



プライマリ・データベース・ガイド

Replication Agent™ 15.7.1

Linux、Microsoft Windows、および UNIX 版

ドキュメント ID：DC01885-01-1571-01

改訂：2012年4月

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

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

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

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

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

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

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

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

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

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

目次

表記規則	1
Replication Agent for Oracle	5
Oracle 固有の考慮事項	5
サポート対象外の Oracle ソフトウェア機能	5
サポート対象外の Oracle データ型、データ、 構造	6
サポート対象の Oracle データ型、データ、構 造	8
Replication Agent コネクティビティ	8
Replication Agent のパーミッション	9
複数の Replication Agent	13
高可用性	16
REDO ログとアーカイブ・ログの設定	16
サプリメンタル・ロギング	20
DDL 複写	21
データベース・オブジェクト名の大文字と小 文字	23
オリジン・キュー ID のフォーマット	24
LTL オリジンのコミット時刻の細分性	25
Replication Server と RSSD のスクリプト	25
Oracle データ型の互換性	26
Oracle データ型の制限事項	34
Oracle ラージ・オブジェクト (LOB) のサポー ト	36
Oracle ユーザ定義データ型	41
シーケンスのマーク付けとマーク解除	44
シーケンスの複写の有効化と無効化	47

さまざまなマシン上での Replication Agent と	
Oracle の設定	49
Real Application Clusters (RAC)	49
Automatic Storage Management	52
Replication Serverset autocorrection コマンド	53
分割されたテーブル	54
マテリアライズド・ビュー	55
インデックス構成テーブル	61
レプリケート・データベースでのトリガ実行 の制御	62
プライマリ・データ・サーバからの複写定義 の変更	62
Oracle Data Guard	63
データベース再同期化	64
Oracle トランザクションおよびオペレーショ ンのトラブルシューティング	64
BOOLEAN 引数を持つストアド・プロシー ジャの複写	65
Oracle ウォーム・スタンバイ	68
Oracle Flashback	69
XMLTYPE データ・レプリケーション	72
Oracle 11g Release 2	76
Oracle 9i	78
Oracle プライマリ・データベースでの Replication	
Agent オブジェクト	78
Replication Agent オブジェクト名	79
テーブル・オブジェクト	80
マーカ・オブジェクト	82
シーケンス	82
マーク付きのプロシージャ	82
トランザクション・ログのトランケーション	83
Replication Agent for Microsoft SQL Server	85
Microsoft SQL Server 固有の考慮事項	85

Microsoft SQL Server の要件	85
Microsoft SQL Server の制限	86
サポート対象外のソフトウェア機能	86
サポート対象外のデータ型	87
Microsoft SQL Server パッチの適用	88
DDL 複写	90
Replication Agent コネクティビティ	92
Replication Agent のパーミッションと役割	93
sybfilter ドライバ	93
プライマリ・データ・サーバと Replication Agent の初期化	93
Microsoft sqlcmd ツール	98
データベース・オブジェクト名の大文字と小文字	98
オリジン・キュー ID のフォーマット	99
Microsoft SQL Server とのデータ型互換性	100
ntext データ型複写	104
プライマリ・データ・サーバからの複写定義の変更	106
Replication Serverset autocorrection コマンド . 計算カラム	107
Microsoft SQL Server プライマリ・データベースの Replication Agent オブジェクト	107
Replication Agent オブジェクト名	108
Microsoft SQL Server での Windows 認証の使用	112
さまざまなマシン上での Replication Agent と Microsoft SQL Server のセットアップ	113
Replication Agent for UDB	115
IBM DB2 固有の考慮事項	115
サポート対象外のソフトウェア機能	115
サポート対象外のデータ型	116
Replication Agent for UDB での機能差	116

IBM DB2 要件	118
さまざまなマシン上での Replication Agent と DB2 Server	120
Replication Agent for UDB 接続パラメータ	122
ログでの再配置	122
Replication Agent for UDB の動作	123
データベース・オブジェクト名の大文字と小 文字	125
オリジン・キュー ID のフォーマット	126
DB2 データ型の互換性	127
Replication Serverset autocorrection コマンド .	132
長い識別子	132
圧縮	132
DB2 プライマリ・データベースの Replication Agent オブジェクト	133
Replication Agent オブジェクト名	133
テーブル・オブジェクト	134
Java プロシージャ・オブジェクト	135
Replication Agent オブジェクトの名前の検索 .	135
マーク付けされたオブジェクトのテーブル	136
トランザクション・ログのトランケーション ...	136
Replication Agent のアップグレードとダウングレード ...	139
Replication Agent for Oracle のアップグレードおよび 移行手順	139
Replication Agent for Oracle の 15.7.1 へのアッ プグレード	140
Oracle 10g を 11g にアップグレードする際の Replication Agent for Oracle 15.7.1 の移行 ...	141
Replication Agent for Microsoft SQL Server のアップ グレード手順	141
Replication Agent for Microsoft SQL Server の 15.7.1 へのアップグレード	142

Replication Agent for UDB のアップグレードおよび移行手順	143
Replication Agent for UDB の 15.7.1 へのアップグレード	143
DB2 をバージョン 8.2 または 9.1 からバージョン 9.5 または 9.7 にアップグレードする際の Replication Agent for UDB の移行	144
Replication Agent for Oracle のダウングレード	145
Replication Agent for Microsoft SQL Server のダウングレード	147
Replication Agent for UDB のダウングレード	148
sybfilter ドライバのリファレンス	151
Microsoft Filter Manager Library のバージョンの確認	151
sybfilter ドライバのインストールと設定	151
トラブルシューティング	154
トレース・ログの使用	155
sybfilter コマンド・リファレンス	155
add	155
check	156
exit	156
help	156
list	156
refresh	156
remove	156
start	157
stop	157
trace	157
追加の説明や情報の入手	159
サポート・センタ	159
Sybase EBF と Maintenance レポートのダウンロード	159
Sybase 製品およびコンポーネントの動作確認	160

目次

MySybase プロファイルの作成	160
アクセシビリティ機能	161
用語解説	163
索引	173

表記規則

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

表記規則

構文要素	定義
等幅 (固定幅)	<ul style="list-style-type: none"> SQL およびプログラム・コード 表示されたとおりに入力する必要があるコマンド ファイル名 ディレクトリ名
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"> ファイルおよび変数の名前 他のトピックまたはマニュアルとの相互参照 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) 用語解説に含まれているテキスト内の用語
bold sans serif	<ul style="list-style-type: none"> 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 用語解説のエントリ (用語解説内) メニュー・オプションのパス 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

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。例：

次のコマンドを実行します。

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

installation directory は、アプリケーションのインストール・ロケーションです。

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**、および **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)、二重引用符でオブジェクト名を区切ります。例：**pdb_get_tables "TableName"**
- 識別子および文字データでは、使用しているソート順によっては大文字と小文字が区別されます。
 - “binary” などの大文字と小文字を区別するソート順を使用する場合には、識別子や文字データは、大文字と小文字を正しく入力してください。
 - “nocase” などの大文字と小文字を区別しないソート順を使用する場合には、識別子や文字データは、大文字と小文字をどのような組み合わせでも入力できます。

用語

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

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

Replication Agent for Oracle

Replication Agent for Oracle に特有の Replication Agent の特性を確認します。

「Replication Agent for Oracle」という用語は、Oracle データ・サーバに配置されているプライマリ・データベース用にインストールおよび設定された Replication Agent ソフトウェアのインスタンスを指します。

注意： Replication Agent の基本的な機能とオペレーションの詳細については、『Replication Agent 管理ガイド』と『Replication Agent リファレンス・マニュアル』を参照してください。

