



新機能ガイド

---

**SAP Replication Server® 15.7.1**

**SP200**

ドキュメント ID：DC37519-01-1571200-01

改訂：2014年3月

Copyright© 2014 by SAP AG or an SAP affiliate company. All rights reserved.

このマニュアルの内容を SAP AG の明示的許可を得ずに、いかなる手段によっても、複製、転載することを禁じます。ここに記載された情報は事前の通知なしに変更されることがあります。

SAP AG およびディストリビュータが販売しているソフトウェア製品には、他のソフトウェアベンダー独自のソフトウェアコンポーネントが含まれているものがあります。国内製品の仕様は変わることがあります。

これらの資料は SAP AG および関連会社 (SAP グループ) が情報のみを目的として提供するものであり、いかなる種類の表明または保証も行わないものではなく、SAP グループはこの資料に関する誤りまたは脱落について責任を負わないものとします。SAP グループの製品およびサービスに関する保証は、かかる製品およびサービスに付属している明確な保証文書がある場合、そこで明記されている保証に限定されます。ここに記載されているいかなる内容も、追加保証を構成するものとして解釈されるものではありません。

ここに記載された SAP および他の SAP 製品とサービス、ならびに対応するロゴは、ドイツおよび他の国における SAP AG の商標または登録商標です。その他の商標に関する情報および通知については、<http://www.sap.com/corporate-en/legal/copyright/index.epx#trademark> を参照してください。

# 目次

表記の規則 .....	1
バージョン 15.7.1 SP200 の新機能 .....	5
SAP HANA データベースの複写サポート .....	5
SAP Replication Server による SAP Business Suite データベースのマテリアライゼーションと複写のサポート .....	5
RMA を使用した SAP HANA データベースへの複写の設定 .....	7
プライマリデータサーバとしての SAP HANA データベースのサポート .....	7
異機種間複写環境における DDL 複写のサポート .....	7
テーブル複写定義の排除 .....	8
ユニークキーがないテーブルの LOB カラムの複写 .....	9
SAP Replication Server で定義されているスキーママッピングのチェック .....	9
データ型マッピングの変更点 .....	10
SAP ASE の複写サポート .....	12
RepAgent の scan_verbose パラメータ .....	12
sp_setreplicate のサポート廃止 .....	13
テーブルレベルの引用符付き識別子の複製 .....	13
パフォーマンスの強化 .....	14
SQM リーダがキャッシュ内の全コマンドを読み取るよう設定する .....	14
ワークスレッド数の制御 .....	15
DIST での並列処理 .....	16

HVAR における DSI スレッドでのトランザク ションの分散適用 .....	16
SAP Replication Server コマンドの強化 .....	19
alter connection の変更点 .....	19
create database replication definition の変更点 .....	20
alter database replication definition の変更点 .....	23
create subscription の変更点 .....	25
activate subscription の変更点 .....	26
システム管理ツール .....	27
SAP Control Center for Replication .....	27
Sybase Central の廃止 .....	27
システムテーブル .....	29
<b>バージョン 15.7.1 SP110 の新機能 .....</b>	<b>31</b>
直接ロードマテリアライゼーションの設定パラメー タ .....	31
<b>バージョン 15.7.1 SP102 の新機能 .....</b>	<b>33</b>
ASE 間の直接ロードマテリアライゼーション .....	33
<b>バージョン 15.7.1 SP100 の新機能 .....</b>	<b>35</b>
リリースバージョン番号の変更 .....	35
Replication Server による SAP HANA データベース のサポート .....	35
ExpressConnect for SAP HANA データベース と SAP HANA データベースに対するレプリ ケートデータベースのコネクティビティ .....	36
プロパティ、ファンクション文字列、および データ型変換 .....	36
直接ロードマテリアライゼーション .....	37
Replication Server による Adaptive Server のサポー ト .....	39
Replication Server における圧縮データのサ ポート .....	39

バックリンクポイントとデータベースの縮小	40
データベースオブジェクトの所有権の譲渡	41
事前計算済み結果セット	41
細密なパーミッション	42
ストアドプロシージャの複写と <code>execute as</code> 句	42
RepAgent のサポート	43
Multi-Path Replication	43
フィルタ別分散	44
複数のスキャナ	44
セカンダリトランケーションポイントの管理	44
パフォーマンスの強化	45
インクリメンタル解析	45
利便性の向上	45
キューの処理時間の推定	46
自動でサイズ変更可能なパーティション	46
複写定義でサポートされていない引用符付き 識別子	46
高速複写モードのパフォーマンスに関するト ラブルシューティング	46
データ消失ステータス	47
ユニークキーがないテーブルの複写	47
セキュリティの向上	47
Replication Server のプリンシパル名	47
LDAP ユーザ認証	48
複写定義へのサブスクリプションの制御	49
メンテナンスユーザのセキュリティ	49
Sybase Control Center	50
プラットフォームのサポート	50
<b>Replication Server 15.7.1 の新機能</b>	<b>51</b>
Adaptive Server レプリケーション機能サポートの強 化	51
Adaptive Server データ圧縮	51

ロー内/ロー外の LOB .....	52
マスタキーと rs パスワード .....	52
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 .....	52
Adaptive Server コマンドとシステムプロシー ジャのサポート .....	53
マルチパスレプリケーション .....	53
異機種間 Multi-Path Replication .....	54
コネクション別分散 .....	55
High-Volume Adaptive Replication と Real-Time Loading .....	57
セキュリティの向上 .....	57
入力時のパスワードの非表示 .....	57
パスワードポリシー管理 .....	58
パスワードの暗号化 .....	58
Replication Server 設定のデフォルトパスワー ドの削除 .....	59
sa ユーザパスワードのリセット .....	59
コマンド監査 .....	59
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 .....	60
セキュリティの推奨事項 .....	60
パフォーマンスの強化 .....	61
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 .....	62
利便性の向上 .....	63
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 .....	63
アップグレードの簡素化 .....	64
システム管理ツール .....	64
<b>Replication Server 15.7 の新機能 .....</b>	<b>67</b>

Replication Server ライセンス .....	67
マルチパスレプリケーション .....	67
パフォーマンスの強化 .....	68
SQM コマンドキャッシュ .....	68
エグゼキュータコマンドキャッシュ .....	69
sqm_cache_size の上限値 .....	70
セグメントを削除するための専用デーモン .....	70
利便性とプロセスの向上 .....	71
複写定義の使用の削減 .....	71
rs_functions に加えられた変更 .....	72
メモリ消費の制御 .....	73
Unicode の機能強化 .....	75
SySAM ライセンス情報の要求 .....	76
サブスクリプション名の拡張 .....	76
後続ゼロの削除 .....	77
Sybase Control Center for Replication および Data Assurance .....	77
Adaptive Server レプリケーション機能サポートの強化 .....	80
RepAgent の自動起動 .....	80
Real-Time Loading と High-Volume Adaptive Replication .....	81
メモリ消費の制御 .....	82
Sybase IQ データベースオプションの設定 .....	83
スキーマ変換とデータ型変換 .....	83
パラメータのデフォルト値の変更 .....	84
Replication Server Data Assurance オプション .....	85
<b>Replication Server バージョン 15.6 ESD #1 の新機能 .....</b>	<b>87</b>
Replication Server と Sybase IQ InfoPrimer の統合 .....	87
ライセンス .....	88
Replication Server と SAP IQ InfoPrimer の統合の使用 .....	88

パラメータ .....	94
Replication Server のコンポーネント .....	95
デフォルトのデータ型変換 .....	99
サポートされない機能 .....	99
<b>Replication Server 15.6 の新機能 .....</b>	<b>101</b>
Replication Server ライセンス .....	101
サブキャパシティライセンス .....	101
Replication Server 15.6 の製品エディションと ライセンス .....	101
Real-Time Loading を使用した Oracle から Sybase IQ への複写 .....	103
Real-Time Loading ソリューション .....	103
Sybase IQ のレプリケートデータサーバ .....	108
SAP IQ での複写の干渉と影響 .....	108
SAP 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 .....	110
SAP IQ レプリケートデータベースのパーミッ ション .....	111
Sybase IQ レプリケートデータベースの設定に 関する問題 .....	112
Sybase IQ への複写シナリオ .....	116
参照制約のあるテーブル .....	121
RTL 情報の表示 .....	122
最終的な変更のデータベース .....	123
混合バージョンのサポートと下位互換性 .....	124
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 .....	124
パフォーマンスの強化 .....	126
HVAR と RTL のリトライメカニズムの強化 ....	126
キューブロックサイズの増加の機能強化 .....	128
利便性とプロセスの向上 .....	134

Adaptive Server のレプリケートデータベース の再同期 .....	134
範囲による例外の削除 .....	148
ローカウムの検証を制御する .....	152
ローカウムの検証エラーメッセージにテー ブル名を表示する .....	153
シームレスアップグレード .....	155
Adaptive Server 複写機能サポートの強化 .....	155
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 .....	155
image と Java データ型のバルクコピーイン ...	157
<b>Replication Server バージョン 15.5 の新機能 .....</b>	<b>159</b>
Replication Server 15.5 の製品エディションとライセ ンス .....	159
Real-Time Loading を使用した Sybase IQ の複写 ...	160
異機種間複写サポートの機能強化 .....	161
異機種環境での並列 DSI サポート .....	161
Oracle に対する異機種ウォームスタンバイサ ポート .....	163
Oracle レプリケートデータベースでのトリガ 制御 .....	164
パフォーマンスの強化 .....	164
Replication Server - Advanced Services Option .....	164
動的 SQL の機能強化 .....	167
ファンクション文字列の効率の向上 .....	169
利便性とプロセスの強化 .....	170
複写定義の変更要求プロセスの強化 .....	170
複写タスクのスケジューリング .....	172
複写の遅延 .....	172
レプリケートデータベースの再同期 .....	173
ローカウム検証での変更 .....	174

alter error class の機能強化 .....	175
リファレンス複写環境の実装に使用するツールセット .....	175
admin who コマンドの機能強化 .....	177
データベース世代番号のリセット .....	178
インバウンドキューに rs_ticket マーカを挿入する .....	179
デフォルト設定と予約語での変更 .....	179
パラメータのデフォルト値の変更 .....	179
RSSD ロックスキーマでの変更 .....	180
予約語 .....	181
<b>Adaptive Server 複写機能サポートの強化 .....</b>	<b>181</b>
bigdatetime と bigtime の複写 .....	181
遅延名前解決 .....	182
SQL 文の複写スレッシュホールドの設定 .....	183
増分データ転送 .....	183
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 .....	184
混合バージョン環境 .....	186
新しくサポートされたオペレーティングシステム .....	186
<b>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 .....</b>	<b>186</b>
Replication Server 設定パラメータの変更 .....	187
memory_limit 設定パラメータの変更 .....	188
<b>Replication Manager 15.5 の新機能 .....</b>	<b>189</b>
bigdatetime と bigtime の複写の有効化 .....	189
<b>Replication Server バージョン 15.2 の新機能 .....</b>	<b>191</b>
DSI でのバルクコピーインのサポート .....	191
サブスクリプションのマテリアライゼーションの強化 .....	191
新しいコネクションパラメータ .....	192
バルクコピーイン用の新しいカウンタ .....	193

制限事項 .....	194
非ブロッキングコミット .....	195
Adaptive Server の遅延コミット機能 .....	195
dsi_non_blocking_commit 設定パラメータ .....	196
rs_non_blocking_commit システムファンク ション .....	196
rs_non_blocking_commit_flush システムファン クション .....	197
サポートされる Adaptive Server 以外のデー タベース .....	197
引用符付き識別子 .....	198
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 .....	198
引用符付き識別子としてマークを付けるコマ ンド .....	199
rs_set_quoted_identifier ファンクション文字列 .....	201
rs_helprep の変更 .....	201
Replication Server ゲートウェイ .....	204
カスケードコネクション .....	204
Replication Server ゲートウェイを有効にする コマンド .....	205
コネクションを追跡するコマンド .....	206
コネクションを削除するコマンド .....	206
非 SQL 文の複写ローカウムの検証 .....	207
Replication Server エラークラスを作成するコ マンド .....	208
エラーアクションを割り当てるコマンド .....	208
Replication Server エラークラスを表示するス トアドプロシージャ .....	210
Replication Server システムデータベースの変 更 .....	211

SQL 文の複写 .....	211
SQL 文の複写の有効化 .....	212
システム設定の変更 .....	212
SQL 文の複写設定 .....	214
SQL 文の複写に対するローカウムの検証 ....	217
SQL の複写でのウォームスタンバイデータ ベースの設定 .....	219
SQL の複写でのウォームスタンバイデータ ベースの設定 .....	219
Replication Server システムデータベースの変 更 .....	219
Adaptive Server 以外のためのエラークラスのサポー ト .....	220
デフォルトの ASE 以外のためのエラークラス .....	221
create error class コマンドの変更 .....	221
alter error class コマンドの変更 .....	222
Adaptive Server 以外での複写サポートの機能強化 ..	222
単純化されたインストールと設定 .....	223
接続プロファイル .....	223
<b>Replication Server バージョン 15.1 の新機能 .....</b>	<b>227</b>
動的 SQL の機能強化 .....	227
ファンクション複写の機能強化 .....	228
Adaptive Server 共有ディスククラスタのサポート ..	229
モニタリングとカウンタの機能強化 .....	230
新しいアクティブオブジェクトカウンタ .....	230
新しいプロシージャインタフェース .....	231
ステーブルキューの管理の改善 .....	231
sysadmin dump_queue の変更 .....	232
sysadmin sqt_dump_queue の変更 .....	232
変更された resume connection コマンド .....	233
変更された sysadmin log_first_tran コマンド ..	233

新しい sysadmin sqm_zap_tran コマンド .....	233
新しい sysadmin sqm_unzap_tran コマンド .....	234
新しい sysadmin dump_tran コマンド .....	234
locales ディレクトリに対する変更 .....	234
拡張パスワード暗号化のサポート .....	235
rs_ticket ストアドプロシージャバージョン 2 .....	235
Replication Server の新しいカウンタ .....	237
ラージオブジェクトデータ型の拡張サポート .....	238
ラージオブジェクトデータ型の部分更新 .....	239
timestamp の拡張サポート .....	239
新しい opaque データ型 .....	240
ダンプトランザクションの機能強化 .....	240
dump サブコマンドの新しいパラメータ .....	241
rs_dumptran の変更 .....	241
ディストリビュータステータスの記録 .....	242
テキスト更新の強化 .....	242
Adaptive Server における整数 ID のサポート .....	243
ステابلキューマネージャのパフォーマンスの強 化 .....	243
ステابلキューのキャッシュ .....	243
セグメントの事前割り付け .....	245
ダイレクト I/O によるファイルアクセスのサ ポート .....	245
<b>Replication Manager 15.1 の新機能 .....</b>	<b>247</b>
動的 SQL のサポートの強化 .....	247
ファンクション複写定義のサポートの強化 .....	247
ラージオブジェクトデータ型のサポート .....	248
Sybase Central 6.0 .....	248
opaque データ型のサポート .....	248
timestamp データ型のサポート .....	249
索引 .....	<b>251</b>

# 目次

## 表記の規則

ここでは、SAP® 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
等幅 (固定幅)	<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>
斜体等幅	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)
斜体	<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>
太字体 sans-serif	<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

SAP® Replication Server® はさまざまなコンポーネントと連携して、SAP Adaptive Server Enterprise (SAP ASE)、SAP HANA® データベース、SAP® IQ、Oracle、IBM

DB2 UDB、Microsoft SQL Server など、サポートされているデータベース間の複製を実現します。SAP Replication Server では SAP ASE を Replication Server システムデータベース (RSSD) に使用します。または、SAP® SQL Anywhere® を Embedded Replication Server システムデータベース (ERSSD) に使用します。

Replication Agent™ は、SAP ASE、SAP HANA データベース、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 - SAP ASE 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 Agent for DB2 for z/OS

## 表記の規則

## バージョン 15.7.1 SP200 の新機能

SAP Replication Server 15.7.1 SP200 では、データベースのサポート、パフォーマンス、および利便性が強化されています。

### SAP HANA データベースの複写サポート

SAP Replication Server 15.7.1 SP200 では、SAP HANA データベースの複写をサポートする機能がいくつか強化されています。

### SAP Replication Server による SAP Business Suite データベースのマテリアライゼーションと複写のサポート

SAP Replication Server 15.7.1 SP200 では、サポートされているデータベース上で稼働中の SAP Business Suite アプリケーションから SAP HANA データベースへの、データベース全体またはテーブルのリストのマテリアライゼーションおよび複写をサポートしています。

サポートされている次のどのデータベース上で実行されている SAP Business Suite アプリケーションからも SAP HANA データベースへ複写できます。

- Oracle
- IBM DB2 UDB
- Microsoft SQL Server

---

**注意：** Replication Server では、種類の異なる複数のプライマリデータベースからのデータベースレベルでのマテリアライゼーションはサポートしていません。

---

### Replication Agent for Microsoft SQL Server によるテーブルのトランケーション

Replication Agent for Microsoft SQL Server を複写に使用する SAP Enterprise Resource Planning (ERP) システムでは、**sp\_SybTruncateTable** ストアドプロシージャを使用して、テーブルのトランケーションに関する Microsoft SQL Server の制限を無効にする必要があります。『異機種間複写ガイド』の「SAP Business Suite データベースのマテリアライゼーションと複写」を参照してください。

SAP Replication Server を使用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がすでに複写に参加している場合は、SAP Replication Management Agent Designer を使用して複写環境を設定します。『Replication Management Agent 設定およびユーザズガイド』を参照してください。複写環境を手動で設定するに

は、『Replication Server: SAP HANA データベースクイックスタートガイド』を参照してください。

### **SAP Replication Server によるクラスタテーブルのサポート**

SAP Replication Server 15.7.1 SP200 では、クラスタテーブルを使用する SAP Business Suite アプリケーションから SAP HANA データベースへの複写をサポートしています。

SAP Business Suite では、アプリケーションレイヤで論理テーブルが定義されます。論理テーブルはクラスタ化されて、データベースの物理テーブルに格納されます。クラスタテーブルとは、複数のテーブルをマテリアライズしたナチュラルジョインの圧縮表現です。クラスタテーブルが格納されたデータベースのマテリアライゼーションと複写を実行するとき、SAP Replication Server はデータのクラスタ化を解除してから DML を SAP HANA データベースに送信します。

---

**注意：** SAP Replication Server では、プールテーブルの複写はサポートされていません。

---

クラスタテーブルを複写する場合は、次のことを確認してください。

- SAP HANA のバージョンが 1.00.62 以上である。
- SAP Kernel が 720、721、740、または 741 である。
- ライブラリ `libSQLDBCHDB.dll` (Windows) または `libSQLDBCHDB.so` (UNIX) が Revision 1.00.72.00 以上である。
- ライブラリ `libodbcHDB.dll` (Windows) または `libodbcHDB.so` (UNIX) が Revision 1.00.72.00 以上である。

---

**注意：** クラスタテーブル `CLU4` と `VER_CLUSTR` は、複写対象としてマーク付けしないでください。詳細については、「SAP ノート 902817 - クラスタテーブル (論理項目名/物理項目名) における不整合」を参照してください。 <https://css.wdf.sap.corp/sap/support/notes/902817>

---

『異機種間複写ガイド』の「クラスタテーブルの複写」を参照してください。

### **DDL 複写における SAP Business Suite のサポート**

レプリケートが SAP HANA データベースで、プライマリデータベースが SAP Adaptive Server でない異機種間複写環境において、Replication Server は複数のデータ定義言語 (DDL) コマンドの複写をサポートしています。

「異機種間複写環境における DDL 複写のサポート (7 ページ)」を参照してください。

## **RMA を使用した SAP HANA データベースへの複写の設定**

Replication Management Agent (RMA) は、サポートされている任意のデータベースから SAP HANA データベースへの複写を設定および管理するときに使用できる分散管理エージェントです。

RMA では、SAP Business Suite または Business Suite 以外のシステムの自動設定、マテリアライゼーション、モニタリング、および管理をサポートしています。

RMA を使用して、これらのプライマリデータベースからレプリケート SAP HANA データベースへの複写を設定できます。

- IBM UDB DB2
- Microsoft SQL Server
- Oracle

RMA では、サポートされている任意のデータベースから ERP データベースへの複写にも対応しています。RMA を使用した複写の設定については、『Replication Management Agent 設定およびユーザズガイド』を参照してください。

