューションなどと比べて Sybase IQ への複写パフォーマンスを向上できます。

- 外部コンポーネント数の減少 – ステージング用のデータベースを使用しないので、メンテナンス・コストとオーバーヘッドが削減されます。
- 遅延時間の減少 – ステージング・ソリューションからのオーバーヘッドがなく、Sybase IQ に直接複写されます。
- 利便性の向上 – RTL の設定には、ファンクション文字列のマッピング、DSI のサスペンドとレジューム、ステージング・データベースから Sybase IQ へのデータの移植、ステージング・ソリューション用のアクティビティのスケジューリングのどれも必要ありません。
- コンパイルとバルク適用 – ログに記録されている個々のオペレーションを送信する代わりに、RTL はコンパイルによってグループ化された一連のオペレーションから **insert**、**update**、または **delete** の中間オペレーションを削除し、複写されたトランザクションの最終的なコンパイルされた状態のみを送信します。トランザクション・プロファイルによって異なりますが、これは通常 Replication Server が Sybase IQ に送信して処理させるコマンド数を少なくします。

Sybase IQ には、SQL 言語モードと比べて **insert** オペレーションのパフォーマンスを向上させるバルク・インタフェースが用意されています。RTL は Sybase IQ のバルク・インタフェースを活用して、**insert** だけでなく **update** と **delete** のオペレーションのパフォーマンスを向上させます。

Replication Server が大量のトランザクションを組み合わせてコンパイルし 1 つのグループにまとめるので、バルク・オペレーション処理が向上し、複写スループットとパフォーマンスも向上します。グループ・サイズを調整して、バルク適用のためにグループ化されるデータ量を制御できます。

### RTL のコンパイルとバルク適用

RTL のコンパイルでは、複写するデータがテーブルごとに **insert**、**update**、**delete** オペレーション別に整理してまとめられ、オペレーションが最終的なローのオペレーションにコンパイルされます。

RTL は複写定義内のプライマリ・キーによって異なったデータ・ローを区別しません。複写定義がない場合は、text と image のカラム以外はすべてプライマリ・キーとみなされます。

通常の複写環境で見られるオペレーションの組み合わせでは、同一のプライマリ・キーを持つテーブルとローがあると、RTL は次のオペレーションの組み合わせルールに従います。

- **insert** の後に **delete** があると結果はオペレーションなしになる。
- **delete** の後に **insert** があると結果のオペレーションは減少しない。

- **update** の後に **delete** があると結果のオペレーションは **delete** になる。
- **insert** の後に **update** があると、2つのオペレーションは1つのオペレーションに集約され、結果のオペレーションは **insert** になる。結果のオペレーションの内容は、最初のオペレーション結果に次のオペレーションの相違点を上書きした結果となる。
- **update** の後にもう1つの **update** があると、2つのオペレーションは1つのオペレーションに集約され、結果のオペレーションは **update** になる。結果のオペレーションの内容は、最初のオペレーション結果に次のオペレーションの相違点を上書きした結果となる。

オペレーションのその他の組み合わせでは、コンパイル・ステータスが無効になります。

### 例 1

これはログ順のローごとの変更例です。この例では、Tは "**create table** T(k int , c int)" コマンドによって以前に作成されたテーブルです。

```
1. insert T values (1, 10)
2. update T set c = 11 where k = 1
3. delete T where k = 1
4. insert T values (1, 12)
5. delete T where k = 1
6. insert T values (1, 13)
```

RTLでは、1の **insert** と2の **update** を組み合わせて **insert T values (1, 11)** に変換できます。変換結果の **insert** と3の **delete** は相殺されるので削除できます。4の **insert** と5の **delete** は削除できます。コンパイルされた最終的な RTL オペレーションは、最後の6の **insert** の次の値になります。

```
insert T values (1, 13)
```

### 例 2

ログ順のローごとの変更のもう1つの例です。

```
1. update T set c = 14 where k = 1
2. update T set c = 15 where k = 1
3. update T set c = 16 where k = 1
```

RTLでは、1と2の **update** をまとめると2の **update** になり、2と3の **updates** をまとめると3の **update** になるので、それが k=1 での最終的なロー変更になります。

Replication Serverは最終的な変更を保管するインメモリ・データベース内の **insert**、**delete**、および **update** テーブルを使用して、レプリケート・データベースに適用する最終的なロー変更を保管します。最終的なロー変更がレプリケート・テーブル別およびオペレーションの種類別 (**insert**、**update**、または **delete**) にソートされると、バルク・インタフェースに渡す準備が整います。

RTL は **insert** オペレーションをレプリケート・テーブルに直接ロードします。Sybase IQ は **update** と **delete** のバルク・オペレーションをサポートしないので、RTL は **update** と **delete** オペレーションをテンポラリ・ワークテーブルにロードします。テンポラリ・ワークテーブルは RTL によって IQ のテンポラリ・ストア内に作成されます。次に、RTL は **join-update** または **join-delete** オペレーションをレプリケート・テーブルに対して実行して、最終的な結果を生成します。ワークテーブルは動的に作成され削除されます。

例2では、次の処理によってコンパイル結果が `update T set c = 16 where k = 1` になります。

1. RTL は `#rs_uT(k int, c int)` ワークテーブルを作成します。
2. RTL がワークテーブルに対して **insert** を実行します。

```
insert into #rs_uT(k, c) location 'idemo.db' {select * from rs_uT}
```

3. RTL が次の **join-update** を実行します。

```
update T set T.c=#rs_uT.c from T,#rs_uT where T.k=#rs_uT.k
```

RTL が大量のトランザクションを1つのグループにまとめるので、バルク・オペレーション処理が向上し、複製スループットとパフォーマンスも向上します。RTL サイズを設定パラメータで調整することによって、RTL がバルク適用のためにグループにまとめるデータの量を制御できます。

RTL はロー変更を変更がログされた順序で適用しませんが、データ・ロスはありません。

- 異なったデータ・ローでは、ロー変更が適用される順序は結果に影響しません。
- 同じロー内では、コンパイル後、**delete** の後に **insert** を適用することによって整合性を維持します。

### 参照：

- RTL 設定 (68 ページ)

### RTL の処理と制限事項

RTL は元のコミット順を維持しながら、トランザクションの最終的なロー変更のみを適用します。それによって中間的なロー変更は省略されますが、トランザクションの整合性は保証されます。

このアプローチには次のような問題が伴います。

- **insert** トリガが起動されません。これは、RTL プロセスが最終的な新しいローのバルク・ロードをテーブルに対して直接行うからです。Replication Server がコンパイルの最終結果をレプリケート・データベースに適用すると、**update** と **delete** の各トリガは引き続き起動します。ただし、Replication Server がコンパ



イルし最終結果には含まれないロー変更が、それらのトリガから確認できなくなります。トリガが検出できるのは、最後のロー・イメージのみです。

Replication Server を使用して、ユーザが変更したテーブルの任意のカラムにそのユーザを関連付けるトリガ・ロジックのあるテーブル・スキーマで、`last_update_user` カラムを使用してユーザ更新を監査するとします。userA がテーブルの `colA` と `colC` を変更した後に、userB が `colB` と `colD` を変更した場合、トリガが起動すると、トリガ・ロジックが検出できるのはテーブルを最後に変更したユーザのみです。したがって、トリガ・ロジックはこれら4つのカラムを変更したユーザとして userB を関連付けます。ロー変更を個別に検出する必要がある同様のロジックを含むトリガを定義する場合、そのテーブルの RTL コンパイルを無効にする必要がある場合があります。

- RTL はロー変更を変更がログに記録された順序と同じ順序では適用しません。複製テーブルにログ順に変更を適用するには、そのテーブルに対して RTL コンパイルを無効にします。
- 複製テーブルに参照制約がある場合、その参照制約を複製定義の中で指定する必要があります。制約エラーを避けるために、RTL は複製定義に従ってテーブルをロードします。
- Sybase IQ への複製では、RTL はカスタム・ファンクション文字列や並列 DSI 逐次化メソッド (デフォルトの `wait_for_commit` メソッドを除く) をサポートしません。RTL はカスタム・ファンクション文字列をコンパイルできないコマンドとして扱います。
- RTL は、コンパイルできないコマンドと呼ばれる種類のコマンドと、コンパイルできないテーブルと呼ばれる種類のテーブルをコンパイルしません。そのような種類のコマンド、トランザクション、またはテーブルを検出すると、Replication Server はログ順にローごとに行う連続複製に戻ります。
  - コンパイルできないコマンド – ストアド・プロシージャ、SQL 文、システム・トランザクション、Replication Server の内部マーク。
  - コンパイルできないトランザクション – コンパイルできないコマンドを含んでいるトランザクション。
  - コンパイルできないテーブル – RTL が無効にされているテーブル、カスタム・ファンクション文字列を持つテーブル、RTL がコンパイルできないテーブルと参照制約関係があるテーブル。
- RTL は、後に `insert` が続く `delete` に対するプライマリ・キーの更新を自動的に変更します。
- RTL はトランザクションのグループ化を停止する `dsi_partition_rule` のようなパラメータを無視します。
- RTL 処理中にエラーが発生すると、Replication Server は RTL をリトライします。リトライでは、RTL コンパイルが失敗したトランザクションを特定できるまで、トランザクション・グループを小さなグループに分割していきます。特定されたトランザクションは連続複製を使って適用されます。

RTLのパフォーマンス上の利点を実現するには、プライマリ・データベースとレプリケート・データベースを同期させ、エラー発生による Replication Server への余分な処理オーバーヘッドを避けるようにします。データを同期するためには、**dsi\_command\_convert** を **i2di,u2di** に設定しますが、これも処理オーバーヘッドを発生させます。データベースが同期している場合は、**dsi\_command\_convert** を **none** にリセットします。

- RTL はロー・カウントの検証を行って複製の整合性を確認します。ロー・カウントの検証はコンパイルに基づいて行われます。予期されるロー・カウントはコンパイル後のロー数です。
- 複製定義の中に **identity** データ型のカラムがある場合、Replication Server はレプリケート・データベース内で次の Sybase IQ コマンドを実行します。
  - **set temporary option identity\_insert = 'table\_name'** (ID カラムの挿入および更新前)
  - **set temporary option identity insert = ""** (ID カラムの挿入および更新後)
- デフォルトでは、Oracle は最小限のロギングを行います。したがって、データベース複製定義を使用している場合、テーブル複製定義を作成するか、完全ロギングを有効にして、**update** コマンドが正しく動作するようにします。テーブル複製定義を作成することにした場合、Replication Agent または Replication Server で次のようにしてその定義を作成できます。
  - Replication Agent for Oracle — 複製マークの付いたテーブルが存在する場合に Replication Server で複製定義を自動的に作成するには、テーブルに複製マークを付ける前に **pdb\_auto\_create\_repdefs** を **true** に設定するか、テーブルにマークを付けてから **rs\_create\_repdef** を実行します。詳細については、Replication Server Options の『Replication Agent リファレンス・マニュアル』を参照してください。
  - Replication Server — **create replication definition** を **send standby** 句を指定して実行すると、Replication Server 内で複製定義を直接作成できます。『Replication Server リファレンス・マニュアル』を参照してください。

## Sybase IQ のレプリケート・データ・サーバ

レプリケート Replication Server は、Sybase IQ レプリケート・データベース内にロギングし、複製されたトランザクションを適用することによって、レプリケート Sybase IQ データ・サーバと直接交信します。

## Sybase IQ での複製の干渉と影響

Sybase IQ レプリケート・データベースに対する唯一の実質的な干渉と影響は、接続プロファイルを通して Sybase IQ レプリケート・データベース内に作成されるシステム・テーブルと、RTL のバルク適用のために Sybase IQ レプリケート・データベース内に作成されるテンポラリ・テーブルです。

### システム・テーブル

接続プロファイルは Sybase IQ レプリケート・データベース内に 3 つのテーブルを作成します。

- `rs_threads` – Replication Server はこのテーブルの情報を使ってデッドロックを検出し、並列 DSI スレッド間でトランザクションの逐次化を実行します。このテーブルのエントリは、トランザクションが開始されたときと、コネクションに対して 2 つ以上の DSI スレッドが定義されたときに更新されます。
- `rs_lastcommit` – レプリケート・データベースに適用される複製トランザクションに関する情報を格納します。`rs_lastcommit` テーブル内の各ローは、プライマリ・データベースからレプリケート・データベースに配信されコミットされた、最後のトランザクションを示します。Replication Server は、この情報によってすべてのトランザクションが配信されたことを確認します。  
Replication Server の `rs_get_lastcommit` 関数は、レプリケート・データベース内の最後にコミットされたトランザクションに関する情報を取得します。ASE 以外のレプリケート・データベースについては、`rs_get_lastcommit` 関数は、データベース固有のファンクション文字列クラスで、レプリケート・データベース内の `rs_lastcommit` テーブルへのアクセスに必要なクエリによって置き換えられます。
- `rs_ticket_history` – Replication Server コマンド `rs_ticket` の実行結果を格納します。プライマリ・データベースに対して `rs_ticket` コマンドを発行し、コマンドがプライマリ・データベースからレプリケート・データベースまで移動するために要する時間を測定できます。この情報を使用して、Replication Server のパフォーマンス、モジュールのハートビート、複製の正常性、テーブルレベルのクワイズをモニタできます。`rs_ticket` の各実行結果は、レプリケート・データベース内にある `rs_ticket_history` テーブルの 1 つのローに保存されます。`rs_ticket_history` テーブルの各ローを問い合わせると、個々の `rs_ticket` 実行結果を取得したり、別のローの結果と比較できます。必要に応じて、`rs_ticket_history` テーブル内のデータを手動でトランケートします。

### ワーク・テーブル

RTL は Sybase IQ データベースの IQ テンポラリ・ストア内にテンポラリ・ワークテーブルを作成して、RTL バルク適用をサポートします。ワークテーブルは動的に作成および削除されます。

テンポラリ・テーブルに必要な Sybase IQ 内の領域は Sybase IQ に複製する予定のデータ量に依存します。テンポラリ・ワークテーブルのために Sybase IQ のテンポラリ・データベースの領域を調整するには、Sybase IQ の `alter dbspace` コマンドを使用します。詳細については、該当するバージョンの Sybase IQ マニュアルを参照してください。Sybase IQ 15.0 以降の例:

```
ALTER DBSPACE dbspace-name ADD FILE FileHist3
  \History1\data/file3' SIZE 500MB
```

## Sybase IQ に関するレプリケート・データベースのコネクティビティ

Sybase IQ をレプリケート・データ・サーバとして使用する場合、データベース・ゲートウェイを使用する必要はありません。レプリケート Replication Server は、Sybase IQ レプリケート・データ・サーバに直接接続します。

Replication Server のデータベース・コネクション名は、データ・サーバ名 (**server\_name**) およびデータベース名 (**db\_name**) で構成されています。レプリケート Replication Server は、データベース・コネクションで指定された Sybase IQ レプリケート・データベースの **server\_name** を含む **interfaces** ファイル・エントリを検索します。

**dsedit** を使って Replication Server の **interfaces** ファイルにエントリを作成し、Sybase IQ レプリケート・データ・サーバが受信するホストとポートの指定に使用します。**interfaces** ファイル・エントリの名前は、Replication Server データベース・コネクションの **server\_name** 部分と一致している必要があります。

Replication Server を再起動して Replication Server の **interfaces** ファイル内の新しいエントリを有効にします。詳細については、『**Replication Server 設定ガイド**』の「**rs\_init** による Replication Server の設定とデータベースの追加」の「新しい Replication Server の設定」で、「**interfaces** ファイルの編集」を参照してください。

Sybase IQ レプリケート・サーバの **interfaces** ファイル内にレプリケート Replication Server のエントリを作成して、Replication Server が **INSERT ... LOCATION** 文を Sybase IQ に送信したときに、Sybase IQ が Replication Server に接続してデータを取得できるようにします。

Replication Server は、データベース・コネクションで指定された **user\_name** と **password** を使用して、Sybase IQ レプリケート・データ・サーバにログインします。Sybase IQ レプリケート・データベースでは、**user\_name** と **password** はメンテナンス・ユーザの ID とパスワードです。

## Sybase IQ レプリケート・データベースのパーミッション

レプリケート・データベース内のトランザクションを適用するために、Replication Server と Sybase IQ ではメンテナンス・ユーザ ID が必要となります。

複写を開始できるようにするには、Sybase IQ データ・サーバでメンテナンス・ユーザ ID を定義して、その ID にトランザクションをレプリケート・データベースに適用する権限を与える必要があります。メンテナンス・ユーザ ID には、Sybase IQ レプリケート・データベースにおける、次のパーミッションが必要です。

- **RESOURCE** 権限 (ワークテーブルとテンポラリ・インデックスの作成に必要)
- **EXECUTE** パーミッション (**sp\_iqwho** ストアド・プロシージャの実行に必要)
- **GRANT ALL** パーミッション (すべての複製テーブルに対して必要)
- **UPDATE** 権限 (すべての複製テーブルに対して必要) および **EXECUTE** 権限 (すべての複製ストアド・プロシージャに対して必要)

### メンテナンス・ユーザ ID に権限を与える

単純なセットアップで開始する場合、または Sybase IQ への複製をテストする場合は、DBA と RESOURCE の権限を与えます。

1. Sybase IQ `rssetup.sql` サンプル・スクリプトを使用して必要な権限を持つ Sybase IQ メンテナンス・ユーザを作成します。

---

**警告！** メンテナンス・ユーザ ID が既に存在する場合、このスクリプトはパスワードをデフォルト・パスワードにリセットします。

---

```
grant connect to dbmaint identified by dbmaint
grant DBA to dbmaint
grant membership in group rs_systabgroup to dbmaint

-- Create a user for REPSRV to extract -- materialization data,
etc.
-- Give sa user access to any replicated tables
-- Give sa user access to REPSRV schema
grant connect to sa identified by sysadmin
grant DBA to sa
grant membership in group rs_systabgroup to sa

-- Allow sa and dbmaint to reference replicated tables created by
DBA
grant group to DBA
grant membership in group DBA to dbmaint
grant membership in group DBA to sa
go
```

このスクリプトは Sybase IQ インストール・ディレクトリ内の `scripts` ディレクトリにあります。以下は UNIX プラットフォームでの例です。

- Sybase IQ 15.0 より古いバージョン – `/$ASDIR/scripts`
- Sybase IQ 15.0 以降 – `/$IQDIR15/scripts`

ディレクトリの場所については、『Sybase IQ インストールおよび設定ガイド』を参照してください。

2. Sybase IQ データベースが Transact-SQL® (For IQ DBA) と互換性があることを確認します。

『Sybase IQ リファレンス：文とオプション』の「データベース・オプション」の「Transact-SQL 互換性オプション」、および『Sybase IQ リファレンス：ビル

ディング・ブロック、テーブル、およびプロシージャ』の「他の Sybase データベースとの互換性」を参照してください。

3. 複写に関係するすべてのテーブルとストアド・プロシージャに対する適切なパーミッションを与えます。

## Sybase IQ レプリケート・データベースの設定に関する問題

Sybase IQ サーバの設定に関する問題について説明します。

### Replication Server インストール

Replication Server は必要な接続プロファイルを自動的にインストールすることによって、Sybase IQ への複写をサポートするファンクション文字列とクラスを提供します。

#### *接続プロファイル*

接続プロファイルを使用すると、ファンクション文字列クラスとエラー・クラスを設定し、ユーザ定義データ型 (UDD: user-defined datatypes) と Sybase IQ 変換をインストールして、Sybase IQ レプリケート・データベース内に複写に必要なテーブルを作成することによって、事前に定義されたプロパティのセットでコネクションを設定できます。

**rs\_oracle\_to\_iq** 接続プロファイルは Replication Server のインストール・パッケージの一部であり、Replication Server のインストール時に登録されます。次に接続プロファイルを説明します。

- ファンクション文字列、エラー・クラス、ユーザ定義データ型をカスタマイズします。ファンクション文字列は、いくつかのデフォルト Replication Server ファンクション文字列を、Sybase IQ データ・サーバと通信し、テーブルとプロシージャにアクセスするように設計されたカスタム・ファンクション文字列に置き換えます。これらのファンクション文字列は、Replication Server のデフォルトの **rs\_iq\_function\_class** に追加されます。RTL はカスタム・ファンクション文字列をコンパイルできないコマンドとして扱います。
- クラス・レベル・データ型変換をカスタマイズします。クラス・レベル変換によって、プライマリ・データ型とデータの変換先のレプリケート・データ型が特定されます。クラス・レベルの変換は **rs\_oracle\_to\_iq** 接続プロファイルによって Sybase IQ レプリケート・データベースに供給されます。これは Oracle データ型を Sybase IQ データ型に変換します。
- Sybase IQ レプリケート・データベース内に **rs\_threads**、**rs\_lastcommit**、**rs\_ticket\_history** テーブルを作成します。
- Sybase IQ へのコネクションを設定するために、次のようにしてデフォルトのファンクション文字列クラスとエラー・クラスのコネクション・プロパティを設定します。

```
set error class rs_iq_error_class
set function string rs_iq_function_class
```

### Sybase IQ へのコネクションを作成する

Sybase IQ レプリケート・データベースへのコネクションをセットアップします。

1. **create connection** を **using profile** 句および該当するコネクション・プロファイルを指定して使用し、レプリケート Sybase IQ データ・サーバおよびデータベースを指定します。

たとえば、Oracle プライマリ・データ・サーバからのコネクションを作成するには、次のようにします。

```
create connection to IQSRVR.iqdb
using profile rs_oracle_to_iq;standard
set username to dbmaint
set password to dbmaint
go
```

Sybase IQ データベースへのレプリケーション・パスを複数作成して、レプリケーション負荷を分散できます。各パスにユニークなメンテナンス・ユーザ ID を使用します。

2. **admin who** を使って Replication Server が Sybase IQ に正常に接続されたことを確認します。

### RTL を有効にする

必要なパーミッションを与え、レプリケート Sybase IQ データベースに接続したら、RTL を有効にして Sybase IQ への複写を設定できます。

**dsi\_compile\_enable** を使ってそのコネクションの RTL を有効にします。

**dsi\_compile\_enable** を off にした場合、Replication Server はログ順、ローごとの連続複写モードを使用します。たとえば、テーブル上のすべてのオペレーションをログ順に複写する必要があるトリガがテーブルにあるためコンパイルを使用できない場合のように、最終的なロー変更のみを複写すると問題が発生する場合、問題のテーブルで **dsi\_compile\_enable** を off に設定します。

---

**注意：** **dsi\_compile\_enable** を on に設定すると、Replication Server は **dsi\_cmd\_prefetch** と **dsi\_num\_large\_xact\_threads** を無効にします。

---

特定のデータベースにのみ影響するように、RTL をデータベース・レベルで有効にして設定するには、次のように入力します。

```
alter connection to IQ_data_server.iq_database
set dsi_compile_enable to 'on'
go
```

RTL をサーバまたはテーブル・レベルで有効にして設定することもできます。

- サーバ・レベル — Replication Server へのすべてのデータベース・コネクションに影響します。

```
configure replication server
set dsi_compile_enable to 'on'
```

- テーブル・レベル — 指定した複製テーブルのみに影響します。テーブル・レベルとデータベース・レベルの両方でパラメータを指定している場合は、テーブル・レベルのパラメータがデータベース・レベルのパラメータよりも優先されます。テーブル・レベルでパラメータを指定しなければ、データベース・レベルのパラメータの設定が適用されます。テーブルにパラメータを設定するには、**alter connection** と **for replicate table named** 句を使用します。次に例を示します。

```
alter connection to IQ_data_server.iq_database
for replicate table named dbo.table_name
set dsi_compile_enable to 'on'
```

**for replicate table name** 句の使用によってテーブル・レベルのコネクション設定を変更できます。設定の変更は指定したテーブルのすべてのサブスクリプションからの複製データと複製定義に適用されます。

---

**注意：** テーブル・レベルの設定には、**alter connection** しか使用できません。これは Replication Server が **for** 句を **create connection** に対してサポートしていないためです。

---

**dsi\_compile\_enable** を実行した後、レプリケート Sybase IQ データベースへのコネクションをサスペンドしてレジュームします。

### RTL 設定

トランザクションのグループ化とコンパイルをいつ開始して停止するかを指定するパラメータも使用できます。

- **dsi\_compile\_max\_cmds** — コマンド数単位で、トランザクションのグループの最大サイズを指定します。RTL がコンパイルしている現在のグループで最大グループ・サイズに達すると、RTL は新しいグループを開始します。  
読み込むデータがなくなると、グループが最大コマンド数に達していても、RTL は現在のトランザクションのセットを現在のグループにグループ化する処理を終了します。デフォルトは 10,000 コマンドです。
- **dsi\_bulk\_threshold** — 特定のコマンド・タイプのテーブルでコンパイルが行われた後の最終的なロー変更コマンド数を指定します。その数に達したら、それがトリガになって、Replication Server はそのテーブルの同じコマンド・タイプにバルク・コピー・インを使用します。デフォルトの最終的なロー変更コマンド数は 20 です。
- **dsi\_command\_convert** — 複製コマンドの変換方法を指定します。変換の種類は次のオペレーションの組み合わせによって指定されます。



- **d** – delete
- **i** – insert
- **u** – update
- **t** – truncate
- **none** – オペレーションなし

**dsi\_command\_convert** でのオペレーションの組み合わせには、**i2none**、**u2none**、**d2none**、**i2di**、**t2none**、**u2di**が含まれます。変換前のオペレーションは "2" の前に、変換後のオペレーションは "2" の後ろにあります。次に例を示します。

- **d2none** – **delete** コマンドを複製しません。このオプションでは、**delete** オペレーションを複製しない場合に **rs\_delete** ファンクション文字列をカスタマイズする必要はありません。
- **i2di,u2di** – **insert** と **update** の両方を **delete** とそれに続く **insert** に変換します。これはオートコレクションと同等のオペレーションになります。**dsi\_row\_count\_validation** を off にすることによってロー・カウントの検証を無効にする場合、複製時に重複キー・エラーを避け、データベースの自動同期ができるようにするために、**dsi\_command\_convert** を **i2di,u2di** に設定するようおすすめします。
- **t2none** – **truncate table** コマンドを複製しません。

**dsi\_command\_convert** のデフォルトは **none** です。これは、コマンドの変換がないことを意味します。

- **dsi\_compile\_retry\_threshold** – グループ内のコマンド数に対するスレッシュホールド値。Replication Server 15.6 では、**dsi\_compile\_retry\_threshold** パラメータがリトライ・メカニズムの機能強化の一部として含まれています。

RTL は、Sybase が推奨する **dsi\_compile\_max\_cmds**、**dsi\_bulk\_threshold**、**dsi\_command\_convert**、および **dsi\_compile\_retry\_threshold** のデフォルト値を自動的に設定します。ただし、複製環境のパフォーマンスを調整するために独自の値を指定することもできます。

- ```

• alter connection to IQSRVR.iqdb
  set dsi_compile_max_cmds to '50000'
  go
• alter connection to IQSRVR.iqdb
  set dsi_bulk_threshold to '15'
  go
• alter connection to IQSRVR.iqdb
  set dsi_command_convert to 'i2di,u2di'
  go
• alter connection to IQSRVR.iqdb
  set dsi_compile_retry_threshold to '200'
  go

```

---

**注意：** 変更するパラメータごとに個別の **alter connection** コマンドを実行する必要があります。 **alter connection** を入力した後は、2つ以上のパラメータを入力しないでください。

---

パラメータの詳細な説明については、『Replication Server リファレンス・マニュアル』の「Replication Server コマンド」の「**alter connection**」を参照してください。

**参照：**

- HVAR と RTL のリトライ・メカニズムの強化 (80 ページ)
- ロー・カウントの検証を制御する (106 ページ)

### **Replication Server 15.5 のシステム・テーブル・サポート**

Replication Server では `rs_tbconfig` テーブルをテーブル・レベルの設定パラメータの保管に使用し、`rs_columns` テーブルの `ref_objowner` カラムと `ref_objname` カラムを参照制約のサポートに使用します。

テーブルの詳細については、『Replication Server リファレンス・マニュアル』の「Replication Server システム・テーブル」を参照してください。

## **Sybase IQ への複写シナリオ**

このシナリオを使用して、RTL を使った Sybase IQ への複写のセットアップとテストの方法について説明します。

Oracle データベース管理者 (Oracle DBA)、Sybase IQ データベース管理者 (IQ DBA)、複写システム管理者 (RSA) は、Oracle、Replication Server、Sybase IQ を複写用に準備して、Sybase IQ データベースへの接続をセットアップする必要があります。

このシナリオでは、`dbo` は `ORA_DS` プライマリ Oracle サーバの `pdb1` データベース内にある `testtab` テーブルの所有者です。`c1`、`c2`、`c3` は `testtab` 内のカラムであり、データ型はそれぞれ `int`、`int`、`char(10)` です。IQSRVR はレプリケート Sybase IQ データ・サーバであり、`iqdb` データベースが格納されています。

### **interfaces ファイルのエントリの作成**

レプリケート Replication Server と Sybase IQ データ・サーバの `interfaces` ファイルに、互いのエントリを作成します。

1. Sybase IQ データ・サーバの `interfaces` ファイル (Windows では `sql.ini` ファイル) に、レプリケート Replication Server のエントリを作成します。

---

**注意：** Sybase IQ データ・サーバの `interfaces` ファイルが、Sybase IQ が使用している `$SYBASE` ディレクトリ (Windows では `%SYBASE%` ディレクトリ) にない場合、このファイルを作成します。

---

- レプリケート Replication Server の `interfaces` ファイルに、Sybase IQ データ・サーバのエントリを作成します。  
さまざまな Sybase IQ マルチプレックス・ノードへのコネクションを作成している場合、影響を受ける各ノードのエントリをレプリケート Replication Server の `interfaces` ファイルに作成します。

#### 参照：

- Sybase IQ に関するレプリケート・データベースのコネクティビティ (64 ページ)

#### テスト・テーブルを作成する

プライマリ・データベースとレプリケート・データベース内にテスト・テーブルを作成し、複写作業をテストするためにメンテナンス・ユーザにそのテーブルに対するパーミッションを与えます。

- Oracle データ・サーバ内の Oracle プライマリ・データベース `pdb1` 内に、`c1integer`、`c2integer`、`c3char(10)` という3つのカラムを持つ `testtab` という名前のテーブルを作成します。構文の詳細については Oracle のマニュアルを参照してください。
- Sybase IQ IQSRVR データ・サーバのレプリケート・データベース `iqdb` 内で次のように入力します。

```
use iqdb
go
create table dbo.testtab(c1 int primary key, c2 int,
c3 char(10))
go
grant all on dbo.testtab to public
go
```

#### プライマリ・データベースとレプリケート・データベースへのコネクションを作成する

プライマリ・データベース・コネクションとレプリケート・データベース・コネクションを作成します。

- プライマリ Oracle データベースへのコネクションを作成します。詳細については、『異機種間複写ガイド』と、Replication Server Options 製品のマニュアルを参照してください。
- Sybase IQ レプリケート・データベースへのコネクションを作成します。

**注意：** Oracle から Sybase IQ への接続の作成には、**rs\_init** は使用できません。

この例では、IQSRVR データ・サーバ内の *iqdb* データベースとデフォルト *dbmaint* の Sybase IQ メンテナンス・ユーザを使用します。

```
create connection to IQSRVR.iqdb
using profile rs_oracle_to_iq;standard
set username to dbmaint
set password to dbmaint
go
```

コマンドが正常に実行されると、次のような出力が表示されます。

```
Connection to 'IQSRVR.iqdb' is created.
```

『Replication Server リファレンス・マニュアル』の「Replication Server コマンド」の「**create connection with using profile** 句」を参照してください。

3. 接続が機能していることを確認します。

```
admin who
go
```

接続が機能していれば、次のような出力が表示されます。

| Spid | Name     | State            | Info                |
|------|----------|------------------|---------------------|
| 63   | DSI EXEC | Awaiting Command | 103 (1) IQSRVR.iqdb |
| 62   | DSI      | Awaiting Message | 103 IQSRVR.iqdb     |
| 35   | SQM      | Awaiting Message | 103:0 IQSRVR.iqdb   |

## RTL の有効化

データベース・レベルで RTL を有効にします。

1. 指定したデータベースのみに影響するように、データベース・レベルで RTL を有効にして設定するには、次のように入力します。

```
alter connection to IQSRVR.iqdb
set dsi_compile_enable to 'on'
go
```

2. 接続への変更を有効にするには、レプリケート Sybase IQ データベースへの接続をサスペンドしてレジュームします。

```
suspend connection to IQSRVR.iqdb
go
resume connection to IQSRVR.iqdb
go
```

### 複写テスト準備のためにテーブルにマークを付ける

プライマリ Oracle データベース内で Sybase IQ データベースへ複写するテーブルにマークを付けます。

この例では、dbo は ORA\_DS プライマリ Oracle データ・サーバの pdb1 データベース内にある testtab テーブルの所有者です。c1、c2、c3 は testtab 内のカラムであり、データ型はそれぞれ int、int、char(10) です。

1. 複写をテストし、挿入が正常に行われることを確認するために、Oracle の testtab にデータ・ローを挿入します。
2. **pdb\_setreptable** Replication Agent コマンドを使って testtab に複写マークを付けます。

詳細については、Replication Server Options 15.5 のマニュアルの『Replication Agent Administration Guide』の「Chapter 2 Setting Up and Configuring Replication Agent」で、「Marking tables in the primary database」を参照してください。

### 複写定義とサブスクリプションを作成する

RTL を有効にして設定したら、Sybase IQ への複写用にマークされているテーブルの複写定義とサブスクリプションを作成します。

1. *repdef\_testtab* という名前の複写定義を作成し、RTL をサポートするために必要な参照制約句を複写定義に追加します。

```
create replication definition repdef_testtab
with primary at ORA_DS.pdb1
with primary table named 'TESTTAB'
with replicate table named dbo.'testtab'
(C1 as c1 int, C2 as c2 int, C3 as c3 char(10))
primary key(C1)
go
```

**注意：** デフォルトでは Oracle のオブジェクト名の文字設定はすべて大文字です。例の手順 1 に見られるように、複写定義でオブジェクト名を大文字から小文字に変換できます。Replication Agent for Oracle の設定パラメータ **lrl\_character\_case** を使用して変換することもできます。詳細については、Replication Server Options の『Replication Agent リファレンス・マニュアル』の「設定パラメータ」で、「**lrl\_character\_case**」を参照してください。

2. 各テーブルとストアド・プロシージャの複写定義に一致するサブスクリプションを作成します。

```
create subscription sub_testtab for repdef_testtab
with replicate at IQSRVR.iqdb
go
```

- testttab がマテリアライズされたことを確認するために、Sybase IQ にログインして次のコマンドを実行します。

```
select * from dbo.testttab
go
```

マテリアライゼーションが正常に行われると、次のような出力が表示されます。

```
c1          c2          c3
-----
1           1          testrow 1
2           2          testrow 2
3           3          testrow 3
(3 rows affected)
```

### RTL が機能することを検証する

RTL が機能することを確認する方法について説明します。

- プライマリ Oracle データ・サーバにログインして、testttab に新しいローを挿入するなどのオペレーションを実行します。
- Sybase IQ にログインして、testttab への変更が Sybase IQ データベースに複写されたことを確認します。

```
select * from dbo.testttab
go
```

複写が正常に実行されると、次のような出力が表示されます。

```
c1          c2          c3
-----
1           1          testrow 1
2           2          testrow 2
3           3          testrow 3
4           4          testrow 4
5           5          testrow 5
6           6          testrow 6
(6 rows affected)
```

### 参照制約のあるテーブル

参照制約 (外部キーその他の検査制約など) のあるテーブルの指定には複写定義を使用できます。それによって RTL にそれらのテーブルの存在が通知されます。

通常は、参照元のテーブルには同じプライマリ・データベース内の参照先テーブルに対する参照制約が含まれています。RTL では複数のプライマリ・データベースからの参照先テーブルをサポートするよう参照制約が拡張されています。

各プライマリ・データベースに対する複写定義内で参照元テーブルを指定できます。ただし、複数の参照制約が互いに競合する場合は、Replication Server によってランダムにテーブルが 1 つ選択されます。

**参照：**

- RTL の処理と制限事項 (60 ページ)

**複写定義の作成と変更**

参照制約のあるテーブルの指定には、**references** パラメータを指定して **create replication definition** コマンドを使用します。

```

create replication definition
...
(column_name [as replicate_column_name]
...
[map to published_datatype] [quoted]
[references [table_owner.]table_name [(column_name)]] ...)
....]

```

参照元のテーブルの追加と変更には、**references** パラメータを指定して **alter replication definition** コマンドを使用します。参照を削除するには、**null** オプションを使用します。

```

alter replication definition
.....
add column name [as replicate_column_name]
[map to published_datatype] [quoted]
[references [table_owner.]table_name [(column_name)]]
...
| alter columns with column_name references
{[table_owner.]table_name [(column_name)] | NULL}
[, column_name references {[table_owner.]table_name [(column_name)]
| NULL}
...