Oracle 固有の考慮事項

以下の全般的な問題と考慮事項は、Replication Agent と Oracle データ・サーバの併用に固有です。

サポート対象外の Oracle ソフトウェア機能

Replication Agent でサポートされない Oracle 機能について説明します。

以下の機能はサポートされていません。

- Oracle の仮想カラム
- Oracle の暗号化されたパーティション
- Oracle のラベル・セキュリティ
- Oracle のパッケージのストアド・プロシージャおよびファンクション (スタンドアロンのプロシージャおよびファンクションがサポートされます。)
- 暗号化されたテーブル領域内の Oracle スキーマ・オブジェクト
- Replication Server[®] 並列 DSI
- Replication Server の `rs_init` ユーティリティ
- Replication Server `rs_subcomp` ユーティリティ
- Replication Server の自動マテリアライゼーション
- 他のベンダが複写している環境で複写を行う際の Replication Server

プライマリ・キーでの遅延更新の複写

ユニークなカラム・インデックスを持つテーブルでの更新は、従来の複写ではサポートされていないため、Replication Server はエラーを報告します。

ユニークなカラム・インデックスを持つテーブルの更新の複写は、サポートされていないため、Replication Server はエラーを報告します。たとえば、テーブル t は

カラム *c* 上にユニークなインデックスを持ち、値は 1、2、3、4、および 5 です。このテーブルには単一の **update** 文が適用されます。

```
update t set c = c+1
```

従来 の 複 写 を 使 用 す る と、こ の 文 は 以 下 の よう に な り ま す。

```
update t set c = 2 where c = 1
update t set c = 3 where c = 2
update t set c = 4 where c = 3
update t set c = 5 where c = 4
update t set c = 6 where c = 5
```

最初の更新で、値 *c*=2 をテーブルに挿入しようとしています。ところが、この値はテーブルに既に存在します。Replication Server は、エラー 2601 (重複キーの挿入の試み) を表示します。

ASE はこの問題を SQL 文機能で解決します。この機能により、更新を遅延させ逆順に実行することで、重複キー・エラーを回避します。ただし、Sybase 以外のデータ・サーバではこの機能は提供されていないので、ユニークなカラム・インデックスを持つ Sybase 以外のテーブルへの更新を複写しようとする、Replication Server DSI がクラッシュします。この問題を回避するには、ユニークなインデックス定義を拡充します。

サポート対象外の Oracle データ型、データ、構造

Replication Agent でサポートされない Oracle データ型について説明します。

次のデータ型はサポートされていません。

- Oracle 提供のデータ型：
 - SYS.ANYDATA を除く “Any” 型 (**SYS.ANYTYPE**、**SYS.ANYDATASET**)
 - MLSLABEL
 - 地理空間型 (MDSYS.SDO_GEOMETRY、SDO_TOPO_GEOMETRY、SDO_GEORASTER)
 - メディア型 (ORDSYS.ORDAudio、ORDSYS.ORDImage、ORDSYS.ORDImageSignature、ORDSYS.ORDVideo、ORDSYS.ORDDoc、SI_StillImage、SI_Color、SI_AverageColor、SI_ColorHistogram、SI_PositionalColor、SI_Texture、SI_FeatureList)
 - Oracle Expression Filter で使用されるデータ型
 - ANYDATA (複写先が非 ANYDATA カラムである場合、またはデータのサイズが Replication Server の OPAQUE データ型のサイズ制約である 16KB を超えている場合)
 - ANYDATA カラムに格納されている BFILE、UROWID、または REF データ

- 複写先が Oracle または Adaptive Server® Enterprise (ASE) 以外のレプリケート・データベースである BLOB またはオブジェクト・ユーザ定義データ型のデータ。これらのデータ型は ECDA for Oracle によってサポートされていないため、Oracle に対して複写する際は ExpressConnect for Oracle を使用してください。
- REF
- UROWID
- **LOB** データが含まれているユーザ定義データ型
- **NOT FINAL** として定義されているユーザ定義データ型
- Oracle LOB の部分更新はサポートされていません。
- 仮想カラム – Replication Agent は、Oracle 11g の計算カラム (または仮想カラム) が含まれているテーブルの複写をサポートしています。ただし、個々の計算カラムの複写はサポートされていません。**force** オプションを使用すると、仮想カラムを持つテーブルに複写のマークを付けることができますが、仮想カラムは複写されません。

Oracle XML DB レポジトリに格納されているデータが次のプロトコルまたは方法でアクセスされる場合、そのデータはサポートされません。

- FTP、HTTP、HTTPS、WebDAV などのインターネット・プロトコル
- Oracle XML DB レポジトリ API

以下のユーザ定義オブジェクト・タイプおよび構造はサポートされていません。

- 関連配列
- ネストされたテーブル
- VARRAY
- ANYDATA カラムに格納されているネストされたテーブルまたは VARRAY

事前に定義された PL/SQL 数値データ型

Replication Agent は、事前に定義された PL/SQL 数値データ型である

PLS_INTEGER または BINARY_INTEGER が含まれているプロシージャのマーク付けをサポートしていません。

Replication Agent は、PLS_INTEGER のサブタイプである SIMPLE_INTEGER データ型を含んだプロシージャのマーク付けをサポートしています。ただし、前述のタイプのその他のサブタイプ (NATURAL、NATURALN、POSITIVE、POSITIVEN、SIGNTYPE など) が含まれたプロシージャのマーキングはサポートしていません。

サポート対象の Oracle データ型、データ、構造

Replication Agent は、暗号化データ、圧縮データ、および SecureFile の複写をサポートしています。

暗号化データ

Replication Agent では、Oracle バージョン 11g Release 1 以降における暗号化カラム、パーティション、およびテーブル領域でのデータの複写を透過的にサポートします。

お使いのプライマリ Oracle データベースのバージョンが、11g Release 1 より前の場合でも、**force** オプションを使用して暗号化カラム、パーティション、およびテーブル領域があるテーブルをマーク付けできますが、カラムは複写されません。

圧縮データ

Replication Agent では、Oracle バージョン 11g Release 1 以降について、Oracle 直接ロード・オペレーション (I/O バッファのバイパス) のために圧縮されたデータ、圧縮テーブル領域、および圧縮テーブルの複写をサポートします。

SecureFile データ

Replication Agent は、SecureFile オプションで格納された Oracle データの複写をサポートしています。

Replication Agent は、Oracle バージョン 11g Release 2 以降について、SecureFile オプションで格納された Oracle LOB データの複写をサポートしています。お使いのプライマリ Oracle データベースのバージョンが、11g Release 2 より前の場合でも、SecureFile LOB カラムがあるテーブルを複写するようマーク付けできますが、カラムは複写されません。

Replication Agent コネクティビティ

Replication Agent for Oracle では、すべての複写システム・コンポーネントとの通信に JDBC™ を使用します。

Oracle JDBC ドライバは Replication Agent ホスト・マシンにインストールされ、このドライバのインストール先ディレクトリは CLASSPATH 環境変数に含まれる必要があります。

Replication Agent インスタンスが TNS リスナ・サービスに接続するには、TNS リスナ・サービスをプライマリ・データベースにインストールし、実行しておく必要があります。『Oracle Database Net Services Administrator's Guide』を参照してください。インストールする特定の JDBC ドライバおよびバージョンの詳細については、『Replication Agent インストール・ガイド』を参照してください。

Replication Agent のパーミッション

Replication Agent for Oracle では、Oracle との接続に **pds_username** が使用されており、Oracle パーミッションが必要となります。