## **プライマリデータサーバとしての SAP HANA データベースのサポート**

Replication Agent™ for SAP HANA データベースを使用して複写モデルを設定、管理、構成し、プライマリ SAP HANA データベースからレプリケート SAP HANA データベースへデータを複写できるようになりました。

Replication Agent for SAP HANA は RMA コンテナ内で動作し、初期データ抽出およびロードを実行するほか、プライマリ SAP HANA データベース上のデータ変更をレプリケート SAP HANA データベースに複写します。

複写モデルのインストールと展開の手順については、『Replication Agent for SAP HANA Configuration and Users Guide』を参照してください。

## **異機種間複写環境における DDL 複写のサポート**

レプリケートが SAP HANA データベースである異機種間複写環境において、Replication Server は複数のデータ定義言語 (DDL) コマンドの複写をサポートしています。

Replication Server では、プライマリデータベースが Oracle、Microsoft SQL Server、または IBM DB2 UDB の場合、SAP HANA データベースへの次の DDL コマンドの複写をサポートしています (SAP ASE はプライマリデータベースとしてサポートされていません)。

- **create table**

- **alter table (add column または drop column)**
- **rename table**
- **drop table**

この機能強化により、データベース複写定義が **auto\_update\_table\_list** または **auto\_extend\_table\_list** パラメータを指定して定義されている場合、Replication Server はデータベース複写定義を自動的に変更して、データベースレベルで複写対象としてマーク付けされたデータベースで発生したすべての DDL の変更を反映します。たとえば、あるデータベースが複写対象としてマーク付けされている場合、そのデータベースにテーブルが追加されると、その新しく追加されたテーブルに対して自動的に複写が確立されます。

次の場合、作成、変更、名称変更、または削除されたテーブルは、レプリケートテーブル所有者を除き、テーブル複写定義では正規化されません。

- レプリケートテーブル所有者がテーブル複写定義で定義されており、かつ
- レプリケートサイトがテーブル複写定義にサブスクライブしている。

---

**注意：** 異機種間複写環境では、プライマリデータ型が対応するレプリケートデータ型にマッピングされます。

---

参照:

- 「データ型マッピングの変更点 (10 ページ)」
- 『異機種間複写ガイド』の「データ型の変換とマッピング」
- 『リファレンスマニュアル』の「create database replication definition」
- 『異機種間複写ガイド』の「異機種間複写環境における DDL 複写のサポート」

## テーブル複写定義の排除

異機種間環境でデータベース複写定義とサブスクリプションを使用する場合、テーブル複写定義を使用する必要はありません。

以前のリリースでは、異機種間複写環境においてはテーブルごとにテーブル複写定義が必要でした。このようにテーブル数の多いデータベースが参加する環境では、すべてのテーブル複写定義を管理すると、余計に手間がかかります。DDL 複写では、テーブル複写定義との調整なしに DDL でテーブル定義を変更すると、データに不整合が生じるおそれがあります。データの不整合を回避しながら、テーブルごとにテーブル複写定義を使用する必要性をなくすために、データベース複写定義とサブスクリプションを使用できるようになりました。

『異機種間複写ガイド』の「テーブル複写定義の排除」を参照してください。

## ユニークキーがないテーブルの LOB カラムの複写

サポートされているデータベースから SAP HANA データベースの複写で、Replication Server はユニークキーがないテーブルの LOB カラムの複写をサポートしています。

ユニークキーがなく、LOB カラムがあるテーブルの場合、LOB カラムのステータスを **replicate\_if\_changed** に設定する必要があります。

Replication Agents for Oracle、Microsoft SQL Server、または IBM DB2 UDB の場合、デフォルトですべての LOB カラムのステータスが **replicate\_if\_changed** に設定されています。

SAP Adaptive Server Replication Agent Thread の場合、デフォルトでは **sp\_setreptable** によって複写ステータスが **always\_replicate** に設定されています。 **sp\_setrepcol** を使用して、LOB カラムの複写ステータスを **replicate\_if\_changed** に設定する必要があります。

参照:

- 『リファレンスマニュアル』の「sp\_setrepcol」
- 『異機種間複写ガイド』の「ユニークキーがないテーブルの LOB カラムの複写」
- 『管理ガイド 第 1 巻』の「ユニークキーがないテーブルの複写」

## SAP Replication Server で定義されているスキーママッピングのチェック

**check schema map** を使用すると、SAP Replication Server で定義されているすべてのプライマリおよびレプリケートのスキーママッピングを表示できます。

### 構文

```
check schema map [[from primary_data_server.primary_database.  
[from_schema|NULL]] |  
[to replicate_data_server.replicate_database.[to_schema|NULL]]]
```

### パラメータ

- **primary\_data\_server.primary\_database** - プライマリデータサーバとデータベースの名前。
- **from\_schema** - プライマリデータベースのスキーマ。
- **replicate\_data\_server.replicate\_database** - レプリケートデータサーバとデータベースの名前。
- **to\_schema** - レプリケートデータベースのスキーマ。

- **NULL** – すべてのプライマリスキーマまたはすべてのレプリケートスキーマを指定します。NULL は *from\_schema* と *to\_schema* の代わりに使用します。

## 例

- **例 1** – プライマリ Oracle データベースまたはプライマリ Adaptive Server データベースのスキーママッピングを表示します。

```
check schema map from PDS.PDB.SAPSR3
```

- **例 2** – プライマリ IBM DB2 UDB データベースのスキーママッピングを表示します。

```
check schema map from PDS.PDB.SAP<SID>
```

- **例 3** – レプリケートデータベースのスキーママッピングを表示します。

```
check schema map to RDS.RDB.NULL
```

- **例 4** – レプリケート SAP HANA データベースのスキーママッピングを表示します。

```
check schema map to RDS.RDB.tableowner
```

## 使用法

複写中にスキーママッピングが変更された場合、**check schema map** コマンドを使用して現在のマッピング関係をチェックできます。後で要件に応じてマッピングを変更できます。

『リファレンスマニュアル』の「check schema map」を参照してください。

## データ型マッピングの変更点

Replication Server 15.7.1 SP200 では、Oracle、Microsoft SQL Server、および DB2 UDB から SAP HANA データベースへのデータ型マッピングが次のように変更されました。

変更されていないデータ型マッピングは、ここには掲載していません。データ型マッピングの全一覧については、『異機種間複写ガイド』の「データ型の変換とマッピング」を参照してください。

表 1 : Oracle から SAP HANA データベースへのデータ型マッピング

Oracle データ型	複写定義のデータ型	SAP HANA データベースのデータ型
CHAR	char	varchar
NCHAR	unichar	nvarchar

Oracle データ型	複写定義のデータ型	SAP HANA データベースのデータ型
RAW	rs_oracle_binary	varbinary

表 2 : Microsoft SQL Server から SAP HANA データベースへのデータ型マッピング

Microsoft SQL Server のデータ型	複写定義のデータ型	SAP HANA データベースのデータ型
bigint	rs_msss_bigint	bigint
char	char	varchar
datetime	rs_msss_datetime	timestamp
datetime2	rs_msss_datetime2	timestamp
nchar	unichar	nvarchar
nvarchar(max)	rs_msss_nvarchar_max	nclob
smalldatetime	rs_msss_datetime	seconddate
timestamp	timestamp	varbinary(8)
time	rs_msss_time	timestamp
uniqueidentifier	char(36)	varchar(36)
varbinary(max)	rs_msss_varbinary_max	blob
varchar(max)	rs_msss_varchar_max	clob

表 3 : DB2 UDB から SAP HANA データベースへのデータ型マッピング

DB2 UDB のデータ型	複写定義のデータ型	SAP HANA データベースのデータ型
BLOB	rs_udb_varchar_for_bit	blob
CHAR	char	varchar
CHAR FOR BIT DATA	rs_udb_char_for_bit(推奨) binary	varbinary
DECFLOAT(16)	rs_udb_decfloat	double
GRAPHIC	unichar	nvarchar

DB2 UDB のデータ型	複写定義のデータ型	SAP HANA データベースのデータ型
INTEGER	rs_udb_bigint	integer

## SAP ASE の複写サポート

SAP Replication Server 15.7.1 SP200 は、SAP Adaptive Server Enterprise との互換性が検証されています。

### RepAgent の scan\_verbose パラメータ

`sp_help_rep_agent` の `scan_verbose` パラメータは、`scan` パラメータと同じ出力のほかに、Adaptive Server RepAgent が現在のマーカーからログの末尾までに処理するログページ数を表示します。

『Replication Server リファレンスマニュアル』で `sp_help_rep_agent` の `scan_verbose` パラメータを参照してください。

たとえば、`scan_verbose` は、2つの複写パスとデフォルトパス、および複数のスキャナを持つマルチスレッド RepAgent についてスキャン情報を表示し、現在のマーカーからログの末尾までに処理されるログページ数を表示します。log pages left カラムを参照してください。

```
sp_help_rep_agent pdb2, 'scan_verbose'
```

次のように表示されます。

```
Replication Agent Scan status
```

```

dbname  pathname  scanner_spid  start_marker  end_marker
-----  -
pdb2    default  34            (1099,0)     (1113,1)
pdb2    path1    35            (1099,0)     (1113,1)
pdb2    path2    36            (1099,0)     (1113,1)

current_marker  log_pages_left  log_recs_scanned
-----
(1113,1)        0                125
(1113,1)        0                125
(1113,1)        0                125

oldest_transaction
-----
(0,0)
(0,0)
(0,0)

```

## sp\_setreplicate のサポート廃止

Replication Server 15.7.1 SP200 では、**sp\_setreplicate** Adaptive Server システムプロシージャのサポートが廃止されました。

代わりに、**sp\_setreptable** および **sp\_setrepcol** システムプロシージャを使用して、テーブルとカラムの複写を制御したり、テーブルとカラムの現在の複写ステータスを表示したりできます。『Replication Server リファレンスマニュアル』の「**sp\_setreptable**」と「**sp\_setrepcol**」を参照してください。

## テーブルレベルの引用符付き識別子の複製

オブジェクトに対してテーブルレベルで **dsi\_quoted\_identifier** を設定することにより、引用符付きオブジェクトを正しく複写するよう Replication Server に指示できます。

ある接続でテーブル複写を実行するとします。接続レベルの **dsi\_quoted\_identifier** は **on** に設定されています。その後、引用符付き識別子が出現するテーブルを複写するために、同じ接続でデータベース複写を設定します。これには、接続レベルの **dsi\_quoted\_identifier** を **always** に設定する必要があります。テーブル名とカラム名がすべて引用符で囲まれているため、既存のテーブル複写は失敗する可能性があります。既存のテーブル複写を正常に実行するためには、そのテーブルに対してテーブルレベルの **dsi\_quoted\_identifier** を **on** に設定します。

以前のリリースでは、**dsi\_quoted\_identifier** パラメータは接続レベルかサーバレベルでしか設定できませんでした。Replication Server 15.7.1 SP200 では、テーブルレベルで **dsi\_quoted\_identifier** を設定可能になりました。テーブルレベルの **dsi\_quoted\_identifier** の設定は、既存の接続レベルの **dsi\_quoted\_identifier** の設定に優先します。また、接続レベルの設定はサーバレベルの **dsi\_quoted\_identifier** の設定に優先します。

プライマリテーブルに引用符付き識別子がある場合に、複写が正常に実行されるようにするには、**quoted** 句を指定してテーブル複写定義を作成し、接続レベルの **dsi\_quoted\_identifier** を **on** に設定する必要があります。

プライマリテーブルに引用符付き識別子がある場合で、テーブル複写定義に対して **create subscription** に **init replicate table** オプションを指定して使用するときは、**dsi\_quoted\_identifier** を **always** に設定し、次の条件を確認してください。

- テーブル複写定義で指定するプライマリテーブル名がプライマリデータベース内のテーブル名と同じである。

- Replication Agent パラメータ **lrl\_character\_case** が **asis** に設定されている。これにより、オブジェクト名が、複写定義で指定されているとおりに大文字と小文字が区別されて送信されます。
- テーブル複写定義で指定するカラム名がプライマリデータベース内のカラム名と同じである。
- スキーママッピングで指定する所有者名がプライマリデータベース内の所有者名と同じである。
- データベース複写の作成後にテーブル複写定義を削除する場合は、テーブル複写定義内のすべての識別子 (テーブル名、カラム名など) を引用符で囲む。

『管理ガイド 第 1 巻』の「引用符付き識別子」を参照してください。

## パフォーマンスの強化

---

Replication Server 15.7.1 SP200 では、複写パフォーマンスを強化するためいくつかの変更が加えられました。

### SQM リーダがキャッシュ内の全コマンドを読み取るよう設定する

**sqm\_reader\_first** を有効にすると、SQM リーダがステابلキューマネージャのキャッシュから必ずすべてのコマンドを取得することができます。

#### 前提条件

**sqm\_reader\_first** を有効にする前に、**sqm\_cache\_enable** でステابلキューマネージャ (SQM) のキャッシュを有効にします。コマンドの構文については、『管理ガイド 第 2 巻』の「ステابلキューキャッシュのパラメータの設定」と『リファレンスマニュアル』の「SAP Replication Server のコマンド」を参照してください。

#### 手順

SQM キャッシュはインバウンドキューとアウトバウンドキューに書き込まれたコマンドをキャッシュするので、各読み取りプロセスではアンパックが必要ありません。このため、SQM キャッシュは処理時間を短縮します。キャッシュが満杯に近づいても未読のコマンドが破棄されないようにし、SQM リーダがキャッシュ内のすべてのコマンドを必ず取得するようにするには、**sqm\_reader\_first** を有効にします。このパラメータは、キャッシュが満杯のときは SQM ライタによるキャッシュへの書き込みを禁止します。SQM リーダが遅れを取り戻してキャッシュからコマンドを読み取った後、キャッシュに空き領域が用意された時点で初めて SQM ライタによるキャッシュへの書き込みを許可します。

**sqm\_reader\_first** は、キューレベルでインバウンドキューかアウトバウンドキューのどちらかに対して設定することも、サーバレベルですべてのキューに対して設定することもできます。デフォルトは off です。

1. SQM リーダが SQM キャッシュにあるすべてのコマンドを必ず取得するようにします。

たとえば、キューレベルで **sqm\_reader\_first** を有効にするには、次のように入力します。

```
alter queue, q_number, q_type,
set "sqm_reader_first" to "on"
```

サーバレベルで **sqm\_reader\_first** を有効にするには、次のように入力します。

```
configure replication server
set "sqm_reader_first" to "on"
```

2. キューレベルで **sqm\_reader\_first** を設定した場合は、**resume queue** を使用してキューを再起動します。  
サーバレベルで **sqm\_reader\_first** を設定した場合は、SAP Replication Server を再起動します。

## ワークスレッド数の制御

グローバルタスクマネージャが SAP Replication Server モジュールによってスケジューリングされたタスクの処理に割り当てることができるワークスレッドの数を制御します。

SAP Replication Server が起動すると、グローバルタスクマネージャが作成されます。このグローバルタスクマネージャによって、ユーザが **worker\_thread** パラメータを使用してサーバレベルですべての接続に対して設定した数のワークスレッドが作成されます。グローバルタスクマネージャは、DSI でのトランザクションの分散適用や DIST での並列処理など、各モジュールによってスケジューリングされたタスクがスレッドを使用できるようにします。

デフォルトのワークスレッド数は 4 です。設定できる最大値は 32、最小値は 2 です。 **worker\_thread** の変更を有効にするには、SAP Replication Server を再起動する必要があります。

1. グローバルタスクマネージャが使用できるワークスレッド数を変更します。次のように入力します。

```
configure replication server set worker_thread_num to
'number_of_worker_thread'
go
```

2. (オプション) **admin who** を使用してワークスレッドのステータスを確認します。

たとえば、各ワークスレッドについて次のように表示されます。

```
11 WORKER           Awaiting Task
10 WORKER           Awaiting Task
```

9 WORKER	Awaiting Task
8 WORKER	Awaiting Task

## DIST での並列処理

DIST スレッドで各トランザクションの並列処理を最大化することにより、複写パフォーマンスが向上します。

SAP Replication Server が管理する各プライマリデータベースについて、DIST スレッドはステابلキュートランザクション (SQT) スレッドを使用してインバウンドキューからトランザクションを読み込み、トランザクションを1つずつ処理した後、ステابلキューマネージャ (SQM) を使用してトランザクションをアウトバウンドキューに書き込みます。

並列DIST処理では、SQTがインバウンドキューからトランザクションを読み込んだ後、DISTスレッドが並列でトランザクションを処理します。トランザクションの並列処理は複写を高速化するとともに、レプリケートサイトへのトランザクションの配信順序がプライマリサイトで発生したトランザクションのコミット順序と同じであることを保証します。

特定のプライマリデータベース接続または SAP Replication Server へのすべてのプライマリデータベース接続について DIST スレッドでの並列処理を有効にするには、**parallel\_dist** パラメータを使用します。

『管理ガイド 第2巻』の「DIST での並列処理」を参照してください。

## HVAR における DSI スレッドでのトランザクションの分散適用

トランザクション内の並列処理の強化により、HVAR における DSI スレッドの複写パフォーマンスが向上します。

SAP ASE への High Volume Adaptive Replication (HVAR) におけるデータサーバインタフェース (DSI) のパフォーマンスを向上させるには、SAP Replication Server で分散トランザクションを有効にし、レプリケート SAP ASE データサーバで分散トランザクション管理のサポートを有効にします。トランザクションの分散適用の場合、SAP Replication Server は各コマンドを逐次に適用するのではなく、トランザクション内のコマンドを複数のグループに分け、コマンドグループを並列でレプリケートデータベースに適用します。トランザクションの分散適用は、サーバレベルですべての接続に対して有効にするか、接続レベルで特定の接続に対して有効にすることができます。接続レベルの設定はサーバレベルの設定に優先します。

### 制限事項

- トランザクションの分散適用を有効にするには、先に SAP ASE データベースへの HVAR を有効にする必要があります。『SAP Replication Server 管理ガイド

第2巻』の「Adaptive Server への High-Volume Adaptive Replication」を参照してください。

- SAP Replication Server がトランザクションの分散適用を無効にするのは、ユーザが High Volume Adaptive Replication における大規模なコンパイル可能なトランザクションについてフルインクリメンタルコンパイルを有効にした場合のみです。この制限は、ユーザがフルインクリメンタルコンパイルを有効にしただけにもかかわらず、フルインクリメンタルコンパイルを使用しない小規模なコンパイル可能なトランザクションには適用されません。『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。
- SAP Replication Server は、RSSD における DSI スレッドについてはトランザクションの分散適用を自動的に無効にします。
- SAP Replication Server は、HVAR で処理できないコンパイル不能なトランザクションについてはトランザクションの分散適用を自動的に無効にします。代わりに、ログ順、ローごとの連続複写に戻ります。『管理ガイド 第2巻』の「HVAR の処理と制限事項」を参照してください。
- トランザクションの分散適用は、複数のテーブルの複写が存在し、並列処理を利用できるトランザクションプロファイルに有効です。
- トランザクションの分散適用では、トランザクションの並列処理を行うため、必要なメモリや処理リソースが増えます。

### **HVAR における DSI でのトランザクション分散適用の設定**

SAP ASE データベースへの High Volume Adaptive Replication におけるトランザクションの分散適用のサポートを設定します。

SAP Replication Server では、SAP ASE データベースへの High Volume Adaptive Replication (HVAR) についてトランザクションの分散適用をサポートしています。SAP Replication Server からのトランザクションの分散適用をサポートするには、SAP ASE データサーバごとに分散トランザクション管理 (DTM) の個別のライセンスと設定が必要です。

1. SAP ASE レプリケートデータベースで分散トランザクション管理 (DTM) のサポートを有効にします。
 

『SAP ASE システム管理ガイド 第2巻』の「分散トランザクション管理」を参照してください。

  - a) 有効な DTM ライセンスがあることを確認します。
 

『SAP ASE S システム管理ガイド 第2巻』 > 「分散トランザクション管理」 > 「DTM 機能の有効化」 > 「ライセンスキーのインストール」を参照してください。
  - b) DTM を有効にします。
 