```

**alter replication definition** と **create replication definition** の両方に **reference** 句を指定すると、Replication Server の動作は次のようになります。

- **reference** 句をカラム・プロパティとして扱う。各カラムはテーブルを1つだけ参照できる。
- **reference** 句内の **column\_name** パラメータに指定したカラム名を処理しない。
- 循環参照になる参照制約を許可しない。たとえば、元の参照先テーブルは元の参照元テーブルへの参照制約を持つことはできない。

複写プロセスでは、RTL は次のようにロードします。

- 参照先テーブルへの挿入の後で複写定義で指定した参照元テーブルに挿入する。
- 複写定義で指定したテーブルでの削除の後で参照先テーブルで削除する。

場合によっては、両方のテーブルでの更新が競合によって失敗することがあります。RTL が複写処理のリトライをしないようにして、パフォーマンスの低下を防ぐには、以下を行います。

- 更新を削除と挿入に変換するように、**dsi\_command\_convert** を "u2di" に設定してレプリケーションの更新を停止する。
- **dsi\_compile\_enable** を off にして、影響を受けたテーブルがコンパイルされるのを避ける。

カスタム・ファンクション文字列を持つテーブルと、コンパイルできない既存テーブルへの参照制約を持つテーブルは RTL でコンパイルできません。これらのテーブルにマークを付けることによって、RTL は参照制約エラーによって発生するトランザクションのリトライを避け、複写処理を最適化できます。

## RTL 情報の表示

設定パラメータ・プロパティとテーブル参照の情報を表示できます。

### 設定パラメータ・プロパティの表示

**admin config** を使用して、例に示されているようなデータベース・レベルとテーブル・レベルの設定パラメータを表示します。

- データベース・レベル
  - NY\_DS データ・サーバ (NY\_DS.nydb1) の nydb1 データベースへの接続に使用するデータベース・レベルの設定パラメータをすべて表示するには、次のように入力します。

```
admin config, "connection", NY_DS, nydb1
```
  - NY\_DS.nydb1 への接続で **dsi\_compile\_enable** が on であることを確認するには、次のように入力します。

```
admin config, "connection", NY_DS, nydb1, dsi_compile_enable
```
  - **dsi\_compile\_enable** など、名前の一部に "enable" があるデータベース・レベルの設定パラメータをすべて表示するには、次のように入力します。

```
admin config, "connection", NY_DS, nydb1, "enable"
```

---

**注意：** "enable" は Replication Server の予約語なので、引用符で囲む必要があります。『Replication Server リファレンス・マニュアル』の「トピック」の「予約語」を参照してください。

---

- テーブル・レベル

**dsi\_command\_convert** を使用して NY\_DS データ・サーバの nydb1 データベースにある tb1 テーブルで **d2none** を設定した後、すべての設定パラメータを表示するには、次のように入力します。

```
admin config, "table", NY_DS, nydb1
```

『Replication Server リファレンス・マニュアル』の「Replication Server コマンド」の「**admin config**」を参照してください。



### テーブル参照の表示

テーブル参照の情報と RTL の情報を表示するには、**rs\_helprep** を使用します。これは、Replication Server システム・データベース (RSSD) 上で実行できます。

**create replication definition** を使用して作成した **authors\_repdef** 複写定義に関する情報を表示するには、次のように入力します。

```
rs_helprep authors_repdef
```

『Replication Server リファレンス・マニュアル』の「RSSD ストアド・プロシージャ」の「**rs\_helprep**」を参照してください。

## 最終的な変更のデータベース

Replication Server には、最終的な変更を保管するデータベースがあります。これは、インメモリ・レポジトリとして機能し、トランザクションの最終的なロー変更、つまりコンパイルしたトランザクションを保管します。

各トランザクションに対して 1 つの最終的な変更のデータベース・インスタンスがあります。最終的な変更を保管するデータベース内の各複写テーブルには最高 3 つの追跡テーブルがあります。最終的な変更を保管するデータベースとその中のテーブルを点検することによって RTL 複写のモニタと問題のトラブルシューティングを行うことができます。

### 最終的な変更のデータベースをモニタする

最終的な変更のデータベース・インスタンスにアクセスしてモニタします。

**sysadmin cdb** コマンドを使って最終的な変更のデータベースをモニタします。

『Replication Server リファレンス・マニュアル』の「Replication Server コマンド」の「**sysadmin cdb**」を参照してください。

## 混合バージョンのサポートと下位互換性

RTL では、複写定義で指定されている参照制約を複写できるのは、アウトバウンド・ルートのバージョンが 15.5 以降の場合のみです。

アウトバウンド・ルートのバージョンが 15.5 より古くても RTL は機能します。しかし、バージョン 15.5 以降の場合は、参照制約情報を Replication Server で使用できません。

連続複写モードはサポートされているすべてのバージョンの Replication Server のデフォルト複写モードです。RTL を使用できるのは Replication Server 15.5 以降のみです。

## ステージング・ソリューションから RTL にマイグレートする

Sybase IQ への複写に現在ステージング・ソリューションを使用している場合、Real-Time Loading ソリューションにマイグレートします。

このシナリオでは、pdb がプライマリ Oracle データベース、PRS がプライマリ Replication Server、RRS がレプリケート Replication Server、staging\_db が Oracle ステージング・データベース、iqdb がレプリケート Sybase IQ データベースという複写構成を想定しています。このシナリオでのデータ・フローは、次のとおりです。

pdb -----> PRS -----> RRS -----> staging\_db -----> iqdb

### 前提条件

ステージング・ソリューションからマイグレートする前に、いくつかのタスクを実行する必要があります。

これらのタスクには次のようなものがあります。

1. プライマリおよびレプリケート Replication Servers をバージョン 15.5 以降にアップグレードする必要があります。詳細については、『Replication Server インストール・ガイド』と『Replication Server 設定ガイド』を参照してください。
2. pdb に流れ込むトランザクションがないこと、およびマイグレート中は複写システムがクワイース状態にあることを確認します。
  - a. すべてのプライマリ・データベースとシステム・データベースの Replication Agent を停止するために、Replication Server で次のコマンドを実行します。
 

```
suspend log transfer from all
```
  - b. Adaptive Server を RSSD として使用している場合は、次のコマンドで RSSD の RepAgent を停止します。
 

```
sp_stop_rep_agent rssid_name
```
  - c. Replication Server のキューが空になっていること、そして Replication Server がクワイース状態になっていることを、次のコマンドを実行して確認します。
 

```
admin quiesce_check
```

Replication Server がまだクワイース状態になっていない場合は、**admin quiesce\_force\_rsi** でリトライします。Replication Server がクワイース状態になっていなければ、データが失われる可能性があります。
3. pdb と iqdb が同期していることを確認します。データベース間の再同期は、ステージング・データベースにすべてのデータが複写されてから、iqdb にステージング・データベースのデータをロードすることによって行います。デー

データベース間の再同期を行わない場合は、iqdb のページとマテリアライズを行う必要があります。

4. Sybase IQ サーバがレプリケート Replication Server に接続してデータを抽出できるように、Sybase IQ の interfaces ファイルにそのレプリケート Replication Server のエントリを追加します。

### Real-Time Loading ソリューションにマイグレートする

ステージング・ソリューションから RTL にマイグレートします。

1. レプリケート Sybase IQ データ・サーバでメンテナンス・ユーザを作成します。または、既存のメンテナンス・ユーザを使用することもできます。
2. `rs_oracle_to_iq` 接続プロファイルと手順 1 のメンテナンス・ユーザ (`dbmaint` など) を使って、レプリケート Replication Server からレプリケート Sybase IQ データベースへのコネクションを作成します。

```
create connection to IQSRVR.iqdb
using profile rs_oracle_to_iq;standard
set username to dbmaint
set password to dbmaint
go
```

3. `dbo` は Sybase IQ 内に存在しないので、プライマリ・データベースで、`dbo` が所有するテーブルが `owner_on` としてマークされていない場合、Sybase IQ がそのテーブルを見つけることができるように、そのテーブルの `owner` を有効にする必要があります。

```
pdb_setreptable testtab, mark, owner
go
```

4. `owner` を有効にしたので、複写定義を作成し直して所有者情報を含めます。
5. テーブル間に参照制約がある場合、Replication Server がその参照制約の存在を計算に入れてバルク適用を適切な順序で実行できるように、参照制約を定義して複写定義を変更する必要があります。
6. レプリケート・データベースへのコネクションを RTL で有効にします。

```
alter connection to iqserver_name.iqdb
set dsi_compile_enable to 'on'
```

コネクションをサスペンドしてレジュームしたら、接続の変更が有効になります。

7. 各テーブルにサブスクリプションを作成します。プライマリ・データベースとレプリケート・データベースが同期している場合は、`without materialization` 句をサブスクリプションに含めます。それ以外の場合は、マテリアライゼーション時にオートコレクションを有効にする必要があります。

これで、Oracle から Sybase IQ に直接複写できるようになります。

### マイグレーション後のクリーンアップ

RTL を使用して複写を有効にして設定した後、ステージング・ソリューションのシステムをクリーンアップします。

1. ステージング・データベースのサブスクリプションを削除します。
2. 使用しない複写定義を削除します。
3. レプリケート Replication Server からステージング・データベースへのコネクションを削除します。
4. ステージング・データベースから Sybase IQ にデータを抽出する環境を終了します。

## パフォーマンスの強化

Replication Server 15.6 では、数種のパフォーマンス強化機能が組み込まれました。

### HVAR と RTL のリトライ・メカニズムの強化

リトライ・メカニズムの強化により、High-Volume Adaptive Replication (HVAR) と Real-Time Loading (RTL) の複写パフォーマンスが向上します。

HVAR と RTL はできるだけ多くのコンパイル可能なトランザクションをグループ化して、グループ内のトランザクションをまとめた最終的な変更としてコンパイルしてから、レプリケート・データベースでバルク・インタフェースを使用してその変更をレプリケート・データベースに適用しようとしています。HVAR と RTL の処理結果から発生する複写トランザクションが失敗すると、HVAR と RTL はリトライ・メカニズムを呼び出します。グループ内のトランザクションが失敗すると、RTL と HVAR はそのグループを同じサイズの 2 つのグループに分割し、コンパイルとバルク適用を各グループに対して試みます。リトライ・メカニズムは失敗したトランザクションを特定し、Replication Server がエラー・アクションのマッピングを実行できるようにします。また、DSI が停止する場合もあるので、失敗したトランザクションの前にあるすべてのトランザクションが適用されます。

HVAR と RTL 内の最終的な変更を保管するデータベースは、トランザクションの最終的なロー変更、つまりコンパイルしたトランザクションを保管するインメモリ・レポジトリとして機能します。最終的な変更を保管するデータベースの内容は、複数のプライマリ・トランザクションからのコマンドを集約したものであり、HVAR と RTL ではログ順に適用されません。したがって、リトライ・メカニズムがないと失敗したトランザクションを特定する方法がありません。グループ内のトランザクションが失敗したら、リトライ・メカニズムはそのグループを分割してコンパイルとバルク適用を繰り返します。このような連続したリトライ・プロセスはパフォーマンスの低下の原因となります。

リトライ・メカニズムの強化によって、HVAR または RTL でトランザクションが失敗したグループが検出された場合にグループが3等分され、失敗したトランザクションを含むグループの特定がより効率的に行われるようになりました。

さらに Replication Server 15.6 では、**dsi\_compile\_retry\_threshold** パラメータを使用してグループ内のコマンド数にスレッシュホールド値を指定できます。失敗したトランザクションを含むグループ内のコマンド数が **dsi\_compile\_retry\_threshold** の値より小さい場合、Replication Server はそのグループのリトライ処理を行わないので、処理時間を節約してパフォーマンスを向上できます。代わりに、Replication Server は連続複写モードに切り替わります。連続複写モードでは、プライマリ・データベースのログ順に従って変更がレプリケート・データベースに送信されます。

**configure replication server** は **dsi\_compile\_retry\_threshold** をサーバ・レベルで設定してすべてのレプリケート・データベース・コネクションに適用するときに使用し、**alter connection** は指定したデータベースとデータ・サーバへのコネクションに対して **dsi\_compile\_retry\_threshold** を設定するときに使用します。

---

**注意：** **dsi\_compile\_retry\_threshold** を使用するには、RTL または HVAR を **dsi\_compile\_enable** で有効にする必要があります。変更するパラメータごとに個別の **configure replication server** または **alter connection** コマンドを実行する必要があります。 **configure replication server** または **alter connection** を入力した後は、複数のパラメータを入力しないでください。

---

- サーバ・レベル

```
configure replication server
set dsi_compile_enable to 'on'
go
...
configure replication server
set dsi_compile_retry_threshold to 'value'
go
```

- データベース・コネクション・レベル

```
alter connection to data server.database
set dsi_compile_enable to 'on'
go
...
alter connection to data server.database
set dsi_compile_retry_threshold to 'value'
go
```

**dsi\_compile\_retry\_threshold** の有効な値は、0 ~ 2,147,483,647 の整数です。デフォルト値は 100 です。

**dsi\_compile\_retry\_threshold** を設定するときに、データベース・コネクションをサスペンドしてレジュームする必要はありません。このパラメータはコマンドを実行するとすぐに有効になります。

**参照：**

- RTL 設定 (68 ページ)

## キュー・ブロック・サイズの増加の機能強化

Replication Server を再起動しないでキュー・ブロック・サイズを変更できます。

キュー・ブロック・サイズはステーブル・キュー構造で使用される連続メモリ・ブロックのバイト数です。Replication Server 15.5 では、複写パフォーマンスを向上させるために、キュー・ブロック・サイズをデフォルトの 16KB から 32KB、64KB、128KB、または 256KB に増やすことができるようになりました。パフォーマンスの向上はトランザクション・プロファイルと環境にも依存します。ただし、バージョン 15.5 では、キュー・ブロック・サイズの変更を有効にするには Replication Server を再起動する必要があります。バージョン 15.6 では、キュー・ブロック・サイズの変更を有効にするために Replication Server を再起動する必要はありません。

---

**注意：** キュー・ブロック・サイズの増加機能を使用するには、REP\_HVAR\_ASE という名前の Advanced Services Option ライセンスが必要です。

---

### *推奨事項*

以下を行うことを強くおすすめします。

- キュー・ブロック・サイズを増やす前に十分なメモリがあることを確認する。
- その複写システムに最適な値を決めるために異なったキュー・ブロック・サイズを試す。

### *制限*

- キュー・ブロック・サイズの変更を実行している最中に Replication Server にデータが流れ込まないようにしてください。
- サブスクリプションのマテリアライゼーション、マテリアライゼーション解除、またはルートの作成や破棄を実行している最中にキュー・ブロック・サイズは変更できません。Replication Server は処理を続けますが、キュー・ブロック・サイズの変更はエラー・メッセージで終了します。
- いったんキュー・ブロック・サイズを変更する手順を開始すると、Replication Server はその処理が完了するまで、別のキュー・ブロック・サイズの変更コマンドを受け付けません。
- RSSD で直接キュー・ブロック・サイズを変更する別の手順を使用しないでください。異なった手順の使用によってキュー・ブロック・サイズの設定に一貫性がなくなり、Replication Server が停止する場合があります。

---

**注意：** キュー・ブロック・サイズの変更後はすべてのキューが空になります。

---

**参照：**

- Replication Server 15.6 の製品エディションとライセンス (55 ページ)

**キュー・ブロック・サイズを変更する**

キュー・ブロック・サイズの変更は Replication Server の設定における主要な変更であり、Replication Server へのすべてのコネクションに影響します。ログ転送をサスペンドして、Replication Server をクワイス状態にする必要があります。

Replication Server のキュー・ブロック・サイズの変更手順では、「アップストリーム」は Replication Server にデータを送るすべての複製システム・コンポーネントを、「ダウストリーム」は Replication Server からデータを受け取るコンポーネントを意味します。

1. データの整合性を維持するために、キュー・ブロック・サイズを変更する前に変更対象の Replication Server にデータが流れ込むのを止める必要があります。
  - a) すべての Replication Agent から設定を変更する Replication Server へのログ転送をサスペンドします。
  - b) Replication Agent からのアップストリームのログ転送をすべてサスペンドします。
  - c) すべてのアップストリーム Replication Server をクワイス状態にします。
  - d) 設定を変更する Replication Server への受信ルートをすべてサスペンドします。
  - e) 設定を変更する Replication Server をクワイス状態にします。
2. **configure replication server** の **set block\_size to 'value'** 句に設定変更対象の Replication Server のキュー・ブロック・サイズを設定して実行します。このコマンドは、次の処理を実行します。
  - a. 進行中のサブスクリプション・マテリアライゼーションが存在しないことを確認します。
  - b. すべてのログ転送がサスペンドされていることを確認します。
  - c. すべての受信ルートがサスペンドされていることを確認します。
  - d. Replication Server がクワイス状態であることを確認します。
  - e. キューをパージします。
  - f. `rs_locator` RSSD システム・テーブル内の値がゼロになって、キュー・ブロック・サイズ変更手順を開始したときにレプリケート・データベースに適用されていない可能性のあるトランザクションを Replication Agent が再送信できるようになります。
  - g. キュー・ブロック・サイズが入力された値に設定されます。
  - h. (省略可能) **with shutdown** オプションを含めた場合は、Replication Server が停止します。キュー・ブロック・サイズの変更は Replication Server を再起動す

ると有効になります。停止によって Replication Server は確実にすべてのメモリをクリアします。

3. キュー・ブロック・サイズが変更されたら、データ・フローをレジュームします。
  - a) **with shutdown** オプションを使用した場合は、Replication Server を再起動します。
  - b) Replication Agent からのログ転送をレジュームします。
  - c) 受信ルートをすべてレジュームします。

4. すべてのダウンストリーム Replication Server の RSSD とデータ・サーバでデータ・ロス調べます。通常は、設定を変更した Replication Server の RSSD でデータ・ロスがあります。設定を変更した Replication Server の RSSD からデータを受け取るレプリケート RSSD でのデータ・ロスは無視します。

データ・サーバでデータ・ロスを修復する手順に従います。RSSD でデータ・ロスがあった場合、影響を受けた Replication Server のログに次のようなメッセージが表示されます。

```
E. 2010/02/12 14:12:58. ERROR #6067 SQM(102:0 primaryDS.rssd) - /
sqmoqid.c(1071)
Loss detected for replicateDS.rssd from primaryDS.RSSD
```

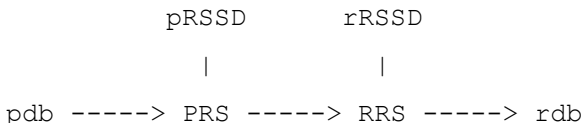
*replicateDS* はレプリケート・データ・サーバの名前、*primaryDS* はプライマリ・データ・サーバの名前です。

**単純な複写システムでキュー・ブロック・サイズを増やす**

次の単純な複写システムの例で、プライマリ Replication Server とレプリケート Replication Server のキュー・ブロック・サイズを設定する方法について説明します。

複写システムの構成は次のとおりです。

- プライマリ・データベース - pdb
- レプリケート・データベース - rdb
- プライマリ Replication Server - PRS
- プライマリ Replication Server の RSSD - pRSSD
- レプリケート Replication Server - RRS
- レプリケート Replication Server の RSSD - rRSSD



この例では、RSSD は、Replication Server システム・データベース (RSSD) として機能する Adaptive Server と Embedded Replication Server システム・データベース



(ERSSD) として機能する SQL Anywhere® の両方を意味します。すべてのコマンドの完全な構文、例、使用方法の詳細については、『Replication Server リファレンス・マニュアル』を参照してください。

## 1. プライマリ Replication Server を設定する

- a) すべての Replication Agent からのログ転送をサスペンドします。プライマリ Replication Server で次のコマンドを実行します。

```
suspend log transfer from all
```

- b) プライマリ Replication Server をクワイース状態にします。

```
admin quiesce_force_rsi
```

- c) プライマリ Replication Server のキュー・ブロック・サイズを 64KB に設定します。

```
configure replication server
set block_size to '64'
```

(省略可能) ブロック・サイズの設定で **with shutdown** オプションを使用して、プライマリ Replication Server を停止します。次に例を示します。

```
configure replication server
set block_size to '64' with shutdown
```

- d) トランザクション・ログを調べて、プライマリ Replication Server がマテリアライズ中でないこと、ログ転送とルートがサスペンドされていること、プライマリ Replication Server がクワイース状態であることを確認します。
- e) プライマリ Replication Server を停止した場合は、再起動します。  
『Replication Server 管理ガイド第 1 巻』の「複製システムの管理」の「Replication Server の起動」を参照してください。
- f) プライマリ Replication Server のトランザクション・ログを調べて、ブロック・サイズが変更されたことを確認します。
- g) Replication Agent がプライマリ Replication Server に接続できるように、ログ転送をレジュームします。プライマリ Replication Server で次のコマンドを実行します。

```
resume log transfer from all
```

- h) レプリケート Replication Server のログ・ファイルでデータ・ロスに関する情報を調べます。レプリケート Replication Server で **ignore loss** コマンドを実行して、プライマリ Replication Server の RSSD からレプリケート Replication Server の RSSD へのデータ・ロスを無視します。

```
ignore loss from PRS.pRSSD to RRS.rRSSD
```

詳細については、『Replication Server 管理ガイド第 2 巻』の「複製システム・リカバリ」の「ロスの無視」を参照してください。

## 2. レプリケート Replication Server を設定する

- a) すべての Replication Agent からのログ転送をサスペンドします。プライマリ Replication Server とレプリケート Replication Server で次のコマンドを実行します。

```
suspend log transfer from all
```

- b) プライマリ Replication Server をクワイース状態にします。

```
admin quiesce_force_rsi
```

- c) レプリケート Replication Server へのルートを持つすべての Replication Server でルートをサスペンドします。

```
suspend route to RRS
```

- d) レプリケート Replication Server をクワイース状態にします。

```
admin quiesce_force_rsi
```

- e) レプリケート Replication Server のブロック・サイズを 64KB に設定します。

```
configure replication server  
set block_size to '64'
```

(省略可能) **with shutdown** オプションを使用して、レプリケート Replication Server を停止します。次に例を示します。

```
configure replication server  
set block_size to '64' with shutdown
```

- f) トランザクション・ログを調べて、レプリケート Replication Server がマテリアライズ中でないこと、ログ転送とルートがサスペンドされていること、レプリケート Replication Server がクワイース状態であることを確認します。

- g) レプリケート Replication Server を停止した場合は、再起動します。

- h) レプリケート Replication Server のトランザクション・ログを調べて、ブロック・サイズが変更されたことを確認します。

- i) Replication Agent がレプリケート Replication Server に接続できるように、ログ転送をレジュームします。レプリケート Replication Server で次のコマンドを実行します。

```
resume log transfer from all
```

- j) Replication Agent がプライマリ Replication Server に接続できるように、ログ転送をレジュームします。プライマリ Replication Server で次のコマンドを実行します。

```
resume log transfer from all
```

- k) サスペンドしたルートをレジュームします。

```
resume route to RRS
```

- l) プライマリ Replication Server とレプリケート Replication Server のログ・ファイルでデータ・ロスに関する情報を調べます。レプリケート RSSD がプライマリ RSSD に複写されている場合は、プライマリ Replication Server で **ignore loss** コマンドを実行して、プライマリ RSSD とレプリケート RSSD の間のデータ・ロスを無視します。

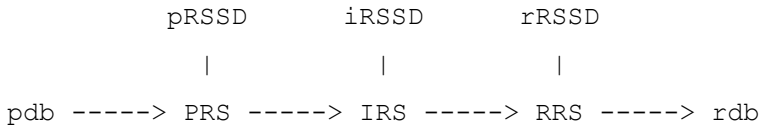
```
ignore loss from RRS.rRSSD to PRS.pRSSD
```

### 中間ルートを持つ複写システムでキュー・ブロック・サイズを増やす

次の中間ルートを持つ複写システムの例で、プライマリ Replication Server のキュー・ブロック・サイズを設定する方法について説明します。

複写システムの構成は次のとおりです。

- プライマリ・データベース – pdb
- レプリケート・データベース – rdb
- プライマリ Replication Server – PRS
- プライマリ Replication Server の RSSD – pRSSD
- レプリケート Replication Server – RRS
- レプリケート Replication Server の RSSD – rRSSD
- 中間 Replication Server – IRS
- 中間 Replication Server の RSSD – iRSSD



この例では、RSSD は、Replication Server システム・データベース (RSSD) として機能する Adaptive Server と Embedded Replication Server システム・データベース (ERSSD) として機能する SQL Anywhere の両方を意味します。すべてのコマンドの完全な構文、例、使用方法の詳細については、『Replication Server リファレンス・マニュアル』を参照してください。

1. すべての Replication Agent からのログ転送をサスペンドします。プライマリ Replication Server で次のコマンドを実行します。

```
suspend log transfer from all
```

2. PRS をクワイース状態にします。

```
admin quiesce_force_rsi
```

3. プライマリ Replication Server のブロック・サイズを 64KB に設定します。

```
configure replication server
set block_size to '64'
```

(省略可能) ブロック・サイズの設定で **with shutdown** オプションを使用して、プライマリ Replication Server を停止します。次に例を示します。

```
configure replication server
set block_size to '64' with shutdown
```

4. トランザクション・ログを調べて、プライマリ Replication Server がマテリアライズ中でないこと、ログ転送とルートがサスペンドされていること、プライマリ Replication Server がクワイース状態であることを確認します。
5. プライマリ Replication Server を停止した場合は、再起動します。『Replication Server 管理ガイド第1巻』の「複写システムの管理」の「Replication Server の起動」を参照してください。
6. プライマリ Replication Server のトランザクション・ログを調べて、ブロック・サイズが変更されたことを確認します。
7. Replication Agent がプライマリ Replication Server に接続できるように、ログ転送をレジュームします。プライマリ Replication Server で次のコマンドを実行します。

```
resume log transfer from all
```

8. 中間 Replication Server とレプリケート Replication Server のログ・ファイルでデータ・ロスに関する情報を調べます。中間 Replication Server で **ignore loss** コマンドを2回実行して、プライマリ Replication Server の RSSD からレプリケート Replication Server の RSSD、およびプライマリ Replication Server の RSSD から中間 Replication Server の RSSD へのデータ・ロスを無視します。

```
ignore loss from PRS.pRSSD to RRS  
go  
ignore loss from PRS.pRSSD to IRS.iRSSD
```

詳細については、『Replication Server 管理ガイド第2巻』の「複写システム・リカバリ」の「ロスの無視」を参照してください。

## 利便性とプロセスの向上

---

Replication Server 15.6 では各種の利便性とプロセスが強化されました。

### Adaptive Server のレプリケート・データベースの再同期

Replication Server を使用すると、レプリケート・データベースを再同期してマテリアライズできます。また、プライマリ・データベースのクワイースを強いることなく、データの損失や整合性を失うリスクなしで複写をレジュームできます。

データベース再同期化は、信頼されたソースから取得したデータ・ダンプを同期先のデータベースに適用することをベースとしています。

Oracle データベースを再同期するには、『Replication Server 異機種間複写ガイド』の「Oracle レプリケート・データベースの再同期」を参照してください。

### データベースの再同期を設定する

データベースの再同期を設定するには、Replication Server と RepAgent の両方のコマンドとパラメータを使用します。

1. RepAgent をサスペンドして複製プロセスを停止します。
2. Replication Server を再同期モードにします。  
再同期モードになると、Replication Server はトランザクションをスキップします。さらに、プライマリ・データベースまたは信頼されたソースから取得したダンプを使ってレプリケート・データベースにデータを再移植する準備として、複製キューから複製データをパージします。
3. RepAgent を再開して、データベース再同期マーカを Replication Server に送信し、再同期処理が進行中であることを示します。
4. DSI がデータベース再同期マーカを受け取ったことを確認します。
5. プライマリ・データベースからダンプを取得します。

Replication Server がプライマリ・データベース・ダンプが完了したことを示すダンプ・マーカを検出すると、Replication Server はトランザクションのスキップを停止し、どのトランザクションをレプリケート・データベースに適用するかを判定できるようになります。

6. DSI がデータベース・ダンプ・マーカを受け取ったことを確認します。

---

**注意：** データベース・ダンプのマーカの送信は、再同期マーカに `init` 命令を付けて送信した場合には適用されません。

---

7. ダンプをレプリケート・データベースに適用します。
8. 複製をレジュームします。

### Replication Server にトランザクションをスキップさせる

指定されたレプリケート・データベースで DSI アウトバウンド・キュー内のトランザクションを Replication Server にスキップさせるには、`skip to resync` パラメータを `resume connection` コマンドに付けて使用します。これは Replication Server が RepAgent からのデータベース・ダンプ・マーカを受け取るまで有効です。

レプリケート・データベース内のデータはダンプの内容によって置き換えられることになっているので、Replication Server はアウトバウンド・キュー内のレコードを処理しません。

『Replication Server リファレンス・マニュアル』の「Replication Server コマンド」の「`resume connection`」を参照してください。

次のコマンドを実行します。

```
resume connection to data_server.database skip to resync
marker
```

**警告！ resume connection を skip to resync marker オプションを付けて間違っ**たコ  
ネクションで実行すると、レプリケート・データベースのデータが非同期されま  
す。

**skip to resync marker** を設定すると、Replication Server は Replication Server ログ内ま  
たはデータベース例外ログ内でスキップされたトランザクションをログに記録し  
ません。**skip [n] transaction** を設定すると、Replication Server はスキップされたトラ  
ンザクションをログに記録します。

#### データベース再同期マーカを Replication Server に送信する

RepAgent を使用してデータベース再同期マーカを Replication Server に送信し、再  
同期処理が進行中であることを示します。

再同期モードで RepAgent を再開すると、RepAgent はデータベース再同期マーカを  
最初のメッセージとして Replication Server へ送信してから、SQL データ定義言語  
(DDL: data definition language) またはデータ操作言語 (DML: data manipulation  
language) のトランザクションを送信します。同じプライマリ・データベースの複  
数のレプリケート・データベースはそれぞれに DSI アウトバウンド・キューがあ  
るので、すべて同じ再同期マーカを受け取ります。

**skip to resync marker** パラメータでレジュームする各 DSI に対して、DSI が再同期  
マーカを受け取ったことが、DSI アウトバウンド・キューによって Replication  
Server システム・ログに記録されます。また、その時点からデータベース・ダン  
プ・マーカを受け取るまで DSI がコミットされたトランザクションを拒否するこ  
とも記録されます。

Adaptive Server では、データベース再同期マーカの送信で各オプションをサポート  
するために、**resync**、**resync purge**、または **resync init** パラメータを指定して  
**sp\_start\_rep\_agent** を使用します。

#### オプションを指定しないで再同期マーカを送信する

トランケーション・ポイントに変更がなく、RepAgent が最後に処理したところか  
らトランザクション・ログの処理を続けることになっているときは、  
**sp\_start\_rep\_agent** を使用してオプションを指定しないで再同期マーカを送信し  
ます。

構文：**sp\_start\_rep\_agent database\_name, 'resync'**

各アウトバウンド DSI スレッドとキューはデータベース再同期マーカを受け取り  
処理します。再同期マーカを受け取ったとき、DSI は skip to resync マーカ要求に  
従って Replication Server システム・ログにレポートを送ります。その後、ダン

ブ・データベース・マーカを待つ間、DSI はコミットされたトランザクションを拒否します。このメッセージと、ダンプ・データベース・マーカを待つ動作の変更によって、レプリケート・データベースにダンプを適用できるようになります。

#### パージ命令付きで再同期マーカを送信する

再同期マーカを送信するために **purge** オプションを指定して **sp\_start\_rep\_agent** を使用すると、新しいインバウンド・トランザクションを受け取る前にインバウンド・キュー内のすべてのオープン・トランザクションをパージして重複の検出をリセットするよう、Replication Server に指示できます。

構文： **sp\_start\_rep\_agent database\_name, 'resync purge'**

プライマリ・データベースのトランケーション・ポイントが移動した場合は、**purge** オプションを使用します。これは次の操作を行った場合に発生します。

- トランケーション・ポイントを手動で変更する。
- RepAgent を無効にする。
- **dbcc dbrepair** などの Adaptive Server コマンドを実行する。

トランケーション・ポイントが変更されると、Replication Server のインバウンド・キュー内にあるオープン・トランザクションは、新しいセカンダリ・トランケーション・ポイントから送られたアクティビティと一致しないため、パージされる必要があります。変更されたトランケーション・ポイントが以前のオリジン・キュー ID (OQID) を持つレコードを送信する可能性があるため、Replication Server は重複検出をリセットします。以前のデータがキューからパージされると、Replication Server は RepAgent からのすべてのアクティビティを重複アクティビティとして扱いません。したがって、新しいアクティビティが拒否されることはありません。ダンプ・データベース・マーカを受け取るまで、Replication Server はアウトバウンド・キュー・コマンドを拒否し続けるので、パージ・オプションは DSI の処理を変更しません。

#### init コマンド付きで再同期マーカを送信する

再同期マーカを **init** コマンド付きで送信するには、**init** オプションを指定して **sp\_start\_rep\_agent** を使用します。これによって、インバウンド・キュー内のすべてのオープン・トランザクションをパージして重複の検出をリセットし、アウトバウンド DSI をサスペンドするよう、Replication Server に指示できます。

構文： **sp\_start\_rep\_agent database\_name, 'resync init'**

このオプションはプライマリ・データベースにレプリケート・データベースと同じダンプを再ロードするときに使用します。プライマリ・データベースから取得したダンプはないので、RepAgent はダンプ・データベース・マーカを送りません。再同期マーカの後に来るダンプ・データベース・マーカを待つ代わりに、**init** オプションは Replication Server が再同期マーカを処理したらすぐに DSI コネクションをサスペンドします。

DSI がサスペンドされたら、それ以降 DSI を通るすべてのアクティビティは新しいトランザクションのみになります。プライマリで使用したダンプをレプリケート・データベースに再ロードしたら、DSI をレジュームできます。

**参照：**

- ダンプ・データベース・マーカを Replication Server に送信する (92 ページ)
- プライマリ・データベースとレプリケート・データベースを同じダンプから再同期する (98 ページ)

データベースのダンプを取得する

Adaptive Server の **dump database** コマンドを使用します。

Adaptive Server Enterprise の『システム管理ガイド 第 2 巻』の「バックアップおよびリカバリ・プランの作成」の「**dump** コマンドおよび **load** コマンドの使用方法」を参照してください。

ダンプ・データベース・マーカを Replication Server に送信する

RepAgent はプライマリ・データベースからダンプを取得すると、Replication Server にダンプ・データベース・マーカを自動的に生成して送信します。

---

**注意：** ダンプ・データベース・マーカの送信は、再同期マーカに **init** 命令を付けて送信した場合には適用されません。

---

レプリケート・データベースにダンプを適用したら、手動で DSI をレジュームできます。ダンプ・データベース・マーカが示すダンプ・ポイントの後にコミットされたトランザクションは、レプリケートされます。

DSI スレッド情報をモニタする

データベースの再同期中に DSI についての情報を提供するには、**admin who** コマンドを使用します。

| 状態               | 説明                                                                             |
|------------------|--------------------------------------------------------------------------------|
| SkipUntil Resync | <b>skip to resync</b> を実行した後 DSI がレジュームする。このステータスは DSI がデータベース再同期マーカを受け取るまで続く。 |
| SkipUntil Dump   | DSI がデータベース再同期マーカを受け取った。このステータスは DSI がその後のダンプ・データベース・マーカを受け取るまで続く。             |

再同期するデータベースにダンプを適用する

プライマリ・データベースのダンプをレプリケート・データベースに適用できるのは、関連するメッセージがシステム・ログに表示された後だけです。

- Replication Server が **purge** オプション付き、またはなしの再同期データベース・マーカと、ダンプ・データベース・マーカを受け取る時のメッセージ。



```
DSI for data_server.database received and processed
Resync Database Marker. Waiting for Dump Marker.
```

```
DSI for data_server.database received and processed
Dump Marker. DSI is now suspended. Resume after database has been
reloaded.
```

- Replication Server が **init** マーカ付きの再同期データベースを受け取る時のメッセージ。

```
DSI for data_server.database received and processed
Resync Database Marker. DSI is now suspended. Resume after
database has been reloaded.
```

再同期するデータベースにダンプをロードする方法の詳細については、『Adaptive Server Enterprise リファレンス・マニュアル：コマンド』の「コマンド」の「**load database**」を参照してください。

### データベース再同期化シナリオ

データベースの再同期手順はそのシナリオによって異なります。再同期手順を完了すると、プライマリ・データベースとレプリケート・データベースはトランザクションの一貫性が保たれた状態になります。

手順を実行するには次の要件があります。

- 複写システム管理者であること。
- 正常に稼動する複写環境が存在すること。
- プライマリ・データベースからレプリケート・データベースへデータをコピーするためのメソッドやプロセスがあること。

Adaptive Server と Replication Server の RepAgent のコマンドと構文については、『Replication Server リファレンス・マニュアル』、および『Replication Server 管理ガイド 第 1 巻』の「RepAgent の管理と Adaptive Server のサポート」を参照してください。

### 1つ以上のレプリケート・データベースをプライマリ・データベースから直接再同期する

1つ以上のレプリケート・データベースを1つのプライマリ・データベースから再同期します。

この手順では、多少の違いはありますが、次のことを実行できます。

- プライマリ・データベースとレプリケート・データベース間の複写の遅延時間が、複写によるデータベースの回復が不可能で、複写データに基づくレポートの作成が実用的でなくなった場合に、レプリケート・データベースにデータを再移植する。
- プライマリ・データベースから信頼されたデータをレプリケート・データベースに再移植する。

- プライマリ・データベースが複数のレプリケート・データベースのソースになっている場合に、再同期を調整する。
- プライマリ・サイトが一对のウォーム・スタンバイ・データベースで構成されている論理データベースであり、それに1つまたは複数のレプリケート・データベースを再同期する場合に、再同期を調整する。ウォーム・スタンバイのペアでは、アクティブ・データベースがプライマリ・データベースとして、スタンバイがレプリケート・データベースとして機能する。したがって、1つまたは複数のレプリケート・データベースからはアクティブ・データベース(プライマリ・サイトのウォーム・スタンバイ・ペアの1つ)がプライマリ・データベースに見える。

### 参照：

- ウォーム・スタンバイ・アプリケーションのアクティブ・データベースとスタンバイ・データベースの再同期 (99 ページ)

#### プライマリ・データベースから直接再同期する

レプリケート・データベースをプライマリ・データベースから直接再同期します。

1. RepAgent による複製プロセスを停止します。Adaptive Server で次のコマンドを実行します。

```
sp_stop_rep_agent database
```

2. レプリケート・データベースとの Replication Server DSI コネクションをサスペンドします。

```
suspend connection to dataserver.database
```

3. レプリケート・データベースのアウトバウンド・キューからデータを削除し、プライマリ・データベースの RepAgent からの再同期マーカを待機するように Replication Server に指示します。

```
resume connection to data_server.database skip to  
resync marker
```

4. RepAgent に再同期モードで起動するよう指示し、再同期マーカを Replication Server に送信します。

- トランケーション・ポイントが元の位置から移動していない場合は、Adaptive Server で次のコマンドを実行します。

```
sp_start_rep_agent database, 'resync'
```

- トランケーション・ポイントが元の位置から移動している場合は、Adaptive Server で次のコマンドを実行します。

```
sp_start_rep_agent database, 'resync purge'
```

5. Replication Server システム・ログで次のメッセージを検索して、DSI が RepAgent から再同期マーカを受信して受け入れていることを確認します。

```
DSI for data_server.database received and processed
Resync Database Marker. Waiting for Dump Marker.
```

**注意：** 複数のデータベースを再同期する場合は、再同期する各データベースの DSI コネクションが再同期マーカを受け入れていることを確認します。

6. プライマリ・データベース・コンテンツのダンプを取得します。詳細については、『Adaptive Server Enterprise リファレンス・マニュアル：コマンド』の「コマンド」の「**dump database**」を参照してください。Adaptive Server は自動的にダンプ・データベース・マーカを生成します。

7. Replication Server システム・ログで次のメッセージを検索して、Replication Server がダンプ・データベース・マーカを処理していることを確認します。

```
DSI for data_server.database received and processed
Dump Marker. DSI is now suspended. Resume after database has been
reloaded.
```

Replication Server がダンプ・マーカを受け取ると、DSI コネクションが自動的にサスペンドされます。

8. プライマリ・データベースのダンプをレプリケート・データベースに適用します。詳細については、『Adaptive Server Enterprise リファレンス・マニュアル：コマンド』の「コマンド」の「**load database**」を参照してください。
9. レプリケート・データベースにダンプを適用したら、次のコマンドを使用して DSI をレジュームします。

```
resume connection to data_server.database
```