以下のパーミッションが必要です。

- **GRANT ALTER ANY PROCEDURE** – 複写対象プロシージャを管理するために必要となります。
- **GRANT ALTER DATABASE** – Replication Agent が Data Guard スタンバイ・データベースのトランザクション・ログからの読み取りを行うために必要となります。
- **GRANT ALTER ON *TABLE_NAME*** – 指定した *TABLE_NAME* に対してテーブル・レベルのサプリメンタル・ロギングが有効になっていない場合にユーザ定義データ型を複写するうえで必要となります。
- **GRANT ALTER SYSTEM** – ログ・アーカイブ・オペレーションのやり直しを実行するために必要となります。
- **GRANT CREATE ANY PROCEDURE** – 複写対象プロシージャにマークを付けるために必要となります。
- **GRANT CREATE PROCEDURE** – **rs_marker** および **rs_dump proc** プロシージャを作成するために必要となります。
- **GRANT CREATE PUBLIC SYNONYM** – プライマリ・データベースに作成されたテーブルの同義語を作成するために必要となります。
- **GRANT CREATE SEQUENCE** – 複写をサポートするうえで必要となります。
- **GRANT CREATE SESSION** – Oracle との接続で必要となります。
- **GRANT CREATE TABLE** – プライマリ・データベースでのテーブルの作成に必要となります。
- **GRANT DROP PUBLIC SYNONYM** – 作成された同義語を削除するうえで必要となります。
- **GRANT EXECUTE_CATALOG_ROLE** – Oracle LogMiner を使用するために必要となります。
- **GRANT EXECUTE ON DBMS_FLASHBACK** – **DBMS_FLASHBACK.get_system_change_number** を実行するために必要となります。
- **GRANT EXECUTE ON SYS.DBMS_LOCK** – プライマリ・データベースで **commit** ログ・レコードを生成するために必要となります。
- **GRANT SELECT ANY TRANSACTION** – Oracle LogMiner を使用するうえで必要となります。
- **GRANT SELECT_CATALOG_ROLE** – **DBA_*** ビューからの選択に必要となります。

- **GRANT SELECT ON SYS.OPQTYPE\$** – DDL の複写と XMLTYPE データの複写で必要となります。
- **GRANT SELECT ON SYS.RECYCLEBIN\$** – Replication Agent で Oracle Flashback を使用するために必要となります。
- **GRANT SELECT ON SYS.ARGUMENT\$** – プロシージャの DDL コマンドを処理するために必要となります。
- **GRANT SELECT ON SYS.CCOL\$** – テーブルの複写 (カラム制約情報) をサポートするうえで必要となります。
- **GRANT SELECT ON SYS.CDEF\$** – テーブルの複写 (制約情報) をサポートするうえで必要となります。
- **GRANT SELECT ON SYS.COL\$** – テーブルの複写 (カラム情報) をサポートするうえで必要となります。
- **GRANT SELECT ON SYS.COLLECTION\$** – テーブルの複写をサポートするうえで必要となります。
- **GRANT SELECT ON SYS.COLTYPE\$** – テーブルの複写をサポートするうえで必要となります。
- **GRANT SELECT ON SYS.CON\$** – テーブルの複写 (制約情報) をサポートするうえで必要となります。
- **GRANT SELECT ON SYS.DEFERRED_STG\$** – LogMiner によって圧縮テーブルがサポートされていない Oracle 11g Release 2 で圧縮テーブルの複写を抑制するうえで必要となります。
- **GRANT SELECT ON SYS.IND\$** – インデックスの識別に必要となります。
- **GRANT SELECT ON SYS.INDCOMPART\$** – インデックスの識別に必要となります。
- **GRANT SELECT ON SYS.INDPART\$** – インデックスの識別に必要となります。
- **GRANT SELECT ON SYS.INDSUBPART\$** – インデックスの識別に必要となります。
- **GRANT SELECT ON SYS.LOB\$** – LOB の複写をサポートするうえで必要となります。
- **GRANT SELECT ON SYS.LOBCOMPPART\$** – 分割された LOB の複写をサポートするうえで必要となります。
- **GRANT SELECT ON SYS.LOBFrag\$** – 分割された LOB の複写をサポートするうえで必要となります。
- **GRANT SELECT ON SYS.MLOG\$** – マテリアライズド・ビューのログ・テーブルをフィルタするうえで必要となります。
- **GRANT SELECT ON SYS.NTAB\$** – テーブルの複写をサポートするうえで必要となります。
- **GRANT SELECT ON SYS.OBJ\$** – レポジトリ内のプロシージャ DDL コマンドを処理するために必要となります。
- **GRANT SELECT ON SYS.PROCEDUREINFO\$** – プロシージャの複写をサポートするうえで必要となります。

- **GRANT SELECT ON SYS.SEG\$** – LogMiner によって圧縮テーブルがサポートされていない Oracle のバージョンで圧縮テーブルの複写を抑制するうえで必要となります。
- **GRANT SELECT ON SYS.SEQ\$** – シーケンスの複写をサポートするうえで必要となります。
- **GRANT SELECT ON SYS.SNAP\$** – マテリアライズド・ビューのテーブルを除外するうえで必要となります。
- **GRANT SELECT ON SYS.TAB\$** – テーブルの複写をサポートするうえで必要となります。
- **GRANT SELECT ON SYS.TABCOMPART\$** – 分割されたテーブルの複写をサポートするうえで必要となります。
- **GRANT SELECT ON SYS.TABPART\$** – 分割されたテーブルの複写をサポートするうえで必要となります。
- **GRANT SELECT ON SYS.TABSUBPART\$** – 分割されたテーブルの複写をサポートするうえで必要となります。
- **GRANT SELECT ON SYS.TS\$** – Oracle 11g でテーブル領域の暗号化を識別するうえで必要となります。
- **GRANT SELECT ON SYS.TYPE\$** – Oracle の事前定義型とユーザ定義型を処理するうえで必要となります。
- **GRANT SELECT ON SYS.USER\$** – Oracle ユーザの識別に必要となります。
- **GRANT SELECT ON SYS.ATTRIBUTE\$** – Oracle の型を処理するうえで必要となります。
- **GRANT SELECT ON V_\$LOGMNR_CONTENTS** – Oracle LogMiner を使用するうえで必要となります。
- **GRANT SELECT ON V_\$LOGMNR_LOGS** – Oracle LogMiner を使用するうえで必要となります。

古いアーカイブ・ファイルを削除するように Replication Agent が設定されている場合は、ディレクトリとアーカイブ・ログ・ファイルに対する **UPDATE** 権限がユーザにあることが必要となります。

Replication Agent for Oracle では、**ALTER SYSTEM ARCHIVE LOG** コマンドを発行するための **ALTER SYSTEM** 権限が必要です。オンラインの Oracle REDO ログにのみアクセスするよう Replication Agent が設定されている場合に、(ログのすべてのデータが複写されたために) オンラインの REDO ログが複写で不要になると、Replication Agent は **ALTER SYSTEM ARCHIVE LOG SEQUENCE** コマンドを発行します。オンライン・ログまたはアーカイブ・ログの処理にかかわらず、Replication Agent は Oracle ログの末尾に処理を移動するよう指示を受けると、**ALTER SYSTEM** 権限を使用して **ALTER SYSTEM ARCHIVE LOG CURRENT** コマンドを発行します。Replication Agent は **ALTER SYSTEM ARCHIVE LOG CURRENT** コマンドを発行することにより、現在の REDO ログ・ファイルに古いデータが含まれていないことを確

認めます。Replication Agent は、**ra_locator** コマンドの **move_truncpt** オプションによって要求されると、Oracle REDO ログの末尾に処理を移動します。また Replication Agent は、あるバージョンの Replication Agent を別のバージョンに移行する際にも、Oracle REDO ログの末尾に処理を移動します。

DDL 複写に関する Oracle 10g および 11g の権限

注意： **GRANT ALL PRIVILEGES TO DDLUSER** を発行すると、DDL ユーザは SYS ユーザや SYSTEM ユーザなどのスーパーユーザになります。