レプリケート SAP ASE データサーバで次のように入力します。

```
sp_configure 'enable dtm', 1
```

- c) SAP ASE データサーバを再起動します。
- d) SAP Replication Server メンテナンスユーザに分散トランザクションの役割を付与します。

```
sp_role "grant", dtm_tm_role, maintenance user
```

## 2. SAP Replication Server でトランザクションの分散適用を有効にします。

- a) レプリケート SAP ASE データベースへの接続に対して HVAR を有効にします。  
たとえば、単一の接続に対して HVAR を有効にするには、次のように入力します。

```
alter connection to data_server.database
set dsi_compile_enable to 'on'
go
```

HVAR 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。『システム管理ガイド 第2巻』の「HVAR の有効化」を参照してください。

- b) **dt\_apply** パラメータを 1 より大きい値に設定して、レプリケートデータベースへの特定の接続に対して、またはサーバレベルですべての接続に対してトランザクションの分散適用を有効にします。

**dt\_apply** は、SAP Replication Server が 1 つのトランザクションを適用するとき使用する最大のタスク数を制御します。**dt\_apply** のデフォルト値は 1 です。このときトランザクションの分散適用は行われません。設定できる最大値は 8、最小値は 1 です。

たとえば、レプリケートデータベースへの単一の接続に対してトランザクションの分散適用を有効にするには、次のように入力します。

```
alter connection to data_server.database
set dt_apply to "number_of_distributed_application_tasks"
go
```

サーバレベルでトランザクションの分散適用を有効にするには、次のように入力します。

```
configure replication server
set dt_apply to "number_of_distributed_application_tasks"
go
```

*number\_of\_distributed\_application\_tasks* の値は二重引用符で囲ってください。

- c) 接続レベルで **dt\_apply** を設定した場合は、そのデータベースへの接続を再起動します。サーバレベルで変更した場合は、SAP Replication Server を再起動します。

## SAP Replication Server コマンドの強化

SAP Replication Server 15.7.1 SP200 では、いくつかの SAP Replication Server コマンドが強化されています。

### alter connection の変更点

**alter connection** の構文が変更され、新しいパラメータがいくつか追加されました。

```
alter connection to data_server.database {
[for replicate table named [table_owner.]table_name
[set table_param [to] 'value']] |
set function string class [to] function_class |
set error class [to] error_class |
set replication server error class [to] rs_error_class |
set password [to] passwd |
set dsi_connector_sec_mech [to] hdbuserstore |
set log_transfer [to] {on | off} |
set database_param [to] 'value' |
set security_param [to] 'value' |
set security_services [to] 'default' |
set dataserver and database name [to] new_ds.new_db |
set trace [to] 'value' |
    set schemamap [with | without decluster] from
data_server.db.{from_schema | NULL} to {to_schema | NULL} |
    set sap_trim_len to {'on' | 'off'} |
    set sapsystemname to 'sid_adm_value' |
    set reblock_ddntf to {'on' | 'off'}}}
```

Replication Server 15.7.1 SP200 では、次のパラメータが追加されています。

- **schemamap** - プライマリスキーマまたは所有者 (*from\_schema*) をスキーマまたは所有者 (*to\_schema*) にマッピングします。  
DML または 異機種間 DDL 複写では、レプリケートデータベースがテーブル所有者を含むテーブル複写定義にサブスクライブしている場合は、そのテーブル複写定義のレプリケート所有者が有効になります。**schemamap** を適用できるのは、テーブルにテーブルサブスクリプションがない場合か、テーブル複写定義にレプリケートテーブル所有者が含まれていない場合のみです。
- **data\_server.db** - データサーバとプライマリデータを格納するデータベースを指定します。
- **from\_schema** - プライマリデータベースのスキーマを指定します。
- **to\_schema** - レプリケートデータベースのスキーマを指定します。
- **with decluster** - スキーマに対してクラスタテーブルのクラスタ化解除を有効にします。

- **without decluster** - スキーマに対してクラスタテーブルのクラスタ化解除を無効にします。
- **NULL** - すべてのユーザを指定します。 *from\_schema* と *to\_schema* の両方に **NULL** を指定すると、 *data\_server.db* のマッピング関係が削除されます。
- **sapsystemname** - (SAP Business Suite アプリケーションからの複写の場合のみ) SAP システム名 (<*sid*>) の環境変数を指定します。
- **sap\_trim\_len** - (Oracle データベース上の SAP Business Suite からサポートされている任意のデータベースへの複写の場合のみ) 次の場合に、 **Replication Server** で複写時に最初の 2 バイトをトランケートする必要があるかどうかを指定します。
  - カラムのアプリケーションデータ型 (DDIC 型) が RAW または LRAW であり、かつ
  - DDIC 型の長さが 255 バイトを超えている
 有効な値は 'on' または 'off' (デフォルト) です。
- **reblock\_ddntf** - (DB2 UDB データベース上の SAP Business Suite からサポートされている任意のデータベースへの複写の場合のみ) **Replication Server** でレプリケートデータベースの各 **FIELDS** 列の長さを、最後のロー以外に 32768 に調整するかどうかを指定します。 有効な値は 'on' (デフォルト) または 'off' です。

『リファレンスマニュアル』の「alter connection」を参照してください。

## create database replication definition の変更点

**create database replication definition** の構文が変更され、新しいパラメータがいくつか追加されました。

```
create database replication definition db_repdef
  with primary at server_name.db
  [not replicate DDL] |
  [replicate DDL [{with | without} auto_update_table_list] |
  [{with | without} auto_extend_table_list]]
  [[not] replicate functions setcont]
  [[not] replicate transactions setcont]
  [[not] replicate system procedures setcont]
  [[not] replicate tables [[setcont [except setcont]] | in files
  ('file_path')]
  [[not] replicate {SQLDML | DML_options} [in table_list]]
  [user username password pass]]
setcont ::= [[in] ([owner1.]name1[, [owner2.]name2 [, ... ]]]]
```

**Replication Server 15.7.1 SP200** では、次のパラメータが追加されています。

- **[replicate DDL [{with | without} auto\_update\_table\_list] | [{with | without} auto\_extend\_table\_list]]** - **Replication Server** に対し、サブスクライブするデータベースに **DDL** を送信するよう指示します。また、テーブルリストを更新または拡張するかどうかを指示します。オプションなしで **replicate DDL** を指定する

と、DDL はレプリケートデータベースに送信されますが、テーブルは複写パスに追加されません。

- **replicate DDL with auto\_update\_table\_list** - DDL コマンドをレプリケートデータベースに送信します。テーブルで **pdb\_automark\_tables** が true の場合、DDL コマンドの **drop table** または **rename table** が検出されると、そのテーブルはテーブルリストで自動的に更新されます。
- **replicate DDL without auto\_update\_table\_list** - DDL コマンドをレプリケートデータベースに送信しますが、DDL コマンドの **drop table** または **rename table** が検出されても、テーブルリスト内のテーブルを更新しません。
- **replicate DDL with auto\_extend\_table\_list** - DDL コマンドをレプリケートデータベースに送信します。テーブルで **pdb\_automark\_tables** が true の場合、DDL コマンドの **create table** が検出されると、そのテーブルは自動的にテーブルリストに追加されます。
- **replicate DDL without auto\_extend\_table\_list** - DDL コマンドをレプリケートデータベースに送信しますが、DDL コマンドの **create table** が検出されても、テーブルをテーブルリストに追加しません。
- **[[not] replicate tables [[setcont [except setcont]]]** - テーブルをレプリケートデータベースに送信するかどうかを指定します。テーブルを複写するとき、テーブルごとに最大 1 個の句を使用できます。句を省略すると、Replication Server によりすべてのテーブルが複写されます。  
例外リストはテーブルリストとのみ使用できます。例外リストを使用すると、データベース複写定義に定義されたテーブルリストをさらに詳細に調整できます。例外リストはテーブルリストに優先します。
- **[[not] replicate tables in files ('file\_path')]** - *file\_path* で指定したテーブルをレプリケートデータベースに送信するかどうかを指定します。  
*file\_path* には、対象テーブルリストまたは除外テーブルリストが記述されています。一度に 1 つのファイルのみを指定できます。*file\_path* には絶対パスを指定する必要があります。

---

**注意：** プライマリ Replication Server を起動するユーザにこのファイルの読み取りパーミッションが必要です。

---

テーブル名の書式は、ファイルでもテーブルリストでも同じです。テーブルリストでは次の書式でテーブル名を指定できます。

- *ownername.tablename*
- *tablename* (テーブル名は *dbo.tablename* として格納されます)
- *\*.tablename*
- *ownername.\**
- *'\*x\*y'.a\*b'* (文字列にワイルドカードが埋め込まれています)

---

**注意：** **create database replication definition** を発行すると、部分ワイルドカードはテーブルリストや例外リストを含むすべてのリストで展開されます。単純

ワイルドカードは、例外リストで展開されてからシステムテーブルに格納されます。ワイルドカードの展開が必要な場合は、**user** と **password** の値を指定する必要があります。

対象テーブルリストまたは除外テーブルリストを使用するときは、次の次のガイドラインに従ってください。

- ファイル内ではテーブル名のデリミタとして改行文字を使用します。
- # で始まる行はコメントとして無視されます。
- 所有者名またはテーブル名の前の空白はトランケートされます。
- 所有者名の最大長は 30 文字です。

例外リストはテーブルリストとのみ使用できます。例外リストを使用すると、データベース複写定義に定義されたテーブルリストをさらに詳細に調整できます。例外リストはテーブルリストに優先します。

- **[user usernamepasswordpass]** - プライマリ Adaptive Server database または Replication Agent への接続用のユーザ ID とパスワード。プライマリテーブルから選択します。

テーブル名にワイルドカードを使用する場合は、*username* と *password* を指定する必要があります。

テーブル名にワイルドカードを使用する場合は、**username** と **password** の値を指定する必要があります。

**注意：** **user** と **password** の値は一度だけ使用され、RSSD には保存されません。

## 例 1

データベース複写定義 *dbrepdef* を作成します。DDL が複写され、テーブルがプライマリで複写対象としてマーク付けされている場合は、`create table` DDL コマンドが検出されると、テーブルが自動的に *table\_list* に追加されます。このコマンドは、テーブル *USER1.TABLE1* のデータと *TABLE2* のすべてのデータも複写します。ただし、*USER2* が所有する *TABLE2* は除きます。

```
create database replication definition dbrepdef
with primary at ds1.pdb1
replicate DDL
with auto_extend_table_list
replicate tables in (USER1.TABLE1, *.TABLE2) except in
(USER2.TABLE2)
```

## 例 2

データベース複写定義 *db\_repdef* を作成します。DDL が複写されますが、新しい DDL コマンドが検出されても *table\_list* は更新されません。このコマンドは、ファイル `/sap/user/tablelist.txt` で指定されたテーブルのデータも複写します。これには、テーブル *USER1.TABLE1* と、*TABLE2* の全データが含まれます。

たとえば、`tablelist.txt` には次のように記述されているとします。

```
=====
#user tables:
USER1.TABLE1
*.TABLE2
=====
```

```
create database replication definition db_repdef
with primary at ds1.pdb1
replicate DDL
replicate tables in files ('\sap/user/table_list.txt')
```

『リファレンスマニュアル』の「create database replication definition」を参照してください。

## alter database replication definition の変更点

**alter database replication definition** の構文が変更され、新しいパラメータがいくつか追加されました。

```
alter database replication definition db_repdef
with primary at data_server.database
{[not replicate DDL] |
[replicate DDL [{with | without} {auto_update_table_list |
auto_extend_table_list}]}] |
[not] replicate setname setcont |
[not] replicate {{SQLDML | DML_options} [in table_list]} |
[alter owner from current_table_owner to new_table_owner [for
table_name]] |
[{add | remove} tables {setcont}]}
[with dsi_suspended]
[user username password pass]
setcont ::= [[in] ([owner1.]name1[, [owner2.]name2 [, ... ])] | [in
files ('file_path')]]
setname ::= {tables | functions | transactions | system procedures}
```

Replication Server 15.7.1 SP200 では、次のパラメータが追加されています。

- **not replicate DDL** - Replication Server に対し、サブスクライブするデータベースに DDL を送信しないよう指示します。
- **replicate DDL [{with | without} {auto\_update\_table\_list | auto\_extend\_table\_list}]** - Replication Server に対し、サブスクライブするデータベースに DDL を送信するよう指示します。また、テーブルリストを更新または拡張するかどうかを指示します。オプションなしで **replicate DDL** を指定すると、DDL はレプリケートデータベースに送信されますが、テーブルは複写パスに追加されません。
- **replicate DDL with auto\_update\_table\_list** - DDL コマンドをレプリケートデータベースに送信します。テーブルで **pdb\_automark\_tables** が **true** の場合、DDL コマンドの **drop table** または **rename table** が検出されると、そのテーブルはテーブルリストで自動的に更新されます。

- **replicate DDL without auto\_update\_table\_list** - DDL コマンドをレプリケートデータベースに送信しますが、DDL コマンドの **drop table** または **rename table** が検出されても、テーブルリスト内のテーブルを更新しません。
- **replicate DDL with auto\_extend\_table\_list** - DDL コマンドをレプリケートデータベースに送信します。テーブルで **pdb\_automark\_tables** が **true** の場合、DDL コマンドの **create table** が検出されると、そのテーブルは自動的にテーブルリストに追加されます。
- **replicate DDL without auto\_extend\_table\_list** - DDL コマンドをレプリケートデータベースに送信しますが、DDL コマンドの **create table** が検出されても、テーブルをテーブルリストに追加しません。
- **add tables in {(table\_list) | files 'file\_path'}** - テーブルを複製パスに追加します。テーブルのリストを指定するか、またはファイルでテーブル名を指定できます。テーブルを複製パスに追加すると、そのテーブルのデータの複製が開始します。

---

**注意：** テーブルを追加または削除するときに指定できるのは、一度に 1 つのファイルのみです。 *file\_path* には絶対パスを指定する必要があります。

---

- **remove tables in {(table\_list) | files 'file\_path'}** - テーブルを複製パスから削除します。テーブルのリストを指定するか、ファイルにテーブル名を指定できます。
- **[in files ('file\_path')]** - 対象テーブルリストまたは除外テーブルリストが記述されているファイル。一度に 1 つのファイルのみを指定できます。 *file\_path* には絶対パスを指定する必要があります。

---

**注意：** プライマリ Replication Server を起動するユーザにこのファイルの読み取りパーミッションが必要です。

---

テーブル名の書式は、ファイルでもテーブルリストでも同じです。テーブルリストでは次の書式でテーブル名を指定できます。

- *ownername.tablename*
- *tablename* (テーブル名は *dbo.tablename* として格納されます)
- *\*.tablename*
- *ownername.\**
- *'\*x\*y'.a\*b'* (文字列にワイルドカードが埋め込まれています)

---

**注意：** **create database replication definition** を発行すると、部分ワイルドカードはテーブルリストや例外リストを含むすべてのリストで展開されます。単純ワイルドカードは、例外リストで展開されてからシステムテーブルに格納されます。ワイルドカードの展開が必要な場合は、**user** と **password** の値を指定する必要があります。

---

対象テーブルリストまたは除外テーブルリストを使用するときは、次の次のガイドラインに従ってください。

- ファイル内ではテーブル名のデリミタとして改行文字を使用します。
- # で始まる行はコメントとして無視されます。
- 所有者名またはテーブル名の前の空白はトランケートされます。
- 所有者名の最大長は 30 文字です。
- **[user usernamepasswordpass]** - プライマリ Adaptive Server database または Replication Agent への接続用のユーザ ID とパスワード。プライマリテーブルから選択します。  
テーブル名にワイルドカードを使用する場合は、**username** と **password** の値を指定する必要があります。

---

**注意：** **user** と **password** の値は一度だけ使用され、RSSD には保存されません。

『リファレンスマニュアル』の「alter database replication definition」を参照してください。

## create subscription の変更点

**create subscription** の構文が変更され、新しいパラメータがいくつか追加されました。

```
create subscription sub_name
for {table_repdef | func_repdef | publication pub |
    database replication definition db_repdef }
    [ with primary at server_name.db ]
with replicate at data_server.database
[where {column_name | @param_name}
    {< | > | >= | <= | = | &} value
[and {column_name | @param_name}
    {< | > | >= | <= | = | &} value]...]
[without holdlock [direct_load [init replicate table with {create |
create_or_truncate | truncate | recreate}]]
[user username password pass][num_of_selects selects]
[hold_resource_on_error]]
incrementally | without materialization]
[subscribe to truncate table] [for new articles]
```

Replication Server 15.7.1 SP200 では、次のパラメータが追加されています。

- **init replicate table with create** - レプリケートデータベースにテーブルを作成します。テーブルがすでに存在する場合、マテリアライゼーションは失敗します。
- **init replicate table with create\_or\_truncate** - レプリケートデータベースにテーブルを作成します。テーブルがすでに存在する場合、Replication Server はトランケーション後、既存のテーブルを使用します。
- **init replicate table with create\_or\_truncate** - レプリケートデータベースのテーブルをトランケートします。テーブルが存在しない場合、マテリアライゼーションは失敗します。
- **init replicate table with recreate** - レプリケートデータベースのテーブルを削除して、テーブルを作成しなおします。

---

**注意：**レプリケートテーブルがまだ存在してなくても、マテリアライゼーションは失敗しません。

---

- **hold\_resource\_on\_error - direct\_load** マテリアライゼーションで作成するサブスクリプションがエラーになった場合、サブスクリプションリソースを保持します。デフォルトでは、**direct\_load** オプションを指定して作成するサブスクリプションがエラーになった場合、リソースは保持されません。
- **num\_of\_selectsselects - direct\_load** オプションを指定して作成するサブスクリプションのマテリアライゼーションパフォーマンスを向上させるために、複数の選択スレッドを有効にします。デフォルト値は 1 です。有効な値は 1 ~ 10 です。このオプションを使用できるのは、**direct\_load** オプションを指定する場合のみです。また、プライマリデータベースが IBM DB2 9.7 以降か Oracle のどちらかであり、ファンクション文字列 **rs\_select** がカスタマイズされていない場合にしか使用できません。これらの条件を満たさない場合や、プライマリテーブルのロー数があまり多くない場合、この数値は 1 に下げられます。

『リファレンスマニュアル』の「create subscription」を参照してください。

## activate subscription の変更点

**activate subscription** の構文が変更され、新しいパラメータが追加されました。

```
activate subscription sub_name
for {table_rep_def | function_rep_def |
publication pub_name
with primary at data_server.database}
with replicate at data_server.database
[with suspension [at active replicate only] | with catchup_queue]
```

Replication Server 15.7.1 SP200 では、**with catchup\_queue** パラメータが追加されます。**with catchup\_queue** 句を使用すると、キャッチアップキューを開始してプライマリテーブルの DML を格納するよう Replication Server に指示できます。その後のプライマリテーブルの更新がキャッチアップキューに格納されます。バルクマテリアライゼーションの完了後に **validate subscription** コマンドを発行すると、サブスクリプションが VALID になる前に、キャッチアップキューにあるすべての DML オペレーションがレプリケートテーブルに適用されます。このため、バルクマテリアライゼーションの実行中にプライマリテーブルでレプリケート DSI をサスペンドしたり、DML を制限したりする必要がありません。

参照:

- 『管理ガイド 第 1 巻』の「activate subscription」コマンド
- 『リファレンスマニュアル』の「activate subscription」

## システム管理ツール

Replication Management Agent (RMA) を使用して SAP HANA データベースへの複製を設定したり、SAP Control Center の各ツールを使用して複製システムを管理したりできます。

### 参照：

- RMA を使用した SAP HANA データベースへの複製の設定 (7 ページ)

## SAP Control Center for Replication

SAP Control Center for Replication バージョン 3.3 を使用して複製環境の管理とモニタリングを行います。

## Sybase Central の廃止

Sybase Central は、SAP または Sybase のどの Web サイトからもダウンロードできなくなりました。複製環境の管理には SAP Control Center を使用します。

SAP Replication Server のマニュアル群で「SAP Control Center for Replication」を参照してください。

### SAP Control Center for Replication と Sybase Central for Replication の比較

複製システム管理のサポートについて、SAP Control Center for Replication バージョン 3.3 (SCC Replication) と、バージョン 15.0 以降の Sybase Central コンポーネント、Replication Monitoring Services (RMS) および Replication Manager Plug-in (RMP) を比較します。