### サードパーティ・ダンプ・ユーティリティを使用して再同期する

ディスク・スナップショット用ツールのようなサードパーティ・ダンプ・ユーティリティを使ってプライマリ・データベースをダンプした後、再同期を調整します。

サードパーティ・ツールでは、プライマリ・データベースとのやり取りをネイティブのデータベース・ダンプ・ユーティリティほど密接には行うことができません。RepAgent がダンプ・データベース・マーカの生成に使用できるような記録をサードパーティ・ツールがプライマリ・データベースのトランザクション・ログに書き込まない場合は、独自のダンプ・データベース・マーカを生成して再同期処理を完了できるようにします。詳細については、サードパーティ・ツールのマニュアルを参照してください。

1. RepAgent による複製プロセスを停止します。Adaptive Server で次のコマンドを実行します。

```
sp_stop_rep_agent database
```

2. レプリケート・データベースとの Replication Server DSI コネクションをサスペンドします。

```
suspend connection to dataserver.database
```

- レプリケート・データベースのアウトバウンド・キューからデータを削除し、プライマリ・データベースの RepAgent からの再同期マーカを待機するように Replication Server に指示します。

```
resume connection to data_server.database skip to  
resync marker
```

- サードパーティ・ユーティリティを使用してプライマリ・データベースのコンテンツのダンプを取得します。
- サードパーティ・ツールからダンプまたは情報を取得したら、プライマリ・データベースからの情報に基づいてダンプ・ポイントを決定します。サードパーティ・ツールを使用する場合、ユーザはダンプ・ポイントを決定する責任があります。たとえば、ディスク複写ツールを使用する場合、プライマリ・データベースでアクティビティを一時的に停止してディスク・スナップショットから実行中のトランザクションを消去し、ダンプ・データベース・マーカとして「トランザクション・ログの末尾」ポイントを使用できます。
- 手順 5 で取得したダンプの位置の末尾に RepAgent 用のマークを付けるには、プライマリ・データベースで **rs\_marker** ストアド・プロシージャを実行します。

```
rs_marker "dump database database_name 'current date' oqid"
```

ここで、*current date* は datetime 形式の任意の値、*oid* は任意の有効な 16 進値です。『Replication Server リファレンス・マニュアル』の「トピック」の「データ型」の「日付と時間のデータ型」の「日付および時間の値の入力フォーマット」を参照してください。

たとえば、`rdbl1` 上のダンプ位置の末尾を日付および時間値 "20110915 14:10:10" と *oid* の値 0x0003 でマーク付けすることができます。

```
rs_marker "dump database rdbl1 '20110915 14:10:10' 0x0003"
```

RepAgent は、手順 6 でマーク付けしたポイントのダンプ・データベース・マーカを自動的に生成し、それを Replication Server に送信します。

- 再同期モードで RepAgent を開始し、再同期マーカを Replication Server に送信します。

- トランケーション・ポイントが元の位置から移動していなければ、Adaptive Server で次のコマンドを実行します。

```
sp_start_rep_agent database, 'resync'
```

- トランケーション・ポイントが元の位置から移動している場合は、Adaptive Server で次のコマンドを実行します。

```
sp_start_rep_agent database, 'resync purge'
```

- Replication Server システム・ログで次のメッセージを検索して、DSI が Replication Agent から再同期マーカを受信して受け入れていることを確認します。

```
DSI for data_server.database received and processed
Resync Database Marker. Waiting for Dump Marker.
```

9. Replication Server システム・ログで次のメッセージを検索して、Replication Server がダンプ・データベース・マーカを処理していることを確認します。

```
DSI for data_server.database received and processed
Dump Marker. DSI is now suspended. Resume after
database has been reloaded.
```

Replication Server がダンプ・マーカを受け取ると、DSI コネクションが自動的にサスペンドされます。

10. プライマリ・データベースのサードパーティ・ツールからのダンプをレプリケート・データベースに適用します。詳細については、Adaptive Server のマニュアルとサードパーティ・ツールのマニュアルを参照してください。
11. レプリケート・データベースにダンプを適用したら、次のコマンドを使用して DSI をレジュームします。

```
resume connection to data_server.database
```

データベース再同期マーカに対するサポートがない場合に再同期する  
RepAgent またはプライマリ・データベースが再同期マーカを自動生成するように更新されていない場合に再同期を調整します。

---

**注意：** この手順は Adaptive Server でしか実行できません。

---

1. レプリケート・データベースとの Replication Server DSI コネクションをサスペンドします。

```
suspend connection to dataserver.database
```

2. レプリケート・データベースのアウトバウンド・キューからデータを削除し、プライマリ・データベースの RepAgent からの再同期マーカを待機するように Replication Server に指示します。

```
resume connection to data_server.database skip to
resync marker
```

3. システム・ログにオープン・トランザクションがないことを確認してから、プライマリ・データベースで **resync marker** を手動生成します。

```
execute rs_marker 'resync database'
```

4. Replication Server システム・ログで次のメッセージを検索して、DSI が RepAgent から再同期マーカを受信して受け入れていることを確認します。

```
DSI for data_server.database received and processed
Resync Database Marker. Waiting for Dump Marker.
```

5. プライマリ・データベース・コンテンツのダンプを取得します。

Adaptive Server は自動的にダンプ・データベース・マーカを生成します。  
『Adaptive Server Enterprise リファレンス・マニュアル：コマンド』の「コマンド」の「**dump database**」を参照してください。

6. Replication Server システム・ログで次のメッセージを検索して、Replication Server がダンプ・データベース・マーカを処理していることを確認します。

```
DSI for data_server.database received and processed  
Dump Marker. DSI is now suspended. Resume after database has been  
reloaded.
```

Replication Server がダンプ・マーカを受け取ると、DSI コネクションが自動的にサスペンドされます。

7. プライマリ・データベースのダンプをレプリケート・データベースに適用します。詳細については、『Adaptive Server Enterprise リファレンス・マニュアル：コマンド』の「コマンド」の「**load database**」を参照してください。
8. レプリケート・データベースにダンプを適用したら、次のコマンドを使用して DSI をレジュームします。

```
resume connection to data_server.database
```

### プライマリ・データベースとレプリケート・データベースを同じダンプから再同期する

再同期を調整して、プライマリ・データベースとレプリケート・データベースを同じダンプまたはデータのコピーから再ロードします。プライマリ・データベースからダンプを取得しないので、ダンプ・データベース・マーカは必要ありません。

1. RepAgent による複製プロセスを停止します。トランケーション・ポイントを変更しないでください。

Adaptive Server で次のコマンドを実行します。

```
sp_stop_rep_agent database
```

2. レプリケート・データベースとの Replication Server DSI コネクションをサスペンドします。

```
suspend connection to data_server.database
```

3. レプリケート・データベースのアウトバウンド・キューからデータを削除し、プライマリ・データベースの RepAgent からの再同期マーカを待機するように Replication Server に指示します。

```
resume connection to data_server.database skip to  
resync marker
```

4. ダンプを適用する前に RepAgent の設定を取得します。

**注意：** Adaptive Server はデータベース内に RepAgent が使用する接続設定およびその他の設定を保存します。別のデータベースから取得したダンプをプライマリ・データベースにロードすると、RepAgent はその設定を失い、設定はダンプを取得した元のデータベースの設定に一致するように変更されます。

5. プライマリ・データベースに外部ソースからのデータ・ダンプを適用します。ダンプを適用したら、RepAgent の設定をダンプを適用する前の設定にリセットします。
6. プライマリ・データベースのトランザクション・ログにレプリケート・データベース・テーブルに影響するオペレーションが含まれていないことを確認するために、プライマリ Adaptive Server データベースで次のコマンドを実行します。

```
rs_update_lastcommit 0, 0, 0, ""
go_100
```

7. プライマリ・データベースのトランザクション・ログの最後までトランザクション・ポイントを移動します。Adaptive Server で次のコマンドを実行します。

```
dbcc settrunc('ltm', 'end')
go
```

8. **init** 命令を付けて再同期モードで RepAgent を開始します。Adaptive Server で次のコマンドを実行します。

```
sp_start_rep_agent database, 'resync init'
```

9. Replication Server システム・ログで次のメッセージを検索して、DSI が RepAgent から再同期マーカを受信して受け入れていることを確認します。

```
DSI for data_server.database received and processed
Resync Database Marker. DSI is now suspended. Resume
after database has been reloaded.
```

Replication Server が **init** マーカ付きのデータベース再同期指示を受信して処理すると、DSI コネクションはサスペンドします。

10. レプリケート・データベースに外部ソースからのデータ・ダンプを適用します。
11. レプリケート・データベースにダンプを適用したら、レプリケート・データベースで DSI をレジュームして Replication Server がプライマリ・データベースからのトランザクションを適用できるようにします。

```
resume connection to data_server.database
```

### ウォーム・スタンバイ・アプリケーションのアクティブ・データベースとスタンバイ・データベースの再同期

ウォーム・スタンバイ・ペアが単一プライマリ・データベースのレプリケート・サイトになっているときに、ウォーム・スタンバイ環境でアクティブ・データベースとスタンバイ・データベースを再同期します。

このシナリオでは、レプリケート・サイトはウォーム・スタンバイ・ペアです。ウォーム・スタンバイ・ペアは、アクティブ・データベースとスタンバイ・データベースから構成され、1つの論理データベースとして機能します。

```
Primary ---> Replication ---> Replicate logical database
database      Server          [Active+Standby warm standby
                               pair]
```

この再同期化シナリオ手順は、ウォーム・スタンバイ・ペアのアクティブなレプリケート・データベースをプライマリ・データベースからのダンプに再同期するプロセスと、その次のウォーム・スタンバイ・ペアのスタンバイ・レプリケート・データベースをアクティブ・データベースからのダンプまたはプライマリ・データベースからの既存のダンプに再同期するプロセスの2段階の再同期プロセスです。

1. プライマリ・データベースの RepAgent とウォーム・スタンバイ・アクティブ・データベースの RepAgent による複製プロセスを停止します。

Adaptive Server で次のコマンドを実行します。

```
sp_stop_rep_agent database
```

2. アクティブ・データベースとスタンバイ・データベースとの Replication Server DSI コネクションをサスペンドします。

```
suspend connection to dataserver.database
```

3. アクティブ・データベースとスタンバイ・データベースのアウトバウンド・キューからデータを削除し、プライマリ・データベースの RepAgent からの再同期マーカを待機するように Replication Server に指示します。

```
resume connection to data_server.database skip to
resync marker
```

4. 再同期モードでプライマリ・データベースの RepAgent を開始し、再同期マーカを Replication Server に送信します。

- トランケーション・ポイントが元の位置から移動していなければ、Adaptive Server で次のコマンドを実行します。

```
sp_start_rep_agent database, 'resync'
```

- トランケーション・ポイントが元の位置から移動していれば、Adaptive Server で次のコマンドを実行します。

```
sp_start_rep_agent database, 'resync purge'
```

5. Replication Server システム・ログで次のメッセージを検索して、アクティブ・データベースの DSI がプライマリ・データベース RepAgent から再同期マーカを受信して受け入れていることを確認します。

```
DSI for data_server.database received and processed
Resync Database Marker. Waiting for Dump Marker.
```



6. プライマリ・データベース・コンテンツのダンプを取得します。詳細については、『Adaptive Server Enterprise リファレンス・マニュアル：コマンド』の「コマンド」の「**dump database**」を参照してください。Adaptive Server は自動的にダンプ・データベース・マーカを生成します。
7. ダンプを適用する前に RepAgent の設定を取得します。

**注意：** Adaptive Server はデータベース内に RepAgent が使用する接続設定およびその他の設定を保存します。別のデータベースから取得したダンプをプライマリ・データベースにロードすると、RepAgent はその設定を失い、設定はダンプを取得した元のデータベースの設定に一致するように変更されます。

8. Replication Server システム・ログでアクティブ・データベースから次のメッセージを検索して、アクティブ・データベースの Replication Server DSI がダンプ・データベース・マーカを処理していることを確認します。

```
DSI for data_server.database received and processed
Dump Marker. DSI is now suspended. Resume after database has been
reloaded.
```

9. プライマリ・データベースのダンプをアクティブ・データベースに適用します。詳細については、『Adaptive Server Enterprise リファレンス・マニュアル：コマンド』の「コマンド」の「**load database**」を参照してください。

ダンプを適用したら、RepAgent の設定をダンプを適用する前の設定にリセットします。

10. プライマリ・データベースのトランザクション・ログにレプリケート・データベース・テーブルに影響するオペレーションが含まれていないことを確認するために、プライマリ Adaptive Server データベースで次のコマンドを実行します。

```
rs_update_lastcommit 0, 0, 0, ""
go 100
```

11. アクティブ・データベースのトランザクション・ログの最後までトランケーション・ポイントを移動します。Adaptive Server で次のコマンドを実行します。

```
dbcc settrunc('\ltm', 'end')
go
```

12. **init** 命令を付けて再同期モードで RepAgent を開始します。Adaptive Server で次のコマンドを実行します。

```
sp_start_rep_agent database, 'resync init'
```

13. Replication Server システム・ログで次のメッセージを検索して、スタンバイ・データベースの DSI がアクティブ・データベース RepAgent から再同期マーカを受信して受け入れていることを確認します。

```
DSI for data_server.database received and processed
Resync Database Marker. DSI is now suspended. Resume
after database has been reloaded.
```

Replication Server が **init** マーカ付きのデータベース再同期指示を受信して処理すると、DSI コネクションはサスペンドします。

14. アクティブ・データベースのコンテンツのダンプを取得し、スタンバイ・データベースにダンプを適用します。ダンプにデータベース設定情報が含まれない場合は、手順 6 からプライマリ・データベースのダンプを適用することもできます。
15. アクティブ・データベースとスタンバイ・データベースの DSI をレジュームします。

```
resume connection to data_server.database
```

### 参照：

- 1 つ以上のレプリケート・データベースをプライマリ・データベースから直接再同期する (93 ページ)

## 範囲による例外の削除

RSSD 例外ログ内のトランザクションを個別にではなく範囲を指定して削除するには、ストアド・プロシージャを使用します。

**rs\_delexception** ストアド・プロシージャでは、例外ログから削除するトランザクションは一度に 1 つしか指定できません。Replication Server 15.6 では、**rs\_delexception\_id**、**rs\_delexception\_date**、**rs\_delexception\_range** を使用して、削除するトランザクションの範囲を指定できます。

トランザクションの範囲を指定すると、Replication Server はその範囲のトランザクションに対する一時的なテーブルを 1 つだけ作成するので、個々のトランザクションに 1 つのテーブルを作成するのに比べて、リソースが節約されパフォーマンスが向上します。

### rs\_delexception\_id

**rs\_exceptscmd**、**rs\_exceptshdr**、**rs\_systext** システム・テーブル内の例外ログで、トランザクション ID によって指定された範囲のトランザクションを削除します。

### 構文

```
rs_delexception_id transaction_id_start [,transaction_id_end]
```

### パラメータ

- **transaction\_id\_start** – 削除するトランザクションの範囲の最初の ID 番号。

- **transaction\_id\_end** – 削除するトランザクションの範囲の最後の ID 番号。範囲の最後のトランザクションの指定は省略可能。

## 例

- **例 1** – 例外ログから ID 番号が 1234 のトランザクションを削除します。トランザクションを 1 つ削除する場合は、**rs\_delexception\_id** も使用できます。

```
rs_delexception_id 1234
```

- **例 2** – ID 番号 1234 ~ 9800 のトランザクションをすべて例外ログから削除します。

```
rs_delexception_id 1234, 9800
```

## 使用法

- **rs\_delexception\_id** は *transaction\_id\_start* ~ *transaction\_id\_end* の範囲にあるトランザクションを例外テーブルから削除します。このとき、*transaction\_id\_start* と *transaction\_id\_end* も削除されます。
- パラメータを指定しないと、**rs\_delexception\_id** はエラー・メッセージを表示します。例外ログ内に現在ある有効なトランザクションのリストを取得するには、**rs\_helpexception** または **rs\_delexception** をパラメータを指定しないで実行します。
- トランザクション ID の有効な値を 1 つだけ *transaction\_id\_start* で指定して、2 つ目のトランザクション ID 番号を *transaction\_id\_end* で指定しないと、**rs\_delexception\_id** は *transaction\_id\_start* で指定されたトランザクションのみを削除します。
- トランザクション ID 番号として 0 (ゼロ) を入力して、2 つ目のトランザクション ID 番号を入力しないと、**rs\_delexception\_id** は例外ログ内のすべてのトランザクションを削除します。
- 123.456 のような浮動小数点数を入力すると、次のように処理されます。
  - **ERSSD** – **rs\_delexception\_id** は整数部 123 のみを処理して小数点以下の数は無視します。
  - **RSSD** – **rs\_delexception\_id** はエラー・メッセージを返すので、コマンドを再入力できます。
- 入力したコマンドによって削除されたトランザクションがない場合、**rs\_delexception\_id** はエラー・メッセージを表示します。

### **rs\_delexception\_date**

rs\_exceptscmd、rs\_exceptshdr、rs\_systext システム・テーブル内の例外ログで、トランザクションの日付によって指定された範囲のトランザクションを削除します。

### **構文**

```
rs_delexception_date transaction_date_start [,transaction_date_end]
```

### **パラメータ**

- **transaction\_date\_start** – 削除するトランザクションの範囲の最初の日付 (最も古いトランザクション開始の日付)。日付は二重引用符で囲む。
- **transaction\_date\_end** – 削除するトランザクションの範囲の最後の日付 (最も新しいトランザクション開始の日付)。範囲の最後の日付 (トランザクション開始日付) の指定は省略可能。日付は二重引用符で囲む。

### **例**

- **例1** – 開始の日付が "10/01/2010" であるトランザクションを例外ログから削除します。

```
rs_delexception_date "10/01/2010"
```

- **例2** – 開始の日付が "10/01/2010" ~ "10/31/2010" の範囲にあるトランザクションを例外ログから削除します。

```
rs_delexception_date "10/01/2010", "10/31/2010"
```

### **使用法**

- *transaction\_date\_start* と *transaction\_date\_end* に入力する日付の形式には、RSSD のホスト Adaptive Server または ERSSD として機能する SQL Anywhere データベースがサポートする形式を使用できます。使用できる日付と時刻の形式については、以下を参照してください。
  - 『Adaptive Server Enterprise リファレンス・マニュアル：ビルディング・ブロック』：「システム・データ型とユーザ定義データ型」の「日付と時刻のデータ型」の「日付および時刻データの入力」
  - 『SQL Anywhere サーバー SQL リファレンス』：「SQL データ型」の「日付と時刻データ型」の「日付と時刻をデータベースに送信する」
- **rs\_delexception\_date** は *transaction\_date\_start* ~ *transaction\_date\_end* の範囲にあるトランザクションを例外テーブルから削除します。このとき、*transaction\_date\_start* と *transaction\_date\_end* も削除されます。
- パラメータを指定しないと、**rs\_delexception\_date** はエラー・メッセージを表示します。**rs\_helpexception** または **rs\_delexception** をパラメータを指定しないで

実行した場合は、"org date" カラムを参照して例外ログにおける現在の有効なトランザクションと開始日付を取得します。

- 有効な日付を *transaction\_date\_start* に指定し、2 つ目の有効な日付を *transaction\_date\_end* に指定しなかった場合、**rs\_delexception\_date** は *transaction\_date\_start* に指定されたトランザクションのみを削除します。
- 入力したコマンドによって削除されたトランザクションがない場合、**rs\_delexception\_date** はエラー・メッセージを表示します。

### **rs\_delexception\_range**

トランザクションの範囲を、システム・テーブル *rs\_exceptscmd*、*rs\_exceptshdr*、*rs\_systext* にある例外ログ内の元のサイトまたはユーザ、または送信先サイトで指定します。

### **構文**

```
rs_delexception_range
{{"origin"|"org"}, "origin_data_server.origin_database" |
, {"destination"|"dest"},
"destination_data_server.destination_database" |
, "user", "origin_user"}
```

### **パラメータ**

- **"origin"|"org"**, ***origin\_data\_server.origin\_database*** – **origin** か短縮形の **org** を入力して、例外ログから削除するトランザクションを開始したデータ・サーバおよびデータベースを指定します。パラメータは二重引用符で囲み、カンマでパラメータを区切ります。
- **"destination"|"dest"**, ***destination\_data\_server.destination\_database*** – **destination** か短縮形の **dest** を入力して、例外ログから削除するトランザクションを受け取ったデータ・サーバおよびデータベースを指定します。パラメータは二重引用符で囲み、カンマでパラメータを区切ります。
- **"user"**, ***origin\_user*** – **user** を入力して、例外ログから削除するトランザクションを開始したユーザを指定します。パラメータは二重引用符で囲み、カンマでパラメータを区切ります。

### **例**

- **例 1** – SYDNEY\_DS というデータ・サーバの *south\_db* というデータベースが開始したトランザクションを例外ログから削除します。

```
rs_delexception_range "org", "SYDNEY_DS.south_db"
```

- **例 2** – TOKYO\_DS というデータ・サーバの *east\_db* というデータベースが受け取ったトランザクションを例外ログから削除します。

```
rs_delexception_range "destination", "TOKYO_DS.east_db"
```

- **例 3** – rsuser1 というユーザが開始したトランザクションを例外ログから削除します。

```
rs_delexception_range "user", "rsuser1"
```

## 使用法

- 一度に入力できるパラメータとその値は 1 つだけです。たとえば、**"org"** と **"origin\_dataserver.origin\_database"** に続けて **"user"** と **"origin\_user"** を入力することはできません。
- パラメータを入力して値を指定する必要があります。パラメータを指定しないと、**rs\_delexception\_range** はエラー・メッセージを表示します。  
**rs\_helpexception** または **rs\_delexception** をパラメータを指定しないで実行した場合は、Origin Site、Dest. Site、Dest. User カラムを参照してそれぞれのカラムの現在の値のリストを取得し、例外ログ内の有効なトランザクションを調べます。
- **rs\_delexception\_range** で **"origin"**、**"destination"**、または **"user"** のみを入力して、対応する値を指定しなかった場合、**rs\_delexception\_range** はエラー・メッセージを表示します。
- 入力したコマンドによって削除されたトランザクションがない場合、**rs\_delexception\_range** はエラー・メッセージを表示します。

## ロー・カウントの検証を制御する

ロー・カウントの検証をオフにするには、**dsi\_row\_count\_validation** を使用します。

バージョン 15.2 以降の Replication Server では、ロー・カウントの検証がデフォルトで有効になっているので、ロー・カウントの不一致などのさまざまなロー・カウント検証エラーに対して自動的にエラー・メッセージが表示され、デフォルト・エラー・アクションが実行されます。Replication Server エラー・クラスを設定してさまざまなエラー・アクションを有効にできます。

Replication Server 15.6 では、同期されていないテーブル・ローがあり、デフォルトのエラー・アクションとメッセージをバイパスする場合、**dsi\_row\_count\_validation** を **off** に設定してロー・カウントの検証を無効にできます。

デフォルトでは **dsi\_row\_count\_validation** が **on** に設定されているので、ロー・カウントの検証は有効になっています。

**configure replication server** は **dsi\_row\_count\_validation** をサーバ・レベルで設定してすべてのレプリケート・データベース・コネクションに適用するときに使用し、**alter connection** は指定したデータベースとデータ・サーバへのコネクションに対してそのパラメータを設定するときに使用します。次に例を示します。

- すべてのデータベース・コネクションに対してロー・カウントの検証を無効にする。

```
configure replication server
set dsi_row_count_validation to 'off'
```

**configure replication server** を **dsi\_row\_count\_validation** を指定して実行した後、Replication Server へのすべてのデータベース・コネクションをサスペンドしてレジュームする必要があります。設定の変更はデータベース・コネクションをレジュームした後で有効になります。

- 特定のコネクションのロー・カウント検証を有効にする (次の例では SYDNEY\_DS データ・サーバの pubs2 データベースを指定)

```
alter connection to SYDNEY_DS.pubs2
set dsi_row_count_validation to 'on'
```

特定のコネクションに対して **dsi\_row\_count\_validation** を設定した場合は、データベース・コネクションをサスペンドしてレジュームする必要はありません。パラメータはただちに有効になります。ただし、新しい設定は、このコマンドを実行した後で Replication Server が処理する複写オブジェクトのバッチに影響します。設定の変更は Replication Server が現在処理している複写オブジェクトのバッチには影響しません。

『Replication Server 管理ガイド第 2 巻』の「エラーと例外の処理」の「データ・サーバのエラー処理」を参照してください。

## ロー・カウントの検証エラー・メッセージにテーブル名を表示する

Replication Server 15.6 では、ロー・カウントの検証エラー・メッセージにテーブル名が表示されます。

使用しているものに応じて次のようにします。

- 連続モードのログ順、ローごとの複写 — Replication Server はテーブル名、テーブル所有者名、およびそのトランザクション・エラーの元となった出力コマンドを特定する番号をログに記録して表示します。Replication Server はテーブル名の最初の 30 バイトしかログに記録しません。

**DSI\_CHECK\_ROW\_COUNT\_FULL\_NAME** トレースを有効にすると、表示されるテーブル名の最大の長さが 255 バイトになります。

- High Volume Adaptive Replication (HVAR) または Real-Time Loading (RTL) — Replication Server は HVAR と RTL のコンパイルの結果できる内部の **join-update** 文と **join-delete** 文をログに記録して表示します。HVAR または RTL が HVAR と RTL の処理の一部としてコマンドを既にコンパイルした後なので、トランザクション・エラーの原因になったコマンド自体を取得することはできません。表示できる **join-update** 文と **join-delete** 文の最大の長さは 128 バイトです。これには末尾の "...¥0" も含まれます。

この例は次のもので構成されます。

- プライマリ・サイト - pdb1 というプライマリ・データベース。3 カラム 3 ローからなる ThisTableHasANameLongerThan30Characters という名前のテーブルがあります。

| id | 名前     | 年齢 |
|----|--------|----|
| 1  | John   | 40 |
| 2  | Paul   | 38 |
| 3  | George | 37 |

- レプリケート・サイト - rdb1 というプライマリ・データベース。ローが 2 つあり、id カラムの値がそれぞれ 1 と 3 になっている ThisTableHasANameLongerThan30Characters というテーブルがあります。

次のコマンドをこの pdb1 に対して実行します。

```
update ThisTableHasANameLongerThan30Characters set age = 20
```

エラー・メッセージは複写モードによって異なります。

- 連続モードのログ順、ローごとの複写では次のようになります。
  - I. 2010/06/07 01:30:21. DSI received Replication Server error #5185 which is mapped to WARN by error action mapping.
  - W. 2010/06/07 01:30:21. WARNING #5185 DSI EXEC(103(1) ost\_replnx6\_61.rdb1) - /dsiexec.c(11941)
  - Row count mismatch for the command executed on 'ost\_replnx6\_61.rdb1'. The command impacted 0 rows but it should impact 1 rows.
  - I. 2010/06/07 01:30:21. The error was caused by output command #3 of the failed transaction on table 'dbo.ThisTableHasANameLongerThan30C'.

---

**注意：** テーブル名はデフォルトの 30 バイトで切り落とされています。

---

エラー・メッセージが表示できるテーブル名の最大の長さを 255 バイトにするために **DSI\_CHECK\_ROW\_COUNT\_FULL\_NAME** 追跡を on にした場合、エラー・メッセージの最後の行に完全なテーブル名が表示されます。

```
I. 2010/06/07 02:22:55. The error was caused by output
command #3 of the failed transaction on table
'dbo.ThisTableHasANameLongerThan30Characters'.
```

- HVAR または RTL の複写では次のようになります。
  - W. 2010/06/07 02:06:56. WARNING #5185 DSI EXEC(103(1) ost\_replnx6\_61.rdb1) - i/hqexec.c(4047)



```

Row count mismatch for the command executed on
'ost_replnx6_61.rdb1'. The command impacted 1 rows but it
should impact 2 rows.
I. 2010/06/07 02:06:56. (HQ Error): update
ThisTableHasANameLongerThan30Characters set age = w.age
from ThisTableHasANameLongerThan30Characters
t,#rs_uThisTab...
I. 2010/06/07 02:06:57. The DSI thread for database
'ost_replnx6_61.rdb1' is shutdown.

```

## シームレス・アップグレード

Replication Server バージョン 15.6 では、ルートのアップグレードに簡略化されがプロセスを提供されています。

簡略化されたルート・アップグレード・プロセスによって、Sybase Central™ の Replication Manager プラグインを使用する必要がなくなります。『Replication Server 設定ガイド』の「Replication Server のアップグレードまたはダウングレード」の「Replication Server のアップグレード」で、「ローカル・サイトの新しいバージョン・レベルへのコミット」の「Replication Server のルート・バージョン」の「ルートのアップグレード」を参照してください。

## Adaptive Server 複写機能サポートの強化

Replication Server 15.6 では Adaptive Server 複写機能のサポートが強化されました。

### インメモリ・データベースおよびリラックス持続性データベース

Replication Server 15.5 では、インメモリ・データベースおよびリラックス持続性データベースをレプリケート・データベースとして使用できます。Replication Server 15.6 では、インメモリ・データベースおよびリラックス持続性データベースをプライマリ・データベースとして使用できます。また、データベース再同期化とバルク・マテリアライゼーションを使用して、インメモリ・データベースおよびリラックス持続性データベースをリストアできます。

インメモリ・データベースはキャッシュ内のみ存在するので、それをサポートするホストが停止したりデータベース障害が発生したりすると、そのデータとデータベース・オブジェクトは保存されません。インメモリ・データベースまたはリラックス持続性データベースをリストアするには、次のいずれかを使用できます。

- データベース再同期化 — 「1つ以上のレプリケート・データベースをプライマリ・データベースから直接再同期する」の順に従ってください。データベースの dump と load を実行するためのディスク領域と時間が十分あることを確認

し、Replication Server がトランザクションをスキップする期間が許容範囲内であることを確認します。許容できる時間の長さは、アウトバウンド・キュー内のセグメントを **admin who, sqm** でモニタすることによって見積もることができます。『Replication Server リファレンス・マニュアル』の「Replication Server コマンド」の「**admin who**」を参照してください。

- バルク・マテリアライゼーション – 「バルク・マテリアライゼーションを使用してインメモリ・レプリケート・データベースまたはリラックス持続性データベースを再同期する」の手順に従ってください。

テンプレート・データベースまたはデータベース・ダンプを使用して初めて複写をセットアップする場合、またはインメモリ・データベースおよびリラックス持続性データベースをリストアする場合は、『Replication Server 管理ガイド第1巻』の「RepAgent の管理と Adaptive Server のサポート」の「インメモリ・データベースおよびリラックス持続性データベース」を参照してください。

### 参照：

- 1つ以上のレプリケート・データベースをプライマリ・データベースから直接再同期する (93 ページ)

### バルク・マテリアライゼーションを使用してインメモリ・レプリケート・データベースまたはリラックス持続性データベースを再同期する

2つのバルク・マテリアライゼーション・メソッドのどちらかを使用して、インメモリ・データベースまたはリラックス持続性データベースをリストアできます。

### 前提条件

バルク・マテリアライゼーションを開始する前に、複写定義とサブスクリプションが存在することを確認します。

### 手順

1. インバウンド・キューとアウトバウンド・キューをすばやく空にするには、インメモリ・データベースまたはリラックス持続性データベースがレプリケート・データベースになっているサブスクリプションのアクティブ化を解除します。

```
deactivate subscription subscription_name
for {table_repdef_name | func_repdef_name | {publication pub_name |
database replication definition db_repdef_name}
with primary at dataserver.database}
with replicate at dataserver.database
go
```

サブスクリプションのアクティブ化を解除した後は、Replication Server はインバウンド・キュー内のすべてのトランザクションをインメモリ・データベースまたはリラックス持続性データベースのアウトバウンド・キューに送信しません。

それに対して、サブスクリプションを削除すると、インバウンド・キューに書き込まれていたコミットされたすべてのトランザクションが Replication Server のダウストリームに分配されます。アクティブ化解除はプライマリ・サイトでのみ行われるので、サブスクリプションのアクティブ化解除は DSI が実行されていなくても行うことができます。アクティブ化解除のマーカがアウトバウンド・キューに到達すると、Replication Server ログにそのエントリが表示されます。

The deactivate marker for subscription *subscription\_name* arrives at outbound queue: *data\_server\_name.database\_name*.

アクティブ化解除のマーカがアウトバウンド・キューに到達した後、**sysadmin sqm\_purge\_queue** を使用してレプリケート・サイトのアウトバウンド・キューをパージし、アウトバウンド・キューを空にします。『Replication Server リファレンス・マニュアル』の「Replication Server コマンド」の「**sysadmin sqm\_purge\_queue**」を参照してください。

2. プライマリ Replication Server とレプリケート Replication Server の両方で **check subscription** を実行して、サブスクリプションのステータスがプライマリ Replication Server では DEFINED に、レプリケート Replication Server では VALID になっていることを確認します。
3. インメモリ・データベースまたはリラックス持続性データベースを作成する場合は、『Replication Server 管理ガイド 第1巻』の「サブスクリプションの管理」の「サブスクリプション・マテリアライゼーション・メソッド」の「バルク・マテリアライゼーション」に記載されている "アトミック・マテリアライゼーションのシミュレート" または "ノンアトミック・マテリアライゼーションのシミュレート" バルク・マテリアライゼーション・メソッドを使用します。次のいずれかの手順を実行します。
  - アトミック・マテリアライゼーションをシミュレートする — 手順 4～9 を実行する
  - 非アトミック・マテリアライゼーションをシミュレートする — 手順 4～13 を実行する

## image と Java データ型のバルク・コピー・イン

Replication Server 15.6 と Adaptive Server 15.0.3 ESD #1 では、Adaptive Server テーブル内の image と Java データ型のカラムをバルク・コピー・インを使って複写できます。これらのデータ型をレプリケート・データベースとウォーム・スタンバイ・データベースに複写するには、そのデータ型を複写定義、ファンクション複写定義、サブスクリプション内で指定します。

---

**注意：** image と Java データ型のバルク・コピー・インを使用するには、Adaptive Server 15.0.3 ESD #1 以降にアップグレードする必要があります。

---

## Replication Server 15.6 の新機能

バルク・コピー・インを設定するには、『Replication Server 管理ガイド第2巻』の「パフォーマンス・チューニング」の「DSI バルク・コピー・イン」を参照してください。

## Replication Server バージョン 15.5 の新機能

Replication Server 15.5 では、パフォーマンス、利便性、プロセス、データベース・サポートが強化されました。

### Replication Server 15.5 の製品エディションとライセンス

Replication Server 15.5 は、Enterprise Edition (EE) と Real-Time Loading Edition (RTLE) の 2 つの異なる製品エディションとしてリリースされました。これらは、異なるベースとオプション機能で構成されており、別々のライセンスが必要です。

**注意：**“Replication Server – Real-Time Loading Edition” を使用して Adaptive Server に複製することはできません。

表 8 : Replication Server 製品エディションの機能とライセンス

| エディション                    | 機能の種類 | 機能                        | 説明                                                                                               | ライセンス        |
|---------------------------|-------|---------------------------|--------------------------------------------------------------------------------------------------|--------------|
| Enterprise Edition        | ベース   | Replication Server        | Advanced Services Option、ExpressConnect for Oracle、Real-Time Loading 以外の Replication Server の機能。 | REP_SERVER   |
|                           | オプション | Advanced Services Option  | Replication Server のパフォーマンス強化機能。                                                                 | REP_HVAR_ASE |
|                           |       | ExpressConnect for Oracle | Replication Server を Oracle に直接接続できるようにする。Replication Server Options 15.5 のマニュアルを参照。             | REP_EC_ORA   |
| Real-Time Loading Edition | ベース   | Replication Server        | Advanced Services Option、ExpressConnect for Oracle、Real-Time Loading 以外の Replication Server の機能。 | REP_SERVER   |
|                           |       | Real-Time Loading (RTL)   | Sybase® IQ への複製を可能にする。                                                                           | REP_RTL_IQ   |

| エディション | 機能の種類 | 機能                       | 説明                               | ライセンス        |
|--------|-------|--------------------------|----------------------------------|--------------|
|        |       | Advanced Services Option | Replication Server のパフォーマンス強化機能。 | REP_HVAR_ASE |
|        | オプション | なし。                      |                                  |              |

Replication Server をインストールする前に、有効な SySAM ライセンスを取得します。Sybase Software Asset Management (SySAM) は Sybase 製品のライセンス管理とアセット管理を行います。詳細については、『Replication Server インストール・ガイド』の「作業を始める前に」の「インストール前の作業」で、「ライセンスの取得」を参照してください。

#### 参照：

- Real-Time Loading を使用した Sybase IQ の複写 (114 ページ)
- Replication Server – Advanced Services Option (118 ページ)

## Real-Time Loading を使用した Sybase IQ の複写

Real-Time Loading (RTL) を使用した Sybase IQ の複写によってパフォーマンスが向上します。これは Replication Server – Real-Time Loading Edition の機能として提供されています。

バージョン 15.5 より古い Replication Server では、連続複写モードでローごとにログの順序で、個々の複写処理がレプリケート・データベースに直接送信されます。

Replication Server バージョン 15.5 からは、Real-Time Loading を使って Adaptive Server から Sybase IQ に複写できます。同一のデータベース・スキーマを持つ Sybase IQ のレプリケート・データベースへ複写するとき、Replication Server は連続複写モードより高いパフォーマンスを達成します。RTL は次の処理を行うことによってデータを削減します。

- コンパイルテーブルごとに複写データを insert、update、delete オペレーション別に整理してまとめ、オペレーションを最終的なローのオペレーションにコンパイルする。
- バルク適用 – コンパイルされた最終的な結果に対して最も効率の良いバルク・インタフェースを使用して、最終的な結果をバルク適用する。Replication Server は、メモリ内の最終的な変更が保管されるデータベースを使って最終的なロー変更を保管し、それをレプリケート・データベースに適用する。

ログに記録されている個々のオペレーションを送信する代わりに、中間オペレーションをすべて削除し、複写されたトランザクションの最終的な状態のみを送信

します。アプリケーションによって異なりますが、一般に、これははるかに少ないデータが処理されることを意味します。

Replication Server が大量のトランザクションを組み合わせることでコンパイルし 1 つのグループにまとめるので、バルク・オペレーション処理が向上し、複製スループットとパフォーマンスも向上します。データ量を制御できます。バルク適用のためにグループ化されるデータ量は、グループ・サイズを調整することによって制御できます。

『Replication Server 異機種間複製ガイド』の「レプリケート・データ・サーバとしての Sybase IQ」を参照してください。

#### ライセンス

RTL を使用した Sybase IQ へのレプリケーションは、Real-Time Loading Edition 製品エディションで実行可能です。

#### データベースとプラットフォームのサポート

Sybase IQ 12.7 ESD #3 以降への複製では RTL を使用できます。64 ビットのハードウェア・プラットフォームを使用すると、最適なパフォーマンスを得ることができます。

Replication Server 15.5 は、Adaptive Server バージョン 15.0.3 またはバージョン 15.5 以降のプライマリ・データベースからのみ Sybase IQ への複製をサポートします。

#### 参照：

- 64 ビット・コンピューティング・プラットフォームのサポート (140 ページ)
- Replication Server 15.5 の製品エディションとライセンス (113 ページ)