Oracle のバージョンが異なると、パーミッション要件も異なります。Oracle 10g および 11g では、DDL ユーザ・パーミッションが付与されると、次のコマンドを実行できます。

- **GRANT ALTER ANY INDEX**
- **GRANT ALTER ANY INDEXTYPE**
- **GRANT ALTER ANY PROCEDURE**
- **GRANT ALTER ANY TABLE**
- **GRANT ALTER ANY TRIGGER**
- **GRANT ALTER ANY TYPE**
- **GRANT ALTER SESSION**
- **GRANT BECOME USER**
- **GRANT CREATE ANY INDEX**
- **GRANT CREATE ANY INDEXTYPE**
- **GRANT CREATE ANY PROCEDURE**
- **GRANT CREATE ANY SYNONYM**
- **GRANT CREATE ANY TABLE**
- **GRANT CREATE ANY TRIGGER**
- **GRANT CREATE ANY TYPE**
- **GRANT CREATE ANY VIEW**
- **GRANT CREATE INDEXTYPE**
- **GRANT CREATE MATERIALIZED VIEW**
- **GRANT CREATE PROCEDURE**
- **GRANT CREATE PUBLIC SYNONYM**
- **GRANT CREATE SYNONYM**
- **GRANT CREATE TABLE**
- **GRANT CREATE TRIGGER**
- **GRANT CREATE TYPE**
- **GRANT CREATE VIEW**
- **GRANT DELETE ANY TABLE**
- **GRANT DROP ANY INDEX**

- GRANT DROP ANY INDEXTYPE
- GRANT DROP ANY MATERIALIZED VIEW
- GRANT DROP ANY PROCEDURE
- GRANT DROP ANY SYNONYM
- GRANT DROP ANY TABLE
- GRANT DROP ANY TRIGGER
- GRANT DROP ANY TYPE
- GRANT DROP ANY VIEW
- GRANT DROP PUBLIC SYNONYM
- GRANT INSERT ANY TABLE
- GRANT SELECT ANY TABLE
- GRANT UPDATE ANY TABLE

DDL からユーザのパーミッションが取り消されると、次のコマンドを実行できません。

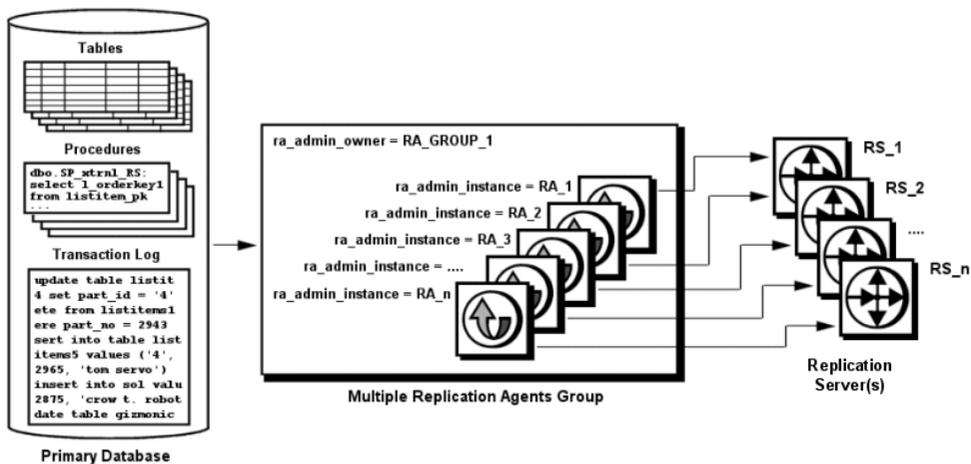
- GRANT ALTER DATABASE
- GRANT ALTER ROLLBACK SEGMENT
- GRANT ALTER SYSTEM
- GRANT ALTER TABLESPACE
- GRANT ANALYZE ANY
- GRANT AUDIT ANY
- GRANT AUDIT SYSTEM
- GRANT CREATE DATABASE LINK
- GRANT CREATE ROLLBACK SEGMENT
- GRANT CREATE TABLESPACE
- GRANT DROP PUBLIC DATABASE LINK
- GRANT DROP ROLLBACK SEGMENT
- GRANT DROP TABLESPACE
- GRANT LOCK ANY TABLE

複数の Replication Agent

複数の Replication Agent インスタンスを使用すると、独立した複数のパスに並列してトランザクションを複製できるため、複製のスループットおよびパフォーマンスが向上し、リソースの競合が低減されます。

Replication Server のマルチパス・レプリケーションで複数の Replication Agent インスタンスを使用すると、さまざまなストリームを介したデータの複製が有効になり、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。Replication Agent for Oracle では、複数の Replication Agent インスタンスを使用した Oracle テーブルとプロシージャの複製がサポートされます。

図 1 : マルチパス・レプリケーション



注意：一般的な Replication Agent グループの Replication Agent インスタンスはすべて、同じバージョンでなければなりません。

エンドツーエンドのマルチパス・レプリケーションのシナリオの説明については、『Replication Server 異機種間複写ガイド』を参照してください。

設定

Replication Server のマルチパス・レプリケーション・シナリオでは、複数の Replication Agent for Oracle インスタンスをまとめてグループ化し、一般的なシステム・リソースを共有して、マルチパス・レプリケーション・ソリューションで連携します。

- Replication Agent グループの各 Replication Agent for Oracle インスタンスは、**ra_admin_owner** パラメータと **ra_admin_prefix** パラメータに対して同じ値のセットを使用する必要があります。
- Replication Agent グループの各 Replication Agent for Oracle インスタンスは、**ra_admin_instance_prefix** パラメータに対してユニークな値のセットを使用する必要があります。このパラメータにより、グループ内の Replication Agent for Oracle インスタンスが識別されます。

『Replication Agent リファレンス・マニュアル』を参照してください。

複数の Replication Agent による DDL 複写

プライマリ・データベースですべての DDL オペレーションを複写するよう各 Replication Agent インスタンスを設定することも、どの DDL オペレーションも複写しないよう設定することもできます。

Replication Agent グループで、以下を複写するよう各 Replication Agent インスタンスを設定することもできます。

- インスタンスによって複写するようマーク付けされたすべてのオブジェクトの DDL オペレーション
- インスタンスによって複写するようマーク付けされたすべてのオブジェクトの DDL オペレーションと、どのインスタンスによっても複写するようマーク付けされていないオブジェクトの DDL オペレーション

複数の Replication Agent を使用することによって生じる DDL 同期エラーを避けるには、**pdb_setrepddl** コマンドで、以下のキーワードを使用します。

- **all** は、すべての DDL オペレーションがインスタンスによって複写されるよう指定します。
- **marked** は、インスタンスによって複写するようマーク付けされたオブジェクトの DDL オペレーションだけが複写されるよう指定します。
- **unmarked** は、どのインスタンスによっても複写するようマーク付けされていないオブジェクトの DDL オペレーションが複写されるよう指定します。
Nonschema DDL オペレーションも、このキーワードを指定するインスタンスによって複写されます。

例

3つのインスタンスを含む Replication Agent グループが、プライマリ・データベースのこれらのテーブル上で実行される DDL オペレーションを複写します。

- `table_A` は、インスタンス `ra_instance_1` によって複写するようマーク付けされます。
- `table_B` は、インスタンス `ra_instance_2` によって複写するようマーク付けされます。
- `table_C` は、インスタンス `ra_instance_3` によって複写するようマーク付けされます。
- `table_D` は、どのインスタンスによっても複写するようマーク付けされていません。
-

Replication Agent インスタンスは、**pdb_setrepddl** コマンドを使用して複写する DDL を指定します。

```

ra_instance_1>pdb_setrepddl enable, marked
ra_instance_1>pdb_setrepddl enable, unmarked
ra_instance_2>pdb_setrepddl enable, marked
ra_instance_3>pdb_setrepddl enable, marked
    
```