機能	SCC for Replication 3.3	RMP 15.x	RMS 15.x
SAP Replication Server と Replication Agent のステータスのモニタリング	X	X	X
SAP Replication Server と Replication Agent の可用性のモニタリング	X	X	X
複製パフォーマンスのモニタリング	X		

機能	SCC for Repli- cation 3.3	RMP 15.x	RMS 15.x
エンドツーエンド遅延のモニタリング (rs_ticket ハートビート)	X		X
SAP 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		
SAPIQ 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		
SAP 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		
SAP 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		
パフォーマンス統計履歴情報の図示	X		
複写トポロジの表示	X		
複写パスのトレース	X		
ヒートチャートの表示	X		
アラート通知の設定	X		X
SAP Replication Server のユーザセキュリティと役割の設定	X		
SAP 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		
SAP Replication Server と Replication Agent の設定	X	X	X
SAP Replication Server と Replication Agent のサスペンドと再開	X	X	X
SAP 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

機能	SCC for Repli- cation 3.3	RMP 15.x	RMS 15.x
RCL と SQL エディタを使用した SAP 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 SAP ASE 複写環境の設定の簡略化		X	
SAP ASE RepAgent スレッドの設定		X	
SAP ASE 自動マテリアライゼーションの設定		X	
SAP Replication Server 接続の作成と削除		X	
SAP Replication Server 論理接続の作成と削除		X	
SAP Replication Server ルートの作成と削除		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	
SAP 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	
SAP Replication Server ユーザの作成、変更、削除		X	
SAP Replication Server キューデータの表示		X	
SAP Replication Server 例外ログの管理		X	

## システムテーブル

SAP Replication Server 15.7.1 SP200 では、いくつかの新しいシステムテーブルが追加されたほか、既存のシステムテーブルに変更が加えられています。

テーブルおよび関連するインデックスについては、『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。

### 新しいシステムテーブル

- `rs_dependtbls` - データベース複製定義への追加または削除が保留中のテーブルに関する情報を格納します。
- `rs_schemamap` - プライマリデータベースとレプリケートデータベースとのスキーママッピングに関する情報を格納します。

### 既存のシステムテーブルの変更点

- `rs_articles` - (`objid`) に対するインデックスが追加されました。
- `rs_dbsubsets` - `status` カラムと `priority` カラムが追加されました。
- `rs_dictionary` - (`words`) に対するユニークなクラスタードインデックスが追加されました。
- `rs_encryptionkeys` - (`name`) に対するユニークなクラスタードインデックスが追加されました。
- `rs_passwords` - 次のものが追加されました。
  - `rs_passwords` の `uid` カラムと `rs_users` の `uid` カラムとの 1 対 1 関係
  - (`uid`) のインデックス
- `rs_recovery` - `text` カラムのデータ型の長さが `binary (512)` に変更されました。
- `rs_subscriptions` - 次のものが追加されました。
  - `released`, `mechanism`, `tableowner`, `tablename`, `prsid`, `num_selects` の各カラム
  - `rs_subscriptions` の `prsid` カラムと `rs_dbsubset` の `prsid` カラムとの 1 対 1 関係
  - `released` カラムが (`subid`, `recovering`, `error_flag`, `materializing`, `dematerializing`, `primary_sre`, `replicate_sre`) に対するインデックスに追加されました。
  - (`recovering`, `requestdate`) に対するインデックス
- `rs_targetobjs` - (`dbid`, `objname`, `objowner`, `objtype`) に対するユニークなクラスタードインデックスが追加されました。

## バージョン 15.7.1 SP110 の新機能

SAP Replication Server 15.7.1 SP110 では、直接ロードマテリアライゼーション用の新しい設定パラメータが追加されています。

### 直接ロードマテリアライゼーションの設定パラメータ

**alter connection** が変更され、直接ロードマテリアライゼーションの設定パラメータが追加されました。

この設定の詳細については、『管理ガイド 第1巻』の「直接ロードマテリアライゼーション」と『リファレンスマニュアル』の「**alter connection**」を参照してください。

設定パラメータ	説明
<b>dsi_check_unique_key</b>	<p>直接ロードマテリアライゼーション時にプライマリデータベーステーブルのユニークキーをチェックする。このパラメータをオン設定した場合、ユニークキーのないプライマリテーブルではアクティビティをいっさい実行できない。実行すると、キャッチアップフェーズでサブスクリプションが失敗する可能性があるため、サブスクリプションがアボートされ、エラーとマークされる。</p> <p>サブスクリプションがアボートされたら、それを削除して作成しなおす必要がある。サブスクリプションを再作成する前に(テーブルのトランケートと同様に)レプリケートテーブルをクリーンアップする必要がある。</p> <p>デフォルト値: オン</p>



## バージョン 15.7.1 SP102 の新機能

SAP Replication Server 15.7.1 SP102 では、ASE 間の直接ロードマテリアライゼーションがサポートされています。

### ASE 間の直接ロードマテリアライゼーション

SAP Replication Server 15.7.1 SP102 では、プライマリ SAP ASE からレプリケート SAP ASE への直接ロードマテリアライゼーションを使用したデータのマテリアライズがサポートされています。

直接ロードマテリアライゼーションの設定と SAP ASE から SAP ASE への複写の有効化については、『管理ガイド第 1 巻』の「Adaptive Server 間の直接ロードマテリアライゼーション」を参照してください。



## バージョン 15.7.1 SP100 の新機能

Replication Server 15.7.1 SP100 では、データベースサポート、パフォーマンス、利便性、およびセキュリティが強化されています。

### リリースバージョン番号の変更

---

これまで Sybase® のお客様にはメジャーリリースまたはマイナーリリースに続く ESD (Electronic Software Delivery) として知られてきたソフトウェアパッチを SP (サポートパッケージ) と呼ぶようになりました。SP には最大 3 桁までの数字が割り振られます。

『SAP® 製品リリース方針 - すべての主要ソフトウェアリリース』 (<https://service.sap.com/releasestrategy>) を参照してください。このバージョン番号の変更によるアップグレード手順やダウングレード手順の変更はありません。

### Replication Server による SAP HANA データベースのサポート

---

Replication Server ではレプリケートデータベースとして SAP HANA データベースをサポートしています。

Replication Server では、ExpressConnect for SAP HANA データベースを介して次のプライマリデータベースから SAP HANA データベースインスタンスへのデータの複写をサポートしています。

- Adaptive Server
- Oracle
- Microsoft SQL Server
- DB2 UDB (Linux、UNIX、Windows)

Replication Server では、これらのデータベースから SAP HANA データベースへのデータの直接ロードマテリアライゼーションもサポートしています。

## ExpressConnect for SAP HANA データベースと SAP HANA データベースに対するレプリケートデータベースのコネクティビティ

Replication Server では、ExpressConnect for SAP HANA データベースを使用して SAP HANA レプリケートデータベースに接続できます。

### *ExpressConnect for SAP HANA データベースの使用*

Replication Server データベースから SAP HANA データベースへの接続には次の 2 種類があります。

- セキュア。この接続ではデータベース接続で指定された **hdbuserstore** キーを使用します。
- 標準。この接続では interfaces ファイルにある SAP HANA データベースのホストとポート番号のエントリを使用します。

---

**注意：** ExpressConnect for SAP HANA データベースは、pSeries Linux を除き、Replication Server と同じオペレーティングシステムをサポートしています。Linux では Red Hat Enterprise Linux 6 と SuSE Linux Enterprise Server (SLES) 11 のみにサポートが限定されます。

---

**注意：** ExpressConnect for SAP HANA データベースには、必要な SAP HANA データベース ODBC ドライバが付属していません。Replication Server のインストール後、ExpressConnect for SAP HANA データベースを使用する前に、これらのライブラリを SAP Support MarketPlace からダウンロードしてインストールしてください。『Replication Server リリースノート』>「インストールに関する特別な指示」>「ExpressConnect for SAP HANA データベース用 ODBC ライブラリのインストール」を参照してください。

---

## プロパティ、ファンクション文字列、およびデータ型変換

Replication Server には、SAP HANA データベースをサポートするための新しいファンクション文字列クラス、新しい接続プロファイル、およびレプリケートデータベースオブジェクトが用意されています。

新しいファンクション文字列 `rs_hanadb_function_class` が Replication Server に追加されました。これらのファンクション文字列は、SAP HANA データベースサーバと通信し、テーブルやプロシージャにアクセスできるように設計されています。

Replication Server には、SAP HANA への複写用として次のような新しい接続プロファイルが用意されています。

- `rs_ase_to_hanadb` - Adaptive Server から SAP HANA データベースへのクラスレベル変換をインストールします。

- `rs_oracle_to_hanadb` - Oracle から SAP HANA データベースへのクラスレベル変換をインストールします。
- `rs_udb_to_hanadb` - DB2 UDB から SAP HANA データベースへのクラスレベル変換をインストールします。
- `rs_msss_to_hanadb` - Microsoft SQL Server から SAP HANA データベースへのクラスレベル変換をインストールします。

## 直接ロードマテリアライゼーション

直接ロードマテリアライゼーションを使用すると、各種のプライマリデータベースと SAP HANA データベースの間でデータをマテリアライズできます。

直接ロードマテリアライゼーションを使用してマテリアライズできるデータは次のとおりです。

- Adaptive Server から SAP HANA データベース
- Microsoft SQL Server から SAP HANA データベース
- Oracle から SAP HANA データベース
- DB2 UDB から SAP HANA データベース

---

**注意：** 直接ロードマテリアライゼーションでは、Adaptive Server データベースへのデータのマテリアライズはサポートされていません。

---

直接ロードマテリアライゼーションを有効にするには、**create subscription** コマンドの **direct\_load** オプションを使用します。直接ロードマテリアライゼーションを使用するときは、**create subscription** に関する次の制約に注意してください。

- **direct\_load** オプションを使用すると、同じレプリケートテーブルに対して同時に他のサブスクリプションを作成したり定義したりできません。
- **direct\_load** オプションはテーブル複写定義へのサブスクリプション専用であり、**without holdlock** とともに使用します。**without materialization** オプションや **incrementally** オプションとは併用できません。
- **user** オプションと **password** オプションは、**direct\_load** とのみ一緒に使用できません。
- 論理接続や代替接続に対しては **direct\_load** オプションを使用できません。複写定義でのプライマリ接続とサブスクリプションでのレプリケート接続は物理接続であることが必要です。
- プライマリデータベースのメンテナンスユーザを **user** オプションと **password** オプションで使用してサブスクリプションを作成することはできません。
- プライマリデータベースが Adaptive Server でない場合は、他の自動マテリアライゼーション方法を使用できません。Oracle などのデータベースの自動マテリアライゼーション方法は直接ロードマテリアライゼーションのみです。レプ

リケートデータベースが Adaptive Server でない場合、**with purge** オプションでサブスクリプションを削除することはできません。

- **direct\_load** オプションが使用できるのは、レプリケート Replication Server のサイトバージョンとルートバージョンが 1571100 以降の場合のみです。
- **direct\_load** オプションを使用して作成されたサブスクリプションでは、ローフィルタリング、名前マッピング、カスタムファンクション文字列、およびデータ型マッピングを使用できます。
- **direct\_load** オプションでサブスクリプトを作成しようとしても、作成されるサブスクリプションの数が **num\_concurrent\_subs** 以上になる場合は、Replication Server によって拒否されます。

#### プライマリデータベースに関する注意事項

- Replication Server はプライマリデータベースからデータを直接マテリアライズするとき、Adaptive Server 以外のデータベースの場合は Replication Agent に接続し、Adaptive Server の場合はプライマリデータベースに直接接続します。
- 直接ロードマテリアライゼーションによって Adaptive Server 以外のプライマリデータベースからデータをマテリアライズするには、Replication Agent バージョン 15.7.1 SP100 以降が必要です。
- **create subscription** コマンドを呼び出すとき、Replication Server は Replication Agent 管理者のログイン名で Replication Agent に接続します。

#### 直接ロードマテリアライゼーションの設定パラメータ

直接ロードマテリアライゼーションでは次の設定パラメータを使用します。

Replication Server のインストール後、**rs\_init** によってデフォルトの設定パラメータが設定されます。

**configure replication server** を使用してこれらのパラメータを変更する方法については、『Replication Server 管理ガイド 第 1 巻』>「複写システムの管理」>「Replication Server 設定パラメータの設定」>「Replication Server パラメータの変更」を参照してください。

表 4：直接ロードマテリアライゼーションの設定パラメータ

設定パラメータ	説明
<code>mat_load_tran_size</code>	<p>直接ロードマテリアライゼーションでプライマリデータをレプリケートテーブルに初期コピーするときの最適なトランザクションサイズまたはバッチサイズを指定する。</p> <p>有効な値: 10 ~ 2147483646</p> <p>デフォルト値: 10000</p>
<code>max_mat_load_threads</code>	<p>マテリアライズされるテーブルごとの最大ロードスレッド数を指定する。</p> <p>有効な値: 1 ~ 20</p> <p>デフォルト値: 5</p> <p>Replication Server は、テーブルごとに 1 個のロードスレッドで直接ロードマテリアライゼーションを開始し、必要に応じてこのパラメータで指定された数までスレッドを追加生成する。</p> <p><code>max_mat_load_threads</code> は、ローカルの Replication Server とデータベースの接続パラメータである。</p> <p>このパラメータと <code>num_concurrent_subs</code> の値によって直接ロードマテリアライゼーションのリソース使用が制御される。</p>

## Replication Server による Adaptive Server のサポート

Replication Server 15.7.1 SP100 は、Adaptive Server Enterprise 15.7 SP100 との互換性が検証されています。

## Replication Server における圧縮データのサポート

Replication Server は Adaptive Server データの圧縮を解除してからレプリケートデータベースに送信します。

Adaptive Server プライマリデータベースと、LOB スキーマ、文字セット、エンディアン、バージョン、およびページサイズが同じか異なる Adaptive Server データベース、または種類の異なるレプリケートデータベースの間のデータ圧縮を Replication Server がサポートするかどうかは、データがロー内かロー外かによって異なります。『Replication Server 管理ガイド 第1巻』の「Replication Server における圧縮データのサポート」を参照してください。

LOB 圧縮データのサブスクリプションマテリアライゼーションのサポートは、次の項目に依存します。

- 複写定義でのカラムデータ型の指定方法
- バージョン 15.7.1 SP100 以降の Replication Server が 1 つ以上あるかどうか、および複数の Replication Server からなる複写システムの中でその Replication Server が占める他の Replication Server に対する相対位置

『Replication Server 管理ガイド 第 1 巻』の「LOB 圧縮データのサブスクリプションマテリアライゼーション」を参照してください。

## バックリンクポインタとデータベースの縮小

Replication Server では、Adaptive Server のバックリンクポインタとデータベースの縮小をサポートしています。

Transact-SQL の `writetext` コマンドを複写するには、データベースが LOB データを格納するテキストページを指すデータローにアクセスする必要があります。このデータローへのアクセスを許可にするために、Adaptive Server は最初のテキストページのバックリンクポインタまたは複写用に作成されたインデックスを使用します。カラム、テーブル、またはデータベースレベルでインデックスを作成するプロセスでは、複写をサポートするための情報を提供する負荷の高い操作が必要になります。

以前のバージョンからアップグレードしたのではない、Adaptive Server version 15.7 SP100 以降のデータベースでは、Adaptive Server はデフォルトでデータベースへの LOB バックリンクポインタを作成して管理するので、`sp_reptostandby` がすぐに有効になります。したがって、テーブルの複写を設定するときにインデックスを作成する必要はありません。LOB カラムの複写に必要な情報がバックリンクポインタの形ですでに利用できる場合、Adaptive Server は `sp_reptostandby`、`sp_setrepcol`、および `sp_setreptable` の `use_index` パラメータを無視します。

ただし、Adaptive Server 15.7 SP100 より前のバージョンで作成したデータベースを使用している場合や、そのデータベースからアップグレードした場合は、インデックスの作成によって複写の設定に時間がかかることがあります。処理時間を短縮するには、該当するレベル(カラム、テーブル、またはデータベース)で `dbcc shrinkdb_setup` を実行してバックリンクポインタを作成し、バックリンクを最新の状態にします。

`dbcc shrinkdb_setup` は、以前に `use_index` でマーク付けしたカラム、テーブル、またはデータベースの複写インデックスを `suspect` (疑わしい) としてマーク付けします。これらのオブジェクトのインデックスは、`dbcc shrinkdb_setup` の実行後には必要ないため、`dbcc reindex` を使用して削除できます。

『Adaptive Server Enterprise 15.7 SP100 新機能ガイド』の「データベースの縮小」を参照してください。

## データベースオブジェクトの所有権の譲渡

Replication Server では、Adaptive Server におけるデータベースオブジェクトの所有権の譲渡をサポートしています。

Adaptive Server の従業員ライフサイクル管理をサポートするために、データベース管理者、システムセキュリティ担当者、データベース所有者は従業員の変化に応じてデータベースオブジェクトの割り当てを管理できます。それには、**alter...**

**modify owner** Adaptive Server コマンドを使用してデータベースオブジェクトの所有権を譲渡します。また、データベース管理者は **alter... modify owner** を使用してオブジェクトの所有権からオブジェクトの作成を分離できます。『Adaptive Server Enterprise セキュリティ管理ガイド』>「ユーザパーミッションの管理」>「データベースオブジェクトの所有権の変更」を参照してください。

サイトバージョン 1550 以降の Replication Server 15.7.1 SP100 では、複写定義内のオブジェクト所有権の譲渡もサポートしています。Replication Server は **alter... modify owner** によって生成された DDL の変更を複製します。これにより、プライマリ Adaptive Server データベースでのオブジェクト所有権の譲渡がレプリケート Adaptive Server データベースに適用されます。ただし、所有者名が必要とされる複写定義において所有者で修飾されたオブジェクトのオブジェクト所有権が変更された場合、Replication Server で自動的に複写定義を変更してその変更を適用することはありません。データが継続的に正しく複製されるように、オブジェクト所有権の変更と影響を受ける複写定義の対応する変更を整合させてください。

---

**注意：** 複写定義内のオブジェクト所有権の譲渡をサポートするには、Replication Server のサイトバージョンが 1550 以降である必要があります。

---

『Replication Server 管理ガイド 第 1 巻』の「データベースオブジェクトの所有権の譲渡」を参照してください。

## 事前計算済み結果セット

Replication Server では、事前計算済み結果セットをサポートするプライマリ Adaptive Server とレプリケート Adaptive Server 間で事前計算済み結果セット DDL コマンドの複製をサポートしています。

Adaptive Server 15.7 ESD #2 以降では、すべてのクエリ式に対して作成可能な事前計算済み結果セットとともに、事前計算済み結果セットをメンテナンスするための一連のポリシーをサポートしています。『Adaptive Server Enterprise 新機能ガイド バージョン 15.7 ESD #2』の「事前計算済み結果セット」を参照してください。

Replication Server は、現在 Adaptive Server データベース間の DDL コマンドを複製している環境では、事前計算済み結果セットの DDL コマンドを複製します。ただし、事前計算済み結果セットを複製対象としてマークすることはできません。ま

た、データ操作言語 (DML) コマンドで事前計算済み結果セットを直接更新することもできません。

『Replication Server 管理ガイド 第 1 巻』の「事前計算済み結果セット」を参照してください。

## 細密なパーミッション

Adaptive Server の細密なパーミッションの複写をサポートするよう複写システムを設定します。

Adaptive Server の細密なパーミッションを使用すると、システム権限を付与する際に、サイト固有の役割を要件に一致する権限で構成したり、システム管理者やデータベース所有者によるユーザデータへのアクセスを制限したりできます。『Adaptive Server Enterprise セキュリティ管理ガイド』の「細密なパーミッションの使用」を参照してください。

Replication Server では Adaptive Server の **sp\_restore\_system\_role** システムプロシージャをサポートしています。このプロシージャを使用すると、細密なパーミッションをシステム標準の役割に適用した後で、その役割をデフォルトの権限の状態に戻すことができます。