## 異機種間複製サポートの機能強化

---

Replication Server 15.5 では、異機種データベースのサポートが拡張されています。

### 異機種環境での並列 DSI サポート

異機種環境では、並列データ・サーバ・インタフェース (DSI: Data Server Interface) スレッドの使用によって、レプリケート・データ・サーバへトランザクションが適用されるように Replication Server を設定できます。トランザクションを並列で適用すると、複製速度を上げて、なおかつプライマリ・サイトで適用されるトランザクションの順序を維持できます。

表 9 : Replication Server の ASE 以外のデータベースに対する並列 DSI サポート

| データベース               | 内部コミット制御メソッド | 外部コミット制御メソッド |
|----------------------|--------------|--------------|
| Oracle               | Yes          | No           |
| Microsoft SQL Server | Yes          | Yes          |
| IBM DB2 UDB          | Yes          | Yes          |

ASE 以外のデータベースに対する並列 DSI の使用の詳細については、『Replication Server 異機種間複製ガイド』を参照してください。

### 新しい逐次化メソッド

**wait\_after\_commit** はトランザクションの逐次化メソッドです。これによって並列 DSI での異機種間複製におけるデータの整合性とパフォーマンスが向上します。

**wait\_after\_commit** では、各スレッドの最初のバッチは、その前のスレッドが完全にコミットされてから開始されます。Oracle データベースなど、多版型同時実行制御 (MVCC: multiversion concurrency control) またはオプティミスティック同時制御 (OCC: optimistic concurrency control) を使用するデータベースでは、**wait\_after\_commit** 逐次化メソッドを使用することをおすすめします。それ以外の場合は、**wait\_for\_commit** をデフォルト・メソッドとして使用できます。

### 新規および更新された設定パラメータ

Replication Server 15.5 では、並列データ・サーバ・インタフェース (DSI) プロセスをサポートするために、**dsi\_max\_cmds\_in\_batch** が追加され、**dsi\_max\_xacts\_in\_group** と **dsi\_serialization\_method** が更新されています。

表 10 : 新しい設定パラメータ

| パラメータ                        | 値       | デフォルト値 | 説明                                                                                                       |
|------------------------------|---------|--------|----------------------------------------------------------------------------------------------------------|
| <b>dsi_max_cmds_in_batch</b> | Integer | 100    | 出力コマンドのバッチ処理の対象にできるソース・コマンドの最大数を定義します。<br><br>パラメータの変更を反映させるには、接続をサスペンドして再開する必要があります。<br><br>範囲：1 - 1000 |



表 11 : 更新された設定パラメータ

| パラメータ                           | 値                                                                                             | デフォルト値                 | 説明                                                                                        |
|---------------------------------|-----------------------------------------------------------------------------------------------|------------------------|-------------------------------------------------------------------------------------------|
| <b>dsi_max_xacts_in_group</b>   | Integer                                                                                       | 20                     | グループ化できるトランザクションの最大数を指定する。大きい値を指定するほど、レプリケート・データベースでのデータ遅延時間が短縮される。<br><br>範囲：1 - 1000    |
| <b>dsi_serialization_method</b> | <b>no_wait</b><br><b>wait_for_start</b><br><b>wait_for_commit</b><br><b>wait_after_commit</b> | <b>wait_for_commit</b> | トランザクションをレプリケート・データ・サーバに適用するときに、並列 DSI スレッド間で順序一貫性を保つために使用されるメソッドを指定する。どの場合でもコミット順は保持される。 |

**Adaptive Server 内部コミット制御のファンクション文字列の変更**

Replication Server は **rs\_dsi\_check\_thread\_lock** 関数を使用して、現在の DSI エグゼキュータ・スレッドが他のレプリケート・データベース・プロセスをブロックしているかどうかを調べます。Replication Server 15.5 では、デッドロックを検出するように **rs\_dsi\_check\_thread\_lock** ファンクション文字列が変更されました。

『Replication Server リファレンス・マニュアル』の「第 4 章 Replication Server システム・ファンクション」を参照してください。

**Oracle に対する異機種ウォーム・スタンバイ・サポート**

Replication Server 15.5 では、Oracle データベース用のウォーム・スタンバイ・データベース・アプリケーションを作成して維持できます。

Oracle データベース用のウォーム・スタンバイ・セットアップを作成するために Replication Server システム管理者が手動で行う必要のあるタスクがあります。詳細については、『Replication Server 異機種間複写ガイド』の「Oracle に対する異機種ウォーム・スタンバイ」を参照してください。

新しい設定パラメータ **ra\_standby** が、Oracle のウォーム・スタンバイをサポートするために追加されました。このパラメータは Replication Agent for Oracle がスタンバイ・モードで動作するかどうかを指定します。詳細については、『Replication Agent 15.5 Reference Manual』の「Chapter 2 Configuration Parameters」を参照してください。

Adaptive Server には、引き続き **rs\_init** ユーティリティを使ってウォーム・スタンバイ環境を設定できます。『Replication Server 管理ガイド 第 2 巻』を参照してください。

### 製品の互換性

次の表は、Oracle 用のウォーム・スタンバイ機能をサポートする追加の複製コンポーネントを示します。

表 12 : Oracle のウォーム・スタンバイ・サポートの製品互換性

| データベース・サーバのバージョン | Replication Agent のバージョン          | ECDA Option のバージョン | ExpressConnect のバージョン          |
|------------------|-----------------------------------|--------------------|--------------------------------|
| Oracle 10g、11g   | Replication Agent for Oracle 15.5 | ECDA 15.0 ESD #3   | ExpressConnect for Oracle 15.5 |

## Oracle レプリケート・データベースでのトリガ制御

トリガの制御は、PL/SQL コマンドが Oracle 10g または 11g のレプリケート・データベースに対して実行されるたびに、セッション・レベルまたは接続・レベルで行うことができます。レプリケート・データベースでのトリガ実行の制御によって、古いバージョンではレプリケート・データベースでのトリガ制御が存在しなかったために発生した重複と誤りを排除できます。

**RS\_TRIGGER\_CONTROL** パッケージではこの機能がサポートされています。このパッケージは、接続プロファイルを通して Oracle レプリケート・データベースへの接続が作成されたときに自動的にインストールされます。

**rs\_triggers\_reset** システム関数もこの機能をサポートするように変更されたので、**dsi\_keep\_triggers** コネクション・パラメータを off に設定して Oracle でのトリガを無効にできるようになりました。トリガ・アクションの開始時にトリガ制御文を追加して、レプリケート・データベースで制御する必要のある各トリガを作成し直してください。

詳細については、『Replication Server 異機種間複製ガイド』の「レプリケート・データ・サーバとしての Oracle」の「トリガの起動の設定」を参照してください。

## パフォーマンスの強化

Replication Server 15.5 では、複数のパフォーマンス強化機能が組み込まれました。

### Replication Server – Advanced Services Option

Replication Server – Advanced Services Option は、別途ライセンスする Replication Server の製品オプションです。これには、Replication Server のパフォーマンス強化機能が組み込まれています。

このオプションは、

- Replication Server – Enterprise Edition のオプションとして、別途 REP\_HVAR\_ASE ライセンスで提供されます。  
Replication Server – Enterprise Edition をご使用の場合は、REP\_HVAR\_ASE ライセンス・ファイルを Sybase 製品ダウンロード・センタ (SPDC) からダウンロードして Advanced Services Option の強化機能をアクティブにします。
- Replication Server – Real-Time Loading Edition にバンドルされます。  
Replication Server – Real-Time Loading Edition をアクティブにするには、Replication Server – Real-Time Loading Edition の製品エディション・ライセンスを SPDC からダウンロードします。

#### 参照：

- Replication Server 15.5 の製品エディションとライセンス (113 ページ)

### **High Volume Adaptive Replication**

ローごとの個々の複写オペレーションをログの順序でレプリケート・データベースに直接送信する連続複写モードに比べて、High-Volume Adaptive Replication (HVAR) では、データの削減をもたらすコンパイルおよびバルク適用プロセスの使用によって、より良いパフォーマンスを得ることができます。

- コンパイル – テーブルごとに複写データを insert、update、delete オペレーション別に整理してまとめ、オペレーションを最終的なローのオペレーションにコンパイルする。
- バルク適用 – コンパイルされた最終的な結果に対して最も効率の良いバルク・インタフェースを使用して、最終的な結果をバルク適用する。Replication Server は、メモリ内の最終的な変更が保管されるデータベースを使って最終的なロー変更を保管し、それをレプリケート・データベースに適用する。

ログに記録されている個々のオペレーションを送信する代わりに、中間オペレーションをすべて削除し、複写されたトランザクションの最終的な状態のみを送信します。アプリケーションによって異なりますが、一般に、これははるかに少ないデータが処理されることを意味します。

Replication Server が大量のトランザクションを組み合わせることでコンパイルし 1 つのグループにまとめるので、バルク・オペレーション処理が向上し、複写スループットとパフォーマンスも向上します。バルク適用のためにグループ化されるデータ量は、グループ・サイズを調整することによって制御できます。

HVAR は、プライマリ・データベースと同じスキーマを持つレプリケート・データベースのあるシステムのアーカイブとレポートを行うオンライン・トランザクション処理 (OLTP: creating online transaction processing) の作成に特に役立ちます。

詳細については、『Replication Server 管理ガイド第 2 巻』の「パフォーマンス・チューニング」の「Adaptive Server への High-Volume Adaptive Replication」を参照してください。

### システム・テーブルのサポート

Replication Server は `rs_tbconfig` テーブルをテーブル・レベルの設定パラメータの保管に使用し、`rs_columns` テーブルの `ref_objowner` と `ref_objname` カラムを参照制約のサポートに使用します。

テーブルの詳細については、『Replication Server リファレンス・マニュアル』の「Replication Server システム・テーブル」を参照してください。

### データベースとプラットフォームのサポート

HVAR では Adaptive Server 12.5 以降への複写がサポートされています。64 ビットのハードウェア・プラットフォームを使用すると、最適なパフォーマンスを得ることができます。

### 参照：

- 64 ビット・コンピューティング・プラットフォームのサポート (140 ページ)

### DSI 効率の向上

データ・サーバ・インタフェース (DSI) 効率の向上機能では、データ複写の遅延時間を短縮することによってパフォーマンスを向上させます。これにより、Replication Server が `ct_results` ルーチンを通してレプリケート・データ・サーバからの結果を待つ時間が短縮され、その結果データ・サーバが Replication Server を待つ時間が短縮されます。

詳細については、『Replication Server 管理ガイド第 2 巻』の「パフォーマンス・チューニング」の「DSI 効率の向上」を参照してください。

### RepAgent エグゼキュータ・スレッドの効率の向上

RepAgent エグゼキュータ・スレッドの効率の向上機能では、NRM スレッドを使用してログ転送言語 (LTL: Log Transfer Language) コマンドを正規化してバックし、それと並行して RepAgent エグゼキュータ・スレッドによる解析を行うことによって、パフォーマンスを向上させます。

NRM スレッドとの並列処理によって RepAgent エグゼキュータ・スレッドは応答時間を短縮します。NRM スレッドは RepAgent エグゼキュータ・スレッドから分離したスレッドです。

NRM スレッドを有効にして RepAgent エグゼキュータ・スレッドが使用できるメモリを指定するには、『Replication Server 管理ガイド第 2 巻』の「パフォーマンス・チューニング」の「RepAgent エグゼキュータ・スレッドの効率の向上」を参照してください。

### ディストリビュータ・スレッドの読み込み効率の向上

Replication Server 15.5 で、ディストリビュータ (DIST) スレッドはステーブル・キュー・トランザクション (SQT) スレッド・キャッシュから SQL 文を直接読み取ります。これにより、SQT からの負荷、および SQT と DIST の間の依存性が減少し、SQT と DIST の両方の効率が向上します。

詳細については、『Replication Server 管理ガイド第 2 巻』の「パフォーマンス・チューニング」の「ディストリビュータ・スレッドの読み込み効率の向上」を参照してください。

### メモリ割り付けの強化

強化されたメモリ割り付けでは、メモリ割り付けを大きな単位で行うことによってメモリ割り付け回数が減少し、パフォーマンスが向上します。

詳細については、『Replication Server 管理ガイド第 2 巻』の「パフォーマンス・チューニング」で「メモリ割り付けの強化」を参照してください。

### キュー・ブロック・サイズの増加

キュー・ブロック・サイズが増加され、1つのブロックでより多くのトランザクションを処理できるようになりました。

キュー・ブロック・サイズはステーブル・キュー構造で 사용되는連続メモリ・ブロックのバイト数です。古いバージョンの Replication Server では、キュー・ブロック・サイズは 16KB に固定されていました。

---

**注意：** キュー・ブロック・サイズを変更する前に、受信データ・フローとルートをサスペンドし、Replication Server をクワイース状態にする必要があります。ブロック・サイズを設定するコマンドを実行した後、Replication Server は自動的に停止します。新しいパラメータを有効にするために、Replication Server を再起動してください。

---

推奨事項、前提条件、キュー・ブロック・サイズの設定手順については、『Replication Server 管理ガイド第 2 巻』の「パフォーマンス・チューニング」の「キュー・ブロック・サイズの増加」を参照してください。

## 動的 SQL の機能強化

Replication Server 15.5 では動的 SQL の機能が強化されました。

Replication Server の動的 SQL により、複製パフォーマンスが強化されます。これは、Replication Server データ・サーバ・インタフェース (DSI) モジュールを使用して、レプリケート・データベースで動的 SQL 文を作成し、繰り返し実行できるようにすることで実現されます。詳細については、『Replication Server 管理ガイド第

2 巻』の「Chapter 4 Performance Tuning」の「動的 SQL で強化された Replication Server のパフォーマンス」を参照してください。

### 最適化された動的 SQL 文の実行

Replication Server 15.5 では動的 SQL 文の実行が強化されました。

バージョン 15.5 以前の Replication Server では、DSI が動的 SQL 文を実行するたびに言語コマンド、準備文、実行文が生成されますが、動的 SQL コマンドが失敗したときには言語コマンドのみが使用されます。

バージョン 15.5 では、動的 SQL 文が次の方法で最適化されます。

- 動的 SQL コマンドが失敗したときにのみ言語コマンドを生成する。
- 準備文の生成は準備文が初めて使用されたときに一度だけ行う。

### replicate minimal columns 句を動的 SQL で使用して最低限必要なカラムのみを複写する。

Replication Server 15.5 では、複写処理が動的 SQL をスキップすることはありません。**replicate minimal columns** 句が有効になっていて Replication Server が **replicate minimal columns** と動的 SQL を同時に効率よく使用する場合でもスキップしません。

バージョン 15.5 以前の Replication Server では、**replicate minimal columns** 句が複写定義で使用されると、変更されていないカラムは DSI では使用できず、動的 SQL はスキップされます。

### replicate\_minimal\_columns の拡張

Replication Server バージョン 15.5 では、**replicate\_minimal\_columns** パラメータが常にコネクッションにまで拡張されるので、データ・サーバ・インタフェース (DSI) では複写定義がないときまたは複写定義に **replicate minimal columns** 句がないときに最小カラムを使用するかどうかをこのパラメータを使用して決定します。

Replication Server のバージョン 15.5 以前では、**replicate\_minimal\_columns** はウォーム・スタンバイのときにしか使用できませんでした。

デフォルトでは、**replicate\_minimal\_columns** はすべてのコネクッションに対してオンです。コネクッションの **replicate\_minimal\_columns** 設定は、**replicate all columns** 句で設定された複写定義よりも優先されます。

あるコネクッションに対して **replicate\_minimal\_columns** を on に設定すると、現在の複写環境の動作が変わる可能性があります。レプリケートに送信されるコマンドに依存するレプリケート・コネクッションおよびトリガ・プロセスのカスタム・ファンクション文字列は、値が変更されなくても、影響を受ける可能性があります。元の動作をリストアするには、そのコネクッションの **replicate\_minimal\_columns** を off に設定します。

たとえば、`replicate_minimal_columns` を SYDNEY\_DS データサーバの `pubs2` データベースに対するコネクションで有効にするには、次のように入力します。

```
alter connection to SYDNEY_DS.pubs2
set replicate_minimal_columns to 'on'
```

`admin config` を使用して、`replicate_minimal_columns` 設定を表示できます。

---

**注意：** `dsi_compile_enable` を on に設定すると、Replication Server は `replicate_minimal_columns` の設定を無視します。

---

## ファンクション文字列の効率の向上

Replication Server 15.5 では、ファンクション文字列処理コマンド、ストアド・プロシージャ、システム・テーブルの機能が強化され、レプリケート・データベースに適用される必要のないファンクション文字列を指定できるようになりました。

バージョン 15.5 より古い Replication Server では、すべてのファンクション文字列がすべてのレプリケート・データベースで実行されますが、これらのファンクション文字列 (出力コマンドを含まないファンクション文字列など) の多くは ASE 以外のデータベースには適用されません。これらのファンクション文字列が実行されないようにすることによって、処理オーバーヘッドが削減され、複写環境が単純化されます。

### ファンクション文字列処理コマンドの変更

Replication Server 15.5 では、`none` パラメータのスコープがすべての関数に適用できるように拡張されており、Replication Server がレプリケート・データベースでの実行を省くことができる関数の指定を柔軟に行うことができます。

15.5 より古いバージョンでは、`alter function string` コマンドと `create function string` コマンドの `none` パラメータは、Replication Server が `text`、`unitext`、または `image` カラムの値を複製しないように、`rs_writetext` 関数にのみ適用されます。

`none` パラメータは、出力コマンドを持たないクラス・レベルとテーブル・レベルのファンクション文字列を指定するために使用します。Replication Server はそれらのファンクション文字列をレプリケート・データベースで実行しません。

詳細については、『Replication Server 管理ガイド第2巻』の「データベース・オペレーションのカスタマイズ」の「出力テンプレート」を参照してください。

`alter function string` コマンドと `create function string` コマンドの構文に変更はありません。『Replication Server リファレンス・マニュアル』の「Replication Server コマンド」を参照してください。

### ストアド・プロシージャの変更

Replication Server 15.5 では、ファンクション文字列処理の機能強化をサポートするために、**rs\_helpfstring** および **rs\_helpclassstring** ストアド・プロシージャが拡張されています。

- **rs\_helpfstring** — テーブル・レベルのファンクション文字列 (出力コマンドを持たないものも含む) を表示する。
- **rs\_helpclassstring** — クラス・レベルのファンクション文字列 (出力コマンドを持たないものも含む) を表示する。

『Replication Server リファレンス・マニュアル』の「Adaptive Server ストアド・プロシージャ」を参照してください。

### rs\_funcstrings System テーブルの変更

Replication Server 15.5 では、ファンクション文字列の機能強化をサポートするために、0x08 ビットが **rs\_funcstrings** テーブルの *attributes* カラムに追加されています。

詳細については、『Replication Server リファレンス・マニュアル』の「Replication Server システム・テーブル」の「rs\_funcstrings」を参照してください。

## 利便性とプロセスの強化

---

Replication Server 15.5 では、利便性とプロセスに関する機能強化が複数組み込まれています。

### 複写定義の変更要求プロセスの強化

Replication Server 15.5 では、複写定義に対する変更要求が強化され、複写定義の変更とデータ複写の伝達が自動的に調整されるようになりました。

これらの強化によって、プライマリ・データベースのダウン時間がなくなり、レプリケート・データベースのダウン時間もほとんどなくなって、データベース・スキーマの変更と複写定義の変更の調整がより便利になりました。複写定義の変更要求を発行すると、Replication Server は要求された変更の種類に基づいて、新しいバージョンの複写定義を作成する必要があるかどうかを決定します。

Replication Server が新しいバージョンの複写定義を作成する場合、複写定義が変更される前に自動的に古いバージョンの複写定義を使ったプライマリの更新が行われ、複写定義の変更後には新しいバージョンの複写定義を使ったプライマリの更新が行われます。

これらの機能強化がない場合にスキーマの変更と複写定義の変更を調整するには、プライマリの更新をクワイースタートにし、プライマリ・テーブルまたはストアド・プロシージャに関連付けられているすべてのデータが複写システム全体で処理されるのを待って Replication Agent を停止する、プライマリ・スキーマを変更する、



複写定義を変更する、カスタム・ファンクション文字列があれば変更する、変更が複写されるのを待つ、複写スキーマを変更する、Replication Agent を再起動してプライマリの更新をレジュームするという手順を踏む必要があります。

強化された複写定義の変更要求プロセスのコマンド、プロシージャ、ユーザ・シナリオについては、『Replication Server 管理ガイド第1巻』の「複写テーブルの管理」の「複写定義の変更要求プロセス」を参照してください。

### 製品の互換性

Adaptive Server のプライマリ・データベース、および Replication Server がサポートするすべてのバージョンの Microsoft SQL Server と Oracle のプライマリ・データベースでは、複写定義を変更できます。サポートされているバージョンについては、Replication Server Options のマニュアルを参照してください。

### 混合バージョンのサポート

**drop column name** 句を指定して **alter replication definition** を実行すると、サイト・バージョンが 1550 より古いレプリケート・サイトからのサブスクリプションを受け取る複写定義がある場合、プライマリ Replication Server は **alter replication definition** コマンドを拒否します。

**with DSI\_suspended** パラメータを指定して複写定義の変更要求を発行しても、サイトバージョンが 1550 より古いレプリケート DSI はサスペンドされません。

### 複写定義関連の強化項目リスト

複写定義関連の強化によって、データベース・スキーマを変更しながら、プライマリ・データベースで直接複写定義の変更を要求できます。それには、**alter replication definition**、**alter applied replication definition**、または **alter request function replication definition** コマンドを使用します。

次のことができます。

- プライマリ・データベースから直接複写定義コマンドを発行する。
- ターゲット・データベースの古いバージョンの複写定義で、Replication Server がすべてのデータを適用してから、Replication Server がターゲット DSI をサスペンドするように指示する **alter replication definition** コマンドを使用する。これにより、新しいバージョンの複写定義が到着する前にターゲット・スキーマの変更とカスタム・ファンクション文字列の変更を行うためのウィンドウが表示される。
- Replication Server が複写定義の変更要求を実行できることを確認するために、データの変更なしで変更要求を実行する。
- 複写定義からカラムを削除するために **alter replication definition** を実行する。
- Replication Agent から送信された複写定義要求で失敗したものを Replication Server に省略させる。プライマリ Replication Server で複写定義コマンドが失敗すると、Replication Agent が停止する。Replication Server がそのコマンドを省略

しない限り、Replication Agent を再起動すると、失敗したコマンドが再び実行される。

### システム・テーブルの変更

強化された複写定義の変更プロセスをサポートするために、Replication Server では *rs\_columns*、*rs\_locator*、*rs\_objects* システム・テーブルが変更されています。

| システム・テーブル         | 変更の説明                                                                  |
|-------------------|------------------------------------------------------------------------|
| <i>rs_columns</i> | <i>version</i> カラムが追加された。                                              |
| <i>rs_locator</i> | <i>C</i> 、 <i>F</i> 、および <i>S</i> の値が <i>type</i> カラムに追加された。           |
| <i>rs_objects</i> | <i>active_inbound</i> 、 <i>attributes2</i> 、 <i>version</i> カラムが追加された。 |

詳細については、『Replication Server リファレンス・マニュアル』の「Replication Server システム・テーブル」を参照してください。

### 複写タスクのスケジューリング

Replication Server 15.5 では複写タスクのスケジュールを設定できます。

たとえば、レプリケート・データベースがプライマリ・データベースからデータを受信していないときに、レプリケート・データベースの特定のステータスに関するレポートを作成できます。複写が夜中の指定された期間にのみ行われるようにスケジュールを設定することによって、次の日の処理でレプリケート・データベースが変更されないようにし、前の日のデータに対してレポートが行われるようにすることができます。これは、レプリケート・データベースへの接続が1日の特定の時刻にサスペンドおよびレジュームされるようにスケジュールを設定することによって行うことができます。作成されたスケジュールは、*rs\_schedule* および *rs\_scheduledtxt* システム・テーブルに保存されます。

詳細については、『Replication Server 管理ガイド第1巻』の「複写スケジュールの管理」の「複写タスクのスケジュール」、および『Replication Server リファレンス・マニュアル』の「Replication Server システム・テーブル」を参照してください。

### 複写の遅延

Replication Server 15.5 では複写を一定時間遅らせることができます。

プライマリ・データベースにコミットされた人為的なエラーから回復できるように、プライマリ・データベースからの更新を一定時間遅らせることによって、レプリケート・データベースをフェールバック・システムとして使用できます。

詳細については、『Replication Server 管理ガイド第1巻』の「複写スケジュールの管理」の「複写の遅延」を参照してください。

## レプリケート・データベースの再同期

データベース再同期化を使用すると、データや整合性を損なうことなく、またプライマリ・データベースのクワイズを強制しないで、レプリケート・データベースをマテリアライズして複写をレジュームできます。

### Replication Agent のサポート

再同期マーカの自動生成などのデータベース再同期化の全機能を使用するには、Replication Agent のサポートが必要です。Replication Agent 15.5 for Oracle では、データベース再同期化の全機能がサポートされています。詳細については、『Replication Server 異機種間複写ガイド』の「Oracle レプリケート・データベースの再同期」と Replication Agent のマニュアルを参照してください。

RepAgent (Adaptive Server の複写エージェント) では、Adaptive Server 15.5 より後のバージョンでデータベース再同期化の全機能がサポートされる予定です。RepAgent からのサポートなしで Adaptive Server データベースを再同期化するには、『Replication Server 管理ガイド 第2巻』の「第7章 複写システム・リカバリ」で「Adaptive Server のレプリケート・データベースの再同期」を参照してください。

### 製品の互換性

次の表は、Oracle データベースの再同期化をサポートする Oracle、Replication Agent for Oracle、ECDA Option for Oracle、および ExpressConnect for Oracle のバージョンを示します。Replication Server Options 15.5 では ExpressConnect for Oracle が ECDA Option for Oracle の代わりに使用されます。

詳細については、Replication Server Options 製品のマニュアルと『Replication Server 異機種間複写ガイド』を参照してください。

表 13 : Oracle データベースの再同期の製品互換性

| データベース・サーバのバージョン | Replication Agent のバージョン          | ECDA Option のバージョン | ExpressConnect のバージョン          |
|------------------|-----------------------------------|--------------------|--------------------------------|
| Oracle 10g、11g   | Replication Agent for Oracle 15.5 | ECDA 15.0 ESD #3   | ExpressConnect for Oracle 15.5 |

### システム・テーブルのサポート

*rs\_databases* テーブルでは、*dist\_status* カラムと *src\_status* カラムのデータ型が、*tinyint* から *cs\_int* に変更されました。また、“0x100 - waiting for a resync marker” ステータスが *dist\_status* に追加されました。

### レプリケート・データベースを再同期する

信頼されたソースからデータ・ダンプを取得して、そのダンプを再同期するターゲット・データベースに適用します。

1. Replication Agent をサスペンドして複製プロセスを停止します。
2. Replication Server を再同期モードにします。  
再同期モードになると、Replication Server はトランザクションをスキップします。さらに、プライマリ・データベースまたは信頼されたソースから取得したダンプを使ってレプリケート・データベースにデータを再移植する準備として、複製キューから複製データをパージします。
3. Replication Agent を再開して、データベース再同期マーカを Replication Server に送り、再同期処理が進行中であることを示します。
4. プライマリ・データベースからダンプを取得します。  
Replication Server はプライマリ・データベース・ダンプが完了したことを示すダンプ・マーカを検出すると、トランザクションのスキップを停止し、どのトランザクションをレプリケート・データベースに適用するかを判定できるようになります。
5. ダンプをレプリケート・データベースに適用します。
6. 複製をレジュームします。

### ロー・カウント検証での変更

Replication Server 15.5 では、エラー番号 5185 と 5187 のエラーに対するデフォルト・エラー・アクションが、"warn" から "stop replication" に変更され、Replication Server エラー・クラスに 5203 が追加されました。

表 14 : 新しいエラーと変更されたエラー

| server_error | エラー・メッセージ                                                                                                                       | デフォルトのエラー・アクション  | 説明                                                                                                                |
|--------------|---------------------------------------------------------------------------------------------------------------------------------|------------------|-------------------------------------------------------------------------------------------------------------------|
| 5185         | Row count mismatch for the command executed on 'dataserver.database'. The command impacted x rows, but it should impact y rows. | stop_replication | このメッセージは、SQL 文の複製、ストアド・プロシージャ、またはオートコレクションが有効になっているロー変更の一部ではないコマンドがデータ・サーバに送られた後、影響を受けたロー数が予想されたロー数とは異なる場合に表示される。 |

| server_error | エラー・メッセージ                                                                                                                                            | デフォルトのエラー・アクション  | 説明                                                                                  |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|-------------------------------------------------------------------------------------|
| 5187         | Row count mismatch for the autocorrection delete command executed on 'dataserver.database'. The command deleted x rows, but it should delete y rows. | stop_replication | このメッセージは、オートコレクションが有効な場合、delete コマンドがデータ・サーバに送られた後、影響を受けたロー数が予期されたロー数とは異なる場合に表示される。 |
| 5203         | Row count mismatch on 'dataserver.database'. The delete command generated by dsi_command_convert deleted x rows, whereas it should delete y rows.    | stop_replication | このメッセージは、削除されたロー数が予期された削除ロー数とは異なる場合に表示される。                                          |

Replication Server エラー・クラスにデフォルトのエラー・アクションをオーバーライドさせる場合に、**assign action** コマンドをプライマリ・サイトで使用します。

詳細については、『Replication Server 管理ガイド第2巻』の「エラーと例外の処理」を参照してください。

コマンド、パラメータ、ストアド・プロシージャ、システム・テーブルの詳細については、『Replication Server リファレンス・マニュアル』を参照してください。

#### 参照：

- 非 SQL 文の複写ロー・カウントの検証 (161 ページ)

## alter error class の機能強化

**alter error class** は Replication Server エラー・クラスの変更に使用できます。

Replication Server エラー・クラスを指定するには **replication server** パラメータを **alter error class** コマンドの構文で使用します。詳細については、『Replication Server リファレンス・マニュアル』、および『Replication Server 管理ガイド第2巻』の「エラーと例外の処理」を参照してください。

## リファレンス複写環境の実装に使用するツールセット

Replication Server 15.5 には、その環境内で使用可能な製品を使って、Adaptive Server から Adaptive Server への、および Oracle から Oracle への複写のリファレンス実装をすばやくセットアップするためのツールセットが含まれています。リ

ファレンス複製環境では、パフォーマンスに関する問題の特定に使用する統計が収集でき、Replication Server の機能を試すことができます。

ツールセットを使用すると、次の手順を実行できます。

1. Replication Server、プライマリ・データベース、レプリケート・データベースを構築します。
2. 複製環境を設定します。
3. データベース・レベルの複製によってプライマリ・データベースとレプリケート・データベースに対する単純なトランザクションを実行します。
4. 手順 3 の複製処理から統計とモニタ・カウンタを収集します。
5. リファレンス複製環境をクリーンアップします。

リファレンス複製環境の構築、設定、使用の詳細については、『Replication Server 管理ガイド第 2 巻』の「リファレンス複製環境の実装」を参照してください。

---

**注意：**リファレンス実装は、単一の Replication Server、プライマリ・データベース・サーバ、レプリケート・データベース・サーバを含む複製環境を構築します。複数の複製システム・コンポーネント用のリファレンス環境トポロジは設定できません。

---

### プラットフォームのサポート

リファレンス実装は Replication Server 15.5 がサポートするすべてのプラットフォームで使用できます。ただし、リファレンス環境を Replication Server がサポートする Microsoft Windows プラットフォームでセットアップするには、Cygwin を使ってリファレンス実装スクリプトを実行する必要があります。 <http://www.cygwin.com/> を参照してください。

### Adaptive Server に必要なコンポーネント

Adaptive Server から Adaptive Server への複製のためのリファレンス実装環境には、サポートされているバージョンの Replication Server と Adaptive Server が必要です。

表 15 : Adaptive Server のリファレンス実装でサポートされている製品コンポーネントのバージョン

| Replication Server | Adaptive Server |
|--------------------|-----------------|
| 15.5               | 15.0.3, 15.5    |

**Oracle に必要なコンポーネント**

Oracle から Oracle への複製のためのリファレンス実装環境には、サポートされているバージョンの Replication Server、Oracle、Replication Agent for Oracle、ECDA Option for Oracle が必要です。

表 16 : Oracle のリファレンス実装でサポートされている製品コンポーネントのバージョン

| Replication Server | Oracle | Replication Agent for Oracle | ECDA Option for Oracle |
|--------------------|--------|------------------------------|------------------------|
| 15.5               | 10.2   | 15.2                         | 15.0 ESD #3            |

**admin who コマンドの機能強化**

Replication Server 15.5 では、すべてのスレッド・モジュールで **admin who** を実行するときに接続識別子を指定できます。

1つのスレッド・モジュールのすべてのコネクションに関する情報を表示する代わりに、スレッド・モジュールの接続識別子を指定することによって特定のコネクションに対する **admin who** の実行結果のみを表示できます。

- DIST — ディストリビュータ
- DSI — データ・サーバ・インタフェース
- RSI — Replication Server インタフェース
- SQM — ステابل・キュー・マネージャ
- SQT — ステابل・キュー・トランザクション

接続識別子を指定しても、Replication Server がそれに一致する情報を見つけることができない場合、出力にはレコードが 1 つも表示されません。

完全な構文と例については、『Replication Server リファレンス・マニュアル』の第 3 章で、「admin who」を参照してください。

---

**注意：** 接続識別子を使用すると **no\_trunc** オプションは使用できません。

---

**DIST と DSI のスレッド・モジュールの新しいカラム**

**admin who** は DIST と DSI のスレッド・モジュールに対して追加のカラムを表示します。

表 17 : DIST と DSI のスレッド・モジュールで追加されるカラム

| スレッド | カラム名                        | 説明                                                                                                                                 | 値                                                                                          |
|------|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------|
| DIST | <i>RSTicket</i>             | Replication Server の <b>stats_sampling</b> パラメータが on の場合に DIST スレッドによって処理された <b>rs_ticket</b> サブコマンド数。                             | 最小値 : 0<br>最大値 : $2^{63}-1$<br>デフォルト値は 0                                                   |
| DIST | <i>SqtMaxCache</i>          | データベース・コネクション用の最大 SQT キャッシュ・メモリ (バイト単位)。                                                                                           | デフォルト値の 0 は、 <b>sqt_max_cache_size</b> の現在の設定値が、コネクションの最大キャッシュ・サイズとして使用されることを示す。デフォルト値は 0。 |
| DSI  | <i>RSTicket</i>             | Replication Server の <b>stats_sampling</b> パラメータが on の場合に DSI キュー・マネージャによって処理された <b>rs_ticket</b> サブコマンド数。                         | 最小値 : 0<br>最大値 : $2^{63}-1$<br>デフォルト値は 0                                                   |
| DSI  | <i>dsi_rs_ticket_report</i> | ファンクション文字列 <b>rs_ticket_report</b> が呼び出されたかどうかを判断する。 <b>rs_ticket_report</b> は <b>dsi_rs_ticket_report</b> が on に設定されているときに呼び出される。 | on または off<br>デフォルト値は off                                                                  |

完全な構文と例については、『Replication Server リファレンス・マニュアル』の第 3 章で、「admin who」を参照してください。

**データベース世代番号のリセット**

複製システム内の各プライマリ・データベースには、データベース世代番号が格納されます。この番号は、プライマリ・データベースと、プライマリ・データベースを管理する Replicatin Server の RSSD に格納されます。

リカバリのためにプライマリ・データベースをロードする場合は、使用しているリカバリ手順の指示に従って、データベース世代番号を変更する必要があります。

データベース世代番号の最大値は 65,535 です。どうしても必要なとき以外は、高い番号にならないようにすることをおすすめします。Replication Server 15.5 以降で



は、最大値の 65,535 に達する前に、データベース世代番号を 0 にリセットできません。Replication Server 15.5 より古いバージョンでは、データベース世代番号をリセットしたら複製環境を再構築する必要があります。

詳細については、『Replication Server 管理ガイド第 2 巻』の「複製システム・リカバリ」の「データベース生成番号のリセット」を参照してください。

## インバウンド・キューに `rs_ticket` マーカを挿入する

Replication Server 15.5 では、データ複製のパフォーマンスに関する問題を特定するためのシステム・コマンドが追加されました。

`sysadmin issue_ticket` は `rs_ticket` マーカをインバウンド・キューに挿入して、プライマリ・データベース上の RepAgent がそのチケットを処理する必要をなくします。`rs_ticket` はその時点から、マーカが Replication Server 上のモジュールを通過するときに、システム時刻をマーカの末尾に追加します。このマーカが収集した情報は、複製データベース内の `rs_ticket_history` テーブルに格納されます。

『Replication Server リファレンス・マニュアル』の「Replication Server コマンド」を参照してください。

## デフォルト設定と予約語での変更

Replication Server 15.5 では、Replication Server の強化とパフォーマンスの向上のためにデフォルト設定と値の変更が行われました。

### パラメータのデフォルト値の変更

`memory_limit` と `smp_enable` を除いて、Replication Server 15.5 にアップグレードすると、古いバージョンでデフォルトを使用していた値のみが新しいデフォルト値に設定されます。

表 18 : パラメータのデフォルト値の変更

| パラメータ                                 | 古い値        | 新しい値                   | バージョン 15.5 からのダウングレード     |
|---------------------------------------|------------|------------------------|---------------------------|
| <code>exec_cmds_per_timeslice</code>  | 5          | 2,147,483,647          | ダウングレードでは設定されていた値は変更されない。 |
| <code>init_sqm_write_delay</code>     | 1,000 ミリ秒  | すべてのプラットフォームで 100 ミリ秒  | ダウングレードでは設定されていた値は変更されない。 |
| <code>init_sqm_write_max_delay</code> | 10,000 ミリ秒 | すべてのプラットフォームで 1000 ミリ秒 | ダウングレードでは設定されていた値は変更されない。 |

| パラメータ                                                                                                                            | 古い値                       | 新しい値                                                                                                                                                           | バージョン 15.5 からのダウングレード                                                                    |
|----------------------------------------------------------------------------------------------------------------------------------|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------|
| <b>memory_limit</b>                                                                                                              | 80MB                      | 2,047MB<br>以前の値が 2,047MB より小さかった場合、アップグレードによって新しいデフォルト値にまで引き上げられる。                                                                                             | 2,047MB より大きい値に設定されていた場合は、ダウングレードすると、オーバフローから保護するために 2,047MB にリセットされる。                   |
| <b>smp_enable</b>                                                                                                                | off                       | on<br>元の設定が off だった場合は、アップグレードによって on に変更されない。                                                                                                                 | ダウングレードでは設定されていた値は変更されない。                                                                |
| <b>sqt_max_cache_size</b>                                                                                                        | 1,048,576<br>バイト<br>(1MB) | <ul style="list-style-type: none"> <li>32 ビット・プラットフォームフォーム：<br/>1,048,576 バイト<br/>(1MB)</li> <li>64 ビット・プラットフォームフォーム：<br/>20,971,520 バイト<br/>(20MB)</li> </ul> | 2,147,483,647 バイトより大きい値に設定されていた場合は、ダウングレードすると、オーバフローから保護するために 2,147,483,647 バイトにリセットされる。 |
| <b>sts_full_cache_system_table_name</b><br>(次のシステム・テーブル用)<br><br><i>rs_columns,</i><br><i>rs_functions,</i><br><i>rs_objects</i> | off — 完全にキャッシュされない。       | on — 完全にキャッシュされる。                                                                                                                                              | ダウングレードによって設定は変更されない。                                                                    |