DDL は以下のように複写されます。

Replication Agent インスタンス	複写される DDL
ra_instance_1	table_A、table_D、および nonschema DDL オペレーション向け DDL
ra_instance_2	table_B 向け DDL
ra_instance_3	table_C 向け DDL

高可用性

プライマリ・データ・サーバでは、ハードウェアまたはソフトウェアでエラーが発生したときのダウン時間を最小にとどめるために、フェールオーバー・クラスタリングなどの高可用性ソリューションを使用することがあります。

Replication Agent には専用的高可用性ソリューションは用意されていませんが、次のような場合、プライマリ・データベースに対してサードパーティの高可用性ソリューションを使用できます。

- Replication Agent が OCFS、Network File System (NFS)、Veritas Cluster Server (VCS) などの共有ファイル・システムにインストールされている。Replication Agent バイナリ、設定ファイル、RASD ファイルは、システムにインストールする必要があります。
- Sun Cluster Manager、Veritas Cluster Manager、Oracle Cluster Ready Services (CRS) などのサードパーティ・クラスタ管理ソリューションを使用しているは、フェールオーバー発生時に Replication Agent が自動的に起動するために使用します。

REDO ログとアーカイブ・ログの設定

オンライン・ログとアーカイブ・ログのいずれにもデフォルトでアクセスできません。オンライン・ログにのみアクセスするよう Replication Agent を設定できますが、それを行うには、自動アーカイブをオフにする必要があります。Oracle に対する手動アーカイブ・ログ・コマンドの発行が Replication Agent で必要となります。

注意： Oracle アーカイブ・ログを直接トランケートするよう Replication Agent for Oracle が設定されている場合、つまり `rman_enabled` が `false` に設定され、`pdb_archive_remove` が `true` に設定されている場合は、Oracle アーカイブ・ログ・

ファイルに直接アクセスできるホストに Replication Agent for Oracle がインストールされている必要があります。

アーカイブ・ログへのアクセス

デフォルト設定を使用している場合に、アーカイブ・ログ・ファイルにアクセスするには、アーカイブ・ログ・ファイルが保存されているディレクトリ・パスを使用するよう Replication Agent を設定します。Oracle インスタンスのデフォルトでは、Oracle **ALTER SYSTEM** コマンドの **DB_RECOVERY_FILE_DEST** パラメータによって指定されたフラッシュ・リカバリ領域に複数のディレクトリが作成されます。各ディレクトリはそれぞれ別の日に対応し、その日付に基づいた名前が付けられます。ただし、Replication Agent では、アーカイブされた REDO ログ・ファイルを単一のディレクトリに配置する必要があります。そのため、Replication Agent によって読み取られる単一のディレクトリをアーカイブ先にするよう Oracle を設定しなければなりません。

注意： 他のアーカイブ・ファイル・プロセスとの競合を防ぐため、複写専用の追加の宛先ディレクトリにアーカイブ・ログ・ファイルを複製するよう Oracle を設定することが必要となる場合があります。

Oracle 環境向けにログのアーカイブ先を指定する方法については、Oracle の **ALTER SYSTEM** コマンドと **LOG_ARCHIVE_DEST_n** パラメータの説明を参照してください。

注意： ここでは、ファイルシステム・ファイルとして格納されている Oracle アーカイブ REDO ログへのアクセスについて説明しています。アーカイブ REDO ログの格納に Oracle ASM を使用している場合は、Automatic Storage Management の説明を参照してください。

Replication Agent for Oracle では、REDO ログのアーカイブが Oracle データベースで有効になっていることが必要となります。

```
alter database ARCHIVELOG;
```

注意： Oracle Real Application Clusters (RAC) を使用している場合は、REDO ログのアーカイブをクラスタ内の各インスタンスに対して有効にする必要があります。

ログ・アーカイブが有効になっていることを確認します。

```
select log_mode from v$database;
```

Oracle RAC を使用している場合は、次の SQL 文を使用して、ログ・アーカイブが有効になっていることを確認します。

```
select instance, name, log_mode from gv$database;
```

ARCHIVELOG (Oracle 10g では ARCHIVELOG または MANUAL) が返される場合は、ログ・アーカイブが有効になっています。

アーカイブ・ログ・ファイルへのアクセス

Replication Agent で、アーカイブされた REDO ログ・ファイルが格納されると考えられる場所に **pdb_archive_path** 設定プロパティを設定します。Replication Agent の **pdb_archive_remove** 設定パラメータを true に設定すると、複写をサポートするうえで必要でなくなったアーカイブ・ログ・ファイルを Replication Agent が削除できるようになります。

rman_enabled パラメータを設定すると、Replication Agent が Oracle **RMAN** ユーティリティを使用して古いアーカイブ・ログ・ファイルをトランケートできるようになります。『Replication Agent リファレンス・マニュアル』を参照してください。

Replication Agent アーカイブ設定

pdb_include_archives が true (デフォルト) に設定されている場合、Replication Agent はアーカイブを行わないため、REDO ログの自動アーカイブを実行するよう Oracle を設定しておくことをおすすめします。

pdb_include_archives 設定パラメータが false に設定されている場合、Replication Agent for Oracle では、Oracle REDO ログの自動アーカイブも無効にしておくことが必要となります。アーカイブは、オンライン REDO ログ・ファイルのデータが複写されるときに Replication Agent によって手動で実行されます。

Replication Agent for Oracle では、Oracle のバージョンにもよりますが、Oracle データベースでこれらの設定が指定されていることが必要となります。

参照：

- Automatic Storage Management (52 ページ)

Oracle 10g の場合の自動アーカイブの無効化

Oracle 10g の場合の自動アーカイブを無効にします。

1. **sysdba** 管理者権限があることを確認して、データベースを閉じます。
2. 次のように入力します。

```
alter database ARCHIVELOG MANUAL;
```

3. ログのアーカイブが無効になっていることを確認するには、次のように入力します。

```
select log_mode from v$database;
```

MANUAL が返される場合は、自動ログ・アーカイブが無効になっています。

Oracle 11g の場合の自動アーカイブの無効化

Oracle 11g の場合の自動アーカイブを無効にします。

1. LOG_ARCHIVE_START パラメータを変更するには、サーバ起動パラメータ・ファイルを手動で編集するか、次のように入力します。

```
alter system set log_archive_start=false scope=spfile;
```

2. LOG_ARCHIVE_START パラメータの設定を確認するには、次のように入力します。

```
select value from v$system_parameter where name =  
'log_archive_start';
```

3. false が返される場合は、Oracle サーバの再起動時に自動アーカイブを防ぐよう、サーバ・パラメータ・ファイル内の値が正しく変更されています。ALTER SYSTEM コマンドの LOG_ARCHIVE_START パラメータについては、『Oracle Database Reference Guide』を参照してください。
4. 自動アーカイブは Oracle サーバの再起動時とアクティブ・サーバの両方で無効にしておく必要があります。アクティブ・サーバで自動アーカイブを停止するには、次のように入力します。

```
alter system archive log stop;
```

5. Oracle サーバの再起動時に自動アーカイブを無効にするには、サーバの LOG_ARCHIVE_START パラメータの値を false に変更します。

注意： `pdb_include_archives` が false に設定されている場合：Replication Agent for Oracle が初期化された後の REDO ログ・ファイル処理については、自動アーカイブを一時的であっても有効にはなりません。自動アーカイブが有効になっている場合、または手動アーカイブが実行された場合は、Replication Agent によってまだ処理されていない REDO ログ・ファイルが上書きされ、失われた REDO ログ・ファイル内のデータは複製されません。アーカイブ・ログ・ファイルにアクセスするよう Replication Agent を再設定すると、この状況から回復できます。`pdb_include_archives` を true に設定し、上書きされたファイルのアーカイブが含まれているディレクトリのロケーションに `pdb_archive_path` を設定して、処理をレジュームします。処理が追いついたら、Replication Agent をサスペンドして、`pdb_include_archives` を false に再設定してください。