複写システム内のデータベースで細密なパーミッションを使用し、次のことを行う場合、複写をサポートするよう複写システムを設定します。

- 次の目的で **rs\_init** を使用する
  - ユーザデータベースメンテナンスユーザ ID と RSSD プライマリユーザ ID を設定する
  - Replication システムにデータベースを追加する
  - ユーザデータベースをアップグレードする
- Adaptive Server の master データベースをレプリケートデータベースとして複写システムに追加する
- データベース複写定義を使用して DDL 文を複写する

複写をサポートするには、『Replication Server 管理ガイド 第 1 巻』の「細密なパーミッション」を参照してください。

## ストアドプロシージャの複写と execute as 句

Adaptive Server **create procedure** コマンドの **execute as {owner | caller}** 句を使用してストアドプロシージャを作成するときは、ストアドプロシージャの複写を確実にサポートしてください。

『Adaptive Server Enterprise セキュリティ管理ガイド』の「execute as owner または execute as caller を含むプロシージャの実行」を参照してください。

**execute as {owner | caller}** 句を使用してプロシージャを作成した場合、プライマリデータベースでの呼び出し元の ID がレプリケートデータベースでは必ずしも保持

されません。ID が異なると、ストアドプロシージャの複写が失敗する可能性があります。

ストアドプロシージャの複写を正常に実行するには、『Replication Server 管理ガイド 第 1 巻』の「ストアドプロシージャの複写と **execute as** 句」を参照してください。

## RepAgent のサポート

Replication Server は、RepAgent for Adaptive Server Enterprise 15.7 SP100 との互換性が検証されています。

### RepAgent スキーマスキーマキャッシュメモリの管理

Adaptive Server 15.7 SP100 では、各 RepAgent スキーマスレッドが複写に必要なオブジェクトスキーマを格納するために使用できる最大のメモリ量を設定することで、Adaptive Server RepAgent のメモリ消費を最適化し、RepAgent のパフォーマンスを向上させることができます。

Adaptive Server RepAgent **sp\_config\_rep\_agent** システムプロシージャの **max schema cache per scanner** パラメータを使用すると、各 RepAgent スキーマスレッドが格納および処理できるトランザクション数を増やすことができます。これにより、複写のパフォーマンスが向上します。キャッシュが満杯になると、古いオブジェクトスキーマをフラッシュして新しいオブジェクトスキーマのための領域を確保します。

『Replication Server 管理ガイド 第 2 巻』の「最大スキーマスキーマキャッシュサイズの設定」を参照してください。

### *schema cache grow factor* パラメータの廃止

RepAgent スキーマキャッシュメモリ管理をより効果的に調整するパラメータとして、**schema cache grow factor** は **max schema cache per scanner** に置き換えられました。

## Multi-Path Replication

Adaptive Server RepAgent では、Multi-Path Replication™ のサポートが強化されています。

## フィルタ別分散

Adaptive Server 15.7 SP100 では、Multi-Path Replication の RepAgent フィルタ別分散モデルを選択できます。

フィルタ別分散では、1つのプライマリテーブル内のローにある1つ以上のカラムのデータ値に基づいてフィルタを定義し、そのフィルタを特定の複製パスにバインドします。RepAgent はフィルタ条件に一致するローデータのみを、そのパスを介して送信します。

---

**注意：**カラムフィルタ別分散を選択する前に複数のスキャナを有効にする必要があります。

---

『Replication Server 管理ガイド 第2巻』の「フィルタ別分散」を参照してください。

## 複数のスキャナ

Adaptive Server 15.7 SP100 では、Adaptive Server RepAgent に対して複数のスキャナスレッドを有効にすることにより、使用可能なすべての複製パスを利用して、複製パフォーマンスを向上させることができます。

パスごとに複数の専用スキャナスレッドを用意することで、すべてのスキャナスレッドが並列でデータベースログをスキャンし、LTL コマンドを生成できるようになります。これにより、次のことが実現できます。

- 送信元スレッドがスキャナスレッドからの LTL コマンドを待機する時間が短縮し、スループットが向上する。
- 各スキャナはそれぞれのパスが必要とする LTL コマンドをのみを生成するので、メモリ消費量が減少する。

---

**注意：**RepAgent で Multi-Path Replication のフィルタ別分散モデルを使用するよう設定する前に、複数のスキャナを有効にする必要があります。

---

『Replication Server 管理ガイド 第2巻』の「複数のスキャナ」を参照してください。

## セカンダリトランケーションポイントの管理

Multi-Path Replication 環境でセカンダリトランケーションポイントの管理を強化するには、Adaptive Server RepAgent のトランケーションポイント要求間隔を設定します。

スキャナが 1 つの Multi-Path Replication 環境では、複製パスによってデータを受信する頻度が異なるため、プライマリ Adaptive Server データベースログにおけるデータ分布が不均一になる可能性があります。したがって、RepAgent が各バッチで Replication Server に送信する最大ログレコード数の指定に基づいてトランケーションポイント要求を行うと、要求頻度が低くなります。結果として、トランザ

クシオン量が最も少ない複製パスと同じ頻度でしかセカンダリトランケーションポイントが移動しません。最初のパスで複製されるコマンド数にかかわらず、2番目のパスによって最低数のコマンドの処理が完了するまでトランケーションは発生しません。トランケーションポイントの移動が遅い場合や、まったく移動しない場合、Adaptive Server のトランザクションログはトランケート可能になるまで拡大します。

『Replication Server 管理ガイド 第1巻』の「セカンダリトランケーションポイントの管理」を参照してください。

## パフォーマンスの強化

---

Replication Server 15.7.1 SP100 では、複製パフォーマンスの向上を目的とした機能強化が行われています。

### インクリメンタル解析

インクリメンタル解析を有効にすると、HVAR、RTL、および DSI バルクコピーインの複製環境で、特に複製ルートに複数の Replication Server がある場合に、複製のパフォーマンス向上とメモリ消費量の低減を実現できます。

すべてのデータを解析し、DSI エグゼキュータ (DSI-E) の各スレッドを順番に待機させる代わりに、インクリメンタル解析では DSI スケジューラ (DSI-S) スレッドでテーブルレベルの情報のみを解析できます。これは、DSI-S が処理する DML コマンドの最初の部分にあたります。DSI-S スレッドは、DML コマンドの2番目の部分に対する解析タスクを DSI-E スレッドに転送します。

『Replication Server 管理ガイド 第2巻』の「インクリメンタル解析」を参照してください。

『Replication Server 管理ガイド 第2巻』の「Adaptive Server への High-Volume Adaptive Replication」と「DSI バルクコピーイン」を参照してください。

『Replication Server 異機種間複製ガイド』の「Real-Time Loading ソリューション」を参照してください。

## 利便性の向上

---

Replication Server 15.7.1 SP100 では、利便性の向上を目的とした変更がいくつか加えられています。

## キューの処理時間の推定

`admin sqm_process_time` を使用して、Replication Server でキュー内の残りのデータの処理にかかる推定時間を表示できます。

例と使用法については、『Replication Server リファレンスマニュアル』の「`admin sqm_process_time`」を参照してください。

## 自動でサイズ変更可能なパーティション

使用状況に応じて自動的に拡大したり縮小したりする Replication Server パーティションを作成します。

自動でサイズ変更可能なパーティションでは、ステアブルキューメッセージの格納により消費が増大したとき、手動でパーティションのサイズを大きくしたり、新しいパーティションを作成する必要がありません。また、消費が減少した場合にも、ディスク領域を節約するために手動でパーティションサイズを小さくする必要はありません。

自動でサイズ変更可能なパーティションを作成するには、『Replication Server 管理ガイド 第 1 巻』の「自動でサイズ変更可能なパーティション」と『Replication Server リファレンスマニュアル』の「`create auto partition path`」を参照してください。

## 複写定義でサポートされていない引用符付き識別子

Replication Server では、複写定義またはプライマリ Adaptive Server データベースで識別子が引用符で囲まれているかどうかを確認せずに、すべての識別子に引用符を追加できます。

参照:

- 『Replication Server リファレンスマニュアル』の「`alter connection`」の `dsi_quoted_identifier` パラメータ
- 『Replication Server 管理ガイド 第 1 巻』の「引用符付き識別子」

## 高速複写モードのパフォーマンスに関するトラブルシューティング

高速複写モードの使用中に発生した複写エラーの原因を調べるには、Replication Server が連続ログ順複写モードに戻る前に複写をサスペンドします。

`dsi_retry` を使用して、HVAR、RTL、動的 SQL、DSI バルクコピーイン、並列 DSI、または連続ログ順言語複写モードで複写を継続できなかったら複写をサスペンドするかどうかを指定します。その後、Replication Server ログファイルを分析するかキューをダンプして、失敗したトランザクションに関する情報を取得し、トランザクションを適用できなかった原因を調べることができます。

『トラブルシューティングガイド』の次の項目を参照してください。

- **dsi\_retry** の使用が必要になる可能性のあるシナリオについては、「高速モードで複写を継続できない」を参照してください。
- **dsi\_retry** を使用した場合に表示される可能性があるエラーメッセージの例については、「**dsi\_retry** の設定オプションとエラーメッセージの例」を参照してください。

## データ消失ステータス

データの整合性をモニタリングし、キューにおけるデータ消失の可能性をチェックするために、Replication Server 15.7.1 SP100 では **admin health** と **admin who, sqm** の出力に Loss Status カラムが追加されています。

以前のバージョンの Replication Server では、Replication Server ログで消失ステータスを示すメッセージがないかどうか確認する必要がありました。

これらのコマンドの出力例については、『Replication Server リファレンスマニュアル』の「**admin health**」と「**admin who**」を参照してください。

## ユニークキーがないテーブルの複写

いくつかの制限がありますが、Replication Server ではテーブルにユニークキーがない場合でも Adaptive Server へのテーブルの複写をサポートしてします。

ユニークキーがないテーブルの複写をサポートするには、**alter connection** または **configure replication server** を使用して **dsi\_top1\_enable** パラメータを設定します。

制限については、『Replication Server 管理ガイド 第1巻』の「ユニークキーがないテーブルの複写」を参照してください。

## セキュリティの向上

Replication Server 15.7.1 SP100 では、セキュリティの向上を目的とした変更がいくつか加えられています。

## Replication Server のプリンシパル名

Replication Server は、ユーザが定義した Replication Server プリンシパル名を使用した Kerberos Open Client™ 接続を受け入れます。

Replication Server は、プリンシパル名を Kerberos KDC (Key Distribution Center) で認証します。デフォルトのプリンシパル名は Replication Server の名前です。複数の Replication Server インスタンスを実行している場合は、Replication Server ごとに異なるプリンシパル名を指定する必要があります。

異なるプリンシパル名を指定する手順は次のとおりです。

- Replication Server の起動時に、**repserver** 実行プログラムに **-k rs\_principal\_name** オプションを設定する。または
- Replication Server を起動する前に、**isql** コマンドラインツールで **SYBASE\_RS\_PRINCIPAL** 環境変数を設定します。

Replication Server のプリンシパル名を表示するには、**admin show\_principal\_name** を使用します。

いずれかの Replication Server のプリンシパル名を変更したら、**sysadmin principal\_users[,reload]** を実行して、**rs\_principal\_users.cfg** 設定ファイルに格納されているすべての Replication Servers のプリンシパル名を再ロードしてください。

『Replication Server 管理ガイド: 第 1 巻』の「Replication Server プリンシパル名の指定」と「Kerberos シナリオによる Adaptive Server 間の複写」を参照してください。

『Replication Server リファレンスマニュアル』の「repserver」を参照してください。

『Replication Server リファレンスマニュアル』の「**admin show\_principal\_name**」と **sysadmin principal\_users[,reload]**」の各コマンドを参照してください。

## LDAP ユーザ認証

Replication Server では、ユーザ認証のための追加セキュリティメカニズムとして LDAP の使用をサポートしています。

LDAP ユーザ認証を使用すると、クライアントアプリケーションは Replication Server にユーザ名とパスワードの情報を送信し、Replication Server ではなく LDAP サーバによる認証を行えるようになります。

LDAP ユーザ認証を使用すると、ユーザ管理を簡素化および一元化したり、ユーザ管理に関する無用なログインの問題を防ぐことができます。

Replication Server は OpenLDAP クライアント API を使用してユーザ認証用の LDAP サーバと交信します。OpenLDAP API は OpenSSL でプリコンパイルされており、**\$SYBASE/\$OCS/lib3p** または **\$SYBASE/\$OCS/lib3p64** ディレクトリにある共有ライブラリ **libsylbaseldap\*** から動的にロードされます。

LDAP ユーザ認証を使用するための手順は次のとおりです。

1. **configure replication server** コマンドで、**user\_authentication\_source** パラメータを使用して認証メカニズムを設定します。

2. **sysadmin ldap** コマンドで、LDAP URL とともに LDAP サーバ管理者のログイン名、識別名 (DN: Distinguished Name)、およびパスワードの検索フィルタを設定します。

---

**注意：** LDAP 認証を有効にする場合、有効なユーザアカウントが Replication Server に必要です。

---

『Replication Server 管理ガイド: 第 1 巻』の「LDAP ユーザ認証の設定」と「Replication Server と LDAP のユーザアカウント管理」を参照してください。

『Replication Server リファレンスマニュアル』の「**configure replication server**」と「**sysadmin ldap**」を参照してください。

## 複写定義へのサブスクリプションの制御

Replication Server 15.7.1 SP100 では、アーティクルおよびパブリケーションで使用されているサブスクリプションを複写定義から直接無効にできます。

複写定義へのサブスクリプションを許可またはブロックするには、**configure replication server** の **block\_sub\_for\_repdef\_in\_pub** パラメータを使用します。

---

**注意：** プライマリ Replication Server で **block\_sub\_for\_repdef\_in\_pub** パラメータを有効にしてください。

---

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

## メンテナンスユーザのセキュリティ

データベースへの不正なアクセスを防ぐには、メンテナンスユーザをパスワード暗号化によってセキュリティ保護し、パスワードの有効期限を設定します。

**hide\_maintenance\_pwd** パラメータを on に設定すると、次のことが可能になります。

- レプリケート Adaptive Server データベースへの既存の接続のメンテナンスユーザに対して新しいパスワードを定期的に生成し、パスワードを暗号化する。
- 新しく作成するすべてのデータベース接続に対してパスワードを生成して暗号化し、定期的にパスワードを再生成して暗号化する。
- メンテナンスユーザに再入力したすべてのパスワードを変更して暗号化する。

Replication Server で特定の接続に **hide\_maintenance\_pwd** を設定する場合は **alter connection** または **create connection** を使用します。すべてのデータベース接続に **hide\_maintenance\_pwd** を設定する場合は **configure replication server** を使用します。

**configure replication server** で **maintuser\_pwd\_expiration** を使用すると、すべての接続のメンテナンスユーザにパスワードの有効期限を設定できます。

『Replication Server 管理ガイド 第 1 巻』の「メンテナンスユーザのセキュリティ」を参照してください。

## Sybase Control Center

---

Sybase Control Center 3.2.8 には Sybase Control Center 3.2.6 for Replication が含まれています。Sybase Control Center 3.2.6 を Replication Server 15.7.1 SP100 と併用することは可能ですが、Sybase Control Center 3.2.6 for Replication では Replication Server 15.7.1 SP100 の新機能がサポートされていません。

Sybase Control Center 3.2.8 の Sybase Control Center 3.2.6 for Replication を参照してください。

## プラットフォームのサポート

---

Replication Server 15.7.1 SP100 では、VMware vSphere のサポートが追加されました。VMware vSphere は VMware ESXi 製品に基づくプラットフォームです。vSphere ESXi 上で動作するゲストオペレーティングシステムは Windows、Solaris、および Linux です。

サポートされるバージョンについては、『Replication Server インストールガイド』と『Replication Server リリースノート』を参照してください。

# 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すすめします。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\_droptremotelogin**
- **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 Multi-Path Replication

Replication Server 15.7.1 では、Multi-Path Replication 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 間の専用ルートを通じてプライマリ Adaptive Server データベースから Replication Server への複数の複写パスと、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』>「パフォーマンスチューニング」>「Multi-Path Replication」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複写パスと専用ルートを作成できます。

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 5: 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複写では、Replication Server Options に搭載されている Replication Agent for Oracle が必要です。

表 6: 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

データベース	サポートされているバージョン
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。
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。

### ライセンス

Multi-Path Replication 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 への複写は、Real-Time Loading Edition (RTLE) で実行可能です。『Replication Server インストールガイド』>「インストールの計画」>「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) を特定の複写パスにバインドすることで、これらのオブジェクトの並列複写が可能になります。『Replication Server 管理ガイド 第2巻』>「パフォーマンスチューニング」>「Multi-Path Replication」>「複数のプライマリレプリケーションパス」>「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』 > 「パフォーマンスチューニング」 > 「Multi-Path Replication」 > 「並列トランザクションストリーム」 > 「Multi-Path Replication の分散モード」 > 「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 Multi-Path Replication について、接続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

Multi-Path Replication 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』 > 「インストールの計画」 > 「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 7 : 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 DSI SQT 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 を起動して、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

代わりに、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』>「`rs_init`による Replication Server の設定とデータベースの追加」>「リソースファイルを使用した `rs_init`」>「`rs_init` でのリソースファイルの使用」>「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』>「Replication Server のセキュリティ管理」>「Replication Server ユーザのセキュリティ管理」>「Replication Server ログイン名とパスワードの管理」>「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』>「Replication Server のセキュリティ管理」>「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、監査人にモニタリング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、役割を付与されていない状態で、サポートの役割に十分な RSSD テーブルへのモニタリング専用アクセス権を持っています。
- ステータス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 はデータをステータス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接続またはルートには、SSL を使用することをおすすめします。Replication Server の Secure Sockets Layer (SSL) 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` パーミッション、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の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複写を使用して、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複写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、ステーブル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ーブル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 1 巻』の「ターゲットスコープカスタムファンクション文字列」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\_helpobjfstring** 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\_helpobjfstring**」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せん。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、複製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 (64 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 (81 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。

Multi-Path Replication™ 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ابل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ابلキュー」の「ステابل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ホ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ホ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ホ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。

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。

『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。これにより、Replication Server における UTF-8 文字セットの制限がなくなります。

以前のバージョンの Replication Server では、`unichar`、`univarchar`、`unitext` などの Unicode データ型をすべて複製するには、文字セットを UTF-8 に設定する必要がありました。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 - Replication Server で (15.7 より前のバージョンと同様に) UTF-8 文字セットを使用する場合は、このオプションを使用します。
- `U&"` - Replication Server で UTF-8 以外の文字セットを使用する場合は、このオプションを使用します。

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 15.7 の新機能

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。

- **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。この場合、Replication Server では UTF-8 が必須です。
- **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。この場合、Replication Server では UTF-8 以外の文字セットを使用できます。

**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コネクション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、スレッシュ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、および矛盾が表示されます。

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。

## Replication Server 15.7 の新機能

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、サーバ間コネクション、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 Replication 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 (rs_ticket 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		

機能	SCC for Repli- cation 3.2.3	RMP 15.x	RMS 15.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		
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	

機能	SCC for Repli- cation 3.2.3	RMP 15.x	RMS 15.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	
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 に設定しません。デフォルトは false です。

参照:

- 『Replication Server リファレンスマニュアル』の「RepAgent 設定パラメータ」
- 『Replication Server 管理ガイド 第 1 巻』の「RepAgent の起動」と「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

- SAP IQ - Real-Time Loading を使用して SAP IQ バージョン 12.7 ESD #3 以降に複写できます。SAP 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 SAP IQ への複写をサポートしています。
- Oracle - Replication Server は、Oracle 10g および 11g から SAP IQ への複写をサポートしています。Replication Server Options 15.5 の『Replication Agent 15.5 リリースノート』の「製品の概要」の「互換性のある製品」を参照してください。

参照:

- Replication Server ライセンス (67 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ホ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ホールドに達すると、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します。
- カラムとテーブルをレプリケートします。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します。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 8：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 SAP 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>
<b>num_msgs</b>	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。	<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_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_cache_size	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 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。これらのステージング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ージャを使用し

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ジング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 9 : 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する。  <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る。	SY_INFO-PRIMER_SERVER

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

SAP 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 SAP 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:

- SAP 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。

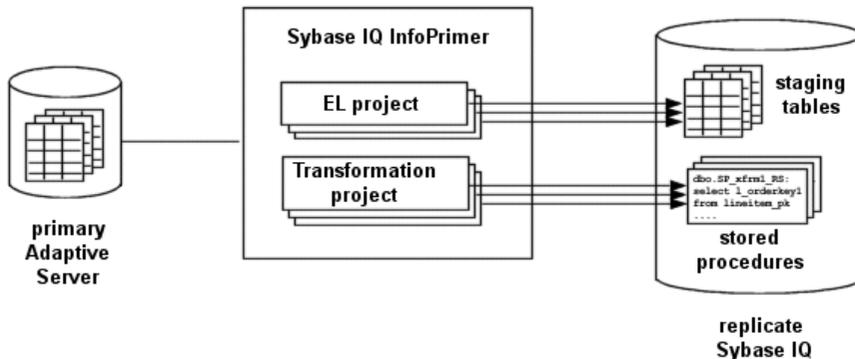
EL プロジェクトエディタの [RepServer] タブで、プライマリ Replication Server とレプリケート Replication Server (プライマリと異なる場合のみ) の接続情報も指定する必要があります。SAP IQ InfoPrimer によって、[処理] タブにコマンドが追加されます。このコマンドは、変更も削除もしないでください。

各ソーステーブルに対して、SAP IQ InfoPrimer は必要なステージング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 SAP 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 SAP 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 SAP IQ データベースでストアー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ー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。

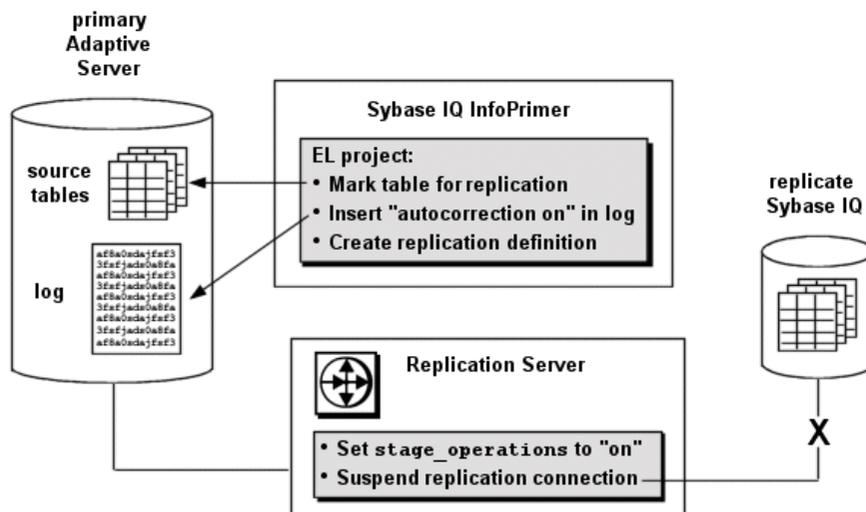


- Replication Server インスタンスで、**stage\_operations** 接続パラメータを使用して、EL プロジェクトで指定されたテーブルのステージオペレーションへのレプリケートデータベース接続を設定します。

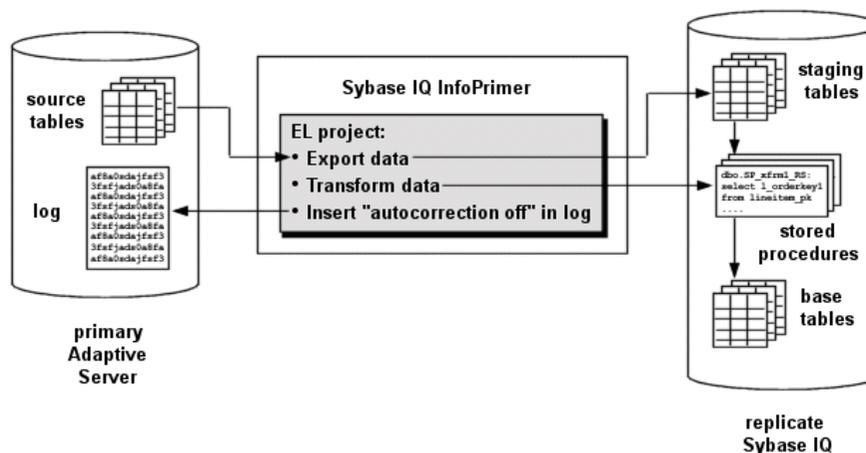
**注意：**stage\_operations が on に設定されている場合、Replication Server は **dsi\_compile\_enable** の設定を無視し、その接続の RTL を有効にします。オペレーションはコンパイルされ、**dsi\_compile\_enable** が on に設定されると、ステージングされます。

SAP 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。
- 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接続がサスペンドされます。
- RSSD でテーブル複写定義を作成します。

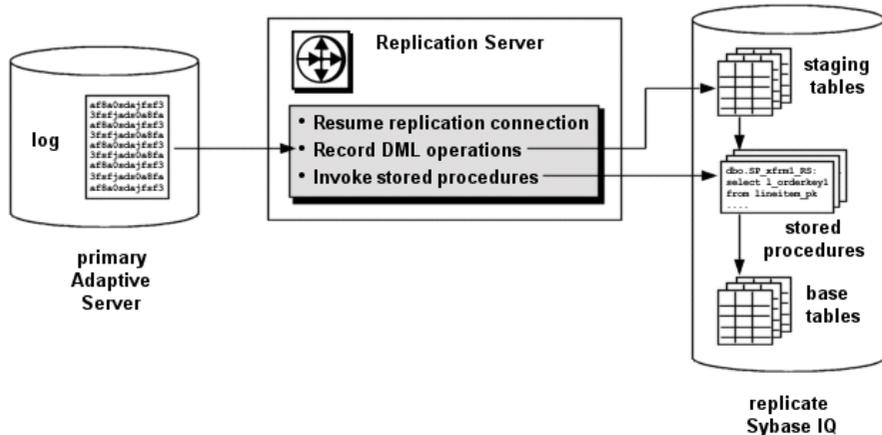


3. SAPIQ InfoPrimer EL プロジェクトは、各テーブルのプライマリデータをレプリケート SAP IQ 上にある対応するステージングテーブルにエクスポートし、変換ストアードプロシージャを実行し、autocorrection off レコードをプライマリデータベースログに挿入します。



4. Replication Server のレプリケートデータベース接続が再開され、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 SAP IQ データベースでステージングテーブルと変換ストアー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：** SAP IQ InfoPrimer は、データの移行と、ステージ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。SAP IQ InfoPrimer は、複製には関与しません。



### ベーステーブル

ベーステーブルには、レプリケート SAP 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接続が設定されると、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接続を設定すると、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 SAP 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。

---

SAP 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 SAP 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複写されます。

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接続を設定しても、レプリケート SAP 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接続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すすめします。

---

### 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、SAP IQ InfoPrimer の SQL 変換プロジェクトには表示されません。

---

### 変換ストアードプロシージャ

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ードプロシージャがレプリケート SAP 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ー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。

SAP IQ InfoPrimer の SQL 変換プロジェクトでこれらのストアードプロシージャによって実行される変換を指定し、それらのストアードプロシージャをレプリケート SAP 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 SAP IQ データベースに存在しないストアードプロシージャを使用しようとするか、またはストアー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接続がサスペンドされます。

---

**注意：** SQL 変換プロジェクトに關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 SAP 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オ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オ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複写します。

---

**注意：** **stage\_operations** パラメータは、SAP 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)。SAP IQ 接続プロファイルを使用して接続を作成すると、**dsi\_dataserver\_make** 接続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、そのコネクシオンの 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。

### 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
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。

データサービインタフェース (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

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ー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

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

- SAP 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
- 複写ストアプロシージャ
- 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 SAP IQ データベースで変換ストア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 10 : 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 11 : 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。	RTL 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プ、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す。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。

レプリケートテーブルに複数のユニークキーがある場合は、ユニークインデックスで指定されているすべてのカラムがテーブル複写定義内のプライマリキーに含まれている必要があります。含まれていないと、複写で重複キーエラーが発生するおそれがあります。

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ります。
- **delete** の後に **insert** があると結果のオペレーションは減少しません。
- **update** の後に **delete** があると結果のオペレーションは **delete** になります。
- **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** になります。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ります。
- **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、6の最後の **insert** 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。SAP 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、複写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 (114 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 の内部マーカ。
  - 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写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写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。
- 複写定義の中に `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、データベース複写定義を使用している場合、テーブル複写定義を作成するか、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。テーブル複写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 - 複写マークの付いたテーブルが存在する場合に Replication Server で複写定義を自動的に作成するには、テーブルに複写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 Agent リファレンスマニュアル』の「Replication Server オプション」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 内で複写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、複写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。

### SAP IQ での複写の干渉と影響

SAP IQ レプリケートデータベースに対する重大な干渉または影響は、接続プロファイルによって SAP IQ レプリケートデータベースに作成されるシステムテーブル

ると、RTL バルク適用を格納するために SAP IQ レプリケートデータベースに作成されるテンポラリテーブルのみです。

#### システムテーブル

接続プロファイルによって3つのテーブルが SAP IQ レプリケートデータベースに作成されます。

- **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写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、複写の正常性、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 バルク適用をサポートするために、RTL によって SAP 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が作成されます。このワークテーブルは動的に作成され削除されます。

テンポラリテーブルに必要な SAP IQ 内の領域は SAP IQ に複写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 SAP IQ のテンポラリデータベースの領域を調整するには、SAP 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 SAP IQ マニュアルを参照してください。  
SAP IQ 15.0 以降の例:

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

### SAP 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

SAP 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 SAP IQ レプリケートデータサーバに直接接続します。

Replication Server のデータベース接続名は、データサーバ名 (**server\_name**) とデータベース名 (**db\_name**) で構成されています。レプリケート Replication Server は、データベース接続で指定された SAP IQ データベースゲートウェイ **server\_name** を含むエントリを `interfaces` ファイルで検索します。

`dsedit` を使用して、SAP IQ レプリケートデータサーバが受信を行うホストとポートを指定するエントリを Replication Server の `interfaces` ファイルに作成します。 `interfaces` ファイルエントリの名前は、Replication Server データベース接続の **server\_name** 部分と一致している必要があります。Replication Server を再起動すると、Replication Server の `interfaces` ファイル内の新規エントリが有効になります。『Replication Server 設定ガイド』>「`rs_init` による Replication Server の設定とデータベースの追加」>「新しい Replication Server の設定」>「`interfaces` ファイルの編集」を参照してください。

SAP IQ レプリケートサーバの `interfaces` ファイルにレプリケート Replication Server のエントリを作成すると、Replication Server が **INSERT ... LOCATION** 文を SAP IQ に送信したときに、SAP IQ が Replication Server に接続してデータを取得できるようになります。

SAP IQ 15.0 以降では、「`set option public.STRING_RTRUNCATION = "OFF"`」を入力すると、Replication Server が **INSERT ... LOCATION** を使用して SAP IQ にデータを挿入するときの「Right truncation of string data」エラーを回避できます。

Replication Server は、データベース接続で指定された **user\_name** と **password** を使用して、SAP IQ レプリケートデータサーバにログインします。SAP IQ レプリケートデータベースの場合、**user\_name** と **password** はメンテナンスユーザの ID とパスワードです。

## SAP 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に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。

複写を開始する前に、SAPIQ データサーバでメンテナンスユーザ ID を定義し、レプリケートデータベースにトランザクションを適用するための権限をその ID に付与する必要があります。メンテナンスユーザ ID には、SAP IQ レプリケートデータベースで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 (すべての複写テーブルに対して必要)
- **UPDATE** 権限 (すべての複写テーブルに対して必要) および **EXECUTE** 権限 (すべての複写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 SAP IQ への複写をテストする場合は、DBA と RESOURCE の権限を与えます。

1. SAPIQ の `rssetup.sql` サンプルスクリプトを使用して、適切な権限を持つ SAPIQ 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、SAPIQ 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。たとえば、UNIX プラットフォームでは、次のディレクトリにあります。

- SAP IQ 15.0 より前のバージョン - /\$ASDIR/scripts
- SAP IQ 15.0 以降のバージョン - /\$IQDIR15/scripts

ディレクトリの位置については、『SAP 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. SAP IQ データベースに Transact-SQL® (IQ DBA 用) との互換性があることを確認します。

『SAP IQ リファレンス: 文とオプション』 > 「データベースオプション」 > 「Transact-SQL 互換性オプション」と『SAP 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』 > 『他の Sybase データベースとの互換性』を参照してください。

3. 複製に関係するすべてのテーブルとストアー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 への複製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製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
```

### SAP IQ への接続の作成

レプリケート SAP IQ データベースへの接続を設定します。

1. **create connection** を **using profile** 句および該当する接続プロファイルを指定して使用し、レプリケート SAP 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
```

SAP IQ データベースへの複製パスを複数作成して、複製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** を使用して、Replication Server が SAP IQ に正常に接続することを確認します。

### RTL を有効にする

該当するパーミッションを付与し、レプリケート SAP IQ データベースに接続したら、SAP IQ への複製について RTL を有効して設定できます。

接続の RTL を有効にするには、**dsi\_compile\_enable** を使用します。

**dsi\_compile\_enable** を off に設定した場合、Replication Server はログ順、ローごとの連続複製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製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、最終ロー変更を複製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。

**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。

---

**留意：** SAP IQ への Real-Time Loading (RTL) 複写を有効にする前に、**dsi\_bulk\_copy** と **dynamic\_sql** を **off** に設定する必要があります。

---

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 - 指定した複写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写データと複写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 SAP IQ データベースへの接続をサスペンドして再開します。

### **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なくても、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** は、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。

---

**留意：** **dsi\_bulk\_copy** を off に設定してください。

---

- **dsi\_command\_convert** - 複写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** コマンドを複写しません。このオプションでは、**delete** オペレーションを複写しない場合に **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、複写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** コマンドを複写しません。

**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ョルド値。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、複写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。

## Replication Server 15.6 の新機能

- 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 (126 ページ)
- ローカウントの検証を制御する (152 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 データベースへのコネクション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 SAP 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. SAP 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：** SAP IQ データサーバの interfaces ファイルが、SAP IQ が使用している SAP \$\$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。

---

2. レプリケート Replication Server の interfaces ファイルに、SAP IQ データサーバのエントリを作成します。  
さまざまな SAP IQ マルチプレックスノードへの接続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：**

- SAP 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 (110 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。

1. 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。
2. 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。

1. 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。
2. 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 へのコネクションの作成には、**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. コネクションが機能していることを確認します。

```
admin who
go
```

コネクションが機能していれば、次のような出力が表示されます。

```
SpidNameStateInfo
-----
63DSI EXECAwaiting Command103(1) IQSRVR.iqdb
62DSIAwaiting Message103 IQSRVR.iqdb
35SQMAwaiting Message103:0 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。

#### 前提条件

SAP IQ への Real-Time Loading (RTL) 複写を有効にする前に、**dsi\_bulk\_copy** と **dynamic\_sql** を **off** に設定します。

## 手順

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. レプリケート SAP IQ データベースへの接続をサスペンドして再開し、接続の変更を有効にします。

```
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* 複写定義を作成し、RTL をサポートする複写定義に必要なすべての参照制約句を追加します。

```
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 **ltl\_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、「**ltl\_character\_case**」を参照してください。

---

2. 各テーブルとストア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
```

3. Sybase IQ にログインして次を実行し、testtab がマテリアライズされたことを確認します。

```
select * from dbo.testtab
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。

```
c1c2c3
-----
11testrow 1
22testrow 2
33testrow 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。

1. プライマリ Oracle データサーバにログインして、testtab 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。
2. Sybase IQ にログインして、testtab 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.testtab
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。

```
c1c2c3
-----
11testrow 1
22testrow 2
33testrow 3
44testrow 4
55testrow 5
66testrow 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 (106 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います。各カラムはテーブルを1つだけ参照できます。
- **reference** 句内の **column\_name** パラメータに指定したカラム名を処理しません。
- 循環参照になる参照制約を許可しません。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ません。

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します。
- 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します。

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します。
- **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ます。

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
```
  - To verify that **dsi\_compile\_enable** is on for the connection to NY\_DS.nydb1, enter:

```
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"
```

- 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製されてから、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징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징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징ングデータベースへのコネクションを削除します。
4. 環境を終了すると、ステー징ングデータベースから SAP IQ にデータが抽出されます。

## 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を使用し、**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** を設定するときを使用します。

---

**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、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 (114 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ابل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、複製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。

---

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します。
- その複製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します。

### 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 (101 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 へのすべての接続に影響します。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、「ダウンストリーム」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します。

- g. キューブロックサイズがに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



```
pdb -----> PRS -----> RRS -----> rdb
```

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「複写システムの管理」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「複写システムリカバリ」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見逃します。

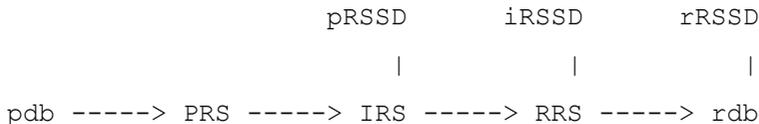
```
ignore loss from RRS.rRSSD to PRS.pRSSD
```

### 中間ルートを持つ複製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製システムの例で、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。

複製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オンで実行すると、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、ダンプ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。

#### ペ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ペ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します。
- RepAgent を無効にします。
- **dbcc dbrepair** などの Adaptive Server コマンドを実行します。

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、ペ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ページされると、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、ページオプションは 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ペ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。プライマリで使っ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 (138 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 (144 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 を再開できます。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、複写されます。

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ときのメッセージ。

```
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します。
- 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します。

- 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、再同期を調整します。
- 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、再同期を調整します。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します。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ます。

### 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を再同期する (145 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めます。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める責任があります。たとえば、ディスク複製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、`rdb1` 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 rdb1 '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
```

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。

### 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を再同期する

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。

プライマリ ---> 複写 ---> レプリケート論理データベース

データベース      サーバ      [アクティブ + スタンバイのウォームスタンバイ

へ

ア]

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 による複写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. プライマリデータベース 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 (139 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みます。
- **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みます。

## 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-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。

## ローカウムの検証を制御する

ローカウム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、ローカウムの検証がデフォルトで有効になっているので、ローカウムの不一致などのさまざまなローカウム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ム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、ローカウム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を使用し、**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を使用します。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ム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 が処理する複製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製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、ローごとの複製 - 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た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」の手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に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 (139 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 のダウン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ク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ク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 "Simulate Atomic Materialization" または "Simulate Nonatomic Materialization" バルクマテリアライゼーションメソッドを使用します。
  - アトミックマテリアライゼーションをシミュレートします - 手順 4～9 を実行
  - 非アトミックマテリアライゼーションをシミュレートします - 手順 4～13 を実行

## 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。

---

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 12 : 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 (160 ページ)
- Replication Server - Advanced Services Option (164 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。

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 (186 ページ)
- Replication Server 15.5 の製品エディションとライセンス (159 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 13 : 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	はい	いいえ
Microsoft SQL Server	はい	はい
IBM DB2 UDB	はい	はい

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 14 : 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する。  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る。  範囲：1 ~ 1000

表 15 : 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。  範囲：1～1000
<b>dsi_serialization_method</b>	no_wait wait_for_start wait_for_commit wait_after_commit	wait_for_commit	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、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 異機種間複製ガイド』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** コ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 16 : 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コネクション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コネクション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 (159 ページ)

### **High Volume Adaptive Replication**

ローごとの個々の複製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製モードに比べて、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 - テーブルごとに複製データを 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、複製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。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 (186 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パックし、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) スレッドはステイ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イ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。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ポートし、Replication Server をクワイイ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のみ言語コマンドを生成します。
- 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います。

### 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製する。

Replication Server 15.5 では、複製処理が動的 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** 句が複製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) では複製定義がないときまたは複製定義に **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** はすべてのコネクションに対してオンです。コネクションの **replicate\_minimal\_columns** 設定は、**replicate all columns** 句で設定された複製定義よりも優先されます。

あるコネクションに対して **replicate\_minimal\_columns** を on に設定すると、現在の複製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、ストアードプロシージャ、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* カラムの値を複写しないように、**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。

### ストアー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\_helpstring** 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\_helpstring** - 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) を表示します。
- **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。

### 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状態にし、プライマリテーブルまたはストアー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、複製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。

強化された複製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「複製テーブルの管理」の「複製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、複製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る複製定義がある場合、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製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。