パラメータ、例、使用方法の情報については、『Replication Server リファレンス・マニュアル』の「Replication Server コマンド」を参照してください。

## RSSD ロック・スキーマでの変更

競合を減らしてパフォーマンスを向上するために、バージョン 15.5 をインストールまたはアップグレードすると、Replication Server システム・データベース (RSSD) のシステム・テーブルのデフォルト・ロック・スキーマはロー・レベルのロックになります。

バージョン 15.5 からダウングレードしてもロック・スキーマは変わりません。

Embedded Replication Server システム・データベース (ERSSD) のデフォルト・ロック・スキーマ (ロー・レベルのロック) には変更はありません。

## 予約語

バージョン 15.5 では Replication Server の予約キーワード・リストに追加がありました。

新しく追加された Replication Server キーワードの予約語については、『Replication Server リファレンス・マニュアル』の「トピック」の「予約語」を参考してください。

## Adaptive Server 複写機能サポートの強化

---

Replication Server 15.5 では Adaptive Server 複写機能のサポートが強化されました。

### bigdatetime と bigtime の複写

Replication Server 15.5 は Adaptive Server 15.5 の *bigdatetime* と *bigtime* の複写をサポートします。これらのデータ型をレプリケート・データベースとウォーム・スタンバイ・データベースに複写するには、そのデータ型を複写定義、ファンクション複写定義、サブスクリプション内で指定します。

*bigdatetime* と *bigtime* によって Adaptive Server はデータおよび時間データをマイクロ秒の精度で格納できます。*bigdatetime* は *TIMESTAMP* データ型に対応し、*bigtime* は Sybase IQ と Sybase SQL Anywhere の *TIME* データ型に対応します。

*bigdatetime* と *bigtime* の使用方法については、『Replication Server 管理ガイド第 1 巻』の「RepAgent の管理と Adaptive Server のサポート」の「*bigdatetime* および *bigtime* データ型のサポート」を参照してください。

#### 混合バージョン情報

*bigdatetime* と *bigtime* は Adaptive Server バージョン 15.5 以降でしかサポートされません。少なくともプライマリ・データ・サーバが Adaptive Server 15.5 以降であれば、次のように対処できます。

- プライマリとレプリケート Replication Server がバージョン 15.5 以降であり、レプリケート Adaptive Server がこれらのデータ型をサポートしない場合、その 2 つのデータ型をそれぞれ *varchar* データ型にマッピングする定義を複写定義に含めます。または、複写定義でその 2 つのデータ型を使用する代わりに *varchar* データ型を使用します。

- プライマリ Replication Server がバージョン 15.5 以降であり、レプリケート Replication Server と Adaptive Server がこれらのデータ型をサポートしない場合、複写定義でその2つのデータ型の代わりに *varchar* データ型を使用します。
- プライマリ Replication Server、レプリケート Replication Server、レプリケート Adaptive Server がこれらのデータ型をサポートしない場合、RepAgent は自動的に *varchar* データ型を Replication Server に送信します。

### 参照：

- *bigdatetime* と *bigtime* の複写の有効化 (143 ページ)

## 遅延名前解決

Replication Server 15.5 では、Adaptive Server 15.5 の遅延名前解決がサポートされます。遅延名前解決は、ストアド・プロシージャを作成するときに、それによって内部で使用されるオブジェクトを解決しないままでストアド・プロシージャを作成できるようにするための Adaptive Server の機能です。

オブジェクト解決のフェーズはそのストアド・プロシージャが Adaptive Server で初めて実行されるまで延期されます。ストアド・プロシージャは初回以降の実行では通常に実行されます。詳細については、『Adaptive Server Enterprise 15.5 新機能ガイド』の「ユーザ定義のストアド・プロシージャ用の遅延名前解決」を参照してください。

### *Replication Server の問題*

15.5 より古いバージョンの Replication Server では、サポートされているデータ定義言語 (DDL) コマンドをスタンバイ・データベースに複写できるように、ウォーム・スタンバイ・アプリケーションをセットアップして、アクティブ・データベースで *sp\_reptostandby* を有効にできます。

ただし、ウォーム・スタンバイではない環境にあるスタンバイ・データベースまたはレプリケート・データベースでは、Replication Server がテンポラリー・テーブルを複写しないので、テンポラリー・テーブルを参照するストアド・プロシージャを作成できません。ストアド・プロシージャを作成するプロセスは、そのストアド・プロシージャが内部で使用されるオブジェクトを解決する必要があります。ただし、レプリケート・データベースにもスタンバイ・データベースにもテンポラリー・テーブルがないので、Replication Server はレプリケート・データベースにもスタンバイ・データベースにもストアド・プロシージャを作成しません。

Replication Server 15.5 では遅延名前解決のサポートがあるので、テンポラリー・テーブル (存在しないテーブルと存在しないプロシージャ) を参照するストアド・プロシージャをレプリケート・データベースにもスタンバイ・データベースにも複写できます。

Replication Server で遅延名前解決を設定するには、『Replication Server 管理ガイド 第 1 巻』の「RepAgent の管理と Adaptive Server のサポート」の「遅延名前解決」を参照してください。

## SQL 文の複写スレッシュホールドの設定

Adaptive Server 15.0.3 ESD #1 以降では、データベース・レベルまたはセッション・レベルで SQL 文の複写をトリガするスレッシュホールドを設定できるので、個々のテーブルでスレッシュホールドを設定する必要はありません。Replication Server 15.5 はこの新しいスレッシュホールド設定をサポートします。

セッション・レベルで設定したスレッシュホールドは、テーブル・レベルとデータベース・レベルのスレッシュホールドよりも優先されます。テーブル・レベルで設定したスレッシュホールドは、データベース・レベルで設定したスレッシュホールドよりも優先されます。古いバージョンの Replication Server では、テーブル・レベルのスレッシュホールドしか設定できませんでした。

詳細については、『Replication Server 管理ガイド第 2 巻』の「パフォーマンス・チューニング」の「SQL 文の複写スレッシュホールドの設定」を参照してください。

## 増分データ転送

Adaptive Server 15.5 では、Adaptive Server から別の Adaptive Server にテーブル全体を転送する代わりに、テーブルのデータを小刻みに転送できます。Replication Server は Adaptive Server 15.5 の増分データ転送機能に関するデータ定義言語をサポートします。これにより、増分転送のマークが付いた複写テーブルで行われるデータ変更オペレーションのための複写は正常に処理されます。

**transfer table** コマンドを使って複写テーブルにデータをロードするときに、そのテーブルにユニーク・インデックス・コマンドがあり、増分転送のデータがそのテーブルに既に存在する場合、Adaptive Server は内部で **insert** コマンドを **update** コマンドに変換します。

**transfer table** コマンドは初回に転送を開始したデータ・サーバとデータベースにのみ適用されます。

ウォーム・スタンバイまたは Multi-Site Availability (MSA) 環境内のアクティブ・データベースでテーブルを増分転送用にマークしてから、アクティブ・データベースが終了してスタンバイ・データベースに切り替わると、増分データ転送がスタンバイ・データベースで正しくレジュームされない可能性があります。アクティブ・データベースと違って、スタンバイ・データベースには増分データ転送アクティビティの記録がありません。したがって、スタンバイ・データベースでも増分データ転送を初期化する必要があります。

詳細については、『Adaptive Server Enterprise Transact-SQL ユーザーズ・ガイド』の「第 8 章 データの追加、変更、転送、削除」を参照してください。

## インメモリ・データベースおよびリラックス持続性データベース

Adaptive Server 15.5 では、インメモリ・データベースとリラックス持続性データベースが導入されています。

インメモリ・データベースはすべてがキャッシュ内に存在し、データにもログにもディスク・ストレージは使用されません。したがって、ディスク I/O の必要もありません。これは従来のディスク常駐型データベースに比べて、より良いパフォーマンスを期待でき、その他にも利点があります。ただし、インメモリ・データベースはキャッシュ内のみ存在するので、それをサポートするホストが停止したりデータベース障害が発生したりすると、データベースのリカバリができません。

リラックス持続性によって、Adaptive Server はインメモリ・データベースから得られるパフォーマンスのメリットをディスク常駐型データベースに拡張します。ディスク常駐型データベースは、ディスクへの書き込みを実行することで、トランザクションの ACID (原子性、一貫性、整合性、持続性) プロパティが維持されるようにします。従来のディスク常駐型データベースは、サーバの障害からトランザクションを確実にリカバリできるように、完全な持続性で動作します。リラックス持続性データベースは、コミットされたトランザクションの完全な持続性と引き換えに、トランザクションの負荷に対する実行時のパフォーマンスを向上させます。**no\_recovery** レベルで作成されたリラックス持続性データベースは、サーバが終了したり停止されたりすると、データもログもリカバリできないという点で、インメモリ・データベースに似ています。リラックス持続性データベースは **at\_shutdown** レベルで作成することもできます。この場合、データベースが適切に停止されると、トランザクションがディスクに書き込まれます。

詳細については、『Adaptive Server Enterprise インメモリ・データベース・ユーザーズ・ガイド』を参照してください。

### Replication Server のサポート

Replication Server は、レプリケート・データベースとして、インメモリ・データベースと、持続性が **no\_recovery** に設定されているリラックス持続性データベースをサポートします。

プライマリ・データベースは、持続性が **full** の従来のディスク常駐型データベースであることが必要です。便宜上、このドキュメントでは、持続性を **non\_recovery** に設定したリラックス持続性データベースを「リラックス持続性データベース」と呼んでいます。

インメモリ・データベースとリラックス持続性データベースを新しいレプリケート・データベースとして初期化できます。それには、データ、オブジェクト・スキーマ、設定情報を次のいずれかから取得します。

- 基本情報の入ったテンプレート・データベース。

- 別のデータベースからのデータベース・ダンプ。ダンプをターゲットのインメモリ・データベースまたはリラックス持続性データベースにロードします。ダンプのソース・データベースには、別のインメモリ・データベース、リラックス持続性データベース、従来のディスク常駐型データベースのいずれも使用できます。

ホスト・データ・サーバがシャット・ダウンまたは再起動すると、インメモリ・データベースおよびリラックス持続性データベースのオブジェクト定義、データ、RepAgent 設定は失われます。インメモリ・データベースまたはリラックス持続性データベースをテンプレートまたはソース・データベースからのデータベース・ダンプで初期化する必要があります。

インメモリまたはリラックス持続性データベースをレプリケート・データベースとして設定するには、『Replication Server 管理ガイド第1巻』の「RepAgent の管理と Adaptive Server のサポート」の「インメモリ・データベースおよびリラックス持続性データベース」を参照してください。

### 最低限の DML ロギングと複写

Adaptive Server では、ディスクのトランザクション・ログに書き込まれるログ・レコードを最適化するために、**insert**、**update**、**delete**、**slow bcp** などの一部のデータ操作言語 (DML) のコマンドをすべて種類の低持続性データベース (持続性を **at\_shutdown** または **no\_recovery** に設定したインメモリ・データベースやリラックス持続性データベースなど) で実行する場合、最低限のロギングを行うか、ロギングを実行しないで済ますことができます。

DML の最低限のロギングは、データベース単位、テーブル単位、セッション単位で実行できます。『Adaptive Server Enterprise インメモリ・データベース・ユーザーズ・ガイド』の「最低限のログを取る DML」を参照してください。

---

**注意：**最低限の DML ロギングのセッションレベルの設定は、データベース・レベルの設定およびテーブル・レベルの設定よりも優先されます。

---

### *Replication Server のサポート*

複写では完全なロギングを使用するため、複写と Adaptive Server 15.5 の最低限のデータ操作言語 (DML) ロギング機能はデータベース・レベルまたはテーブル・レベルなど、同じレベルでは互換性がありません。ただし、最低限の DML ロギングと複写は異なるレベルに共存できるため、あるテーブルで最低限の DML ロギングを実行する一方、他のテーブルで複写を実行することによりパフォーマンスを向上することができます。複写と最低限の DML ロギングの間で互換性の問題が発生するシナリオについては、『Replication Server 管理ガイド第1巻』の「RepAgent の管理と Adaptive Server のサポート」の「最低限の DML ロギングと複写」を参照してください。

## 混合バージョン環境

---

複写システム・ドメインに Replication Server 15.5 以降がある場合は、複写システム・ドメインのシステム・バージョンとすべてのサイトおよびルート・バージョンが 12.6 以降でなければなりません。

バージョン 15.5 にアップグレードするには、その前に Replication Server をバージョン 12.6 以降にアップグレードし、サイト・バージョンを 12.6 以降に設定し、ルートを 12.6 以降にアップグレードする必要があります。

『Replication Server 設定ガイド』の「Replication Server のアップグレードまたはダウングレード」を参照してください。

## 新しくサポートされたオペレーティング・システム

---

Replication Server 15.5 では、次のオペレーティング・システムへのサポートが導入されています。

- Microsoft Windows Server 2008 R2
- Microsoft Windows 7
- SuSe Linux Enterprise Server SLES 11

## 64 ビット・コンピューティング・プラットフォームのサポート

---

Replication Server 15.5 は、大量の仮想メモリ容量を提供し、最大メモリ 2GB という制約を取り除く 64 ビット・コンピューティング・プラットフォームをサポートしています。

さらに、Replication Server のカウンタはすべて 64 ビット定義になり、高精度の演算が可能になっています。

64 ビット・プラットフォームへの移行については、『Replication Server 設定ガイド』の「第 3 章 Replication Server のアップグレードまたはダウングレード」を参照してください。64 ビット・プラットフォームのサポートを設定するには、『Replication Server 管理ガイド第 2 巻』の「パフォーマンス・チューニング」を参照してください。



## Replication Server 設定パラメータの変更

一部の設定パラメータは、変更すると、32 ビット版 Replication Server と 64 ビット版 Replication Server のパフォーマンスに影響を与えます。

表 19 : Replication Server 設定パラメータ

| パラメータ                          | 説明                                                                                                                              | 32 ビットの有効範囲 (バイト単位)        | 64 ビットの有効範囲 (バイト単位)              |
|--------------------------------|---------------------------------------------------------------------------------------------------------------------------------|----------------------------|----------------------------------|
| <b>dsi_sqt_max_cache_size</b>  | データベース・接続の最大ステープル・キュー・トランザクション (SQT) キャッシュ・サイズ。デフォルトの 0 は、 <b>sqt_max_cache_size</b> パラメータの現在の設定値が、接続の最大キャッシュ・サイズとして使用されることを示す。 | 最小： 0<br>最大：<br>2147483647 | 最小： 0<br>最大：<br>2251799813685247 |
| <b>dist_sqt_max_cache_size</b> | DIST 接続の最大ステープル・キュー・トランザクション (SQT) キャッシュ・サイズ。デフォルトの 0 は、 <b>sqt_max_cache_size</b> パラメータの現在の設定値が、接続の最大キャッシュ・サイズとして使用されることを示す。   | 最小： 0<br>最大：<br>2147483647 | 最小： 0<br>最大：<br>2251799813685247 |
| <b>sqt_max_cache_size</b>      | SQT インタフェースの最大キャッシュ・メモリ (バイト単位)。                                                                                                | 最小： 0<br>最大：<br>2147483647 | 最小： 0<br>最大：<br>2251799813685247 |

## memory\_limit 設定パラメータの変更

**memory\_limit** 設定パラメータを変更すると、32 ビット版 Replication Server と 64 ビット版 Replication Server のパフォーマンスに影響を与えます。

表 20 : memory\_limit 設定パラメータ

| パラメータ               | 説明                                                                                                                                                                                                                                                                                                          | 32 ビットでの有効範囲        | 64 ビットでの有効範囲              |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|---------------------------|
| <b>memory_limit</b> | Replication Server が使用できる合計メモリの最大値 (メガバイト単位)。その他のいくつかの設定パラメータの値は、 <b>memory_limit</b> によって示された、メモリ・プールから使用可能なメモリ量に直接関連する。これらの設定パラメータには、 <b>fstr_cachesize</b> 、 <b>md_source_memory_pool</b> 、 <b>queue_dump_buffer_size</b> 、 <b>sqt_max_cache_size</b> 、 <b>sre_reserve</b> 、 <b>sts_cachesize</b> などがある。 | 最小 : 0<br>最大 : 2047 | 最小 : 0<br>最大 : 2147483647 |

# Replication Manager 15.5 の新機能

Replication Manager 15.5 はデータ型 *bigdatetime* と *bigtime* をサポートします。

## bigdatetime と bigtime の複写の有効化

Adaptive Server 15.5 に含まれている *bigdatetime* および *bigtime* データ型を、Replication Manager 15.5 を使って有効にします。

*bigdatetime* および *bigtime* データ型をレプリケート・データベースとウォーム・スタンバイ・データベースに複写するには、そのデータ型を複写定義、ファンクション複写定義、サブスクリプション内で指定します。

[新しいテーブル複写定義の追加] ダイアログ・ボックスの [カラム] タブで、[データ型] 領域の [複写定義] リストから *bigdatetime* または *bigtime* を選択します。

### 参照：

- *bigdatetime* と *bigtime* の複写 (135 ページ)



# Replication Server バージョン 15.2 の新機能

Replication Server 15.2 では、DSI バルク・コピー・イン、非ブロッキング・コミット、引用符付き識別子、Replication Server ゲートウェイ、非 SQL 文の複写でのロー・カウントの検証、SQL 文の複写、非 Adaptive Server エラー・クラスのサポートが導入されています。Replication Server 15.2 では、非 Adaptive Server 複写機能のサポートも強化されました。

## DSI でのバルク・コピー・インのサポート

---

Replication Server バージョン 15.2 では、Adaptive Server® Enterprise 12.0 以降で、大量の **insert** 文を同じテーブルで複写するときのパフォーマンスを向上するバルク・コピー・インのサポートが導入されています。

バージョン 15.1 以前では、Replication Server が Adaptive Server にデータを複写するときに、Replication Server は、SQL の **insert** コマンドを生成し、コマンドを Adaptive Server に送信して、Adaptive Server がローを処理し、オペレーションの結果を送り返してくるのを待機します。このプロセスは、1 日の終わりのバッチ処理や取引の統合などの大量のデータが複写される場合に、Replication Server のパフォーマンスに影響します。

Replication Server 15.2、は Open Client™ Open Server™ Bulk-Library を使用して、レプリケート・データベースにトランザクションを送信する Replication Server モジュールであるデータ・サーバ・インタフェース (DSI) にバルク・コピー・インを実装します。

---

**注意：**バルク・コピー・インは、Adaptive Server データベースでのみサポートされます。DSI のバルク・コピー・インを on にしていて、レプリケート・データベースが Adaptive Server でない場合、DSI がシャット・ダウンされ、エラーが返されます。

---

Open Client Open Server Bulk-Library の詳細については、『Open Client and Open Server Common Libraries Reference Manual』を参照してください。

## サブスクリプションのマテリアライゼーションの強化

---

バルク・コピー・インにより、サブスクリプション・マテリアライゼーションのパフォーマンスも向上します。**dsi\_bulk\_copy** を on にすると、各トランザクションの **insert** コマンドの数が **dsi\_bulk\_threshold** を超えた場合に、Replication Server はバルク・コピー・インを使用してサブスクリプションをマテリアライズします。

**注意：**通常の複製では、テーブルのバルク・オペレーションは、**autocorrection** が有効な場合に無効になります。ただし、マテリアライゼーションでは、**dsi\_bulk\_threshold** に達していて、マテリアライゼーションが障害からリカバリするノンアトミック・サブスクリプションでない場合は、**autocorrection** が有効になっていても、バルク・オペレーションが適用されます。

サブスクリプション・マテリアライゼーションの詳細については、『Replication Server 管理ガイド第 1 巻』を参照してください。

## 新しいコネクション・パラメータ

次のデータベース・コネクション・パラメータによって、DSI でのバルク・オペレーションを制御します。

| パラメータ                     | 説明                                                                                                                                                                                                                                                                                                                     |
|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>dsi_bulk_copy</b>      | コネクションのバルク・コピー・イン機能を on または off にする。<br><b>dynamic_sql</b> と <b>dsi_bulk_copy</b> の両方を on にすると、DSI によってバルク・コピー・インが適用される。バルク・コピー・インが使用されない場合は、動的 SQL が使用される。<br><br>デフォルト値は off                                                                                                                                         |
| <b>dsi_bulk_threshold</b> | <b>insert</b> コマンドの数。この数に到達すると、バルク・コピー・インを使用するように Replication Server をトリガする。ステابل・キュー・トランザクション (SQT) は、大量の <b>insert</b> コマンドのバッチを検出すると、バルク・コピー・インを適用するかどうかを決定するために、指定された数の <b>insert</b> コマンドをメモリに保持する。これらのコマンドはメモリに保持されるため、この値を <b>dsi_large_xact_size</b> の設定値よりも大きい値に設定しないことが推奨される。<br><br>最小値：1<br><br>デフォルト値は 20 |

### 使用法

**dsi\_bulk\_copy** と **dsi\_bulk\_threshold** の値を設定するには、次のようにします。

- **alter connection** を使用して、バルク・コピー・インのコネクション・パラメータをコネクション・レベルで変更する。

```
alter connection to dataserver.database
set {dsi_bulk_copy | dsi_bulk_threshold} to value
```

- **configure replication server** を使用して、サーバのデフォルトを変更する。

```
configure replication server
set {dsi_bulk_copy | dsi_bulk_threshold} to value
```

**dsi\_bulk\_copy** と **dsi\_bulk\_threshold** の値を調べるには、**admin config** を使用します。

**dsi\_bulk\_copy** を on にすると、SQT によって、トランザクションに含まれる同じテーブルでの連続する **insert** 文の数がカウントされます。この数が **dsi\_bulk\_threshold** に達すると、DSI は次の操作を行います。

1. DSI が **insert** ではないコマンドまたは別の複製テーブルに属するコマンドに到達するまで、データを Adaptive Server にバルク・コピーします。
2. トランザクションの残りのコマンドの実行を続行します。

Adaptive Server が、バルク・オペレーションが成功した場合はその終了時点、またはオペレーションが失敗した時点でバルク・コピー・インの結果を送信します。

**注意：** DSI でのバルク・コピー・インの実装により、複数文のトランザクションがサポートされるため、バルク・コピーに含まれないコマンドがトランザクションに含まれている場合でも、DSI でバルク・コピー・インを実行できます。

## バルク・コピー・イン用の新しいカウンタ

バルク・コピー・イン用の新しいカウンタが追加されました。

| カウンタ              | 説明                                                                                                                   |
|-------------------|----------------------------------------------------------------------------------------------------------------------|
| DSINoBulkDatatype | データに含まれているデータ型にバルク・コピー・インとの互換性がないため、スキップされたバルク・オペレーションの数。                                                            |
| DSINoBulkFstr     | テーブルに <b>rs_insert</b> または <b>rs_writetext</b> のカスタマイズされたファンクション文字列が含まれているため、スキップされたバルク・オペレーションの数。                   |
| DSINoBulkAutoc    | <b>autocorrection</b> が有効になっているテーブルがあるため、スキップされたバルク・オペレーションの数。                                                       |
| DSIEBFBulkNext    | 次のコマンドがバルク・コピーであるために実行されたバッチ・フラッシュの数。                                                                                |
| DSIEBulkSucceed   | ターゲット・データベースでデータ・サーバ・インタフェース・エグゼキュータ (DSI/E: Data Server Interface executor) が <b>blk_done(CS_BLK_ALL)</b> を呼び出した回数。 |
| DSIEBulkCancel    | ターゲットのデータベースで <b>blk_done(CS_BLK_CANCEL)</b> が DSI/E によって呼び出された回数。                                                   |
| DSIEBulkRows      | DSI/E がバルク・コピー・インを使用してレプリケート・データ・サーバに送信したローの数。                                                                       |
| BulkTime          | DSI/E がバルク・コピー・インを使用してレプリケート・データ・サーバにデータを送信するのにかかった時間 (ミリ秒)。                                                         |

## 制限事項

DSI がバルク・コピー・インを使用しない、またはバルク・コピー・イン機能がサポートされていない場合があります。

次の場合、Replication Server の DSI では、バルク・コピー・インが使用されません。

- オートコレクションがオンになっており、データがサブスクリプション・マテリアライゼーションの一部ではない場合。
- `rs_insert` にユーザ定義のファンクション文字列がある場合。
- `text` カラムに `rs_writetext` 用のユーザ定義ファンクション文字列があり、出力の `none` または `rpc` がある場合。
- データ・ローに `opaque` データ型または `rs_datatype.canonic_type` の値が 255 のユーザ定義データ型 (UDD) が含まれている場合。
- データ・ローに `image` または Java データ型が含まれている場合。

バルク・コピー・イン機能は、以下の条件下ではサポートされません。以下の場合は、バルク・コピー・インを無効にしてください。

- レプリケート・データベースが Adaptive Server でない場合。この場合、DSI のバルク・コピー・インを有効にすると、DSI が終了し、エラー・メッセージが返される。
- Replication Server とレプリケート Adaptive Server の文字セット間でデータ・サイズが変化し、データ・ローに `text` カラムが含まれる場合。この場合、DSI のバルク・コピー・インを有効にすると、DSI が終了し、次のメッセージが返される。

```
Bulk-Lib routine 'blk_textxfer' failed.
Open Client Client-Library error: Error: 16843015,
Severity 1 -- 'blk_textxfer(): blk layer: user
error: The given buffer of xxx bytes exceeds the
total length of the value to be transferred.'
```

- `owner.tablename` の長さが 255 バイトを超え、レプリケート・データベースが Adaptive Server 15.0.3 中間リリースよりも前のバージョンである場合。この場合、DSI のバルク・コピー・インを有効にすると、Replication Server が終了し、次のメッセージが返される。

```
Bulk-Lib routine 'blk_init' failed.
```

`owner.tablename` の長さが 255 バイトを超えている場合にバルク・コピー・インを使用しないように指定するには、次の手順に従います。

1. トレースを on にします。

```
trace "on", rsfeature, ase_cr543639
```

2. Replication Server 設定ファイルに以下を追加します。



```
trace=rsfeature,ase_cr543639
```

その他の制限事項：

- **insert** コマンドとは異なり、バルク・コピー・インでは、タイムスタンプは生成されない。複写に *timestamp* カラムが含まれていない場合、*timestamp* カラムには NULL 値が挿入される。バルク・コピー・インを無効にするか、または *timestamp* カラムを含めるように複写定義を設定する。
- **writetext** ファンクション文字列を **no log** に変更した場合でも、*Text* カラムと *image* カラムのログは常に記録される。
- バルク・コピーは、Adaptive Server で **insert** トリガを呼び出さない。
- 設定パラメータ **send\_timestamp\_to\_standby** はバルク・コピー・インに影響しない。*timestamp* データは常に複写される。

## 非ブロッキング・コミット

---

Replication Server 15.2 には、Adaptive Server の遅延コミット機能を使用して複写のパフォーマンスを向上させる非ブロッキング・コミットが含まれています。

**注意：** 非ブロッキング・コミットを使用するには、Sybase Enterprise Connect™ Data Access 15.0 ESD #3 (ECDA) 以降を使用する必要があります。

---

## Adaptive Server の遅延コミット機能

Replication Server 15.0 以降には、トランザクションのコミット・フェーズを遅延させることでパフォーマンスを向上させるように設計された遅延コミット機能が備わっています。

コミット・フェーズには、トランザクションのログ・レコードのディスクへの書き込みと、その後のクライアント・アプリケーションへのトランザクション・ステータスの通知が含まれます。遅延コミットを有効にすると、Adaptive Server は、対応するトランザクション・ログをディスクに書き込む前に、コミットが成功したことをクライアント・アプリケーションに通知します。このディスクへの書き込みの遅延により、最後のアクティブなログ・ページでの競合が減少するため、パフォーマンスが向上します。

ただし、Adaptive Server が終了するか、または **shutdown with no wait** を使用して Adaptive Server を停止した場合、トランザクション・ログの最後のページが失われる場合があります。

『Adaptive Server Enterprise 15.0 システム管理ガイド 第2巻』の「第11章 バックアップおよびリカバリ・プランの作成」の「ログ・レコードをコミットするときを決定するための **delayed\_commit** の使用」、および『Adaptive Server Enterprise 15.0 リファレンス・マニュアル：コマンド』の「第1章 コマンド」の「set」に記

載されている **set** コマンドの **delayed\_commit** パラメータの説明を参照してください。

## dsi\_non\_blocking\_commit 設定パラメータ

**dsi\_non\_blocking\_commit** 設定パラメータは、Replication Server がコミット後にメッセージを保存する期間を延長します。

保存期間を延長するには、より大きなステابل・キューが必要になります。『Replication Server 管理ガイド第 1 巻』の「Replication Server の技術的概要」の「ステابل・キュー」を参照してください。

データベース・コネクシオンに **dsi\_non\_blocking\_commit** を設定するには、**alter connection** を次のように使用します。

```
alter connection to data_server.database
set dsi_non_blocking_commit to 'value'
```

**注意：** このパラメータを **alter connection** で使用して、スタンバイ環境でアクティブ・データベース・コネクシオンを設定することはできません。

**dsi\_non\_blocking\_commit** をサーバのデフォルトとして設定するには、**configure replication server** を次のように使用します。

```
configure replication server
set dsi_non_blocking_commit to 'value'
```

ここで、*value* は、保存期間を延長する分数 (最大 60) です。デフォルトは 0 で、非ブロッキング・コミットは無効になっています。

**dsi\_non\_blocking\_commit** の現在の *value* を調べるには、**admin config** を使用します。

ここで説明するコマンドの詳細については、『Replication Server リファレンス・マニュアル』を参照してください。

### バージョン要件

**dsi\_non\_blocking\_commit** は、Adaptive Server 15.0 以降、および Oracle 10g v2 以降でのみ使用できます。サポートされていないバージョンの Adaptive Server、Oracle、またはその他のデータベースの場合、Replication Server は、コネクシオンの **dsi\_non\_blocking\_commit** 設定パラメータを無効にします。

## rs\_non\_blocking\_commit システム・ファンクション

**dsi\_non\_blocking\_commit** の値が 1 ~ 60 なら、**rs\_non\_blocking\_commit** は DSI がレプリケート・データ・サーバに接続するたびに実行されます。

**dsi\_non\_blocking\_commit** の値がゼロなら **rs\_non\_blocking\_commit** は実行されません。

**rs\_non\_blocking\_commit** には、ファンクション文字列クラス・スコープがありません。

Adaptive Server 15.0 以降では、**rs\_non\_blocking\_commit** 関数は **set delayed\_commit on** ファンクション文字列にマッピングされます。Oracle 10g v2 以降では、対応する **alter** ファンクション文字列にマッピングされます。Sybase 以外のデータベースでは、**rs\_non\_blocking\_commit** は null にマッピングされます。

## rs\_non\_blocking\_commit\_flush システム・ファンクション

**rs\_non\_blocking\_commit\_flush** は **dsi\_non\_blocking\_commit** での間隔の指定に従って 1 ~ 60 分間隔で実行されます。**dsi\_non\_blocking\_commit** の値がゼロの場合、**rs\_non\_blocking\_commit\_flush** は実行されません。

**rs\_blocking\_commit\_flush** には、ファンクション文字列クラス・スコープがありません。

Adaptive Server 15.0 以降と Oracle 10g v2 以降では、**rs\_non\_blocking\_commit\_flush** は対応するファンクション文字列にマッピングされます。Sybase 以外のデータベースでは、**rs\_non\_blocking\_commit\_flush** は null にマッピングされます。

### 例 1

Adaptive Server 用に **rs\_non\_blocking\_commit\_flush** ファンクション文字列のインスタンスを作成します。

```
create function string rs_non_blocking_commit_flush
  for sqlserver_derived_class
  output language
  'set delayed_commit off; begin tran; update rs_lastcommit set
  origin_time = getdate() where origin = 0; commit tran;
  set delayed_commit on'
```

### 例 2

Oracle 用に **rs\_non\_blocking\_commit\_flush** ファンクション文字列のインスタンスを作成します。

```
create function string rs_non_blocking_commit_flush
  for oracle_derived_class
  output language
  'alter session set commit_write = immediate; begin tran;
  update rs_lastcommit set origin_time = getdate() where
  origin = 0; commit tran; alter session set commit_write =
  nowait'
```

## サポートされる **Adaptive Server 以外のデータベース**

非ブロッキング・コミットを有効にした Replication Server 15.2 では、Oracle 10g v2 以降への複製がサポートされます。これは、Oracle 10g v2 が、遅延コミットに似た機能をサポートしているためです。

Replication Server 15.2 の異機種データ型サポート (HDS: Heterogeneous Datatype Support) スクリプトには、非ブロッキング・コミット機能をサポートする新しいファンクション文字列があります。Sybase Enterprise Connect Data Access for Oracle では、これらのファンクション文字列がサポートされます。『Replication Server 異機種間複製ガイド』を参照してください。

## 引用符付き識別子

Replication Server バージョン 15.2 では引用符付き識別子が強化されました。

スペースや非英数字などの特殊文字が含まれる、英字以外の文字で始まる、または予約語に相当するオブジェクト名は、正しく解析されるように二重引用符文字で囲む必要があります。これらのオブジェクト名は、引用符付き識別子と呼ばれます。Replication Server バージョン 15.1 以前では引用符付き識別子を受け入れられますが、これらのバージョンでは、データ・サーバへの引用符付き識別子の転送はサポートされていません。

---

**注意:** 引用符付き識別子を使用するには、ECDA 15.0 ESD #3 以降を使用してください。

---

Replication Server 15.2 では、引用符付き識別子のサポートにより、以下が行えます。

- 複製定義で識別子を引用符付きとしてマーク付けする。
- 引用符付き識別子をデータ・サーバに転送できるコネクションを作成する。

識別子での埋め込み二重引用符の使用は現在サポートされていません。

Adaptive Server、SQL Anywhere®、Microsoft SQL Server、Universal Database (UDB)、Oracle などのデータ・サーバでは、サポートされる長さ、特殊文字、予約語の観点から、引用符付き識別子は異なる方法で処理されます。異機種環境では、複製されている引用符付き識別子がプライマリ・データ・サーバとレプリケート・データ・サーバの両方で有効であることを確認してください。

### バージョン要件

引用符付き識別子の複製を成功させるには、プライマリ Replication Server とレプリケート・データ・サーバに接続する Replication Server のバージョンを 15.2 にします。ただし、ルート上の中間 Replication Server は、以前のバージョンでもかまいません。

## 引用符付き識別子のサポートを有効にする設定パラメータ

**dsi\_quoted\_identifier** 設定パラメータは、データ・サーバ・インタフェース (DSI) での引用符付き識別子のサポートを有効化または無効化します。

**create connection** または **alter connection** コマンドを使用して、データ・サーバ・コネクシオンで **dsi\_quoted\_identifier** を on または off に設定します。

**dsi\_quoted\_identifier** のデフォルト値は off です。

**dsi\_quoted\_identifier** の値を確認するには、**admin config** コマンドを使用します。

## 引用符付き識別子としてマークを付けるコマンド

**create replication definition**、**alter replication definition** コマンドにより、新しいパラメータ **quoted** を使用して引用符付き識別子をマーク付けできます。

識別子がマーク付けされ、**dsi\_quoted\_identifier** が on に設定されている場合、複写定義にサブスクリプションを作成するレプリケート・サーバが、マーク付けされた識別子を引用符付き識別子として受け取ります。**dsi\_quoted\_identifier** が off の場合、マーク付けは無視され、レプリケート・サーバは引用符付き識別子を受け取りません。

---

**注意：** ウォーム・スタンバイ・データベースおよび複写定義のサブスクライバへの複写時に、プライマリ・テーブル名は引用符付きとしてマーク付けされているが、レプリケート・テーブル名はマーク付けされていない場合 (またはその逆)、Replication Server は、プライマリ・テーブル名とレプリケート・テーブル名の両方を引用符付きとして送信します。

---

この項で説明するコマンドの詳細については、『Replication Server リファレンス・マニュアル』を参照してください。

### **create replication definition** の構文変更

**create replication definition** の構文は引用符付き識別子をサポートするために変更されました。

```
create replication definition replication_definition
with primary at data_server.database
[with all tables named [table_owner.] 'table_name' [quoted] |
[with primary table named [table_owner.] 'table_name']
  with replicate table named [table_owner.] 'table_name' [quoted]]
(column_name [as replicate_column_name] [datatype [null | not null]
  [map to published_datatype]] [quoted]
[, column_name [as replicate_column_name]
  [datatype [null | not null]
  [map to published_datatype]] [quoted]...)
primary key (column_name [, column_name]...)
[searchable columns (column_name [, column_name]...)]
```

```
[send standby [{all | replication definition} columns]]
[replicate {minimal | all} columns]
[replicate_if_changed (column_name [, column_name]...)]
[always_replicate (column_name [, column_name]...)]
[with dynamic sql | without dynamic sql]
```

**例**

引用符付き識別子としてカラム *foo\_coll* を持つテーブル *foo* を作成します。

```
create replication definition repdef
  with primary at primaryDS.primaryDB
  with all tables named "foo"
  ("foo_coll" int quoted, "foo_col2" int)
  primary key ("foo_coll")
```

**alter replication definition の構文変更**

**alter replication definition** の構文は引用符付き識別子をサポートするために変更されました。

```
alter replication definition replication_definition
{with replicate table named table_owner.'table_name' |
add column_name [as replicate_column_name]
  [datatype [null | not null]]
  [map to published_datatype] [quoted],... |
alter columns with column_name
  [as replicate_column_name] [quoted | not quoted],... |
alter columns with column_name
  datatype [null | not null]
  [map to published_datatype],... |
alter columns column_name {quoted | not quoted}
add primary key column_name [, column_name]... |
drop primary key column_name [, column_name]... |
add searchable columns column_name [, column_name]... |
drop searchable columns column_name [, column_name]... |
send standby [off | {all | replication definition} columns] |
replicate {minimal | all} columns |
replicate_if_changed column_name [, column_name]... |
always_replicate column_name [, column_name]...} |
{with | without} dynamic sql
alter replicate table name {quoted | not quoted}
```

**例 1**

*foo* という名前のテーブルを引用符付きとしてマーク付けします。

```
alter replication definition repdef
  alter replicate table name "foo" quoted
```

**例 2**

カラム *foo\_coll* のマーク付けを解除します。

```
alter replication definition repdef
  with replicate table named "foo"
  alter columns "foo_coll" not quoted
```