データベースに対するすべての変更の強制ロギング

複製する必要のあるすべてのデータが確実にログに記録されるようにするため、データベースに対するすべての変更の Oracle REDO ログ・ファイルへの強制ロギングを有効にしておくことをおすすめします。

1. プライマリ・データベースで次のコマンドを実行します。

```
alter database FORCE LOGGING;
```

2. 次のコマンドを実行して、現在の設定を確認します。

```
select force_logging from v$database;
```

Windows アーカイブの UNC パスおよびオンライン Redo ログのパス

Replication Agent for Oracle が Windows サービスとして実行中で、プライマリ Oracle データ・サーバが別のマシンにインストールされている場合、Microsoft Windows Universal Naming Convention (UNC) に従ってアーカイブおよびオンラインの redo ログのパスを次のように設定します。

```
¥¥oracle_server_machine¥oracle_log_path
```

oracle_server_machine はプライマリ Oracle データ・サーバが存在する場所で、*oracle_log_path* はアーカイブまたは redo ログ・ファイルです。たとえば、アーカイブ redo ログ・ファイルの場所を labratx64 という名前のマシンの oracle ディレクトリに設定する場合、次のように入力します。

```
1> ra_config pdb_archive_path, ¥¥labratx64¥oracle
2> go
```

サブリメンタル・ロギング

プライマリ・キー・データとインデックス・カラムのサブリメンタル・ロギングを有効にします。

サブリメンタル・ロギングを有効にするには、次の Oracle コマンドを実行します。

```
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
```

```
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY, UNIQUE INDEX) COLUMNS;
```

プライマリ・キーおよびユニーク・インデックスに関する情報の最小サブリメンタル・ロギングとサブリメンタル・ロギングが有効になっていることを確認するには、次のように入力します。

```
select SUPPLEMENTAL_LOG_DATA_MIN, SUPPLEMENTAL_LOG_DATA_PK,
SUPPLEMENTAL_LOG_DATA_UI from v$database;
```

各カラムに関して YES が返される場合は、プライマリ・キー情報のサブリメンタル・ロギングが有効になっています。

テーブルレベルのサブリメンタル・ロギング

ユーザ定義オブジェクト・タイプ属性に更新を複製するには、Replication Agent でテーブルレベルのサブリメンタル・ロギングを有効にしておく必要があります。

次のように入力して、テーブルレベルのサブリメンタル・ロギングを手動で有効にします。

```
ALTER TABLE THE_TABLE ADD SUPPLEMENTAL LOG DATA (ALL)
COLUMNS;
```

ここで、*THE_TABLE*には、サブリメンタル・ロギングを有効にするテーブルの名前を指定します。次のコマンドを実行して、テーブルレベルのサブリメンタル・ロギングが有効になっていることを確認します。

```
select count(*) from ALL_LOG_GROUPS where
LOG_GROUP_TYPE='ALL COLUMN LOGGING' and OWNER=THE_OWNER
and TABLE_NAME=THE_TABLE
```

ここで、*THE_OWNER*にはテーブルの所有者を指定します。このコマンドが値1を返す場合は、テーブルレベルのサブリメンタル・ロギングがこのテーブルに対して有効になっています。

サブリメンタル・ロギングは Replication Agent for Oracle から有効にすることもできます。これを行うには、『Replication Agent リファレンス・マニュアル』で説明されているとおりに **ra_set_autocorrection** コマンドを実行します。

DDL 複写

データ定義言語 (DDL) コマンドの複写はサポートされていますが、Oracle データベースのみに限られます。Oracle から Oracle 以外のレプリケート・データベースに DDL コマンドを複写することはできません。

Replication Agent では **pdb_setrep_ddl** コマンドを使用して、DDL コマンドの複写を有効または無効にします。Replication Agent for Oracle は、特定の DDL コマンドの複写をオブジェクト、所有者、文、またはユーザごとに有効または無効にすることができます。Replication Server はプライマリ・データベースで DDL コマンドを実行した同じユーザとして、**ddl_username** パラメータを使用してレプリケート・データベースで DDL コマンドを実行します。

pdb_setrepddl および **ddl_username** の使用の詳細については、『Replication Agent リファレンス・マニュアル』の「コマンド・リファレンス」の「**pdb_setrepddl**」、および『Replication Agent リファレンス・マニュアル』の「設定パラメータ」の「**ddl_username**」を参照してください。

DDL パラメータ

ddl_username および **ddl_password** パラメータを設定します。

DDL を Oracle に複写するには、**pdb_setrepddl** を使用してフィルタリング・ルールを適切に設定します。また、Replication Agent の **ddl_username** および **ddl_password** パラメータも設定する必要があります。**ddl_username** は、複写された DDL コマンドをターゲット・データベースで実行するために使用されるデータベース・ユーザ名です。このユーザには、複写されたすべての DDL コマンドをターゲット・データベースで実行するためのパーミッションが必要です。**ddl_password** は、

ddl_username に対応するパスワードです。また、**ddl_username** データベース・ユーザには、複製対象の DDL コマンドを発行する可能性があるすべてのプライマリ・データベース・ユーザに対して **ALTER SESSION SET CURRENT_SCHEMA** コマンドを発行するためのパーミッションが必要です。『Replication Agent リファレンス・マニュアル』を参照してください。

使用に関する特別な注意事項

ddl_username の値を、Replication Server でレプリケート・コネクション用に定義されているメンテナンス・ユーザと同じにすることはできません。これらの名前が同一の場合は、Replication Server でエラーが発生します。

ddl_username パラメータの値は、複製されたすべての DDL 文に対して LTL で送信されます。DDL を複製する場合、Replication Server は、**ddl_username** および **ddl_password** パラメータによって指定されたユーザ ID とパスワードを使用してレプリケート・データベースに接続します。Replication Server はその後で、次のコマンドを発行します。

```
ALTER SESSION SET CURRENT_SCHEMA=user
```

ここで、*user* はプライマリ・データベースで DDL 操作を生成したユーザ ID です。実際の DDL コマンドがレプリケート・データベースに対して実行されます。

ddl_username で指定されているユーザ ID に、**ALTER SESSION SET CURRENT_SCHEMA** を発行するパーミッションまたはこの DDL コマンドをユーザ・スキーマに対して実行するパーミッションがない場合は、コマンドが失敗します。

注意： DDL を複製するには、Replication Server にデータベースレベルの複製定義が必要であり、その定義で **replicate DDL** が設定されていなければなりません。『Replication Server リファレンス・マニュアル』を参照してください。

複製からフィルタされる DDL コマンドおよびオブジェクト

Oracle DDL コマンドおよびオブジェクトの中には複製されないものがあります。

次の DDL コマンドは複製されません。

- **alter database**
- **alter rollback segment**
- **alter session**
- **alter snapshot**
- **alter snapshot log**
- **alter system**
- **alter tablespace**
- **analyze**

- **audit**
- **create control file**
- **create database link**
- **create pfile from spfile**
- **create rollback segment**
- **create schema authorization**
- **create snapshot**
- **create snapshot log**
- **create spfile from pfile**
- **create tablespace**
- **drop database link**
- **drop rollback segment**
- **drop snapshot**
- **drop snapshot/log**
- **drop tablespace**
- **explain**
- **lock table**
- **no audit**
- **rename**
- **set constraints**
- **set role**
- **set transaction**