### 複製定義関連の強化項目リスト

複製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製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製定義コマンドを発行します。
- ターゲットデータベースの古いバージョンの複製定義で、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** コマンドを使用します。これにより、新しいバージョンの複製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ます。
- Replication Server が複製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します。
- 複製定義からカラムを削除するために **alter replication definition** を実行します。
- Replication Agent から送信された複製定義要求で失敗したものを Replication Server に省略させます。プライマリ Replication Server で複製定義コマンドが失敗すると、Replication Agent が停止します。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ます。

### システムテーブルの変更

強化された複製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。

### 複製タスクのスケジューリング

Replication Server 15.5 では複製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。複製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「複製スケジュールの管理」の「複製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。

### 複製の遅延

Replication Server 15.5 では複製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「複製スケジュールの管理」の「複製の遅延」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 17 : 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 18 : 新しいエラーと変更されたエラー

serv-er_er-ror	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期されたロー数とは異なる場合に表示される。

serv-er_er-ror	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 <code>'dataserver.database'</code> . The command deleted <code>x</code> rows, but it should delete <code>y</code> rows.	<b>stop_replication</b>	このメッセージは、オートコレクションが有効な場合、 <code>delete</code> 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 <code>'dataserver.database'</code> . The delete command generated by <code>dsi_command_convert</code> deleted <code>x</code> rows, whereas it should delete <code>y</code> rows.	<b>stop_replication</b>	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 文の複製ローカウントの検証 (207 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。

## リファレンス複製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 への複製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. 複写環境を設定します。
3. データベースレベルの複写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の複写処理から統計とモニタカウンタを収集します。
5. リファレンス複写環境をクリーンアップします。

リファレンス複写環境の構築、設定、使用の詳細については、『Replication Server 管理ガイド第2巻』の「リファレンス複写環境の実装」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写環境を構築します。複数の複写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 への複写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 19 : 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 への複写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 20 : 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 - ステータスキューマネージャ
- SQT - ステータス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 21 : 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 最大値：2 <sup>63</sup> -1 デフォルト値は 0

スレッド	カラム名	説明	値
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>sq_t_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 最大値：2 <sup>63</sup> -1 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 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 Replication 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 にリセットできます。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。

『Replication Server 管理ガイド第 2 巻』の「プライマリデータベース世代番号のリセット」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 22 : 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
<b>exec_cmds_per_time-slice</b>	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。
<b>init_sqm_write_delay</b>	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。
<b>init_sqm_write_max_delay</b>	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 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、オーバー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 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 バイト (1MB)	<ul style="list-style-type: none"> <li>32 ビットプラットフォーム： 1,048,576 バイト (1MB)</li> <li>64 ビットプラットフォーム： 20,971,520 バイト (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、オーバー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> (次のシステムテーブル用) <i>rs_columns,</i> <i>rs_functions, 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：

- *bigint* と *bigtime* の複製の有効化 (189 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、ストアー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にストアー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ードプロシージャが Adaptive Server で初めて実行されるまで延期されます。ストアー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ー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製できるように、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 がテンポラリテーブルを複製しないので、テンポラリテーブルを参照するストアードプロシージャを作成できません。ストアードプロシージャを作成するプロセスは、そのストアードプロシージャが内部で使用す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ー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) を参照するストアー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製できます。

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、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にのみ存在するので、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、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せん。

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みます。

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させることができます。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 23 : 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)
<code>dsi_sq_max_cache_size</code>	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、 <code>sq_max_cache_size</code> 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 最大： 2147483647	最小： 0 最大： 2251799813685247
<code>dist_sq_max_cache_size</code>	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、 <code>sq_max_cache_size</code> 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 最大： 2147483647	最小： 0 最大： 2251799813685247
<code>sq_max_cache_size</code>	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 最大： 2147483647	最小： 0 最大： 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 24 : 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 最大 : 2047	最小 : 0 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 (181 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、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

次の新しいデータベース接続パラメータは DSI でのバルク操作を制御します。

パラメータ	説明
<b>dsi_bulk_copy</b>	接続のバルクコピーイン機能を on または off にする。 <b>dynamic_sql</b> と <b>dsi_bulk_copy</b> が両方とも on の場合、DSI はバルクコピーインを適用する。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。  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ために 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をおすすめする。  最小値： 1  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します。

```
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します。

```
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 によって、以下が実行されます。

1. DSI が、**insert** でないコマンドまたは異なるレプリケート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ません。
- 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ます。

```
Bulk-Lib routine 'blk_textxfer' failed.
Open 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ます。

```
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します。
- **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ます。
- バルクコピーは、Adaptive Server で **insert** トリガを呼び出しません。
- 設定パラメータ **send\_timestamp\_to\_standby** はバルクコピーインに影響しません。*timestamp* データは常に複写されます。

## 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。

**alter connection** を使用して、データベース接続の **dsi\_non\_blocking\_commit** を設定できます。

```
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接続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、接続の **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す。

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 異機種間複写ガイド』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。

- 複写定義で識別子を引用符付きとしてマーク付けします。
- 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します。

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、複写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写を成功させるには、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** コマンドまたは **alter connection** コマンドを使用して、データサーバ接続の **dsi\_quoted\_identifier** を on または off にできます。 **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。

データサーバによって、引用符付き識別子の受け入れ方が異なります。Adaptive Server、SQL Anywhere、および Microsoft SQL Server では引用符付き識別子を想定しないため、引用符付き識別子について接続を設定するための特別なコマンドが必要になります。Oracle と UDB では、引用符付き識別子を受け入れるために接続を設定する必要ありません。

**rs\_set\_quoted\_identifier** ファンクション文字列を使用して、データサーバタイプごとに 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** が **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** が "" 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す。

## 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写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 NamePRSType Creation Date
```

```
-----
```

```
r1ost_wasatch_09TblNov 11, 2008 2:28PM
```

```
PDS.DBPrimary OwnerPrimary Table
```

```
-----
ost_wasatch_08.pdb1t1
```

```

Replicate OwnerReplicate Table
-----
t1

Send Min Cols. Used by Standby Min Vers Dynamic SQL SQL Stmt. Rep.
-----
NoNo10000nNone

Col. NameRep. Col. NameDatatypeLen.Pri. Col.Searchable
-----
c1c1int410
"c2""c2"int400

Function NameFString ClassFString SourceFString Name
-----
rs_deleters_sqlserver_function_classClass Defaultrs_delete
rs_insertrs_sqlserver_function_classClass Defaultrs_insert
rs_selectrs_sqlserver_function_classClass Defaultrs_select
rs_select_rs_sqlserver_function_classClass Defaultrs_select_
with_lockwith_lock
rs_truncaters_sqlserver_function_classClass Defaultrs_truncate
rs_updaters_sqlserver_function_c_classClass Defaultrs_update

Subscriptions known at this Site 'ost_wasatch_09'.
Subscription NameReplicate DS.DBOwnerCreation 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 NamePRSType Creation Date
-----
rlost_wasatch_09Tb1Nov 11, 2008 2:28PM

PDS.DBPrimary OwnerPrimary Table
-----
ost_wasatch_08.pdb1"t1"

Replicate OwnerReplicate Table
-----
"t1"

Send Min Cols. Used by Standby Min Vers Dynamic SQL SQL Stmt. Rep.
-----
NoNo10000nNone
    
```

```

Col. NameRep. Col. NameDatatypeLen.Pri. Col.Searchable
-----
c1c1int410
"c2""c2"int400

Function NameFString ClassFString SourceFString Name
-----
rs_deleters_sqlserver_function_classClass Defaultrs_delete
rs_inserter_sqlserver_function_classClass Defaultrs_insert
rs_selectrs_sqlserver_function_classClass Defaultrs_select
rs_select_rs_sqlserver_function_classClass Defaultrs_select_
with_lockwith_lock
rs_truncaters_sqlserver_function_classClass Defaultrs_truncate
rs_updaters_sqlserver_function_classClass Defaultrs_update

Subscriptions known at this Site 'ost_wasatch_09'.
Subscription NameReplicate DS.DBOwnerCreation 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\_help* prep 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 NamePRSType Creation Date
-----
r1ost_wasatch_09TblNov 11, 2008 2:28PM

PDS.DBPrimary OwnerPrimary Table
-----
ost_wasatch_08.pdb1"t1"

Replicate OwnerReplicate Table
-----
"t1"

Send Min Cols. Used by Standby Min Vers Dynamic SQL SQL Stmt. Rep.
-----
NoNo1000OnNone

Col. NameRep. Col. NameDatatypeLen.Pri. Col.Searchable
-----
c1c1int410
c2c2int400

Function NameFString ClassFString SourceFString Name
-----
rs_deleters_sqlserver_function_classClass Defaultrs_delete
rs_inserter_sqlserver_function_classClass Defaultrs_insert

```

```
rs_selectrs_sqlserver_function_classClass Defaultrs_select
rs_select_rs_sqlserver_function_classClass Defaultrs_select_
with_lockwith_lock
rs_truncaters_sqlserver_function_classClass Defaultrs_truncate
rs_updaters_sqlserver_function_classClass Defaultrs_update

Subscriptions known at this Site 'ost_wasatch_09'.
Subscription NameReplicate DS.DBOwnerCreation 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製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します。
- **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します。
- **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、カスタム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します。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ります。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ません。

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します。
- **show server** - 現在稼働中のサーバを表示します。

### 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っています。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 文の複写ローカウムの検証

---

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ムの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ム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に探します。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ムの検証 (217 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します。
- **error\_class** – 新しいエラークラスの名前。名前は複製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ます。
- **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』の第 3 章「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 25 : 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期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期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プロシージャ**

**rs\_helpdb**、**rs\_helpclass**、**rs\_helperror** ストア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
<i>rs_classes</i>	<i>classtype</i> カラムには Replication Server エラークラス用に新しい値 "R" が含まれている。
<i>rs_databases</i>	<i>rs_errorclassid</i> 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。

## 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 文の複写には、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 文の複写の有効化

SQL 文を複写できるように 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します。
2. SQLDML を複写するように Replication Server を設定します。
  - a) テーブルと Multi-Site Availability (MSA) の複写のための SQLDML の複写定義を作成します。
  - 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写を設定する。

## 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 文の複写がサポートされます。

### データベースレベルの SQL 文複写

SQL 文の複写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 文の複写を有効にできます。

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
- **S** - select into

データベースの複写モードを **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 文の複写として **delete** 文を複写し、**select into** の複写も有効にするには、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ります。

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 文として複写するには、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 文の複写設定

データベースレベルとテーブルレベルで複写オプションを変更できます。

### データベース複写定義

**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 文を複写します。

### 構文

次のコードセグメントは、データベース複写定義の **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ます。

### 例

- **例 1** – 'options' パラメータを使用して、テーブル *tb1* と *tb2* の SQLDML を複写します。