## rs\_set\_quoted\_identifier ファンクション文字列

**rs\_set\_quoted\_identifier** は、引用符付き識別子を受け取るデータ・サーバの種類ごとに、コネクションを適切に設定します。

データ・サーバは、引用符付き識別子を異なる方法で受け取ります。Adaptive Server、SQL Anywhere、Microsoft SQL Server は、引用符付き識別子を予期しないため、引用符付き識別子のコネクションを設定するための特別なコマンドを必要とします。Oracle と UDB では、引用符付き識別子を受け入れるようにコネクションを設定する必要はありません。

**rs\_set\_quoted\_identifier** ファンクション文字列を使用して、データ・サーバの種類ごとに DSI コネクションを適切に設定します。**dsi\_quoted\_identifier** を on にすると、Replication Server は **rs\_set\_quoted\_identifier** をレプリケート・データ・サーバに送信し、引用符付き識別子を予期するようにデータ・サーバに通知します。レプリケート・データ・サーバが Adaptive Server、SQL Anywhere、または Microsoft SQL Server の場合、**rs\_set\_quoted\_identifier** は **set quoted\_identifiers on** コマンドに設定されます。それ以外の場合は **rs\_set\_quoted\_identifier** は "" に設定されます。

**rs\_set\_quoted\_identifier** には、ファンクション文字列クラス・スコープがありません。

## rs\_helprep の変更

**rs\_helprep** は引用符付き識別子を引用符付きとして表示するように変更されました。下の例は、**create replication definition** と **alter replication definition** が引用符付き識別子をどのように定義し、**rs\_helprep** がこれらの識別子をどのように表示するかを示します。

### 例 1

次のようなテーブルおよび複製定義を前提とします。

```
create table t1 (c1 int, c2 int)

create replication definition r1
  with primary at ost_wasatch_08.pdb1
  with all tables named t1
  (c1 int, "c2" int quoted)
  primary key (c1)
```

**rs\_helprep r1** により、**c2** が引用符付き識別子として表示されます。

| Replication Definition Name | PRS   | Type  | Creation Date |
|-----------------------------|-------|-------|---------------|
| -----                       | ----- | ----- | -----         |

```

r1                ost_wasatch_09          Tbl  Nov 11, 2008 2:28PM
PDS.DB            Primary Owner           Primary Table
-----
ost_wasatch_08.pdb1          t1

Replicate Owner      Replicate Table
-----
                        t1

Send Min Cols. Used by Standby Min Vers Dynamic SQL SQL Stmt. Rep.
-----
No                    No                1000      On          None

Col. Name    Rep. Col. Name  Datatype  Len.  Pri. Col.  Searchable
-----
c1           c1              int       4     1          0
"c2"        "c2"           int       4     0          0

Function Name  FString Class          FString Source  FString
Name
-----
rs_delete     rs_sqlserver_function_class  Class
Default      rs_delete
rs_insert     rs_sqlserver_function_class  Class
Default      rs_insert
rs_select     rs_sqlserver_function_class  Class
Default      rs_select
rs_select_   rs_sqlserver_function_class  Class
Default      rs_select_
with_lock
rs_truncate   rs_sqlserver_function_class  Class
Default      rs_truncate
rs_update     rs_sqlserver_function_class  Class
Default      rs_update
with_lock

Subscriptions known at this Site 'ost_wasatch_09'.
Subscription Name  Replicate DS.DB  Owner  Creation Date
-----
(return status = 0)

```

## 例 2

例 1 で定義されたテーブルおよび複製定義を前提とし、*t1* を引用符付き識別子として定義します。

```

alter replication definition r1
alter replicate table name "t1" quoted

```

rs\_helprep r1 により、*c2* と *t1* が引用符付き識別子として表示されます。

```

Replication Definition Name  PRS          Type Creation Date
-----
r1                ost_wasatch_09          Tbl  Nov 11, 2008 2:28PM

```



```

PDS.DB              Primary Owner      Primary Table
-----
ost_wasatch_08.pdb1              "t1"

Replicate Owner      Replicate Table
-----
                        "t1"

Send Min Cols. Used by Standby Min Vers Dynamic SQL SQL Stmt. Rep.
-----
No                    No                    1000      On          None

Col. Name      Rep. Col. Name      Datatype      Len.      Pri. Col.      Searchable
-----
c1              c1                    int            4          1              0
"c2"            "c2"                  int            4          0              0

Function Name      FString Class          FString Source      FString
Name
-----
rs_delete          rs_sqlserver_function_class  Class
Default           rs_delete
rs_insert          rs_sqlserver_function_class  Class
Default           rs_insert
rs_select          rs_sqlserver_function_class  Class
Default           rs_select
rs_select_        rs_sqlserver_function_class  Class
Default           rs_select_
with_lock
rs_truncate       rs_sqlserver_function_class  Class
Default           rs_truncate
rs_update         rs_sqlserver_function_class  Class
Default           rs_update

Subscriptions known at this Site 'ost_wasatch_09'.
Subscription Name  Replicate DS.DB      Owner      Creation Date
-----
(return status = 0)

```

### 例 3

例 2 で定義された複製定義を前提とし、*c2* を引用符付きではないとして定義します。

```
alter replication definition r1
alter columns c2 not quoted
```

`rs_helprep r1` により、*t1* のみが引用符付き識別子として表示されます。

```

Replication Definition Name  PRS      Type      Creation Date
-----
r1                            ost_wasatch_09      Tbl      Nov 11, 2008 2:28PM

```

## Replication Server バージョン 15.2 の新機能

```

PDS.DB          Primary Owner      Primary Table
-----
ost_wasatch_08.pdb1          "t1"

Replicate Owner      Replicate Table
-----
                        "t1"

Send Min Cols. Used by Standby Min Vers Dynamic SQL SQL Stmt. Rep.
-----
No                   No                   1000      On          None

Col. Name      Rep. Col. Name  Datatype   Len.   Pri. Col.  Searchable
-----
c1             c1              int        4      1          0
c2             c2              int        4      0          0

Function Name    FString Class          FString Source  FString
Name
-----
rs_delete       rs_sqlserver_function_class  Class
Default        rs_delete
rs_insert       rs_sqlserver_function_class  Class
Default        rs_insert
rs_select       rs_sqlserver_function_class  Class
Default        rs_select
rs_select_     rs_sqlserver_function_class  Class
Default        rs_select_
with_lock                               with_lock
rs_truncate     rs_sqlserver_function_class  Class
Default        rs_truncate
rs_update       rs_sqlserver_function_class  Class
Default        rs_update

Subscriptions known at this Site 'ost_wasatch_09'.
Subscription Name  Replicate DS.DB  Owner  Creation Date
-----
(return status = 0)

```

## Replication Server ゲートウェイ

Replication Server 15.2 では、Replication Server ゲートウェイが導入されています。これにより、さまざまなサーバへの明示的なログインが最小限に抑えられます。

複製システムの管理では、複製システム管理者 (RSA) が、複数の Replication Server、ID サーバ、対応する Replication Server システム・データベース (RSSD) にログインします。また、RSA は、Replication Server と RSSD の間でログインを頻繁に切り替えます。

Replication Server ゲートウェイは、RSSD のプライマリ・ユーザ名とパスワードを使用して RSSD にログインし、ID サーバのユーザ名とパスワードを使用して ID サーバに、リモート・サーバ ID (RSI) を使用してリモート Replication Server に、メンテナンス・ユーザ ID を使用してリモート Adaptive Server にログインします。Replication Server 自体にアクセスするとき、この情報を繰り返し提供する必要はありません。

#### 制限事項

Replication Server ゲートウェイを使用する場合、Replication Server は文字セットの変換を実行できないため、クライアントとサーバで同じロケール・セットを使用してください。

### カスケード・コネクション

Replication Server ゲートウェイでは、Replication Server と、Replication Server に直接接続されていないサーバとの通信を可能にするカスケード・コネクションもサポートされます。

カスケード・コネクションでは、クライアント・コネクションを使用して複写ドメインを管理することもできます。たとえば、ID サーバに接続し、その後、ID サーバの RSSD に接続できます。この場合、プライマリの制御 Replication Server と ID サーバの両方がゲートウェイであり、コマンドが ID サーバの RSSD に渡され、結果セットが返されます。

### Replication Server ゲートウェイを有効にするコマンド

Replication Server をその RSSD、ID サーバ、またはリモート Replication Server へのゲートウェイにするために **connect** コマンドが追加されました。

#### 構文

```
connect [to] [rssd | idserver | srv_name | ds_name.db_name]
```

#### パラメータ

- **rssd** – Replication Server をその RSSD のゲートウェイにする。設定ファイルの *RSSD\_primary\_user* エントリと *RSSD\_primary\_pw* エントリをゲートウェイが使用できるようにする。
- **idserver** – Replication Server をその ID サーバのゲートウェイにする (Replication Server 自体が ID サーバでない場合)。設定ファイルの *ID\_user* エントリと *ID\_pw* エントリをゲートウェイが使用できるようにする。
- **srv\_name** – ゲートウェイを接続するリモート Replication Server の名前。Replication Server ゲートウェイは、RSI を使用してリモート・サーバにログインするため、リモート・サーバへの直接ルートが必要。

---

**注意：** Replication Server は、それ自体に直接接続できません。ただし、カスケード・コネクションを使用することで、この問題に対処できます。

---

- **ds\_name . db\_name** – ゲートウェイを接続するリモート・データ・サーバおよびデータベースの名前。Replication Server ゲートウェイは、メンテナンス・ユーザを通じてリモート・データ・サーバにログインする。これにより、指定されたデータベースのメンテナンス・ユーザに許可されているタスクを実行できるようになる。ただし、接続先のデータ・サーバで定義された他のデータベースにはアクセスできない。

Replication Server ゲートウェイは、Adaptive Server と、Enterprise Connect Data Access (ECDA) を必要としない Sybase® IQ データ・サーバに直接接続できるようにします。その他のデータ・サーバの場合、Replication Server ゲートウェイは、ECDA を使用して Replication Server とリモート・データ・サーバに接続する必要があります。

## 使用法

**connect** コマンドの詳細については、『Replication Server リファレンス・マニュアル』を参照してください。

## パーミッション

**connect** コマンドを発行するには、Replication Server への初回のログインのための **sa** の役割が必要です。

## コネクションを追跡するコマンド

ゲートウェイで作成されたカスケード・コネクションは、コネクション・スタックで保持され、最初の **connect** コマンドを発行した Replication Server がスタックの一番下に置かれます。カスケード・コネクションを管理するには、**show connection** コマンドと **show server** コマンドを使用します。

- **show connection** – コネクション・スタックの内容を表示する。
- **show server** – 現在稼働中のサーバを表示する。

## 使用法

コネクション・スタックに Replication Server versions 15.2 と 15.1 以前が含まれている場合、**disconnect** コマンドを発行すると、**show connection** コマンドや **show server** コマンドを実行したときに、想定した出力が表示されない可能性があります。これは、Replication Server 15.1 以前では、**disconnect** コマンドの動作が異なるためです。これらのバージョンでは、**disconnect** コマンドは、ゲートウェイ・モードを終了し、最初の **connect** コマンドを発行した Replication Server に稼働中のサーバのステータスを返します。

## コネクションを削除するコマンド

**disconnect** コマンドを使用して、サーバへのコネクションを終了します。

### 構文

```
{disconnect | disc} [all]
```

```
select @variable = {expression | select_statement}
      [, @variable = {expression | select_statement} ...]
      [from table_list]
      [where search_conditions]
      [group by group_by_list]
      [having search_conditions]
      [order by order_by_list]
      [compute function_list [by by_list]]
```

### 使用法

- **disconnect** により、コネクション・スタックは一度に1つずつ終了します。すべてのコネクションを終了するには、**disconnect all** を使用します。
- **disconnect** コマンドの動作は Replication Server 15.1 までのバージョンでは異なります。これらのバージョンでは、**disconnect** コマンドは、ゲートウェイ・モードを終了し、最初の **connect** コマンドを発行した Replication Server に稼働中のサーバのステータスを返します。コネクション・スタックに Replication Server のバージョン 15.2、15.1、およびそれ以前が含まれているときに **disconnect** コマンドを発行した場合、**show connection** コマンドと **show server** コマンドは期待される出力を表示しない場合があります。
- **disconnect** の詳細については、『Replication Server リファレンス・マニュアル』を参照してください。

## 非 SQL 文の複写ロー・カウントの検証

Replication Server 内でのエラーに対処するために、Replication Server 15.2 には、SQL 文の複写に関係しないロー・カウント検証エラーに関連した Replication Server のエラー・クラスとエラー・アクションに対するサポートが含まれています。

**注意：** Replication Server は、カスタム・ファンクション文字列内のそのようなコマンドに対するロー・カウントの検証を無視します。詳細については、『Replication Server 管理ガイド第2巻』の「パフォーマンス・チューニング」の「SQL 文の複写ではオートコレクションはサポートされない」を参照してください。

Replication Server 15.2 では Replication Server エラー・クラスが追加されました。その結果、バージョン 15.2 からは、データ・サーバ・エラー・クラスと Replication Server エラー・クラスの2つのエラー・クラス・タイプにコネクションが関連付けられます。コネクションの Replication Server のエラー・クラスとの関連付けは、Replication Server がデフォルトの Replication Server エラー・アクションに対してオーバーライドする Replication Server エラー・クラスをクエリする前に行われる必要があります。コネクションとの関連付けは1つの Replication Server エラー・クラスにしかできません。ただし、1つの Replication Server エラー・クラスを複数のコネクションに関連付けることはできます。Replication Server エラー・クラスとコネクションを関連付けるには、**create connection** コマンドと **alter connection** コマンドの **set replication server error class** パラメータを使用します。

Replication Server がエラーに応答するとき、そのコネクションに関連付けられた Replication Server エラー・クラスをまず最初を探します。Replication Server エラー・クラスが見つからなかったときは、そのサーバに指定されているデフォルトの **rs\_repserver\_error\_class** エラー・クラスが使用されます。

### 参照：

- SQL 文の複写に対するロー・カウントの検証 (171 ページ)

## Replication Server エラー・クラスを作成するコマンド

Replication Server 15.2 では、**create error class** を使用して Replication Server 内で発生するエラーに対するエラー・アクションを割り当てるための Replication Server エラー・クラスを作成できます。

### 構文

```
create [replication server] error class error_class  
[set template to template_error_class]
```

### パラメータ

- **replication server** – 新しいエラー・クラスが Replication Server エラー・クラスであり、データ・サーバのエラー・クラスではないことを示す。
- **error\_class** – 新しいエラー・クラスの名前。名前は複写システム内でユニークにし、識別子の規則に従わなければなりません。

---

**注意：** Replication Server エラー・クラスとデータ・サーバのエラー・クラスを同じ名前にすることはできません。

- **set template to template\_error\_class** – この句を使用して、別のエラー・クラスに基づいてエラー・クラスを作成する。**create error class** により、テンプレート

のエラー・クラスのエラー・アクションが新しいエラー・クラスにコピーされる。

## 例

- **例 1** – デフォルトの `rs_repserver_error_class` に基づいて `my_rs_err_class` を作成します。

```
create replication server error class my_rs_err_class
set template to rs_repserver_error_class
```

## 使用法

**drop error class** を使用すると、Replication Server エラー・クラスを削除できます。  
**move primary** を使用すると、Replication Server エラー・クラスのプライマリ Replication Server を変更できます。

## エラー・アクションを割り当てるコマンド

エラー・アクションを指定するには、Replication Server エラー・クラスのプライマリ・サイトで **assign action** コマンドを使用します。

## 構文

```
assign action
  {ignore | warn | retry_log | log | retry_stop | stop_replication}
  error_class
  to server_error1 [, server_error2]...
```

## パラメータ

- **error\_class** – アクションを割り当てるエラー・クラスの名前。Replication Server 15.2 では、デフォルトの `rs_repserver_error_class` エラー・クラスなどの Replication Server のエラー・クラスを指定できる。
- **server\_error** – エラーの番号。

Replication Server では、SQL 文の複製に関係しないエラー・アクションにエラー番号を指定できます。

## 例

- **例 1** – Replication Server でエラー番号 5185 が発生した場合に **ignore** エラー・アクションを割り当てます。

```
assign action ignore for rs_repserver_error_class to 5185
```

- **例 2** – Replication Server でエラー番号 5186 が発生した場合に **warn** エラー・アクションを割り当てます。

```
assign action warn for rs_repserver_error_class to 5186
```

次に示すのは、ロー・カウント・エラーが発生した場合に表示されるエラー・メッセージの例です。

```
DSI_SQLDML_ROW_COUNT_INVALID 5186
Row count mismatch for SQLDML command executed on
'mydataserver.mydatabase'.
The command impacted 1000 rows but it should impact 1500 rows.
```

## 使用法

『Replication Server リファレンス・マニュアル』の「Replication Server コマンド」の「**assign action**」と表 3-17 を参照してください。

## SQL 文複製以外のエラー番号

SQL 文の複製に関係しないエラー・アクションに対する Replication Server のエラー番号を次に示します。

表 21 : Replication Server エラー・クラスのエラー・アクション

| server_error | エラー・メッセージ                                                                                                                                           | デフォルトのエラー・アクション | 説明                                                                                                                |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|-------------------------------------------------------------------------------------------------------------------|
| 5185         | Row count mismatch for the command executed on 'dataserver.database'. The command impacted x rows but it should impact y rows.                      | warn            | このメッセージは、SQL 文の複製、ストアド・プロシージャ、またはオートコレクションが有効になっているロー変更の一部ではないコマンドがデータ・サーバに送られた後、影響を受けたロー数が予想されたロー数とは異なる場合に表示される。 |
| 5187         | Row count mismatch for the autocorrection delete command executed on 'dataserver.database'. The command deleted x rows but it should delete y rows. | warn            | このメッセージは、オートコレクションが有効な場合、delete コマンドがデータ・サーバに送られた後、影響を受けたロー数が予想されたロー数とは異なる場合に表示される。                               |



## Replication Server エラー・クラスを表示するストアド・プロシージャ

`rs_helpdb`、`rs_helpclass`、`rs_helperror` ストアド・プロシージャを使用して、Replication Server エラー・クラスについての情報を表示します。

『Replication Server リファレンス・マニュアル』の「Adaptive Server ストアド・プロシージャ」を参照してください。

## Replication Server システム・データベースの変更

Replication Server のエラー処理をサポートするために、Replication Server システム・データベース (RSSD) 内の 2 つのシステム・テーブルが変更されました

| システム・テーブル                 | 説明                                                                                     |
|---------------------------|----------------------------------------------------------------------------------------|
| <code>rs_classes</code>   | <code>classtype</code> カラムには Replication Server エラー・クラス用に新しい値 "R" が含まれています。            |
| <code>rs_databases</code> | <code>rs_errorclassid</code> はそのデータベースに関連付けられている Replication Server エラー・クラス用の新しいカラムです。 |

## SQL 文の複写

Replication Server 15.2 では、ログベースの複写を補完し、バッチ・ジョブによるパフォーマンスの低下に対処する SQL 文の複写をサポートしています。

SQL 文の複写では、Replication Server は、個々のローの変更ではなく、プライマリ・データを変更した SQL 文をトランザクション・ログから受け取ります。

Replication Server は、SQL 文をレプリケート・サイトに適用します。RepAgent は、SQL データ操作言語 (DML: Data Manipulation Language) と個々のローの変更の両方を送信します。設定に応じて、Replication Server が、個々のローの変更によるログの複写または SQL 文の複写のどちらかを選択します。

SQL 文の複写には、複写後に、変更されたローの数がプライマリ・データベースとレプリケート・データベースで一致していることを確認するためのロー・カウントの検証が含まれます。ローの数が一致しない場合は、Replication Server でこのエラーを処理する方法を指定できます。

SQL 文の複写の詳細については、『Adaptive Server Enterprise 15.0.3 新機能ガイド』を参照してください。

#### 製品および混合バージョンの要件

SQL 文の複製には、Adaptive Server バージョン 15.0.3 以降、プライマリおよびレプリケート Replication Server バージョン 15.2 以降、およびルート・バージョン 15.2 以降が必要です。

## SQL 文の複製の有効化

SQL 文を複製できるように Replication Server サーバとプライマリ・データベースを設定します。

1. SQLDML のログを記録するようにプライマリ・データベースを設定する。
2. SQLDML を複製するように Replication Server を設定する。
  - a) テーブルと Multi-Site Availability (MSA) の複製のための SQLDML の複製定義を作成する。
  - b) Replication Server で、**WS\_SQLDML\_REPLICATION** パラメータを on にしてウォーム・スタンバイ複製を設定する。

## システム設定の変更

いくつかの Adaptive Server ストアド・プロシージャで、SQL 文の複製がサポートされます。

### データベース・レベルの SQL 文複製

SQL 文の複製をサポートするための **sp\_setrepdbmode** が追加されました。

**sp\_setrepdbmode** を使用することで、特定の DML オペレーションについて SQL 文の複製を有効にできます。

SQL 文の複製に該当する DML オペレーションには、次のものがあります。

- **U** – update
- **D** – delete
- **I** – insert select
- **S** – select into

データベースの複製モードを **UDIS** の任意の組み合わせに設定すると、RepAgent は、個々のログ・レコードと Replication Server が SQL 文を作成するために必要な情報の両方を送信します。

たとえば、SQL 文の複製として **delete** 文を複製し、**select into** の複製も有効にするには、次のように入力します。

```
sp_setrepdbmode pdb, 'DS', 'on'
```

SQL 文の複写をデータベース・レベルで設定できるのは、**sp\_reptostandby** が **ALL** または **L1** に設定され、データベースが複写されるようにマーク付けされている場合のみです。

『Adaptive Server Enterprise 15.0.3 新機能ガイド』の「システムの変更」の章にある「**sp\_setrepdbmode**」を参照してください。

### テーブル・レベルの SQL 文複写

**sp\_setrepdefmode** は SQL 文の複写をサポートするように強化されました。

**sp\_setrepdefmode** には次のオプションが含まれています。

- 特定の DML オペレーションについての SQL 文の複写の有効化または無効化。
- スレッシュホールドの設定。SQL 文の複写をアクティブにするには、このスレッシュホールドに達する必要がある。

SQL 文の複写に該当する DML オペレーションには、次のものがあります。

- **U** – update
- **D** – delete
- **I** – insert select

テーブルの複写モードを **UDI** の任意の組み合わせに設定すると、RepAgent は、指定された DML オペレーションでの SQL 文の複写を有効にするための追加の情報を送信します。

たとえば、テーブル *t* での **update**、**delete**、**insert select** オペレーションの SQL 文の複写を有効にするには、次のように入力します。

```
sp_setrepdefmode t, 'UDI', 'on'  
go
```

『Adaptive Server Enterprise 15.0.3 新機能ガイド』の「システムの変更」の章にある「**sp\_setrepdefmode**」を参照してください。

### セッション・レベルの SQL 文複写

複写モードを SQL 文の複写に設定するには、セッション・オプション **set repmode** を使用します。

セッション・レベルの設定を指定できるのは、ログインするとき、またはバッチ・ジョブの開始時です。セッション・レベルの設定はデータベース・レベルおよびオブジェクト・レベルの設定よりも優先されます。

セッション中、指定された DML オペレーションについて SQL 文の複写を有効にするには、**set repmode on** を使用します。セッション・レベルで、すべての SQL 文の複写設定を解除するには、**set repmode off** を使用します。たとえば、セッショ

ン中、**select into** と **delete** のみを SQL 文として複製するには、次のように入力します。

```
set repmode on 'DS'
```

『Adaptive Server Enterprise 15.0.3 新機能ガイド』の「システムの変更」の「**set repmode**」を参照してください。

## SQL 文の複製設定

データベース・レベルとテーブル・レベルで複製オプションを変更できます。

### データベース複製定義

**replicate** 句を コマンドまたは **replication definition** コマンドで使用して、MSA (multisite availability) 環境で SQL 文を複製します。

### 構文

次のコード・セグメントは、データベース複製定義の **create** および **alter** の構文を示しています。

```
[[not] replicate setname [in (table list)] ]
```

構文の説明は次のとおりです。

**setname** = DDL | tables | functions | transactions | system procedures | SQLDML | 'options'

### パラメータ

- 'options' – 次の組み合わせ。
  - **U** – update
  - **D** – delete
  - **I** – insert select
  - **S** – select into
- **SQLDML** – **U**, **D**, **I**, **S** 文の組み合わせとして定義される。

### 例

- **例 1** – 'options' パラメータを使用して、テーブル *tb1* と *tb2* の SQLDML を複製します。

```
replicate 'UDIS' in (tb1,tb2)
```

- **例 2** – **SQLDML** パラメータを使用して、前の例の 'options' パラメータと同じ結果を出します。

```
replicate SQLDML in (tbl, tb2)
```

- **例 3** – すべてのテーブルの **select into** 文を除外します。2 つ目の句 **not replicate 'U' in (T)** は、テーブル *T* での **update** を除外します。

```
create database replication definition dbrepdef
  with primary at ds1.pdb1
  not replicate 'S'
  not replicate 'U' in (T)
go
```

- **例 4** – **replicate 'UD'** 句を使用して、すべてのテーブルで **update** 文と **delete** 文を有効にします。

```
create database replication definition dbrepdef_UD
  with primary at ds2.pdb1
  replicate 'UD'
go
```

- **例 5** – **update** 文と **delete** 文をテーブル *tbl1* と *tbl2* に適用します。

```
alter database replication definition dbrepdef
  with primary at ds1.pdb1
  replicate 'UD' in (tbl1, tbl2)
go
```

## 使用法

- **create database replication** 定義内で複数の **replicate** 句を使用できます。ただし、**alter database replication** 定義で使用できる句は 1 つのみです。
- 複写定義でフィルタを指定しなければ、デフォルトは **not replicate** 句になります。**alter database replication definition** を適用すると SQLDML フィルタを変更できます。**replicate** 句では、1 つまたは複数の SQLDML フィルタを指定できます。
- 複数の句を使用して、同じ定義で 1 つのテーブルを複数回指定できます。ただし、**U**、**D**、**I**、**S** は、それぞれ 1 つの定義につき 1 回しか使用できません。次に例を示します。

```
create database replication definition dbrepdef
  with primary at ds2.pdb1
  replicate tables in (tbl1, tbl2)
  replicate 'U' in (tbl1)
  replicate 'I' in (tbl1, tbl2)
go
```

## テーブル複写定義

**replicate SQLDML** 句を **create** テーブル複写定義に含めて SQL 文の複写をサポートします。

## 構文

次のコード・セグメントは、**create** テーブル複写定義の構文を示します。

```
[replicate {SQLDML ['off'] | 'options' }]
```

## パラメータ

- **'options'** – 次の文の組み合わせ。
  - **U** – update
  - **D** – delete
  - **I** – insert select

## 例

- **例 1** – テーブルに対する **create replication definition** の例を示します。

```
create replication definition repdef1
  with primary at ds3.pdb1
  with all tables named 'tb1'

  (id_col int,
   str_col char(40))

  primary key (id_col)
  replicate all columns
  replicate 'UD'

go
```

## 使用法

- 複写定義に **[replicate {minimal | all} columns]** 句がある場合、**[replicate {minimal | all} columns]** 句は必ず **[replicate {SQLDML ['off'] | 'options'}]** 句の前に置きます。
- **send standby** 句を持つテーブル複写定義では **replicate 'I'** 文を指定できます。**insert select** 文を SQL 複写文として複写できるのは、ウォーム・スタンバイまたは MSA 環境のみです。**send standby** 句を持たないテーブル複写定義では **insert select** 文を複写できません。

### SQL 文の複写の制限

SQL 文の複写ではオートコレクションを実行できません。また、SQL 文の複写がサポートされない場合もあります。

次の場合、SQL 文の複写はサポートされません。

- レプリケート・データベースに、プライマリ・データベースとは異なるテーブル・スキーマがある。
- Replication Server が、データまたはスキーマの変換を実行する必要がある。
- サブスクリプションまたはアーティクルに **where** 句が含まれている。
- *text* または *image* のカラムが含まれている更新。
- ファンクション文字列 *rs\_delete*、*rs\_insert*、*rs\_update* がカスタマイズされている。

#### オートコレクションのサポート

SQL 文の複写ではオートコレクションを実行できません。データ・サーバ・インタフェース (DSI) で、SQL 文の複写対象の DML コマンドが検出され、オートコレクションがデフォルトで on になっている場合、DSI がサスペンドされ、複写が停止します。Replication Server でこのエラーを処理する方法を指定するには、エラー番号 5193 で **assign action** コマンドを使用します。

テーブル・レベルのサブスクリプションが確定化されるまで、Replication Server は SQLDML を複写しません。

### SQL 文の複写に対するロー・カウントの検証

Replication Server 15.2 では、SQL 文の複写中に発生する可能性のある SQLDML のロー・カウント・エラーに Replication Server が対応する方法を指定できます。

SQLDML のロー・カウント・エラーは、SQL 文の複写後に、変更されたローの数がプライマリ・データベースとレプリケート・データベースで一致しない場合に発生します。デフォルトのエラー・アクションは、複写の停止です。SQLDML のロー・カウント・エラーのその他のエラー・アクションを指定するには、Replication Server のエラー・クラスのプライマリ・サイトで **assign action** コマンドを使用できます。

#### 参照：

- 非 SQL 文の複写ロー・カウントの検証 (161 ページ)

## エラー・アクションを割り当てるコマンド

エラー・アクションを指定するには、Replication Server エラー・クラスのプライマリ・サイトで **assign action** コマンドを使用します。

### 構文

```
assign action
  {ignore | warn | retry_log | log | retry_stop | stop_replication}
  error_class
  to server_error1 [, server_error2]...
```

### パラメータ

- **error\_class** – アクションを割り当てるエラー・クラスの名前。Replication Server 15.2 では、デフォルトの **rs\_repserver\_error\_class** エラー・クラスなどの Replication Server のエラー・クラスを指定できる。
- **server\_error** – エラーの番号。

Replication Server では、SQL 文の複製に関係しないエラー・アクションにエラー番号を指定できます。

### 例

- **例 1** – Replication Server でエラー番号 5185 が発生した場合に **ignore** エラー・アクションを割り当てます。

```
assign action ignore for rs_repserver_error_class to 5185
```

- **例 2** – Replication Server でエラー番号 5186 が発生した場合に **warn** エラー・アクションを割り当てます。

```
assign action warn for rs_repserver_error_class to 5186
```

次に示すのは、ロー・カウント・エラーが発生した場合に表示されるエラー・メッセージの例です。

```
DSI_SQLDML_ROW_COUNT_INVALID 5186
Row count mismatch for SQLDML command executed on
'mydataserver.mydatabase'.
The command impacted 1000 rows but it should impact 1500 rows.
```

### 使用法

『Replication Server リファレンス・マニュアル』の「Replication Server コマンド」の「**assign action**」と表 3-17 を参照してください。



SQL 文複写のエラー番号

SQL 文の複写に関するエラー・アクションに対する Replication Server のエラー番号を次に示します。

表 22 : SQL 文複写に対するエラー・アクション

| server_error | エラー・メッセージ                                                                                                                                                                                     | デフォルトのエラー・アクション  | 説明                                                                                      |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|-----------------------------------------------------------------------------------------|
| 5186         | Row count mismatch for the command executed on 'dataserver.database'. The command impacted x rows but it should impact y rows.                                                                | stop_replication | 影響を受けたローの数が想定された数と異なる場合の SQL 文の複写におけるロー・カウントの検証エラー。                                     |
| 5193         | You cannot enable autocorrection if SQL Statement Replication is enabled. Either enable SQL Statement Replication only or disable SQL Statement Replication before you enable autocorrection. | stop_replication | SQL 文の複写が有効になっている場合、オートコレクションを有効にできない。オートコレクションを有効にする前に、SQL 文の複写を有効にするか、SQL 文の複写を無効にする。 |

SQL の複写でのウォーム・スタンバイ・データベースの設定

デフォルトでは、ウォーム・スタンバイ・アプリケーションは、SQL 文の複写をサポートする DML コマンドを複写しません。SQL の複写を使用するには追加の設定が必要です。

SQL の複写を使用するには、以下を行います。

- **replicate SQLDML** 句と **send standby** 句を使ってテーブル複写定義を作成する。
- **WS\_SQLDML\_REPLICATION** パラメータを on に設定する。デフォルト値は **UDIS**。ただし、**WS\_SQLDML\_REPLICATION** の優先度は SQL 複写のテーブル複写定義よりも低くなる。テーブル複写定義にテーブルの **send standby** 句が含まれてい

る場合、**WS\_SQLDML\_REPLICATION** パラメータの設定にかかわらず、その句によって、DML 文を複製するかどうかが決まる。

## SQL の複製でのウォーム・スタンバイ・データベースの設定

デフォルトでは、ウォーム・スタンバイ・アプリケーションは、SQL 文の複製をサポートする DML コマンドを複製しません。SQL の複製を使用するには追加の設定が必要です。

SQL の複製を使用するには次のいずれかを実行します。

- **replicate SQLDML** 句と **send standby** 句を使用して、テーブル複製定義を作成する。
- **WS\_SQLDML\_REPLICATION** パラメータを on に設定する。デフォルト値は **UDIS**。ただし、**WS\_SQLDML\_REPLICATION** の優先度は SQL 複製のテーブル複製定義よりも低くなる。テーブル複製定義にテーブルの **sendstandby** 句が含まれている場合、**WS\_SQLDML\_REPLICATION** パラメータの設定にかかわらず、その句によって、DML 文を複製するかどうかが決まる。

## Replication Server システム・データベースの変更

SQL 文複製をサポートするために、Replication Server システム・データベース (RSSD) 内のシステム・テーブル *rs\_dbreps*、*rs\_dbsubsets*、*rs\_objects* が変更されました。

| システム・テーブル           | 説明                                                                                                                                                                                                                   |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>rs_dbreps</i>    | <i>status</i> カラムに、それぞれが DML フィルタに対応する 2 ビット・セットの新しいセットが 4 つ含まれる。セットの最初のビットは空のフィルタかどうかを示し、2 つ目のビットは否定文のセットかどうかを示す。                                                                                                   |
| <i>rs_dbsubsets</i> | <i>type</i> カラムに 4 つの新しいタイプとして <b>U</b> 、 <b>L</b> 、 <b>I</b> 、 <b>S</b> が含まれる。各タイプは DML <b>UDIS</b> フィルタに対応する。この場合、 <b>D</b> ではなく <b>L</b> が delete に使用される。                                                         |
| <i>rs_objects</i>   | <i>attributes</i> カラムに、新しく 5 ビットが含まれる ( <b>U</b> 、 <b>D</b> 、 <b>I</b> 、または <b>S</b> オペレーションのそれぞれに 1 ビットと、テーブル複製定義に含まれるカラム数が着信データ・ローの数よりも少ないかどうかを示す 1 ビット)。<br>システム・ファンクション複製定義 <b>rs_sqldml</b> も、SQL 文の複製をサポートします。 |

## Adaptive Server 以外のためのエラー・クラスのサポート

Replication Server 15.2 では、Adaptive Server Enterprise 以外 (ASE 以外) のレプリケート・データベースでのエラー・クラスとエラー・アクションのマッピングがサポートされます。

Replication Server 15.2 に含まれるデフォルトの ASE 以外のためのエラー・クラスを使用できます。ASE 以外のレプリケート・データベース用に独自のエラー・クラスを作成および変更することもできます。

ASE 以外のエラー・クラスを、ASE 以外のレプリケート・データベース上の特定の接続に割り当てるには、**create connection** および **alter connection** コマンドを使用できます。

エラー・クラスとエラー処理の詳細については、『Replication Server 管理ガイド第 2 巻』を参照してください。

### ネイティブ・エラー・コード

Replication Server は、ASE 以外のレプリケート・サーバへの接続を確立するときに、接続で ASE 以外のレプリケート・サーバからネイティブ・エラー・コードが返されるオプションが有効になっているかどうかを検証します。オプションが有効になっていない場合、Replication Server は、接続は機能しているが、エラー・アクションのマッピングが正確でない可能性があることを示す警告メッセージをログに記録します。

Enterprise Connect™ Data Access (ECDA) Option for ODBC でレプリケート・サーバ用のオプションを設定するには、Replication Server Options のマニュアルで「**ReturnNativeError**」を参照してください。

### 参照：

- Adaptive Server 以外での複製サポートの機能強化 (177 ページ)

## デフォルトの ASE 以外のためのエラー・クラス

Replication Server 15.2 ではデフォルトの Adaptive Server Enterprise 以外 (ASE 以外) のエラー・クラスが追加されました。これらのデフォルトのエラー・クラスは変更できません。

表 23 : デフォルトの ASE 以外のためのエラー・クラス

| データベース  | クラス名                      |
|---------|---------------------------|
| IBM DB2 | <i>rs_db2_error_class</i> |

| データベース               | クラス名                         |
|----------------------|------------------------------|
| IBM UDB              | <i>rs_udb_error_class</i>    |
| Microsoft SQL Server | <i>rs_mssql_error_class</i>  |
| Oracle               | <i>rs_oracle_error_class</i> |

## create error class コマンドの変更

Replication Server 15.2 では、**create error class** コマンドに **set template to** オプションが含まれています。

### 構文

```
create error class error_class
[set template to template_error_class]
```

### 例

- **例 1** – **rs\_oracle\_error\_class** をテンプレートにして、Oracle データベースに対する **my\_error\_class** エラー・クラスを作成します。

```
create error class my_error_class set template to
rs_oracle_error_class
```

### 使用法

独自のエラー・クラスを作成するには、**create error class** と **set template to**、およびテンプレートとしてもう 1 つ別のエラー・クラスを使用します。**create error class** はテンプレート・エラー・クラスのエラー・アクションを新しいエラー・クラスにコピーします。『Replication Server リファレンス・マニュアル』を参照してください。

## alter error class コマンドの変更

Replication Server 15.2 では、**alter error class** コマンドに **set template to** オプションが含まれています。

### 構文

```
alter error class error_class
set template to template_error_class
```

### 例

- **例 1** – テンプレートとして使用する **rs\_sqlserver\_error\_class** に基づいて Oracle データベースに対する **my\_error\_class** を変更します。

```
alter error class my_error_class set template to  
rs_sqlserver_error_class
```

## 使用法

**alter error class** コマンドと、テンプレートとしての他のエラー・クラスを使用し、エラー・クラスを変更します。**alter error class** は、テンプレートのエラー・クラスから変更対象のエラー・クラスにエラー・アクションをコピーし、同じエラー・コードを持つエラー・アクションを上書きします。『Replication Server リファレンス・マニュアル』を参照してください。

## Adaptive Server 以外での複製サポートの機能強化

Replication Server 15.2 には、インストール、設定、アクティブにサポートされている Adaptive Server Enterprise 以外 (ASE 以外) のデータ・サーバを含む複製環境の設定の全体的な利便性について、機能強化がいくつか組み込まれています。

これらの機能強化により、アクティブにサポートされている ASE 以外のデータ・サーバが関連する複製を迅速に実行できるようにする、事前設定された Replication Server 環境が提供され、インストールおよび設定プロセスが自動化されます。