SYS によって所有されるオブジェクトはいずれも複製されません。複製されないユーザのリストで定義されたユーザが所有するオブジェクトは、どれも複製されません。このリストは、`pdb_ownerfilter` コマンドを使用して変更できます。さらに、複製されないオブジェクトの所有者のデフォルト・リストが提供されています。ただし、SYS 所有者を削除することはできません。複製されないオブジェクト所有者のリストの返却、追加、または削除には、`pdb_ownerfilter` コマンドを使用します。『Replication Agent リファレンス・マニュアル』を参照してください。

注意： `truncate table` コマンドは `rs_truncate` として複製されます。

データベース・オブジェクト名の大文字と小文字

データベース・オブジェクト名は、複製定義に指定したものと同一フォーマットでプライマリ Replication Server に送信する必要があります。そうしないと、複製に失敗します。たとえば、複製定義ではテーブル名がすべて小文字で指定されている場合、Replication Agent がプライマリ Replication Server にこのテーブル名を送信する際は、すべて小文字で表示されている必要があります。

Replication Agent がプライマリ Replication Server にデータベース・オブジェクト名を送信するときの大文字と小文字の区別を制御するには、`ltl_character_case` 設定パラメータを次のいずれかの値に設定します。

- `asis` – (デフォルト) データベース・オブジェクト名は、プライマリ・データ・サーバに格納されているものと同じフォーマットで Replication Server に渡されます。
- `lower` – データベース・オブジェクト名は、プライマリ・データ・サーバにどのように格納されているかにかかわらず、すべて小文字で Replication Server に渡されます。
- `upper` – データベース・オブジェクト名は、プライマリ・データ・サーバにどのように格納されているかにかかわらず、すべて大文字で Replication Server に渡されます。

Oracle データ・サーバのデフォルトでは、データベース・オブジェクト名はすべて大文字で格納されます。ただし、大文字と小文字を区別する名前を作成した場合は、大文字と小文字の区別が Oracle で保持されます。

以下は、**asis** オプションを使用した例です。

- `create table tabA` は TABA として格納される
- `create table TabB` は TABB として格納される
- `create table 'TaBc'` は TaBc として格納される

以下は、**upper** オプションを使用した例です。

- `create table tabA` は LTL で TABA として表示される
- `create table TabB` は LTL で TABB として表示される
- `create table 'TaBc'` は LTL で TABC として表示される

オリジン・キュー ID のフォーマット

トランザクション・ログの各レコードは、64 の 16 進文字 (32 バイト) で構成されるオリジン・キュー ID で識別されます。オリジン・キュー ID のフォーマットは Replication Agent インスタンスによって決定され、プライマリ・データベースの種類によって異なります。

表 1 : Replication Agent for Oracle のオリジン・キュー ID のフォーマット

文字	バイト	説明
0-3	2	データベース世代 ID
4-15	6	システム変更番号

文字	バイト	説明
16-19	2	システム変更番号の世代 ID
20-23	2	REDO ログ・スレッド
24-43	10	REDO ログ・レコード・ブロック・アドレス
44-55	6	最も古いアクティブなトランザクション開始のシステム変更番号
56-63	4	ロケータ ID

LTL オリジンのコミット時刻の細分性

Oracle では、オリジンのコミット時刻の精度はミリ秒ではありません。

Replication Agent は Oracle の redo ログからオリジンのコミット時刻を取り出します。redo ログのタイムスタンプには、ミリ秒ではなく、秒単位のみで細分性が含まれています。

Replication Server と RSSD のスクリプト

Replication Agent には、Oracle データ型に対応した追加の Replication Server ユーザ定義データ型と DDL コマンドの複写をサポートするための補助スクリプトが用意されています。

以下の Replication Server スクリプトは Replication Agent に付属しており、インストールされている Replication Server のバージョンが 15.0.1 以前の場合に適用する必要があります。

- \$SYBASE/RAX-15_5/scripts/oracle/
hds_oracle_new_setup_for_replicate.sql
- \$SYBASE/RAX-15_5/scripts/oracle/
oracle_create_error_class_1_rs.sql
- \$SYBASE/RAX-15_5/scripts/oracle/
oracle_create_error_class_2_rssd.sql
- \$SYBASE/RAX-15_5/scripts/oracle/
oracle_create_error_class_3_rs.sql

インストールされている Replication Server のバージョンが 15.0.1 以前である場合は、以下の Replication Server スクリプトを RSSD に対して手動で実行します。

- \$SYBASE/RAX-15_5/scripts/oracle/hds_oracle_funcstrings.sql
- \$SYBASE/RAX-15_5/hds_oracle_udds.sql
- \$SYBASE/RAX-15_5/hds_clt_ase_to_oracle.sql

ユーザ定義データ型に関するスクリプト変更の適用

Oracle ユーザ定義データ型を使用するには、ここで説明するスクリプト変更を適用します。

Oracle ユーザ定義データ型を使用するには、次の手順に従います。

1. バージョン 15.0.1 以前の Replication Server を使用している場合は、Oracle レプリケート・データベースへの DDL の複写をサポートする次のスクリプトを適用します。

```
$SYBASE/RAX-15_5/scripts/oracle/  
hds_oracle_new_setup_replicate.sql
```

このスクリプトは、レプリケート・データベースに作成する必要がある Replication Server オブジェクトを定義します。Replication Server のインストール・ディレクトリにある hds_oracle_setup_replicate.sql スクリプトではなく、このスクリプトを使用してください。改訂版のこのスクリプトには、Oracle 間の DDL 複写をサポートするための追加の変更が含まれています。

2. バージョン 15.0.1 以前の Replication Server 向けに Oracle エラー・クラスを正しく定義するには、次の手順に従います。

- 次のスクリプトを Replication Server に適用します。

```
$SYBASE/RAX-15_5/scripts/oracle/  
oracle_create_error_class_1_rs.sql
```

- 次のスクリプトを RSSD に対して適用します。

```
$SYBASE/RAX-15_5/scripts/oracle/  
oracle_create_error_class_2_rssd.sql
```

- 次のスクリプトを Replication Server に適用します。

```
$SYBASE/RAX-15_5/scripts/oracle/  
oracle_create_error_class_3_rs.sql
```

『Replication Server 異機種間複写ガイド』の「Oracle プライマリ・データ・サーバの問題」を参照してください。

Oracle データ型の互換性

Replication Agent for Oracle は、Oracle トランザクションを処理し、データをプライマリ Replication Server に渡します。一方のプライマリ Replication Server は、複写定義に指定されているデータ型のフォーマットを使用して、Replication Agent for Oracle からデータを受信します。

表 2 : 推奨される Oracle データ型マッピング

Oracle データ型	Oracle の長さ / 範囲	Sybase データ型	Sybase の長さ / 範囲	注意
BINARY_DOUBLE	9 バイト、64 ビット単精度浮動小数点数データ型	double	8 バイト	<ul style="list-style-type: none"> 正の最大有限値は 1.79769313486231E+308。 正の最小有限値は 2.22507485850720-308。
BINARY_FLOAT	5 バイト、32 ビット単精度浮動小数点数データ型	rs_oracle_float	精度に応じて 4 バイトまたは 8 バイト	<ul style="list-style-type: none"> 正の最大有限値は 3.40282E+38F。 正の最小有限値は 1.17549E-38F。
BFILE	4GB、ラージ・バイナリ・ファイルへのロケータ・ポイント	image	2GB	
BLOB	4GB、可変長バイナリ・ラージ・オブジェクト	image	2GB	