```
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 (tb1,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 *tb1* と *tb2* に適用します。

```
alter database replication definition dbrepdef
with primary at ds1.pdb1
replicate 'UD' in (tb1,tb2)
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 (tb1,tb2)
replicate 'U' in (tb1)
replicate 'I' in (tb1,tb2)
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 'tbl1'

(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場合。
- 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場合。
- サブスクリプションまたはアーティクルに **where** 句が含まれてる場合。
- *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場合。

### 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、複製が停止します。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 を複製しません。

## SQL 文の複製に対するローカウムの検証

Replication Server 15.2 では、SQL 文の複製中に発生する可能性のある SQLDML のローカウムエラーに Replication Server が対応する方法を指定できます。

SQLDML のローカウムエラーは、SQL 文の複製後に、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、複製の停止です。SQLDML のローカウム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 文の複製ローカウムの検証 (207 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ます。
- **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』の第3章「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 26 : 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 複写を使用するには、以下を行います。

- **replicate SQLDML** 句と **send standby** 句を使用して、テーブル複写定義を作成します。
- **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 文を複写するかどうかが決まる。

## 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、テーブル複写定義を作成。
- **WS\_SQLDML\_REPLICATION** パラメータを **on** に設定します。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ります。テーブル複写定義にテーブルの **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 文を複写するかどうかが決まります。

## Replication Server システムデータベースの変更

SQL 文複写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。

システム テーブル	説明
<i>rs_dbsubsets</i>	<i>type</i> カラムに 4 つの新しいカラム <b>U</b> 、 <b>L</b> 、 <b>I</b> 、および <b>S</b> が追加された。それぞれ DML <b>UDIS</b> フィルタに対応している。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、テーブル複写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)。  システムファンクション複写定義 <b>rs_sqldml</b> も、SQL 文の複写をサポートする。

## 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 以外での複写サポートの機能強化 (222 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 27 : 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>
IBM UDB	<i>rs_udb_error_class</i>
Microsoft SQL Server	<i>rs_msss_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 以外での複写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写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写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。

**参照：**

- Adaptive Server 以外のためのエラークラスのサポート (220 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 間のコネクションを作成するには、**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 NameVersionComments
-----
rs_ase_to_db2StandardStandard ASE to DB2 replication
connection profile.
rs_ase_to_udbStandardStandard ASE to DB2 replication
connection profile.
...
rs_db2_to_aseStandardStandard DB2 to ASE replication
connection profile.
rs_db2_to_msssStandardStandard 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 NameVersionComments
-----
rs_ase_to_oracleStandardStandard 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ります。
- 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ります。
- 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ません。 **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ります。

## 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、複製の送信元または複製の送信先としてデータベースを使用できます。RepAgent の設定や複製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ابل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ます。
- 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ابل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ابل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ابل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ابل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ーブルキューから削除し、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ータス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ータス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カ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/$SYBASE_REP/locales` ディレクトリに移動されました。
- すべての `<charset>` サブディレクトリが、新しい `$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ました。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、非対称キー暗号化が使用されるため、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もありません。

**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みません。
- チケットサイズが 255 バイトから 1024 バイトに増えました。
- `timestamp` のフォーマットが変更され、日付が含まれるようになりました。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ました。チケットが複数の RSI モジュールを渡すようになりました。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します。
- 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 名が追加されました。
- 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ました。
- 次の 2 つの DSI カウンタが追加されました。
  - `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
DSINoDsqlColumn-Count	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
DSINoDsqlMissing-Cols	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
DSIEDsqlDealloc-Schema	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ーキーとして定義できません。
- 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ません。
- 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ません。
- 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ません。

### 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製できます。

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* カラムを複製できます。

また、複製定義で *timestamp* をプライマリキーとして定義したり、複製定義やファンクション複製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* の複製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、複製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す。

**注意：**複製定義で *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* カラムの複製については、『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製を処理します。

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製を処理します。

### 制限事項

*opaque* データ型の制限事項は次のとおりです。

- 複製定義、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ません。
- **map to** 句で *opaque* データ型を使用できません。
- 複製定義に *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 を使用できません。
- 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ません。
- 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ません。

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** コマンドの複製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取り出すことができるため、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ません。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] フォルダが共存します。
- [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、[複製定義] 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製定義名

を作成できるようになりました。[複写定義] フィールドで選択したストアド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ます。

- [詳細] タブではなく [一般] タブで、[複写ストアドプロシージャ] オプションを使用できます。

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 データを複製できます。
- 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。



# 索引

## 数字

64 ビットサポート 186-188

## A

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 53

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 52

Replication Server 15.5 でのサポート 181

Replication Server 15.6 でのサポート 155

Replication Server との互換性 12, 39

共有ディスククラスタのサポート 229

事前計算済み結果セット 41

セキュリティ 52

データ圧縮 51

データベースオブジェクト所有権の譲渡 41

複写のサポート 12, 39

マスタキー 52

レプリケーションのサポート 51, 80

レプリケートデータベースの再同期 134

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 134

概要 134

Adaptive Server のサービスキー 52

Adaptive Server のマスタキー 52

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 56

admin config コマンド 192

admin health コマンド 47

admin sqm\_process\_time コマンド 46

admin who

機能強化 177

admin who, sqm コマンド 47

Advanced Services Option 164

ASE 以外での複写のサポート

接続プロファイル 223

接続プロファイルの使用 224

接続プロファイルのリスト表示 225

単純化されたインストール、設定 223

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 221

エラークラスの変更 222

デフォルトの ASE 以外のためのエラークラス 221

## B

bigdatetime、複写サポート 181

bigint、複写サポート 181

## C

configure replication server コマンド 196

connection to SAP IQ

作成 113

create auto partition path コマンド 46

## D

DIST ステータスの記録 242

DIST スレッド

並列処理 16

DIST スレッドでの並列処理

説明 16

dist\_sqrt\_max\_cache\_size 187

DSI 191

DSI 効率の向上 166

DSI でのトランザクションの分散適用、設定 17

DSI でのトランザクションの分散適用、説明 16

DSI のサスペンド 46

DSI のモニタリング、データベースの再同期 138

dsi\_bulk\_copy コネクションパラメータ 191-193

値を調べる 192

## 索引

dsi\_bulk\_threshold コネクションパラメータ 192, 193  
値を調べる 192

dsi\_compile\_enable、RTL 114

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 126

dsi\_max\_cmds\_in\_batch 設定パラメータ 162

dsi\_max\_xacts\_in\_group 設定パラメータ 163

dsi\_quoted\_identifiers 198

dsi\_retry 設定パラメータ 46

dsi\_row\_count\_validation 設定パラメータ 152

dsi\_serialization\_method 設定メソッド 163

dsi\_sqrt\_max\_cache 187

dsi\_top1\_enable 設定パラメータ 47

DSI、サスペンド 46

dt\_apply configuration パラメータ 17

dtm\_tm\_role 17

dump database 138

## E

Electronic Software Delivery、置き換え 35

ESD、置き換え 35

exec\_cmds\_per\_timeslice 179

execute as 句  
ストアードプロシージャの複写への影響 42

## F

fstr\_cache\_size 188

## H

HANA データベース  
Replication Server との互換性 35  
複写のサポート 35

High Volume Adaptive Replication 165

HVAR 165  
分散トランザクション管理 (DTM) 16, 17

HVAR、リトライメカニズムの強化 126

HVAR の機能強化  
Replication Server 15.7 81

## I

image データ型、バルクコピーイン複写サポート 157

IMDB 155, 184, 185

init\_sqm\_write\_delay 179

init\_sqm\_write\_max\_delay 179

interfaces ファイル 110

interfaces ファイル、SAP IQ への複写用に作成 117

## J

Java データ型、バルクコピーイン複写サポート 157

## L

LDAP  
RSSD 48  
認証 48  
ユーザ 48

LOB データ型  
サポート 248  
部分更新 239

locales ディレクトリ  
変更 234

LTL コマンド  
キャッシュ 68

## M

master データベース  
パスワードの有効期間 52

master データベースレプリケーション 53

mat\_load\_tran\_size、設定 39

max schema cache per scanner RepAgent 設定パラメータ 43

max\_mat\_load\_threads、設定 39

md\_source\_\_memory\_pool 188

memory\_limit 180, 188

Multi-Path Replication  
カラム別分散、説明 44  
セカンダリトランケーションポイントの管理 44  
複数の RepAgent スキャナ 44

**N**

none パラメータ、alter function string と create function string のスコープの拡張 169

**O**

opaque データ型 248  
 Oracle から Sybase IQ への複写 103  
 Oracle からの複写 103  
 Oracle データサーバ  
     rs\_lastcommit テーブル 109  
 Oracle、トリガの実行 164

**Q**

queue\_dump\_buffer\_size 188

**R**

Real-Time Loading  
     データベースのサポート 81, 103, 161  
     プラットフォームのサポート 81, 103, 161  
 Real-Time Loading Edition 103, 160  
 Real-Time Loading (RTL) による Sybase IQ への  
     複写 103, 160  
 RepAgent  
     自動起動 80  
     スキヤナスキーマキャッシュ 43  
     複写のサポート 43  
 RepAgent エグゼキュータスレッドの効率の向上  
     166  
 RepAgent の自動起動 80  
 replicate minimal columns 句、動的 SQL の使用  
     168  
 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  
     168  
 Replication Manager  
     新機能 15.1 247  
     新機能 15.5 189  
     廃止 64  
 Replication Manager は bigdatetime および  
     bigtime データ型をサポートします。  
     189  
 Replication Server  
     15.2 の新機能 191

15.5 の新機能 159  
 15.6 の新機能 101  
 15.7.1 の新機能 51  
 interfaces ファイル 110  
 SAP IQ レプリケートデータベース 110  
 新機能 15.6 ESD #1 87  
 新機能の概要 15.7.1 SP100 35  
 新機能の概要 15.7.1 SP200 5  
 ステータス、表示 47  
 プリンシパル名 47

Replication Server と SAP IQ InfoPrimer のデー  
     タフロー 88

Replication Server ゲートウェイ  
     カスケードコネクション 204  
     コネクション、追跡 206  
     コネクションの削除 206  
     制限事項 204  
     製品バージョンの要件 207  
     製品バージョン要件 206  
     有効化 205

Replication Server のアップグレード、シームレ  
     ス 155

Replication Server システムテーブル  
     rs\_asyncfuncs 72  
     rs\_clsfunctions 72  
     rs\_objfunctions 72

Replication Server と Sybase IQ InfoPrimer のデー  
     タフロー 87

resume connection コマンド、skip to resync マー  
     カ 135

RMA 7

rs password 設定パラメータ 52  
 rs\_autoc\_ignore システムファンクション 98  
 rs\_autoc\_off システムファンクション 97  
 rs\_autoc\_on システムファンクション 96  
 rs\_ddlsession\_resetting システム関数 41  
 rs\_ddlsession\_setting システム関数 41  
 rs\_delexception  
     ID 範囲と日付範囲によるトランザクショ  
     ンの削除 148  
 rs\_delexception\_date ストアドプロシージャ  
     149  
 rs\_delexception\_id ストアドプロシージャ 148

## 索引

rs\_delexception\_range ストアドプロシージャ  
151

rs\_dictionary システムテーブル 60

rs\_lastcommit テーブル  
Oracle データベース内 109

rs\_passwords システムテーブル 60

rs\_session\_setting ファンクション文字列 83

rs\_status システムテーブル 95

rs\_subscriptions システムテーブル 76

rs\_ticket  
バージョン 2 235

rs\_ticket マーカの挿入 179

rs\_users システムテーブル 60

RSSD ロックスキーマ、変更 180

RTL 103, 160

- admin config コマンド 122
- dsi\_bulk\_threshold 115
- dsi\_command\_convert 115
- dsi\_compile\_enable 114
- dsi\_compile\_retry\_threshold 115
- dsi\_max\_cmds 114
- rs\_helprep ストアドプロシージャ 123
- SAP IQ への複製の有効化 113
- Sybase IQ への複製の設定 114
- コンパイルできないコマンド、テーブル  
107
- コンパイルとバルク適用 104
- コンパイルの例 105
- コンパイルルール 104
- システムテーブルサポート 116
- ステージングソリューションからのマイ  
グレート 124
- 制限事項 106
- データベースとプラットフォームのサポ  
ート 81, 103
- データベースレベルの設定パラメータの  
表示 122
- テーブルレベルの設定パラメータの表示  
122
- テーブル参照の表示 123
- 複製シナリオ 116
- 利点 103
- 下位互換性 124
- 混合バージョンのサポート 124
- 最終的な変更のデータベースの表示 123

- 参照制約 107, 121
- 情報の表示 122

RTL での参照制約 121

RTL の dsi\_bulk\_threshold 115

RTL の dsi\_command\_convert 115

RTL の dsi\_compile\_retry\_threshold 115

RTL の dsi\_max\_cmds 114

RTL の最終的な変更のデータベース、表示  
123

RTL 複製のシナリオ 116

RTL 複製の例 116

RTL、リトライメカニズムの強化 126

RTL の機能強化  
Replication Server 15.7 81

## S

### SAP IQ

- RTL のコンパイルとバルク適用 104
- RTL の有効化 113
- コネクションの作成 113
- レプリケートデータベースのコネクティ  
ビティ 110
- レプリケートデータベースのパーミッシ  
ョン 111
- 干渉、システムテーブル 109
- 干渉、テンポラリーワークテーブル 109
- 複製の干渉と影響 108

SAP IQ への干渉、テンポラリーワークテーブル  
から 109

sa ユーザパスワード  
リセット 59

sa ユーザパスワードのリセット  
パスワード 59

security  
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 60

select コマンド  
変数 206

skip to resync マーカ、RepAgent から Replication  
Server への送信 136

skip to resync パラメータ 135

smp\_enable 180

sp\_configure 'enable dtm' SAP ASE システムプロ  
シージャ 17

- sp\_help\_rep\_agent Adaptive Server システムプロシージャ 12
  - sp\_help\_rep\_agent の scan\_verbosе パラメータ 12
  - sp\_setreplicate Adaptive Server システムプロシージャ 13
  - SP、置き換え 35
  - SQL 文の複写
    - replicate SQLDML 句 214
    - RSSD の変更 219
    - set repmode 213
    - sp\_setrepdbmode 212
    - sp\_setrepdefmode 213
    - WS\_SQLDML\_REPLICATION パラメータ 219
    - ウォームスタンバイの設定 219
    - オートコレクション 216
    - 制限 216
    - セッションレベル 213
    - データベース複写定義 214
    - データベースレベル 212
    - テーブル複写定義 215
    - テーブルレベル 213
    - 複写スレッシュホルドの設定 183
    - ローカウントの検証 217
  - sqm\_async\_seg\_delete 設定パラメータ 70
  - sqm\_reader\_first 設定パラメータ 14
  - SQM コマンドキャッシュ 68
  - sqt\_max\_cache\_size 180, 187, 188
  - sre\_reserve 188
  - sts\_cachesize 188
  - sts\_full\_cache 180
  - Sybase Control Center 50
  - Sybase Control Center for Replication Server 77
  - Sybase Control Center エージェント 77
  - Sybase IQ
    - RTL の設定 114
    - エラークラスとファンクション文字列クラス 113
    - コネクションパラメータ、設定 83
    - ステージングソリューションからのマイグレート 124
    - 接続プロファイル 112
    - レプリケートデータベースの設定 112
  - 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 124
  - Sybase IQ、複写 103, 160
  - Sybase IQ へのコネクション
    - カスタマイズ 83
- ## T
- timestamp データ型 249
- ## U
- Unicode の機能強化 75
- ## V
- varbinary\_strip\_trailing\_zeros 設定パラメータ 77
  - varbinary 値
    - 後続ゼロの削除 77
  - VMware 50
- ## W
- worker\_thread\_num 設定パラメータ 15
- ## あ
- 暗号化
    - パスワード 58
  - 暗号化パスワード
    - 拡張サポート 235
- ## い
- 異機種並列 DSI 161
  - インクリメンタル解析 45
  - インメモリデータベース 155, 184
  - 引用符付き識別子
    - alter replication definition、変更 200
    - create replication definition、変更 199
    - dsi\_quoted\_identifier 198
    - quoted パラメータ 199
    - rs\_set\_quoted\_identifier 201
    - 埋め込み二重引用符 198
    - 常に引用符を追加 46

## 索引

複写定義でサポートされていない 46

## う

ウォームスタンバイ

Oracle 163

異機種 163

データベースの再同期 145

複写定義の削減 63, 71

## え

エグゼキュータコマンドキャッシュ 69

エラークラス、Sybase IQ 113

エラー処理

機能強化 174

## お

オペレーティングシステムのサポート、  
Replication Server 15.5 186

## か

解析、インクリメンタル 45

カウンタ 237

カスケードコネクション、Replication Server ゲ  
ートウェイ 204

監査コマンド 59

干渉と影響、SAP IQ への複写 108

## き

機能強化

admin who 177

locales ディレクトリ 234

log\_first\_tran 233

Replication Server のパフォーマンス 191

resume connection 233

SQM パフォーマンス 243

sysadmin dump\_queue 232

sysadmin sqt\_dump\_queue 232

エラー処理 174

ステーブルキューの管理 231

ダンプトランザクション 240

動的 SQL 227, 247

ファンクション複写 228, 247

モニタリングとカウンタ 230

リリース領域 234

キャッシュ

SQL コマンドキャッシュでの LTL コマンド  
68

SQM コマンド 68

コマンドを動的に 69

テーブルメタデータ 69

キューの処理時間、推定 46

キューブロックサイズ

制限 128

変更 129

例、単純な複写システム 130

キューブロックサイズの増加 128, 167

強化、メモリ割り付け 167

## く

グローバルタスクマネージャ 15

## こ

向上、DSI 効率 166

向上、RepAgent エグゼキュータスレッドの効率  
166

向上、ディストリビュータスレッドの読み込  
み効率 167

後続ゼロ

削除 77

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 56

コマンド

admin health 47

admin config 192

admin sqm\_process\_time 46

admin who 47

alter replication definition 200

configure replication server 196

connect 205

create replication definition 199

disconnect 206

show connection 206

show server 206

sysadmin dump\_tran 234

sysadmin issue\_ticket 179

sysadmin sqm\_unzap\_tran 234  
 sysadmin sqm\_zap\_tran 233  
 sysadmin\_lmconfig 76  
 混合バージョン環境での制約、バージョン 15.5  
 以降 186  
 コンパイルとバルク適用、RTL 104  
**さ**  
 最低限の DML ロギング 185  
 細密なパーミッション  
   Replication Server によるサポート 42  
 削除、例外 148  
 作成  
   SAP IQ への接続 113  
 サブキャパシティライセンス 101  
 サブスクリプションの制御  
   アーティクル 49  
   パブリケーション 49  
   複写定義 49  
 サブスクリプションの文字数の引き上げ 76  
 サポート  
   Adaptive Server 共有ディスククラスタ 229  
   Adaptive Server における整数 ID 243  
   LOB データ型 238, 248  
   opaque データ型 248  
   timestamp データ型 239, 249  
   ダイレクト I/O 245  
     次も参照：バルクコピーインのサポ  
     ート  
 サポートパッケージ、置き換え 35  
**し**  
 シームレスアップグレード 155  
   Replication Server 15.7 64  
 システムテーブル  
   rs\_status 95  
   rs\_subscriptions 76  
 事前計算済み結果セット  
   Adaptive Server のサポート 41  
 自動起動設定パラメータ 80  
 自動でサイズ変更可能なパーティション 46  
   管理とモニタリング 46  
 シナリオ、データベース再同期化 139  
 シナリオ、データベース再同期化、ウォーム  
   スタンバイ 145

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  
   143  
**新機能**  
   Replication Manager 15.1 247  
   Replication Manager 15.5 189  
   Replication Server 15.2 191  
   Replication Server 15.5 159  
   Replication Server 15.6 101  
   Replication Server 15.6 ESD #1 87  
   Replication Server 15.7.1 51  
**新機能の概要**  
   Replication Server 15.7.1 SP100 35  
   Replication Server 15.7.1 SP200 5  
**す**  
 スキーマ変換 83  
 スキャナスキーマキャッシュ 43  
**複数のスキャナ**  
   セカンダリトランケーションポイントの  
   管理 44  
   説明 44  
 ステابلキューのキャッシュ 243  
 ストアドプロシージャの複写  
   execute as 句の影響 42  
**せ**  
**制限事項**  
   LOB データ型 238  
   opaque データ型 240  
   動的 SQL 228  
   ファンクションの複写 229  
   部分更新 239  
**製品エディション、種類 101, 159**  
**セカンダリトランケーションポイントの管理**  
   説明、複数のスキャナのサポート 44  
**セキュリティ**  
   Adaptive Server のマスタキー 52  
   Adaptive Server でのサポート 52  
   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 52  
   rs password 設定パラメータ 52  
   sa ユーザパスワードのリセット 59  
   コマンド監査 59  
   デフォルトパスワードなし 59

## 索引

- パスワード暗号化 58
- パスワードパラメータ 58
- 推奨事項 47, 60
- セキュリティ変更
  - 監査可能コマンド 57
  - パスワードセキュリティ 57
- セグメントの削除
  - 専用デーモン 70
- セグメントの事前割り付け 245
- 接続プロファイル 223
  - Sybase IQ 112
- 接続プロファイル、Sybase IQ 112
- 設定パラメータ
  - dist\_sqt\_max\_cache\_size 187
  - dsi\_bulk\_copy 191-193
  - dsi\_bulk\_threshold 192, 193
  - dsi\_non\_blocking\_commit 196
  - dsi\_row\_count\_validation 152
  - dsi\_sqt\_max\_cache\_size 187
  - exec\_cmds\_per\_timeslice 179
  - init\_sqm\_write\_delay 179
  - init\_sqm\_write\_max\_delay 179
  - mem\_thr\_dst 73
  - mem\_thr\_exec 73
  - mem\_thr\_sqt 73
  - mem\_warning\_thr1 73
  - mem\_warning\_thr2 73
  - memory\_control 73
  - memory\_limit 180, 188
  - smp\_enable 180
  - sqm\_cache\_size 70
  - sqt\_max\_cache\_size 180, 187
  - sts\_full\_cache 180
  - unicode\_format 75
  - 直接ロードマテリアライゼーション 31, 38

## そ

- 増加、キューブロックサイズ 128, 167
- 挿入、rs\_ticket マーカ 179
- 増分データ転送、サポート 183

## た

- ターゲットスコープファンクション文字列 63
- ダイレクト I/O によるファイルアクセス 245

- タスクのスケジューリング、複写 172
- タスクマネージャ、グローバル 15
- ダンプの適用 138

## ち

- 遅延名前解決、複写サポート 182
- 逐次化メソッド
  - wait\_after\_commit 162
- 直接ロードマテリアライゼーションの設定
  - mat\_load\_tran\_size 39
  - max\_mat\_load\_threads 39

## て

- ディストリビュータスレッドの読み込み効率の向上 167
- データ圧縮 39
  - Adaptive Server でのサポート 51
- データ型
  - bigdatetime 181
  - bigtime 181
  - image 157
  - Java 157
  - opaque 240, 248
  - timestamp 249
- データ型変換 83
- データサーバインタフェース 191, 193
- データ転送、増分 183
- データベースオブジェクト所有権の譲渡
  - Adaptive Server のサポート 41
- データベース再同期化 173
- データベース世代番号のリセット 178
- データベースのサポート、Real-Time Loading 81, 103, 161
- データベースの縮小
  - Replication Server によるサポート 40
- データベースのダンプ、取得 138
- データベースの再同期 135, 173
  - DSI のモニタリング 138
  - resuming connection コマンドと skip to resync パラメータ 135
  - skip to resync パラメータ 135
- シナリオ 139
- シナリオ、ウォームスタンバイ 145

シナリオ、データベース再同期マーカのサポートなし 143

ダンプデータベースマーカの送信 138

データベースのダンプの取得 138

データベースのダンプの適用 138

トランザクションのスキップ 135, 136

再同期マーカ、送信 136

設定 135

データベースの再同期の設定 135

DSI スレッド情報のモニタリング 138

Replication Server へのダンプデータベースマーカの送信 138

Replication Server に対するトランザクションのスキップの指示 135

Replication Server へのデータベース再同期マーカの送信 136

データベースのダンプの取得 138

再同期するデータベースへのダンプの適用 138

データベース再同期化シナリオ 139

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 145

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 141

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 139

再同期マーカに対するサポートがない場合の再同期 143

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 144

テーブル

rs\_lastcommit、Oracle データベース内 109

テーブルの複写

ユニークキーがない 47

テーブルメタデータ

キャッシュ 69

テキスト更新の強化 242

デフォルトパスワードの削除 59

デフォルトパラメータ値、変更 84

デフォルトパラメータの値、変更 179

## と

動的 SQL

機能強化 227, 247

動的 SQL の機能強化 167

動的 SQL、replicate minimal columns 句の使用 168

動的 SQL、replicate\_minimal\_columns パラメータのコネクションへの拡張 168

動的 SQL、文の実行の最適化 168

トリガの実行、Oracle 164

## は

バージョンの採番、変更 35

バージョンのサポート

Adaptive Server の再同期 134

パーティション

自動的に拡大/縮小 46

パーティションの作成

自動でサイズ変更可能 46

パーミッション、レプリケート SAP IQ 111

廃止

Replication Manager 64

パスワード

入力の非表示 57

パスワード暗号化

拡張サポート 235

パスワードセキュリティ管理

rs\_dictionary システムテーブル 60

rs\_passwords システムテーブル 60

rs\_users システムテーブル 60

システムテーブルサポート 60

パスワードパラメータ、設定 58

パスワード入力の非表示 57

パスワードの暗号化

メンテナンスユーザ 49

パスワードの有効期限

メンテナンスユーザ 49

バックリンクポインタ 40

パフォーマンスの強化

Advanced Services Option 164

DSI 効率の向上 166

High Volume Adaptive Replication 165

Replication Server 15.7 68

Replication Server 15.7.1 45, 61

## 索引

- パフォーマンスの強化、Replication Server 15.5 164
- パフォーマンスの強化、Replication Server 15.6 126
- パラメータ、変更、デフォルト値 84
- パラメータ、デフォルト値の変更 179
- 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 192
  - コネクションパラメータの値を調べる 192
  - データサーバインタフェース (DSI)、実装 191
  - 複数文のトランザクションのサポート 193

## 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 197
  - ASE 遅延コミット機能 195
  - dsi\_non\_blocking\_commit 196
  - Oracle、サポート 197
  - rs\_non\_blocking\_commit 196
  - rs\_non\_blocking\_commit\_flush 197
  - 設定 196

## ふ

- ファンクション複写
  - 機能強化 228
- ファンクション文字列
  - rs\_dsi\_check\_thread\_lock 163
  - rs\_non\_blocking\_commit 196
  - rs\_non\_blocking\_commit\_flush 197
  - rs\_set\_quoted\_identifier 201
- ファンクション文字列クラス、Sybase IQ 113
- 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 169
- フェールバックシステム、複写の遅延 172
- 複写
  - データ、大量バッチ 191
- 複写スレシヨルドの設定、SQL 文の複写 183
- 複写タスクのスケジュール 172

- 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 63
- 複写定義の削減
  - MSA 63
  - ウォームスタンバイ 63, 71
- 複写定義の変更要求プロセスの強化 170
- 複写のサスペンド
  - オプション 46
- 複写の遅延 172
- 複数のレプリケーションパス 54, 67
  - Adaptive Server モニタリングテーブル 56
  - monRepSenders モニタリングテーブル 56
- プラットフォームのサポート 50
- プラットフォームのサポート、Real-Time Loading 81, 103, 161
- プロセスの強化、複写定義の変更 170
- ブロックサイズ
  - 変更 129
- プロファイル
  - 接続 112
- 分散トランザクション管理 (DTM) 17

## へ

- 並列 DSI パラメータ
  - dsi\_max\_cmds\_in\_batch 162
  - dsi\_max\_xacts\_in\_group 163
  - dsi\_serialization\_method 163

## ま

- マルチパスレプリケーション 67
  - 分散モード 55

## め

- メモリ消費の制御
  - HVAR 82
  - RTL 82
- メモリ割り付けの強化 167
- メンテナンスユーザ
  - パスワードの暗号化 49
  - パスワードの有効期限 49
  - 権限の付与 111

## ゆ

ユニークキー  
ユニークキーがないテーブルの複写 47

## よ

予約語、新規 181

## ら

ライセンス 67, 101  
取得 102  
種類 101  
ライセンス、種類 159

## り

リセット、データベース世代番号 178  
リトライメカニズム、HVAR と RTL での強化  
126  
リファレンス実装 175  
利便性とプロセスの強化、Replication Server  
15.5 170  
利便性とプロセスの強化、Replication Server  
15.6 134  
利便性とプロセスの向上  
Replication Server 15.7 45, 63, 71  
リラックス持続性データベース 155, 184  
リリースの採番、変更 35

## る

ルートアップグレード 155  
ルートのアップグレード 155

## れ

例外の削除 148  
トランザクション ID の範囲 148

日付 149  
ユーザまたは送信先サイト 151

## 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 148  
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 149  
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  
151

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 53  
レプリケート SAP IQ のコネクティビティ 110  
レプリケート SAP IQ のデータベースパーミ  
ッション 111  
レプリケート Sybase IQ の設定 112  
レプリケートデータベース  
SAP IQ 108

## ろ

ローカウントの検証  
機能強化 152, 153  
テーブル名の表示 153  
無効化 152  
ローカウントの検証、SQL 文の複写 217  
ローカウントの検証の機能強化 174  
ロー内/ロー外の LOB サポート 52  
ログ  
例外 148, 149, 151  
ロックスキーマ、RSSD 180

## わ

ワーカスレッド、数の制御 15