アクティブにサポートされているデータ・サーバとは、データ・サーバがプライマリまたはレプリケート・データ・サーバとして機能するために必要なすべてのソフトウェア、マニュアル、サポートを Sybase が提供しているデータ・サーバです。アクティブにサポートされている ASE 以外のデータ・サーバのリストについては、使用しているプラットフォーム用の『Replication Agent リリース・ノート』を参照してください。

ASE 以外のデータ・サーバのサポートの詳細については、『Replication Server 異機種間複製ガイド』および『Replication Server 管理ガイド第 1 巻』を参照してください。

### 参照：

- Adaptive Server 以外のためのエラー・クラスのサポート (175 ページ)

## 単純化されたインストールと設定

Replication Server 15.2 では、異機種 (ASE 以外) のデータ型をサポートするために、データ型定義、ファンクション文字列、クラス・レベル変換をインストールするためのスクリプトを編集、実行する必要はありません。

スクリプトによって提供されるファンクションは、Replication Server 15.2 のインストールの一部として、または Replication Server 15.2 とともにインストールされる接続プロファイルに含まれています。これらの機能強化により、ASE 以外の環境の

インストールと設定が簡略化されます。使用しているプラットフォーム用の『Replication Server 15.2 設定ガイド』の「第 8 章 ASE 以外のサポート機能のインストールと実装」に示されている簡略化された設定手順に従ってください。

### 接続プロファイル

Replication Server 15.2 では、アクティブにサポートされている ASE 以外の各種データ・サーバに関連する接続の設定とレプリケート・データベース・オブジェクトの定義を含む接続プロファイルを使用できます。接続プロファイルにより、インストールするファンクション文字列クラス、エラー・クラス、クラス・レベル変換を指定します。

これらの接続プロファイルと単純な構文を使用すると、Adaptive Server Enterprise、IBM DB2、Microsoft SQL Server、Oracle など、アクティブにサポートされているデータ・サーバ間の接続を作成できます。Replication Server は、接続プロファイルを使用して接続を設定し、レプリケート・データベース・オブジェクトを作成します。

また、接続プロファイルのオプションを使用して、コマンドをバッチ処理するかどうかなどのその他のアクションや、使用するコマンド・セパレータを指定できます。

---

**注意：** 接続プロファイルを使用して接続を作成するときに、システム・テーブル・サービス (STS: System Table Services) キャッシュがリフレッシュされるため、Replication Server を再起動する必要はありません。

---

ASE 以外のデータ・サーバのサポートの詳細については、『Replication Server 異機種間複写ガイド』および『Replication Server 管理ガイド第 1 巻』を参照してください。更新された設定プロセスについては、使用しているプラットフォームの『Replication Server 設定ガイド』を参照してください。

### using profile 句

接続プロファイルを使用して ASE 以外のデータベースと Adaptive Server 間の接続を作成するには、**using profile** 句を **create connection** コマンドで使用します。

### 構文

以下は、**using profile** 句と **display\_only** 句を示す **create connection** 構文の一部です。

```
create connection to data_server.database
using profile connection_profile;version
set username [to] user
[other_create_connection_options]]
[display_only]
```

## パラメータ

- **connection\_profile** – コネクションの設定、Replication Server システム・データベース (RSSD) の変更、レプリケート・データベース・オブジェクトの作成に使用する接続プロファイルを指定する。
- **version** – 接続プロファイルの特定のバージョンを指定する。
- **other\_create\_connection\_options** – *other\_create\_connection\_options* を使用して、プロファイルで指定されない接続オプション (パスワードの設定など) の設定、またはプロファイルで指定されているオプション (Replication Server に用意されているファンクション文字列クラスをオーバーライドするカスタム・ファンクション文字列クラスの指定など) のオーバーライドを行う。**create connection** で使用できるすべてのパラメータについては、『Replication Server リファレンス・マニュアル』の「Replication Server コマンド」の「**create connection**」を参照。
- **display\_only** – 接続プロファイルとともに使用し、コマンドと、コマンドが実行されるサーバを表示する (ただし、コマンドは実行しない)。**display\_only** を使用した結果を確認するには、クライアント・ログおよび Replication Server ログを使用する。

## 例

- **例 1** – Oracle レプリケート・データベースに対するコネクションを作成します。

```
create connection to oracle.instance
using profile rs_ase_to_oracle;standard
set username to ora_maint
set password to ora_maint_pwd
```

- **例 2** – プライマリ・データベースでもある Microsoft SQL Server レプリケート・データベースに対するコネクションを作成します。この例では、接続プロファイルによって提供されるエラー・クラス設定を、このコマンドが別の **my\_msss\_error\_class** エラー・クラスに置き換えます。

```
create connection to msss_server.msss_db
using profile rs_ase_to_msss;standard
set username to msss_maint
set password to msss_maint_pwd
set error class to my_msss_error_class
with log transfer on
```

- **例 3** – プロファイルの特定のバージョン v9\_1 を使用して、DB2 レプリケート・データベースに対するコネクションを作成します。この例では、接続プロファイルによって提供されているコマンド・バッチのサイズが、このコマンドにより新しい値 16384 で上書きされます。

```
create connection to db2.subsys
using profile rs_ase_to_db2;v9_1
set username to db2_maint
```

```
set password to db2_maint_pwd  
set dsi_cmd_batch_size to '16384'
```

- **例 4 – display\_only** オプションを使用して、特定のプロファイルを使用した場合に実行されるコマンドを表示します。コマンドと画面に表示されるコマンド出力は、Replication Server のログにも書き込まれます。

```
create connection to oracle.instance  
using profile rs_ase_to_oracle;standard  
set username to ora_maint  
set password to ora_maint_pwd  
display_only
```

```
go
```

### 使用法

**create connection** の詳細については、『Replication Server リファレンス・マニュアル』を参照してください。

### 使用できる接続プロファイルをリスト表示するコマンド

Replication Server で定義されている各プロファイルのプロファイル名、バージョン、コメントをリスト表示するには、**admin show\_connection\_profiles** コマンドを使用します。

Adaptive Server と Oracle、Oracle と Adaptive Server、IBM DB2 と /Microsoft SQL Server などの、プライマリ・データベースとレプリケート・データベースの組み合わせごとに接続プロファイルが存在します。

**admin show\_connection\_profiles** の詳細と接続プロファイルのリストについては、『Replication Server リファレンス・マニュアル』の「Replication Server コマンド」の「**admin show\_connection\_profiles**」を参照してください。

### 構文

```
admin show_connection_profiles[, "match_string"]
```

### パラメータ

- **match\_string** – *match\_string* オプションを使用すると、オプションで指定した文字列が名前に含まれている接続プロファイルのみが表示される。

### 例

- **例 1** – Replication Server で現在定義されているすべての接続プロファイルの名前をリストします。

```
admin show_connection_profiles  
go
```



出力の抜粋を以下に示します。

```

Profile Name          Version      Comments
-----
rs_ase_to_db2        Standard    Standard ASE to DB2
replication
                    connection profile.
rs_ase_to_udb        Standard    Standard ASE to DB2
replication
                    connection profile.
...
rs_db2_to_ase        Standard    Standard DB2 to ASE
replication
                    connection profile.
rs_db2_to_msss       Standard    Standard DB2 to Microsoft
SQLServer
                    connection profile.
...

```

- **例 2** – 接続プロファイル名に文字列 "oracle" を含み、Replication Server で現在定義されている、すべての接続プロファイルの名前をリスト表示します。

```

admin show_connection_profiles, "oracle"
go

```

出力を以下に示します。

```

Profile Name          Version      Comments
-----
rs_ase_to_oracle     Standard    Standard ASE to Oracle
replication
                    connection profile.

```

### 接続プロファイルのシステム・テーブル

接続プロファイルは *rs\_profile* および *rs\_profdetail* システム・テーブルでサポートされています。

| システム・テーブル            | 説明                                        |
|----------------------|-------------------------------------------|
| <i>rs_profile</i>    | 現在定義されている Replication Server プロファイルを格納する。 |
| <i>rs_profdetail</i> | プロファイルの詳細を格納する。                           |



# Replication Server バージョン 15.1 の新機能

Replication Server 15.1 ではいくつかの新機能と機能強化が導入されています。その中には、動的 SQL、関数の複写、モニタおよびカウンタ、データ型サポート、テーブル・キューの管理、パスワードの暗号化、timestamp データ型のサポート、ダンプ・トランザクションなどの機能強化が含まれています。新機能には、Adaptive Server 共有ディスク・クラスタ、Adaptive Server 整数 ID、LOB データ型の部分更新、ディストリビュータ・ステータスの記録が含まれています。

## 動的 SQL の機能強化

---

動的 SQL では Oracle、Universal Database (UDB)、DB2、Microsoft SQL を含む異機種データベース間の複写がサポートされるようになりました。

Replication Server の動的 SQL により、複写パフォーマンスが強化されます。これは、Replication Server データ・サーバ・インタフェース (DSI) を使用して、ターゲット・ユーザ・データベースで動的 SQL 文を作成し、繰り返し実行できるようにすることで実現されます。**create/alter replication definition** コマンドでは、複写定義を介して各テーブルで動的 SQL の適用の制御が可能です。**create/alter replication definition** コマンドの詳細については、『Replication Server リファレンス・マニュアル』を参照してください。

次のコマンドを使用することにより、特定のレプリケート・データベースに対する動的 SQL の実行をテーブル・レベルで変更できます。

```
set dynamic_sql {on | off}  
for replication definition with replicate at  
data_server.database
```

動的 SQL の使用状況を確認するには、**stats\_sampling** を有効にして **admin stats, dsi** コマンドを実行してから、DSIEDsqlPrepared、DSIEDsqlExecuted、その他の動的 SQL 関連のカウンタを探します。

各複写定義の動的 SQL 設定を表示するには、ストアド・プロシージャ **rs\_helprep**、**rs\_helpsub**、**rs\_helppubsub** を使用します。

これらのストアド・プロシージャの使用に関する情報については、『Replication Server リファレンス・マニュアル』の「Adaptive Server ストアド・プロシージャ」で「**rs\_helprep**」、「**rs\_helpsub**」、「**rs\_helppubsub**」を参照してください。

### 制限事項

動的 SQL コマンドは、Sybase の範囲内のデータをサポートしています。Sybase の範囲外のデータによって動的 SQL が失敗すると、DSI は、エラー・メッセージをログに記録し、言語コマンドを使用して動的 SQL を再送信します。言語コマンドも失敗した場合のみ、DSI が停止します。

この状態が頻繁に発生する場合は、テーブル複写定義によって動的 SQL を無効にするか、**set dynamic\_sql off** コマンドを使用します。

**dynamic\_sql off** には次のコマンドのいずれかを使用します。

- **alter connection... set dynamic\_sql off** — この接続に複写されるすべてのコマンドに対して動的 SQL を無効にする。
- **create/alter replication definition...without dynamic\_sql** — この複写定義を使用するすべてのコマンドに対して動的 SQL を無効にする。
- **set dynamic\_sql off for replication definition with replicate at...** — このレプリケート・接続でこの複写定義を使用するすべてのコマンドに対して動的 SQL を無効にする。

## ファンクション複写の機能強化

---

Replication Server 15.1 では、ファンクション名とは異なる名前のファンクション複写定義を作成できます。

Replication Server は、要求ファンクション複写定義に対して異なるプライマリ・ファンクション名とレプリケート・ファンクション名を強制します。適用されるファンクション複写定義によってファンクションが複写される場合は、**maint\_user** がレプリケート・データベースでトランザクションを実行します。レプリケート・データベースでファンクション複写要求定義によってファンクションが複写される場合は、**origin\_user** がトランザクションを実行します。

これらの機能強化により、次の作業が可能になります。

- 同じ名前の複数のファンクションを異なるデータベースから複写する。
- 1つのプライマリ・ファンクションに対して複数の複写定義を作成し、各複写定義で、異なるレプリケート・サイトに異なるレプリケート・ファンクションを指定する。

ファンクション複写定義を管理するには、次のコマンドを使用します。

- **create applied function replication definition**
- **create request function replication definition**
- **alter applied function replication definition**

- **alter request function replication definition**

#### 混合バージョンのサポート

この機能強化により、混合バージョン環境がサポートされます。ただし、異なるプライマリ・ファンクション名と複写定義名を持つファンクション複写定義は、15.1 より前のバージョンの Replication Server に複写されません。

---

**警告！** システムに古いバージョンの要求ファンクション複写定義がある場合は、その古いバージョンの定義を削除してから、同じプライマリ・ファンクションに対してバージョン 15.1 の複写定義を作成してください。

---

#### ウォーム・スタンバイと *Multi-Site Availability (MSA)* のサポート

ウォーム・スタンバイ環境や MSA 環境では、同じプライマリ・ファンクションのすべてのファンクション複写定義に対してパラメータ・リストが 1 つだけ存在します。1 つのファンクション複写定義を変更してパラメータを追加すると、このファンクションに作成されているすべてのファンクション複写定義に新しいパラメータが追加されます。

これらのコマンドの詳細については、『Replication Server リファレンス・マニュアル』の「Replication Server コマンド」を参照してください。

#### 制限事項

強化されたファンクション複写には、次の制限事項があります。

- 同じファンクションに対して作成されたすべてのファンクション複写定義には、同じ名前とデータ型を持つ同じパラメータ・リストが存在する必要がある。
- 異なるプライマリ・ファンクション名および複写定義名を持つファンクション複写定義をバージョン 15.1 で作成すると、同じプライマリ・ファンクションに対する以前のバージョンの要求ファンクション複写定義が無効になる。
- 1 つのプライマリ・ファンクションに対して、適用ファンクション複写定義と要求ファンクション複写の両方を作成することはできない。**create function replication definition** コマンドの使用によって作成されたファンクション複写定義は、そのファンクションのプライマリ・サイトでは適用ファンクションとみなされます。
- 各適用ファンクション複写定義および要求ファンクション複写定義に対して、ファンクションを複写するための、対応するサブスクリプションを作成する必要がある。

## Adaptive Server 共有ディスク・クラスタのサポート

---

Replication Server と RepAgent スレッドは、どちらも Adaptive Server 共有ディスク・クラスタ環境をサポートしています。共有ディスク・クラスタ環境は、多数の

Adaptive Server が単一のディスクまたはデータベースのセットを共有する場所です。

Sybase 共有ディスク・クラスタでは、複写の送信元または複写の送信先としてデータベースを使用できます。RepAgent の設定や複写対象テーブルへのマーク付けなど、すべてのタスクをクラスタ内の任意のインスタンスから実行できます。

『Replication Server 管理ガイド第 1 巻』の「RepAgent の管理と Adaptive Server のサポート」を参照してください。

## モニタリングとカウンタの機能強化

---

強化されたモニタリング機能とカウンタ機能により、最もアクティブなテーブル、プロシージャ、および関連する統計情報を収集し、その情報を Replication Server システム・データベース (RSSD) の *rs\_statdetail* テーブルおよび関連するテーブルに格納できます。

この情報を使用すると、プライマリ・キーのインデックスが失われたといったレプリケート・データベースのパフォーマンスに関する問題や、Replication Agent™ およびステープル・キュー・トランザクション (SQT) インタフェース/ディストリビュータ (DIST) の処理における遅延時間の問題を診断できます。

## 新しいアクティブ・オブジェクト・カウンタ

---

テーブルとプロシージャに関する文の実行時間をカウントするための新しいカウンタが追加されました。

- AOBJInsertCommand
- AOBJUpdateCommand
- AOBJDeleteCommand
- AOBJWritetextCommand
- AOBJExecuteCommand

アクティブなオブジェクトのカウンタ・メトリックを RSSD にフラッシュするには、次のコマンドのいずれかを実行します。

- **admin stats, "all", save**
- **admin stats, dsi, save**
- **admin stats, sysmon, save**

カウンタに関する情報の表示方法の詳細については、『Replication Server リファレンス・マニュアル』の「rs\_helpcounter」を参照してください。

## 新しいプロシージャ・インタフェース

最もアクティブなテーブルとプロシージャおよび関連する統計情報を引き出すため、Replication Server 15.1 では、ストアド・プロシージャ **rs\_stat\_populate** および **rs\_stat\_genreport** が導入されました。

**rs\_stat\_populate** は *rs\_statdetail* からデータを読み取り、結果をまとめ、補足し、非正規化して、結果を **rs\_statreport** に保存します。一方、**rs\_stat\_genreport** はデータを読み取ってレポートを生成します。

Replication Server 15.1 にアップグレードした後で、次のスクリプトを RSSD にロードします。

```
$SYBASE/$SYBASE_REP/scripts/  
rs_install_statreport_v1510_[ase|asa].sql
```

スクリプトをロードしたら、ストアド・プロシージャ **rs\_stat\_populate** および **rs\_stat\_genreport** を実行します。これらのストアド・プロシージャを実行すると、次の情報が生成されます。

- Replication Server のパフォーマンスの概要 — DIST 処理や DSI 処理など、Replication Server に関する概要情報。
- Replication Server のパフォーマンス分析 — Replication Server の重要なカウンタに基づいたパフォーマンス分析およびチューニングのためのヒント。詳細な説明は、スクリプト・ファイルに記載されている。
- アクティブなオブジェクトの識別結果 — アクティブなテーブルとプロシージャの名前、所有者名、実行時間などのリスト。

**rs\_stat\_populate**、**rs\_stat\_genreport**、**rs\_statreport**、*rs\_statdetail* の詳細については、スクリプト・ファイルを参照してください。

## ステーブル・キューの管理の改善

Replication Server 15.1 では、ステーブル・キューの管理が簡略化されています。

強化されたキュー **dump** コマンドにより、ステーブル・キューの識別、ダンプするステーブル・キューの内容の制御、追加の出力ファイル・オプションのサポートを柔軟に行うことができます。また、Replication Server 15.1 には、ステーブル・キュー・マネージャ (SQM) から特定のトランザクションを削除したりリストアしたりするための新しいコマンドも用意されています。

ステーブル・キューの管理の詳細については、『Replication Server 管理ガイド第 1 巻』を参照してください。次のコマンドの詳細については、『Replication Server リファレンス・マニュアル』を参照してください。

## sysadmin dump\_queue の変更

**sysadmin dump\_queue** の構文は、ステータブル・キューの管理に柔軟性を与えるために変更されました。

**sysadmin dump\_queue** の機能強化は次のとおりです。

- ダンプするステータブル・キューを指定するときに、キュー番号の代わりにサーバ名またはデータベース名を指定するためのオプション
- ダンプするコマンドの数を指定するためのオプション
- トランザクションの開始/終了コマンドだけをダンプする、SQL 文を除いてキュー内のすべての内容をコメントとしてダンプする、などのフィルタ・オプション
- Replication Server ログまたはユーザ定義のログ・ファイルに出力を書き込むためのオプション
- 特定のキューおよびセッションに対して前回の **sysadmin dump\_queue** コマンドが停止した場所からデータ・ダンプを開始するためのオプション

変更された **sysadmin dump\_queue** の構文を次に示します。

```
sysadmin dump_queue {, q_number | server [,database]},
    {q_type,seg, blk, cnt
    [, num_cmds]
    [, {L0 | L1 | L2 | L3}]
    [, {RSSD | client | "log" | file_name}] |
    "next" [, num_cmds]}
```

## sysadmin sqt\_dump\_queue の変更

**sysadmin sqt\_dump\_queue** の構文は、ステータブル・キューの管理に柔軟性を与えるために変更されました。

**sysadmin sqt\_dump\_queue** の機能強化は次のとおりです。

- ダンプするステータブル・キューを指定するときに、キュー番号の代わりにサーバ名またはデータベース名を指定するためのオプション
- コミットされたトランザクションをすべてダンプし、SQT キャッシュにあるトランザクションを読み取るためのオプション
- ダンプするコマンドの数を指定するためのオプション
- トランザクションの開始/終了コマンドだけをダンプする、SQL 文を除いてキュー内のすべての内容をコメントとしてダンプする、などのフィルタ・オプション
- Replication Server ログまたはユーザ定義のログ・ファイルに出力を書き込むためのオプション

変更された **sysadmin sqt\_dump\_queue** の構文を次に示します。

```
sysadmin sqt_dump_queue {, q_number | server [,database]},
    q_type, reader
```



```
[, {open | closed | read}]
[, num_cmds]
[, {L0 | L1 | L2 | L3}]
[, {RSSD | client | "log" | file_name}]
```

## 変更された `resume connection` コマンド

`resume connection skip transaction` オプションは、コネクション・キュー内の指定した数のトランザクションを省略してからコネクションをレジュームできるように強化されました。

省略されたトランザクションは、データベースの例外ログと、Replication Server ログまたは `sysadmin dump_file` コマンドで指定した代替ログ・ファイルに書き込まれます。このコマンドが省略できるトランザクションの最大数は、データ・サーバ・インタフェース (DSI) のアウトバウンド・キュー内のトランザクションの数です。

変更された `resume connection` の構文を次に示します。

```
resume connection to data_server.database
[skip [n] transaction | execute transaction]
```

## 変更された `sysadmin log_first_tran` コマンド

新しいオプション `n` が `sysadmin log_first_tran` コマンドに追加されました。この新しいオプションを使用すると、データベースの例外ログと、Replication Server ログまたは `sysadmin dump_file` コマンドで指定した代替ログ・ファイルに書き込むトランザクションの数を指定できます。

変更された `sysadmin log_first_tran` の構文を次に示します。

```
sysadmin log_first_tran [,n], data_server, database
```

## 新しい `sysadmin sqm_zap_tran` コマンド

`sysadmin sqm_zap_tran` は特定のトランザクションをステابل・キューから削除し、削除されたコマンドの数を示すメッセージを返します。このトランザクションは、ローカル・キュー ID (LQID) で識別できます。

`sysadmin sqm_zap_tran` の構文を次に示します。

```
sysadmin sqm_zap_tran {, q_number, | server [,database]},
q_type, lqid [, {L0 | L1 | L2 | L3}]
[, {RSSD | client | "log" | file_name}]
```

**注意：** このコマンドを使用するには、Replication Server がスタンドアロン・モードになっている必要があります。

## 新しい `sysadmin sqm_unzap_tran` コマンド

`sysadmin sqm_unzap_tran` は特定のトランザクションをステابل・キューにリストアし、リストアされたトランザクション・コマンドの数を示すメッセージを返します。このトランザクションは、LQID で識別できます。

`sysadmin sqm_unzap_tran` の構文を次に示します。

```
sysadmin sqm_unzap_tran {, q_number, | server [,database]},
                        q_type, lqid [, {L0 | L1 | L2 | L3}]
                        [, {RSSD | client | "log" | file_name}]
```

---

**注意：** このコマンドを使用するには、Replication Server がスタンドアロン・モードになっている必要があります。

---

## 新しい `sysadmin dump_tran` コマンド

`sysadmin dump_tran` を使って特定のステابل・キュー・トランザクションの文をログ・ファイルにダンプします。このトランザクションは、LQID で識別できます。

`sysadmin dump_tran` の構文を次に示します。

```
sysadmin dump_tran {[, q_number, | server [,database]},
                    q_type, lqid
                    [, num_cmds]
                    [, {L0 | L1 | L2 | L3}]
                    [, {RSSD | client | "log" | file_name}] |
                    "next" [, num_cmds]}
```

## locales ディレクトリに対する変更

Replication Server のリリース領域とローライゼーション・ディレクトリ構造が変更されました。これで、Replication Server や Adaptive Server など、複数の Sybase 製品を同じコンピュータの同じディレクトリにインストールまたはアンインストールできます。複数バージョンの Replication Server を同じディレクトリにインストールすることもできます。

locales ディレクトリの変更点は次のとおりです。

- Replication Server 固有の locales ファイルが、`$$SYBASE/locales` から新しい `$$SYBASE/$$SYBASE_REP/locales` ディレクトリに移動された。
- すべての `<charset>` サブディレクトリが、新しい `$$SYBASE/$$SYBASE_REP/locales` ディレクトリ内の各言語の `<utf8>` サブディレクトリ内に統合された。

サポートされているすべての言語に対して、異なる文字セットではなく、UTF-8 文字セットが使用される。UTF-8 から他の文字への変換、またはその逆の変換が可能である。

Replication Server は、ローカライゼーション・ファイルからメッセージを読み取り、実行時に、そのメッセージを指定された文字セット・フォーマットに変換する。

- **rs\_init** ロケール・ファイルが \$SYBASE/\$SYBASE\_REP/locales/<language>/utf8/sybinit に移動された。

## 拡張パスワード暗号化のサポート

---

Replication Server 15.1 では、サーバ認証またはクライアント認証には Sybase 共通セキュリティ・インフラストラクチャ (CSI)、RSSD テーブルに格納されているパスワードの暗号化と復号化には暗号法、拡張パスワード暗号化のサポートにはキー・ペアの生成が使用されます。

拡張パスワード暗号化では、接続プロパティ **CS\_SEC\_EXTENDED\_ENCRYPTION** が有効になっている Open Client アプリケーションが Replication Server に接続できるようにする非対称キー暗号化が使用されます。これにより、Replication Server が他のサーバに接続するときに **CS\_SEC\_EXTENDED\_ENCRYPTION** を有効にすることもできます。

非対称キー暗号化では、パスワードの暗号化にはパブリック・キー、パスワードの復号化にはプライベート・キーが使用されます。プライベート・キーはネットワークを介して共有されないため、安全です。

---

**注意：** 拡張パスワード暗号化機能を使用するには、拡張パスワード暗号化をサポートしているサーバ (Adaptive Server 15.0.2 ESD #2 以降など) が必要です。また、この機能は、15.1 リリースの HP Itanium プラットフォームではサポートされません。

---

## rs\_ticket ストアド・プロシージャ・バージョン 2

---

**rs\_ticket** がバージョン 2 になり、Adaptive Server Enterprise 以外 (ASE 以外) のデータベースをサポートするようになりました。

**rs\_ticket** ストアド・プロシージャは、追加の設定や管理を行うことなく使用できます。チケットは、レプリケート・データベース内にある **rs\_ticket\_history** テーブルに自動的に挿入されます。また、チケットを複数のアプリケーション間で共有でき、チケットの発行が他のアプリケーションのチケットに妨害されることはありません。

**rs\_ticket** バージョン2では、利便性を向上するためにより多くのチケット情報が提供されます。たとえば、Replication Server インタフェース (RSI) のタイムスタンプは、複雑なクエリを作成しなくても取得できます。**rs\_ticket\_history** テーブルに対して、Replication Server のパフォーマンスを直接問い合わせることもできます。複数のタイム・ゾーン間でコンピュータの時間またはチケットの同期がとられていない場合は、タイムスタンプ・カラムを変更することでチケットの日付を調整できます。

以前のバージョンの **rs\_ticket** の名前は **rs\_ticket\_v1** に変更されました。以前のバージョンを使用するには、**rs\_ticket\_report** ファンクション文字列を古い内容またはデフォルトの **exec rs\_ticket\_report ?rs\_ticket\_param!param?** で置き換えます。

---

**注意：** 事前に **dsi\_rs\_ticket\_report** を無効にして Replication Server 15.1 にアップグレードした場合は、アップグレード・プロセスの終了後に **dsi\_rs\_ticket\_report** 設定が再度有効になります。

---

**rs\_ticket** のフォーマットは、次のように変更されています。

- バージョン番号が2に変更された (V=2)。チケットのバージョン番号が1である場合、Replication Server はチケットに日付を書き込まない。
- チケット・サイズが255バイトから1024バイトに増えた。
- *timestamp* のフォーマットが変更され、日付が含まれるようになった。新しい *timestamp* のフォーマットは mm/dd/yy hh:mm:ss:mmm。
- RSI メッセージを解析するように RSI 送信元に指示する Replication Server インタフェース (RSI) モジュール・タイムスタンプが追加された。チケットが複数の RSI モジュールを渡すようになった。ただし、**rs\_ticket\_history** テーブルは最新の RSI タイムスタンプのみを維持する。
- チケットの送信元と送信先を識別できるようにするため、プライマリ Replication Server 名とターゲット Replication Server 名が追加された。
- プライマリ・データベース名とレプリケート・データベース名が追加された。
- 次の2つの DSI カウンタが追加された。
  - DSI\_T=xxx - データ・サーバ・インタフェース (DSI) が読み取るトランザクションの総数
  - DSI\_C=xxx - DSI が読み取るコマンドの総数

**rs\_ticket** バージョン2の使用方法の詳細については、『Replication Server リファレンス・マニュアル』を参照してください。

## Replication Server の新しいカウンタ

REPAGENT、RSIUSER、SQM、DSI、DSIEXEC のための新しいカウンタが追加されました。

| カウンタ                    | 説明                                                                                           |
|-------------------------|----------------------------------------------------------------------------------------------|
| RepAgentExecTime        | Open Client/Server™ によって RepAgent ユーザ・スレッドがスケジュールされているミリ秒単位の時間                               |
| RSIUExecTime            | Open Client/Server によって RSI ユーザ・スレッドがスケジュールされているミリ秒単位の時間                                     |
| SQMWaitSegTime          | セグメントの割り付けを待機している時間                                                                          |
| DSINoDsqlNULL           | <b>where</b> 句に NULL 値が含まれるために動的 SQL 文を使用できないコマンドの数                                          |
| DSINoDsqlDatatype       | <i>text</i> 、 <i>image</i> 、 <i>java</i> 、または不適格なユーザ定義データ型 (UDD) であるために動的 SQL 文を使用できないコマンドの数 |
| DSINoDsqlRepdef         | 複写定義によって動的 SQL から除外されているコマンドの数                                                               |
| DSINoDsqlColumnCount    | パラメータの数が 255 を超えるために動的 SQL から除外されているコマンドの数                                                   |
| DSINoDsqlMissingCols    | 一部のコマンドを DSI で使用できないために動的 SQL から除外されているコマンドの数。これは、最少カラム機能によって発生する可能性がある                      |
| DSIEDsqlPrepared        | データ・サーバ・インタフェース・エグゼキュータ (DSI/E) によってターゲット・データベースで作成された動的 SQL 文                               |
| DSIEDsqlDealloc         | DSI/E によってターゲット・データベースで割り付けが解除された動的 SQL 文                                                    |
| DSIEDsqlExecuted        | DSI/E によってターゲット・データベースで実行された動的 SQL 文                                                         |
| DSIEDsqlDeallocSchema   | スキーマ変更のために DSI/E によってレプリケート・データベースで割り付けが解除された動的 SQL 文                                        |
| DSIEDsqlDeallocExecFail | レプリケート・データベースで割り付けが解除された動的 SQL 文                                                             |

カウンタに関する情報を取得するためのコマンドについては、『Replication Server リファレンス・マニュアル』の「Adaptive Server ストアド・プロシージャ」の「**rs\_helpcounter**」を参照してください。

## ラージ・オブジェクト・データ型の拡張サポート

Replication Server 15.1 では、Microsoft SQL Server 2005 のデータ型 *varchar(max)*、*nvarchar(max)*、*varbinary(max)* の複写をサポートしています。これらのデータ型は、それぞれ 2,147,483,647 バイトまでのデータを格納できます。

Replication Server では、テーブル・レベルの複写環境でのユーザ定義データ型 (UDD) としてラージ・オブジェクト (LOB) データ型が導入されています。また、Replication Server は、新しい LOB 型に対してデータベースレベルの複写をサポートしています。新しい LOB データ型は *text*、*unitext*、*image* データ型に直接マッピングされます。

基本となる UDD の型は次のとおりです。

| 新しい LOB データ型          | 基本となる型         |
|-----------------------|----------------|
| <i>varchar(max)</i>   | <i>text</i>    |
| <i>nvarchar(max)</i>  | <i>unitext</i> |
| <i>varbinary(max)</i> | <i>image</i>   |

新しい LOB データ型の詳細については、『Replication Server リファレンス・マニュアル』を参照してください。

### 制限事項

新しい LOB データ型の制限事項は次のとおりです。

- テーブル複写定義で LOB カラムをプライマリ・キーとして定義できない。
- テーブル複写定義またはファンクション複写定義で LOB カラムをサーチャブルとして定義できない。
- 新しい LOB データ型のいずれかをパラメータとして含むストアド・プロシージャを複写できない。
- テキスト・ポインタを使用して、新しい LOB データ型のデータを操作できない。

### 混合バージョンのサポート

混合バージョン環境では、プライマリ Replication Server とレプリケート Replication Server のサイト・バージョンが 15.1、LTL バージョンが 710 であることが必要です。

## ラージ・オブジェクト・データ型の部分更新

---

部分更新トランザクションでは、**delete** コマンドや **replace** コマンドを発行しなくても、テーブル・カラムに文字列を直接挿入したり、既存の文字列を上書きしたりできます。

Replication Server 15.1 では、サポートされているラージ・オブジェクト (LOB) データ型に部分更新トランザクションを複写できます。

部分更新を実装するには、新しい **rs\_updatetext** LTL コマンドを次のように使用します。

```
{[distribute|_ds] command_tags {[applied|_ap] 'table'.rs_updatetext  
{[partialupd|_pu] [{first|_fi}] [last] [{changed|_ch}] [with log]  
[withouttp|_wo]} [{offset|_os}=offset {deletelen|  
_dln}=deletelength]  
[textlen|_tl]=length] text_image_column
```

### 制限事項

部分更新の制限事項は次のとおりです。

- 複数の文字セットの変換がサポートされていない。
- Microsoft SQL Server 2005 だけがサポートされている。

部分更新の詳細については、『Replication Server デザイン・ガイド』を参照してください。

## timestamp の拡張サポート

---

Replication Server 15.1 では、新しいデータ型として *timestamp* が追加されました。*timestamp* データ型を使用すると、レプリケート・データベース、スタンバイ・データベース、MSA データベースに *timestamp* カラムを複写できます。

また、複写定義で *timestamp* をプライマリ・キーとして定義したり、複写定義やファンクション複写定義でサーチャブル・カラムとして定義したりできます。

*timestamp* は *varbinary*(8) として定義され、ステータス・ビット・インジケータによって *varbinary* と区別されます。

*timestamp* の複写をサポートするために、**send\_timestamp\_to\_standby** 設定パラメータが追加されています。**send\_timestamp\_to\_standby** が有効であり、複写定義が存在しない場合は、*timestamp* カラムはレプリケート・データベースに送信されません。

---

**注意：** 複写定義で *timestamp* をサポートするには、レプリケート Adaptive Server がバージョン 15.0.2 以降であることが必要です。

---

*timestamp* データ型の詳細については、『Replication Server リファレンス・マニュアル』を参照してください。*timestamp* カラムの複写については、『Replication Server 管理ガイド第 1 巻』を参照してください。

## 新しい *opaque* データ型

---

*opaque* データ型は、Replication Server がサポートしていないデータ型の複写を処理します。

RepAgent は、ターゲット・データベースで直接適用できるフォーマット・データを提供します。また、*opaque* データ型は、*anydata* データ型や Microsoft SQL Server の *sql\_variant* データ型など、未指定の値や一貫性のない値を格納できるデータ型の複写を処理します。

### 制限事項

*opaque* データ型の制限事項は次のとおりです。

- 複写定義、サブスクリプション、アーティクルのサーチャブル・カラムと **where** 句で *opaque* データ型を使用できない。
- **map to** 句で *opaque* データ型を使用できない。
- 複写定義に *opaque* データ型のカラムまたはパラメータが存在する場合は、動的 SQL を使用できない。
- ファンクション文字列にリモート・プロシージャ・コール (RPC) が含まれる場合は、*opaque* データ型を使用できない。
- 文字セット変換やバイト順序変換を *opaque* データ型に適用できない。

混合バージョン環境では、プライマリ Replication Server とレプリケート Replication Server のサイト・バージョンが 15.1、LTL バージョンが 710 であることが必要です。

*opaque* データ型の詳細については、『Replication Server リファレンス・マニュアル』を参照してください。

## ダンプ・トランザクションの機能強化

---

**with standby\_access** パラメータを持つ **dump transaction** コマンドの複写をサポートするために、ログ転送言語 (LTL) の **dump** サブコマンドと **rs\_dumptran** ファンクション文字列が強化されました。





**dump transaction** の機能強化の詳細については、『Replication Server デザイン・ガイド』を参照してください。

## ディストリビュータ・ステータスの記録

---

Replication Server 15.1 では、ディストリビュータ・スレッドのディストリビュータ (DIST) ステータスを RSSD に保存できます。

DIST スレッドは、インバウンド・キューからトランザクションを読み取り、アウトバウンド・キューに複製トランザクションを書き込みます。DIST スレッドは、Replication Server がプライマリ・データベースに接続するときに作成され、手動で、または Replication Server 設定を使用して、サスペンドまたはレジュームできます。DIST スレッドをサスペンドまたはレジュームすると、スレッドの DIST ステータスが変更されます。

DIST ステータスの記録によって、DIST スレッドは Replication Server が停止した後でもそのステータスを保持できます。

DIST ステータスの記録の詳細については、『Replication Server リファレンス・マニュアル』を参照してください。

## テキスト更新の強化

---

Replication Server では、**text** や **image** などのラージ・オブジェクトを ASE 以外のサーバに複製できます。これは、**writetext** コマンドを DirectConnect Anywhere™ に渡し、このコマンドを **update** 文に変換することによって行われます。

**writetext** コマンドには、レプリケート・データベースを検索して送信するために **update** 文が使用するラージ・オブジェクト・ポインタが含まれています。ほとんどのデータ・サーバは、ラージ・オブジェクトを更新するユニークな実装を独自に備えています。そのため、これらのサーバにラージ・オブジェクトを複製すると、更新ごとにレプリケート・データベースの完全なテーブル・スキャンが必要になることが頻繁にあるため、時間がかかるうえに効率的ではありません。

Replication Server 15.1 には、DirectConnect Anywhere に送信される **writetext** コマンドにプライマリ・キーを含めるためのオプションが用意されています。プライマリ・キーにより、DirectConnect Anywhere は、レプリケート・データベースの検索と複製を効率的に実行できる **update** 文を作成できます。

Replication Server 15.1 には、データ・サーバ・インタフェース (DSI) の設定パラメータ **dsi\_alt\_writetext** が導入されています。テキスト・ポインタまたは一連のプ

ライマリ・キーを **writetext** コマンドに含めるように Replication Server に指示する場合は、**dsi\_alt\_writetext** を使用します。

---

**注意：** この機能を使用するには、ECDA 15.0 ESD #2 が必要です。

---

詳細については、『Replication Server リファレンス・マニュアル』を参照してください。

## Adaptive Server における整数 ID のサポート

---

Replication Server 15.1 は *identity* 値として使用される Adaptive Server データ型の複写をサポートします。

Adaptive Server 15.0 では、次のデータ型を *identity* 値として使用できます。

- *bigint*
- *int*
- *numeric*
- *smallint*
- *tinyint*
- *unsigned bigint*
- *unsigned int*
- *unsigned smallint*

Replication Server 15.1 は、上記のデータ型の複写をサポートしています。*identity* カラムを含むテーブルに対して複写定義を作成する場合は、該当するカラムのデータ型として *identity* を指定してください。

## ステابل・キュー・マネージャのパフォーマンスの強化

---

ステابل・キュー・マネージャ (SQM) のパフォーマンスは、ステابل・キューのキャッシュ、セグメントの事前割り付け、ダイレクト I/O によるファイル・アクセスを含むように強化されました。

### ステابل・キューのキャッシュ

---

Replication Server は、単純なキャッシュ・メカニズムを使用して I/O を最適化します。このメカニズムにより、通常はキャッシュからデータを高速に読み取ることができるため、書き込みに対する遅延時間が短縮され、読み取り速度が向上します。

キャッシュは複数のページで構成され、各ページは隣接する複数のブロックで構成されています。キャッシュは、起動時に各キューに割り付けられます。ページ・サイズを変更すると、ステابل・キュー・デバイス内の I/O のサイズが変

化します。ページがいっぱいになると、単一の書き込み操作でページ全体が書き込まれます。

ステابل・キュー・キャッシュでは、ページ・ポインタが前進し、キャッシュの終端で先頭に戻ります。ライタがメッセージ・キューを満杯にし、メッセージを待機しているときにブロックされると、SQM によって現在のページがフラッシュされます。いっぱいになっていないページがフラッシュされると、データを含むブロックだけがディスクに書き込まれます。

### ステابل・キューのキャッシュ・パラメータを設定するコマンド

ステابل・キューのキャッシュ・パラメータを設定するコマンドの使用例を次に示します。

#### 例 1

サーバ全体のキャッシュのデフォルト値を設定するには、次のコマンドを実行します。

```
configure replication server set sqm_cache_enable to "on|off"
```

#### 例 2

キューのキャッシュを有効または無効にしてサーバ・レベルの設定を無効にするには、次のコマンドを実行します。

```
alter queue q_number, q_type,  
set sqm_cache_enable to "on|off"
```

**sqm\_cache\_enable** パラメータが無効になると、SQM モジュールは、16K に固定された 1 ブロックのバッファを保持する以前のメカニズムに戻ります。

#### 例 3

サーバ全体のページ・サイズのデフォルト値を設定するには、次のコマンドを実行します。

```
configure replication server set sqm_page_size to "num_of_blocks"
```

#### 例 4

指定されたキューのページ・サイズを設定するには、次のコマンドを使用します。

```
alter queue q_number, q_type, set sqm_page_size to "num_of_blocks"
```

*num\_of\_blocks* はページ内に 16K のブロック数を指定します。ページ・サイズを設定すると、Replication Server の I/O サイズも設定されます。たとえば、ページ・サイズを 4 に設定すると、Replication Server は 64K のまとまりでステابل・キューに書き込みを行います。

**例 5**

サーバ全体のキャッシュ・サイズのデフォルト値を設定するには、次のコマンドを実行します。

```
configure replication server set sqm_cache_size to
"num_pages"
```

**例 6**

指定したキューのキャッシュ・サイズを設定するには、次のコマンドを実行します。

```
alter queue q_number, q_type, set sqm_cache_size to
"num_pages"
```

*num\_pages* はキャッシュ内のページ数を指定します。

すべての SQM 設定コマンドは静的であるため、コマンドを有効にするにはサーバを再起動する必要があります。これらの設定パラメータの詳細については、『Replication Server リファレンス・マニュアル』を参照してください。

**セグメントの事前割り付け**

Replication Server 15.1 では、セグメントの割り付けによる遅延時間を削減するために、バックグラウンドでセグメントが事前割り付けされます。特に、RSSD がリモート Adaptive Server 上にある場合は、セグメントの割り付けによってライター・スレッドに対する遅延時間が大幅に増加します。

新しいセグメントが必要な場合、ライター・スレッドは、事前割り付けされたセグメントを使用できるかどうかをチェックし、使用できない場合はセグメントを割り付けるように要求します。ライター・スレッドが新しいセグメントを取得すると、バックグラウンドでセグメントが割り付けられるように事前割り付けの要求が行われます。その結果、ライター・スレッドに新しいセグメントが必要になったときには、既にセグメントが用意されているようになります。

セグメントの事前割り付けを有効または無効にするには、次のコマンドを実行します。

```
configure replication server set sqm_seg_prealloc to
"on|off"
```

このコマンドは静的であるため、コマンドを有効にするにはサーバを再起動する必要があります。このコマンドでは、サーバレベルの設定だけがサポートされません。

## ダイレクト I/O によるファイル・アクセスのサポート

ファイル・システムのパーティションでは、同期 I/O である DSYNC と比較した場合、ダイレクト I/O によって I/O の遅延時間が減少します。

---

**注意：**ダイレクト I/O は、Sun Solaris SPARC でのみサポートされます。

---

ステーブル・キュー・キャッシュを適切に調整する必要があります。適切なキャッシュ・サイズに調整すると、ほとんどの読み込みトランザクションがキャッシュ内で完了します。ダイレクト I/O を設定するには、次のコマンドを実行します。

```
configure replication server set sqm_write_flush to  
"dio"
```

このコマンドを実行すると、ダイレクト I/O が有効になります。ただし、このコマンドは、ステーブル・キューがファイル・システム上に存在する場合にのみ有効です。ダイレクト I/O メソッドにより、Replication Server は、ファイル・システムをバッファリングせずに直接ディスクに対して読み書きできます。

このコマンドは静的であるため、コマンドを有効にするにはサーバを再起動する必要があります。

# Replication Manager 15.1 の新機能

Replication Manager は、Replication Server 15.1 の機能をいくつかサポートしています。

## 動的 SQL のサポートの強化

---

Replication Manager 15.1 を使用することで、混合バージョンの複製環境 (Replication Server のバージョンは 15.0.1 以降) において、テーブル複製定義で動的 SQL 複製を有効にできます。

Replication Manager の GUI が変更されました。[複製定義の作成/変更] ダイアログ・ボックスの [一般] タブに、[動的 SQL の複製] オプションが追加されました。デフォルトでは、このオプションが選択されています。動的 SQL 複製が必要ない場合は、オプションの選択を解除してください。Replication Manager が動的 SQL をサポートしていないバージョンの Replication Server を使用している場合は、[動的 SQL の複製] オプションは自動的に無効になります。

**注意：**複製定義の動的 SQL 複製プロパティを変更した場合、変更された複製定義にのみ適用されます。同じテーブルの他の複製定義は変更されません。

---

Replication Server での動的 SQL 複製の詳細については、『Replication Server 管理ガイド第 2 巻』の「動的 SQL で強化された Replication Server のパフォーマンス」を参照してください。

## ファンクション複製定義のサポートの強化

---

Replication Manager 15.1 を使用して、プライマリ・ファンクション名とは異なる名前を持つ複数のファンクション複製定義を作成できます。

この機能をサポートするために、Replication Manager の GUI に次の変更が組み込まれました。

- [新しいファンクション複製定義の追加] ダイアログ・ボックスの [一般] タブで、ファンクション複製定義の作成後に、[新しいファンクション複製定義の追加] アイコンが [ファンクション複製定義] フォルダによって置き換えられない。複数のファンクション複製定義を作成できるように、[新しいファンクション複製定義の追加] アイコンと [ファンクション複製定義] フォルダが共存する。

- [新しいファンクション複写定義の追加] ダイアログ・ボックスの [一般] タブで、[複写定義] テキスト・フィールドを使用して、独自のファンクション複写定義名を作成できるようになった。[複写定義] フィールドで選択したストア・プロシージャとは異なるファンクション複写定義名を作成できる。
- [詳細] タブではなく [一般] タブで、[複写ストア・プロシージャ] オプションを使用できる。

ファンクション複写の操作の詳細については、『Replication Server 管理ガイド第1巻』を参照してください。

## ラージ・オブジェクト・データ型のサポート

---

Replication Manager 15.1 を使用して、複写環境で定義されているラージ・オブジェクト (LOB) データ型を管理できます。

Replication Manager は Microsoft SQL Server の LOB データ型 *varchar(max)*、*nvarchar(max)*、*varbinary(max)* をサポートします。ただし、カラムにこれらの LOB データ型のいずれかが含まれている場合、カラムをプライマリ・キーにしたり、サーチャブル・カラムとしてマーク付けしたりすることはできません。

Replication Manager の GUI の変更点は次のとおりです。

- 複写環境に LOB データ型がある場合は、[複写定義の作成] ダイアログ・ボックスの [複写定義データ型] リストおよび [パブリッシュ・データ型] リストに LOB が表示されます。
- [新しいテーブル複写定義の追加] ダイアログ・ボックスの [カラム] タブで、*varchar(max)*、*nvarchar(max)*、*varbinary(max)* の各データ型を含むカラムを選択すると、[プライマリ・キー] オプションおよび [サーチャブル] オプションが無効になります。

LOB データ型の操作の詳細については、『Replication Server 管理ガイド第1巻』を参照してください。

## Sybase Central 6.0

---

Replication Manager 15.1 は、Sybase Central 6.0 で動作します。

## opaque データ型のサポート

---

Replication Manager 15.1 では、opaque データを複写できます。opaque データには Oracle の *anydata* データ型や Microsoft SQL Server の *sql\_variant* データ型のような不特定または一貫性のない値を格納できます。



Replication Manager が opaque データをどのようにサポートするかを、次に示します。

- 混合バージョンの複製環境では、Replication Server のバージョンが 15.0.1 以降の場合、opaque データを複製できる。
- Replication Manager の GUI が変更されている。[新しいファンクション複製定義の追加] ウィンドウの [カラム] タブで、[データ型] 領域の [複製定義] リストに新しいオプション “opaque” が表示される。
- opaque データがあるカラムは、サーチャブル・カラムにできない。
- opaque データ型は、ファンクション複製定義とテーブル複製定義の両方でサポートされる。

サポートされるデータ型の複製の詳細については、『Replication Server リファレンス・マニュアル』を参照してください。

## timestamp データ型のサポート

---

Replication Manager 15.1 では timestamp の複製が追加されました。

- 混合バージョンの複製環境では、Replication Server のバージョンが 15.0.1 以降の場合、timestamp データ型を複製できる。
- Replication Manager の GUI が変更されている。[新しいファンクション複製定義の追加] ウィンドウの [カラム] タブで、[データ型] 領域の [複製定義] リストに新しいオプション “time stamp” が表示される。
- timestamp データ型は、ファンクション複製定義とテーブル複製定義の両方でサポートされる。

サポートされるデータ型の複製の詳細については、『Replication Server リファレンス・マニュアル』を参照してください。



## 追加の説明や情報の入手

Sybase Getting Started CD、製品マニュアル Web サイト、オンライン・ヘルプを利用すると、この製品リリースについて詳しく知ることができます。

- Getting Started CD (またはダウンロード) – PDF フォーマットのリリース・ノートとインストール・ガイド、その他のマニュアルや更新情報が収録されています。
- Sybase 製品マニュアル Web サイト (<http://sybooks.sybase.com/>) にある製品マニュアルは、Sybase マニュアルのオンライン版であり、標準の Web ブラウザを使用してアクセスできます。マニュアルはオンラインで参照することも PDF としてダウンロードすることもできます。この Web サイトには、製品マニュアルの他に、EBFs/Maintenance、Technical Documents、Case Management、Solved Cases、Community Forums/Newsgroups、その他のリソースへのリンクも用意されています。
- 製品のオンライン・ヘルプ (利用可能な場合)

PDF 形式のドキュメントを表示または印刷するには、Adobe の Web サイトから無償でダウンロードできる Adobe Acrobat Reader が必要です。

---

**注意：** 製品リリース後に追加された製品またはマニュアルについての重要な情報を記載したさらに新しいリリース・ノートを製品マニュアル Web サイトから入手できることがあります。

---

## サポート・センタ

---

Sybase 製品に関するサポートを得ることができます。

組織でこの製品の保守契約を購入している場合は、サポート・センタとの連絡担当者が指定されています。マニュアルだけでは解決できない問題があった場合には、担当の方を通して Sybase 製品のサポート・センタまでご連絡ください。

## Sybase EBF と Maintenance レポートのダウンロード

---

EBF と Maintenance レポートは、Sybase Web サイトからダウンロードしてください。

1. Web ブラウザで <http://www.sybase.com/support> を指定します。

2. メニュー・バーまたはスライド式メニューの [Support (サポート)] で [EBFs/Maintenance (EBF/メンテナンス)] を選択します。
3. ユーザ名とパスワードの入力が求められたら、MySybase のユーザ名とパスワードを入力します。
4. (オプション) [Display (表示)] ドロップダウン・リストからフィルタを指定し、期間を指定して、[Go (実行)] をクリックします。
5. 製品を選択します。

鍵のアイコンは、「Authorized Support Contact」として登録されていないため、一部の EBF/Maintenance リリースをダウンロードする権限がないことを示しています。未登録ではあるが、Sybase 担当者またはサポート・センタから有効な情報を得ている場合は、[My Account (マイ・アカウント)] をクリックして、「Technical Support Contact」役割を MySybase プロファイルに追加します。

6. EBF/Maintenance レポートを表示するには [Info] アイコンをクリックします。ソフトウェアをダウンロードするには製品の説明をクリックします。

## Sybase 製品およびコンポーネントの動作確認

---

動作確認レポートは、特定のプラットフォームでの Sybase 製品のパフォーマンスを検証します。

動作確認に関する最新情報は次のページにあります。

- パートナー製品の動作確認については、[http://www.sybase.com/detail\\_list?id=9784](http://www.sybase.com/detail_list?id=9784) にアクセスします。
- プラットフォームの動作確認については、<http://certification.sybase.com/ucr/search.do> にアクセスします。

## MySybase プロファイルの作成

---

MySybase は無料サービスです。このサービスを使用すると、Sybase Web ページの表示方法を自分専用カスタマイズできます。

1. <http://www.sybase.com/mysybase> を開きます。
2. [Register Now (今すぐ登録)] をクリックします。

## アクセシビリティ機能

---

アクセシビリティ機能を使用すると、身体障害者を含むすべてのユーザーが電子情報に確実にアクセスできます。

Sybase 製品のマニュアルには、アクセシビリティを重視した HTML 版もあります。

オンライン・マニュアルは、スクリーン・リーダーで読み上げる、または画面を拡大表示するなどの方法により、視覚障害を持つユーザがその内容を理解できるよう配慮されています。

Sybase の HTML マニュアルは、米国のリハビリテーション法第 508 条のアクセシビリティ規定に準拠していることがテストにより確認されています。第 508 条に準拠しているマニュアルは通常、World Wide Web Consortium (W3C) の Web サイト用ガイドラインなど、米国以外のアクセシビリティ・ガイドラインにも準拠しています。

---

**注意：**アクセシビリティ・ツールを効率的に使用するには、設定が必要な場合もあります。一部のスクリーン・リーダーは、テキストの大文字と小文字を区別して発音します。たとえば、すべて大文字のテキスト (ALL UPPERCASE TEXT など) はイニシャルで発音し、大文字と小文字の混在したテキスト (Mixed Case Text など) は単語として発音します。構文規則を発音するようにツールを設定すると便利かもしれませんが、詳細については、ツールのマニュアルを参照してください。

---

Sybase のアクセシビリティに対する取り組みについては、Sybase Accessibility サイト (<http://www.sybase.com/products/accessibility>) を参照してください。このサイトには、第 508 条と W3C 標準に関する情報へのリンクもあります。

製品マニュアルには、アクセシビリティ機能に関する追加情報も記載されています。

追加の説明や情報の入手

## 索引

## 数字

64 ビット・サポート 140-142

## A

Adaptive Server

master データベース・レプリケーションに  
おけるパスワード有効期間 6

Replication Server 15.5 でのサポート 135

Replication Server 15.6 でのサポート 109

共有ディスク・クラスタのサポート 185

セキュリティ 6

データ圧縮 5

マスタ・キー 6

レプリケーションのサポート 5, 34

レプリケート・データベースの再同期 88

Adaptive Server データベースの再同期

Adaptive Server と RepAgent のサポートさ  
れているバージョン 88

概要 88

Adaptive Server のサービス・キー 6

Adaptive Server のマスタ・キー 6

Adaptive Server コマンドとシステム・プロシ  
ージャのレプリケーションのサポート  
7

Adaptive Server モニタリング・テーブル

複数のレプリケーション・パス 10

admin config コマンド 146

admin who

機能強化 131

Advanced Services Option 118

alter connection コマンド 146

ASE 以外での複写のサポート

接続プロファイル 178

接続プロファイルの使用 178

接続プロファイルのリスト表示 180

単純化されたインストール、設定 177

ASE 以外のためのエラー・クラスのサポート

エラー・クラスの作成 176

エラー・クラスの変更 176

デフォルトの ASE 以外のためのエラー・  
クラス 175

## B

bigdatetime、複写サポート 135

bigtime、複写サポート 135

## C

configure replication server コマンド 150

## D

DIST ステータスの記録 198

dist\_sqt\_max\_cache\_size 141

DSI 145

DSI 効率の向上 120

DSI のモニタリング、データベースの再同期  
92

dsi\_bulk\_copy コネクション・パラメータ  
145-147

値の設定 146

値を調べる 146

次も参照：バルク・コピー・インのサ  
ポート

dsi\_bulk\_threshold コネクション・パラメータ  
146, 147

値の設定 146

値を調べる 146

次も参照：バルク・コピー・インのサ  
ポート

dsi\_compile\_enable、RTL 67

dsi\_compile\_max\_cmds 38

dsi\_compile\_retry\_threshold 設定パラメータ、強  
化されたリトライ・メカニズムの有  
効化 80

dsi\_max\_cmds\_in\_batch 設定パラメータ 116

dsi\_max\_xacts\_in\_group 設定パラメータ 117

dsi\_quoted\_identifiers 153

dsi\_row\_count\_validation 設定パラメータ 106

dsi\_serialization\_method 設定メソッド 117

dsi\_sqt\_max\_cache 141

dump database 92

**E**

exec\_cmds\_per\_timeslice 133

**F**

fstr\_cache\_size 142

**H**

High Volume Adaptive Replication 119

HVAR 119

HVAR、リトライ・メカニズムの強化 80

HVAR の機能強化

Replication Server 15.7 35

**I**

image データ型、バルク・コピー・イン複写サ  
ポート 111

IMDB 109, 138, 139

init\_sqm\_write\_delay 133

init\_sqm\_write\_max\_delay 133

interfaces ファイル 64

interfaces ファイル、Sybase IQ への複写用に作  
成 70

**J**

Java データ型、バルク・コピー・イン複写サ  
ポート 111

**L**

LOB データ型  
サポート 204  
部分更新 195

locales ディレクトリ  
変更 190

LTL コマンド  
キャッシュ 22

**M**

master データベース  
パスワードの有効期間 6

master データベース・レプリケーション 7

md\_source\_\_memory\_pool 142

memory\_limit 134, 142

**N**

none パラメータ、alter function string と create  
function string のスコープの拡張 123

num\_msg\_queues 38

num\_msgs 39

num\_threads 39

**O**

opaque データ型 204

Oracle から Sybase IQ への複写 57

Oracle からの複写 57

Oracle データ・サーバ

rs\_lastcommit テーブル 63

Oracle、トリガの実行 118

**Q**

queue\_dump\_buffer\_size 39, 142

**R**

Real-Time Loading

データベースのサポート 35, 57, 115

プラットフォームのサポート 35, 57, 115

Real-Time Loading Edition 57, 114

Real-Time Loading (RTL) による Sybase IQ への  
複写 57, 114

RepAgent

自動起動 34

RepAgent エグゼキュータ・スレッドの効率の  
向上 120

RepAgent の自動起動 34

replicate minimal columns 句、動的 SQL の使用  
122

replicate\_minimal\_columns パラメータ、動的  
SQL を使用したコネクションへの拡張  
122

Replication Manager

新機能 15.1 203

新機能 15.5 143



- 廃止 19
- Replication Manager は bigdatetime および  
bigtime データ型をサポートします。  
143
- Replication Server
  - 15.2 の新機能 145
  - 15.5 の新機能 113
  - 15.6 の新機能 55
  - 15.7.1 の新機能 5
  - 15.7 の新機能 21
  - interfaces ファイル 64
  - Sybase IQ レプリケート・データベース  
64
  - 新機能 15.6 ESD #1 41
- Replication Server ゲートウェイ
  - カスケード・コネクション 159
  - コネクション、追跡 160
  - コネクションの削除 161
  - 制限事項 159
  - 製品バージョンの要件 161
  - 製品バージョン要件 160
  - 有効化 159
- Replication Server のアップグレード、シームレス 109
- Replication Server システム・テーブル
  - rs\_asyncfuncs 26
  - rs\_clsfuncs 26
  - rs\_objfuncs 26
- Replication Server と Sybase IQ InfoPrimer のデータ・フロー 41, 42
- resume connection コマンド、skip to resync マーカ 89
- rs password 設定パラメータ 6
- rs\_autoc\_ignore システム・ファンクション 52
- rs\_autoc\_off システム・ファンクション 51
- rs\_autoc\_on システム・ファンクション 51
- rs\_delexception
  - ID 範囲と日付範囲によるトランザクションの削除 102
- rs\_delexception\_date ストアド・プロシージャ 104
- rs\_delexception\_id ストアド・プロシージャ 102
- rs\_delexception\_range ストアド・プロシージャ 105
- rs\_dictionary システム・テーブル 14
- rs\_lastcommit テーブル
  - Oracle データベース内 63
- rs\_passwords システム・テーブル 14
- rs\_session\_setting ファンクション文字列 37
- rs\_status システム・テーブル 49
- rs\_subscriptions システム・テーブル 30
- rs\_ticket
  - バージョン 2 191
- rs\_ticket マーカの挿入 133
- rs\_users システム・テーブル 14
- rsi\_packet\_size 39
- RSSD ロック・スキーマ、変更 134
- RTL 57, 114
  - admin config コマンド 76
  - dsi\_bulk\_threshold 68
  - dsi\_command\_convert 68
  - dsi\_compile\_enable 67
  - dsi\_compile\_retry\_threshold 69
  - dsi\_max\_cmds 68
  - rs\_helprep ストアド・プロシージャ 77
  - Sybase IQ への複写の設定 68
  - Sybase IQ への複写の有効化 67
  - コンパイル・ルール 58
  - コンパイルできないコマンド、テーブル 61
  - コンパイルとバルク適用 58
  - コンパイルの例 59
  - 参照制約 61, 74
  - システム・テーブル・サポート 70
  - ステージング・ソリューションからのマイグレート 78
  - 制限事項 60
  - データベース・レベルの設定パラメータの表示 76
  - データベースとプラットフォームのサポート 35, 57
  - テーブル・レベルの設定パラメータの表示 76
  - テーブル参照の表示 77
  - 複写シナリオ 70
  - 利点 57
  - 下位互換性 77
  - 混合バージョンのサポート 77
  - 最終的な変更のデータベースの表示 77

## 索引

情報の表示 76  
RTL での参照制約 74  
RTL の dsi\_bulk\_threshold 68  
RTL の dsi\_command\_convert 68  
RTL の dsi\_compile\_retry\_threshold 69  
RTL の dsi\_max\_cmds 68  
RTL の最終的な変更のデータベース、表示 77  
RTL 複写のシナリオ 70  
RTL 複写の例 70  
RTL、リトライ・メカニズムの強化 80  
RTL の機能強化  
Replication Server 15.7 35

## S

sa ユーザ・パスワード  
リセット 14  
sa ユーザ・パスワードのリセット  
パスワード 14  
select コマンド  
変数 161  
skip to resync パラメータ 89  
skip to resync マーカ、RepAgent から Replication  
Server への送信 90  
smp\_enable 134  
SQL 文の複写  
replicate SQLDML 句 168  
RSSD の変更 174  
set repmode 167  
sp\_setrepdbmode 166  
sp\_setrepdefmode 167  
WS\_SQLDML\_REPLICATION パラメータ  
173, 174  
ウォーム・スタンバイの設定 173, 174  
オートコレクション 171  
制限 171  
セッション・レベル 167  
データベース複写定義 168  
データベース・レベル 166  
テーブル複写定義 170  
テーブル・レベル 167  
複写スレッシュホールドの設定 137  
ロー・カウントの検証 171  
sqm\_async\_seg\_delete 設定パラメータ 24  
SQM コマンド・キャッシュ 22

sql\_max\_cache\_size 134, 141, 142  
sre\_reserve 142  
sts\_cachesize 39, 142  
sts\_full\_cache 134  
Sybase Control Center for Replication Server 31  
Sybase Control Center エージェント 31  
Sybase IQ  
RTL のコンパイルとバルク適用 58  
RTL の設定 68  
RTL の有効化 67  
エラー・クラスとファンクション文字列  
クラス 66  
接続の作成 67  
接続・パラメータ、設定 37  
ステージング・ソリューションからのマ  
イグレート 78  
接続プロファイル 66  
レプリケート・データベースの接続  
イビティ 64  
レプリケート・データベースのパーミッ  
ション 64  
レプリケート・データベースの設定 66  
干渉、システム・テーブル 63  
干渉、テンポラリー・ワークテーブル 63  
複写の干渉と影響 62  
Sybase IQ 複写ステージング・ソリューション  
から RTL へのマイグレート 78  
Sybase IQ への干渉、テンポラリー・ワークテ  
ーブル 63  
Sybase IQ、複写 57, 114  
Sybase IQ への接続  
カスタマイズ 37  
作成 67

## T

timestamp データ型 205

## U

Unicode の機能強化 29

## V

varbinary\_strip\_trailing\_zeros 設定パラメータ  
30

varbinary 値  
後続ゼロの削除 30

## あ

暗号化  
パスワード 12  
暗号化パスワード  
拡張サポート 191

## い

異機種並列 DSI 115  
インメモリ・データベース 109, 138  
引用符付き識別子  
alter replication definition、変更 154  
create replication definition、変更 153  
dsi\_quoted\_identifier 153  
quoted パラメータ 153  
rs\_set\_quoted\_identifier 155  
埋め込み二重引用符 152

## う

ウォーム・スタンバイ  
Oracle 117  
異機種 117  
データベースの再同期 99  
複写定義の削減 17, 25

## え

エグゼキュータ・コマンド・キャッシュ 23  
エラー・クラス、Sybase IQ 66  
エラー処理  
機能強化 128

## お

オペレーティング・システムのサポート、  
Replication Server 15.5 140

## か

カウンタ 193  
カスケード・コネクション、Replication Server  
ゲートウェイ 159

監査コマンド 14

## き

機能強化  
admin who 131  
locales ディレクトリ 190  
log\_first\_tran 189  
Replication Server のパフォーマンス 145  
resume connection 189  
SQM パフォーマンス 199  
sysadmin dump\_queue 188  
sysadmin sqt\_dump\_queue 188  
エラー処理 128  
ステーブル・キューの管理 187  
ダンプ・トランザクション 196  
動的 SQL 183, 203  
ファンクション複写 184, 203  
モニタリングとカウンタ 186  
リリース領域 190

### キャッシュ

SQL コマンド・キャッシュでの LTL コマ  
ンド 22  
SQM コマンド 22  
コマンドを動的に 23  
テーブル・メタデータ 23  
キュー・ブロック・サイズ  
推奨事項 82  
制限 82  
変更 83  
例、単純な複写システム 84  
例、中間ルートがある場合 87  
キュー・ブロック・サイズの増加 82, 121  
強化、メモリ割り付け 121

## こ

向上、DSI 効率 120  
向上、RepAgent エグゼキュータ・スレッドの  
効率 120  
向上、ディストリビュータ・スレッドの読み  
込み効率 121  
後続ゼロ  
削除 30

## 索引

コネクション別分散  
Adaptive Server モニタリング・テーブル・  
フィールド 10

コネクティビティ、レプリケート Sybase IQ  
64

コマンド  
admin config 146, 153  
alter connection 146, 150, 153  
alter replication definition 154  
configure replication server 150  
connect 159  
create connection 153  
create replication definition 153  
disconnect 161  
show connection 160  
show server 160  
sysadmin dump\_tran 190  
sysadmin issue\_ticket 133  
sysadmin sqm\_unzap\_tran 190  
sysadmin sqm\_zap\_tran 189  
sysadmin\_lmconfig 30

混合バージョン環境での制約、バージョン 15.5  
以降 140

コンパイルとバルク適用、RTL 58

**さ**

最低限の DML ロギング 139

削除、例外 102

サブキャパシティ・ライセンス 55

サブスクリプションの文字数の引き上げ 30

サポート  
Adaptive Server 共有ディスク・クラスタ  
185  
Adaptive Server における整数 ID 199  
LOB データ型 194, 204  
opaque データ型 204  
timestamp データ型 195, 205  
ダイレクト I/O 202  
次も参照：バルク・コピー・インのサ  
ポート

**し**

シームレス・アップグレード 109  
Replication Server 15.7 18

システム・テーブル  
rs\_status 49

rs\_subscriptions 30

自動起動設定パラメータ 34

シナリオ、データベース再同期化 93

シナリオ、データベース再同期化、ウォーム・  
スタンバイ 99

シナリオ、データベース再同期化、データベ  
ース再同期マーカのサポートなし 97

## 新機能

Replication Manager 15.1 203  
Replication Manager 15.5 143  
Replication Server 15.2 145  
Replication Server 15.5 113  
Replication Server 15.6 55  
Replication Server 15.6 ESD #1 41  
Replication Server 15.7 21  
Replication Server 15.7.1 5

## す

スキーマ変換 37

ステーブル・キューのキャッシュ 199

## せ

### 制限事項

LOB データ型 194  
opaque データ型 196  
動的 SQL 184  
ファンクションの複写 185  
部分更新 195

製品エディション、種類 55, 113

### セキュリティ

Adaptive Server のマスタ・キー 6  
Adaptive Server でのサポート 6  
master データベース・レプリケーションに  
おけるパスワード有効期間 6  
rs\_password 設定パラメータ 6  
sa ユーザ・パスワードのリセット 14  
コマンド監査 14  
デフォルト・パスワードなし 13  
パスワード暗号化 12  
パスワード・セキュリティ管理、システ  
ム・テーブル・サポート 14  
パスワード・パラメータ 12  
推奨事項 14

セキュリティ変更  
   監査可能コマンド 12  
   パスワード・セキュリティ 12  
 セグメントの削除  
   専用デーモン 24  
 セグメントの事前割り付け 201  
 接続プロファイル 178  
   Sybase IQ 66  
 接続プロファイル、Sybase IQ 66  
 設定パラメータ  
   dist\_sqt\_max\_cache\_size 141  
   dsi\_bulk\_copy 145-147  
   dsi\_bulk\_threshold 146, 147  
   dsi\_compile\_max\_cmds 38  
   dsi\_non\_blocking\_commit 150  
   dsi\_row\_count\_validation 106  
   dsi\_sqt\_max\_cache\_size 141  
   exec\_cmds\_per\_timeslice 133  
   init\_sqm\_write\_delay 133  
   init\_sqm\_write\_max\_delay 133  
   mem\_thr\_dst 27  
   mem\_thr\_exec 27  
   mem\_thr\_sqt 27  
   mem\_warning\_thr1 27  
   mem\_warning\_thr2 27  
   memory\_control 27  
   memory\_limit 134, 142  
   num\_msg\_queues 38  
   num\_msgs 39  
   num\_threads 39  
   queue\_dump\_buffer\_size 39  
   rsi\_packet\_size 39  
   smp\_enable 134  
   sqm\_cache\_size 24  
   sqt\_max\_cache\_size 134, 141  
   sts\_cachesize 39  
   sts\_full\_cache 134  
   unicode\_format 29

**そ**

増加、キュー・ブロック・サイズ 82, 121  
 挿入、rs\_ticket マーカ 133  
 増分データ転送、サポート 137

**た**

ターゲット・スコープ・ファンクション文字列  
   17

ダイレクト I/O によるファイル・アクセス  
   202  
 タスクのスケジューリング、複写 126  
 ダンプの適用 92

## ち

遅延名前解決、複写サポート 136  
 逐次化メソッド  
   wait\_after\_commit 116

## て

ディストリビュータ・スレッドの読み込み効  
   率の向上 121  
 データ圧縮  
   Adaptive Server でのサポート 5  
 データ型  
   bigdatetime 135  
   bigtime 135  
   image 111  
   Java 111  
   opaque 196, 204  
   timestamp 205  
 データ型変換 37  
 データ・サーバ・インタフェース 145, 147  
 データ転送、増分 137  
 データベース再同期化 127  
 データベース世代番号のリセット 132  
 データベースのサポート、Real-Time Loading  
   35, 57, 115  
 データベースのダンプ、取得 92  
 データベースのパフォーマンス、レプリケート  
   Sybase IQ 64  
 データベースの再同期 89, 127  
   DSI のモニタリング 92  
   resuming connection コマンドと skip to  
     resync パラメータ 89  
   skip to resync パラメータ 89  
   シナリオ 93  
   シナリオ、ウォーム・スタンバイ 99  
   シナリオ、データベース再同期マーカの  
     サポートなし 97  
   ダンプ・データベース・マーカの送信 92  
   データベースのダンプの取得 92  
   データベースのダンプの適用 92

## 索引

- トランザクションのスキップ 89, 90
- 再同期マーカ、送信 90
- 設定 89
- データベースの再同期の設定 89
  - DSI スレッド情報のモニタリング 92
  - Replication Server に対するトランザクションのスキップの指示 89
  - Replication Server へのダンプ・データベース・マーカの送信 92
  - Replication Server へのデータベース再同期マーカの送信 90
  - データベースのダンプの取得 92
  - 再同期するデータベースへのダンプの適用 92
- データベース再同期化シナリオ 93
  - ウォーム・スタンバイ・アプリケーションのアクティブ・データベースとスタンバイ・データベースの再同期 99
  - サードパーティ・ダンプ・ユーティリティの使用による再同期 95
  - プライマリ・データベースからのレプリケート・データベースの直接的な再同期 93
  - 再同期マーカに対するサポートがない場合の再同期 97
  - 同じダンプからのプライマリ・データベースとレプリケート・データベースの再同期 98
- テーブル
  - rs\_lastcommit、Oracle データベース内 63
- テーブル・メタデータ
  - キャッシュ 23
- テキスト更新の強化 198
- デフォルト・パスワードの削除 13
- デフォルト・パラメータ値、変更 38
- デフォルト・パラメータの値、変更 133

## と

- 動的 SQL
  - 機能強化 183, 203
- 動的 SQL の機能強化 121
- 動的 SQL、replicate minimal columns 句の使用 122
- 動的 SQL、replicate\_minimal\_columns パラメータのコネクションへの拡張 122

- 動的 SQL、文の実行の最適化 122
- トリガの実行、Oracle 118

## は

- バージョンのサポート
  - Adaptive Server の再同期 88
- パーミッション、レプリケート Sybase IQ 64
- 廃止
  - Replication Manager 19
- パスワード
  - 入力为非表示 12
  - パスワード暗号化
    - 拡張サポート 191
  - パスワード・セキュリティ管理
    - rs\_dictionary システム・テーブル 14
    - rs\_passwords システム・テーブル 14
    - rs\_users システム・テーブル 14
    - システム・テーブル・サポート 14
    - パスワード・パラメータ、設定 12
  - パスワード入力の非表示 12
- パフォーマンスの強化
  - Advanced Services Option 118
  - DSI 効率の向上 120
  - High Volume Adaptive Replication 119
  - Replication Server 15.7 22
  - Replication Server 15.7.1 16
- パフォーマンスの強化、Replication Server 15.5 118
- パフォーマンスの強化、Replication Server 15.6 80
- パラメータ、変更、デフォルト値 38
- パラメータ、デフォルト値の変更 133
- バルク insert
  - 次を参照：バルク・コピー・インのサポート
- バルク・コピー・インのサポート
  - コネクション・パラメータ 146
  - コネクション・パラメータ、値の設定 146
  - コネクション・パラメータの値を調べる 146
  - データ・サーバ・インタフェース (DSI)、実装 145
  - 複数文のトランザクションのサポート 147

## ひ

- 非ブロッキング・コミット
  - ASE 以外のデータベース、サポート 152
  - ASE 遅延コミット機能 149
  - dsi\_non\_blocking\_commit 150
  - Oracle、サポート 152
  - rs\_non\_blocking\_commit 150
  - rs\_non\_blocking\_commit\_flush 151
  - 設定 150

## ふ

- ファンクション複写
  - 機能強化 184
- ファンクション文字列
  - rs\_dsi\_check\_thread\_lock 117
  - rs\_non\_blocking\_commit 150
  - rs\_non\_blocking\_commit\_flush 151
  - rs\_set\_quoted\_identifier 155
- ファンクション文字列クラス、Sybase IQ 66
- ファンクション文字列の効率の向上、ファンクション文字列コマンドの none パラメータの拡張 123
- フェールバック・システム、複写の遅延 126
- 複写
  - データ、大量バッチ 145
- 複写スレッシュホールドの設定、SQL 文の複写 137
- 複写タスクのスケジュール 126
- 複写定義、削減
  - ターゲット・スコープ・ファンクション文字列、使用 17
- 複写定義の削減
  - MSA 17
  - ウォーム・スタンバイ 17, 25
- 複写定義の変更要求プロセスの強化 124
- 複写の遅延 126
- 複数のレプリケーション・パス 8, 21
  - Adaptive Server モニタリング・テーブル 10
  - monRepSenders モニタリング・テーブル 10
- プラットフォームのサポート、Real-Time Loading 35, 57, 115
- プロセスの強化、複写定義の変更 124
- ブロック・サイズ
  - 変更 83

- プロファイル
  - 接続 66

## へ

- 並列 DSI パラメータ
  - dsi\_max\_cmds\_in\_batch 116
  - dsi\_max\_xacts\_in\_group 117
  - dsi\_serialization\_method 117

## ま

- マルチパス・レプリケーション 21
  - 分散モード 9

## め

- メモリ消費の制御
  - HVAR 36
  - RTL 36
- メモリ割り付けの強化 121
- メンテナンス・ユーザ
  - 権限の付与 65

## よ

- 予約語、新規 135

## ら

- ライセンス 21, 55
  - 取得 56
  - 種類 55
- ライセンス、種類 113

## り

- リセット、データベース世代番号 132
- リトライ・メカニズム、HVAR と RTL での強化 80
- リファレンス実装 129
- 利便性とプロセスの強化、Replication Server 15.5 124
- 利便性とプロセスの強化、Replication Server 15.6 88

## 索引

利便性とプロセスの向上

Replication Server 15.7 17, 25

リラックス持続性データベース 109, 138

## る

ルート・アップグレード 109

ルートのアップグレード 109

## れ

例外の削除 102

トランザクション ID の範囲 102

日付 104

ユーザまたは送信先サイト 105

例外ログ

トランザクション ID の範囲によるトランザクションの削除 102

トランザクション日付の範囲によるトランザクションの削除 104

元のユーザ、元のサイト、送信先サイトによるトランザクションの削除 105

レプリケーションのサポート、Adaptive Server  
コマンドとシステム・プロシージャ  
7

レプリケート Sybase IQ の設定 66

レプリケート・データベース  
Sybase IQ 62

## ろ

ロー・カウントの検証

機能強化 106, 107

テーブル名の表示 107

無効化 106

ロー・カウントの検証、SQL 文の複写 171

ロー・カウントの検証の機能強化 128

ロー内/ロー外の LOB サポート 6

ログ

例外 102, 104, 105

ロック・スキーマ、RSSD 134