Oracle データ型	Oracle の長さ / 範囲	Sybase データ型	Sybase の長さ / 範囲	注意
BOOLEAN	1 バイト	rs_oracle_decimal	17 バイト	BOOLEAN データ型は PL/SQL でのみ使用されます。
CHAR	255 バイト	char	32K	
CLOB	4GB、可変長 キャラクタ・ラジ・オブジェクト	image または unitext	2GB	Replication Server バージョン 15.0 以降の場合、CLOB データ型は unitext にマッピングされます。それより前のバージョンの Replication Server の場合、NCLOB データ型は image にマッピングされます。
DATE	8 バイト、固定長	datetime または rs_oracle_datetime	8 バイト	Replication Server は 1753 年 1 月 1 日から 9999 年 12 月 31 日までの日付をサポートしています。 Oracle は紀元前 4712 年 1 月 1 日から紀元後 9999 年 12 月 31 日までの日付をサポートしています。 注意： データ型を変換するには、Replication Server の異機種データ型サポート (HDS) 機能を使用してください。
INTERVAL DAY (n) TO SECOND (n)	可変長	rs_oracle_interval		
INTERVAL YEAR (n) TO MONTH	可変長	rs_oracle_interval		

Oracle データ型	Oracle の長さ／範囲	Sybase データ型	Sybase の長さ／範囲	注意
LONG	2GB、 可変長 文字 データ	text		
LONG RAW	2GB、 可変長 バイナリ・ データ	image		
NCHAR	255 バ イト、 マルチ バイト 文字	unichar または char	32K	
NCLOB	4GB、 可変長 マルチ バイ ト・ キャラ クタ・ ラー ジ・オ ブジェ クト	unitext または text	2GB	Replication Server バージョ ン 15.0 以降の場合、 NCLOB データ型は unitext にマッピング されます。それより前の バージョンの Replication Server の場合、NCLOB データ型は image にマッ ピングされます。

Oracle データ型	Oracle の長さ／範囲	Sybase データ型	Sybase の長さ／範囲	注意
NUMBER (p,s)	21 バイト、可変長数値データ	float、int、real、number、decimal、または rs_oracle_decimal	float は 4 バイトまたは 8 バイト。 int は 4 バイト。 real は 4 バイト。 number および decimal は 2 ～ 17 バイト。	float データ型は、この範囲を超過した場合に浮動小数点表記に変換できません。 整数(int)は、Replication Server の範囲 2,147,483,647 ～ -2,147,483,648 または $1 \times 10^{-130} \sim 9.99 \times 10^{25}$ を超過した場合にトランケートされます。 number および decimal データ型は、 $-10^{38} \sim 10^{38}-1$ の範囲を超過した場合にトランケートされます。 Oracle の精度範囲は 1 ～ 38 桁です。デフォルトの精度は 18 桁です。 Oracle の位取り範囲は -84 ～ 127 です。デフォルトの位取りは 0 です。
NVARCHAR2	2,000 バイト、可変長、マルチバイト文字データ	univarchar または varchar	32K	

Oracle データ型	Oracle の長さ／範囲	Sybase データ型	Sybase の長さ／範囲	注意
RAW	2,000 バイト、可 変長バイ ナリ・ データ	rs_oracle_binary	32K	
ROWID	6 バイ ト、 ロー・ アドレ スを表 すバイ ナリ・ データ	rs_oracle_rowid	32K	
SIMPLE_INTEGER	符号付 き整数 を表す 4 バイ ト	integer		SIMPLE_INTEGER は Oracle 11g で新しく追加さ れたデータ型であり、PL/ SQL でのみ使用されます。 注意： PLS_INTEGER お よび事前定義 PL/SQL の数 値データ型 (SIMPLE_INTEGER を 除く) を使用したプロシー ジャのマーク付けはサ ポートされていません。

Oracle データ型	Oracle の長さ／範囲	Sybase データ型	Sybase の長さ／範囲	注意
TIMESTAMP (n)	21 ～ 31 バイト、可変長	datetime または rs_oracle_timestamp9	8 バイト	<p>Replication Server は 1753 年 1 月 1 日から 9999 年 12 月 31 日までの日付をサポートしています。</p> <p>Oracle は紀元前 4712 年 1 月 1 日から紀元後 4712 年 12 月 31 日までの日付をサポートしています。</p> <hr/> <p>注意： データ型を変換するには、Replication Server の異機種データ型サポート (HDS) 機能を使用してください。</p>
TIMESTAMP (n) WITH [LOCAL] TIME ZONE	可変長	rs_oracle_timestampz		
UDD オブジェクト・タイプ	可変長文字データ	rs_rs_char_raw	32K	
VARCHAR2	2,000 バイト、可変長文字データ	varchar	32K	
XMLTYPE	4GB、可変長キャラクター・オブジェクト	text	2GB	XMLTYPE データは Oracle CLOB データとして暗示的に処理されます。

参照：

- Oracle ユーザ定義データ型 (41 ページ)

Replication Server 15.0 の符号なしデータ型のマッピング

Replication Server バージョン 15.0 以降では、符号なしデータ型がサポートされており、複写定義で指定することができます。

表 3 : 符号なし整数の複写定義データ型のマッピング

RepServer 15.0 符号なしデータ型	複写定義データ型
unsigned bigint	numeric (20)
unsigned int	numeric (10)
unsigned smallint	int
unsigned tinyint	tinyint

Oracle ANYDATA データ型の互換性

Replication Agent は、ANYDATA カラムに格納されたデータの複写をサポートしています。

- プライマリ・データベースとレプリケート・データベースの両方が Oracle データベースでなければならない。
- プライマリ・データベース・テーブルとレプリケート・データベース・テーブルの両方に同じ ANYDATA カラムがなければならない。

pdb_ignore_unsupported_anydata 設定パラメータは、タイプ ANYDATA のカラムに格納されているサポート対象外のデータ型のデータが Replication Agent によってどのように処理されるかを決定します。『Replication Agent リファレンス・マニュアル』を参照してください。

Oracle XMLTYPE データ型の互換性

Replication Agent では、CLOB または XML 型のカラムに格納されている XMLTYPE データを、Oracle 10g または 11g プライマリ・データベースから Oracle 10g または 11g レプリケート・データベースあるいは Adaptive Server Enterprise レプリケート・データベースに複写できます。

Replication Agent ではまた、CLOB または XML として格納されている XMLTYPE テーブル・データを、Oracle 10g または 11g プライマリ・データベースから Oracle 10g または 11g レプリケート・データベースに複写することもできます。

参照：

- Oracle 10g および 11g の XMLTYPE に関する制限事項 (36 ページ)
- XMLTYPE データ・レプリケーション (72 ページ)

Oracle データ型の制限事項

Replication Server および Replication Agent には、Oracle NUMBER データ型に対する制約がいくつかあります。

注意： データ型の制限事項に関する最新情報については、『Replication Agent リリース・ノート』を参照してください。

以下の制約があります。

- 整数表現における制限事項：
 - 整数表現に対応する Sybase の int データ型の方が絶対最大値が小さくなっています。
Oracle の NUMBER 絶対最大値は精度が 38 桁であり、範囲は $9.9 \times 10^{125} \sim 1 \times 10^{-130}$ です。Sybase の int 値の範囲は、 $2^{31} - 1 \sim -2^{31}$ (2,147,483,647 以上、-2,147,483,648 以下) です。
 - Oracle の NUMBER 値が Sybase の int 最大値を上回っている場合、この値は Replication Server によって拒否されます。
- 浮動小数点表現における制限事項：
 - 浮動小数点表現の精度の範囲制限は整数表現と同じです。
 - 浮動小数点値が $2^{31} - 1 \sim -2^{31}$ (2,147,483,647 \sim -2,147,483,648) という Sybase の範囲から外れている場合、Replication Agent for Oracle はその値を指数表記に変換して Replication Server との互換性を保ちます。精度または位取りのロスは発生しません。

Replication Agent は、BINARY_FLOAT および BINARY_DOUBLE データ型の以下の特別値の複写をサポートしていません。

- NaN (数値以外)
- Inf (正の無限値)
- -Inf (負の無限値)

Replication Server および Replication Agent には、Oracle TIMESTAMP WITH [LOCAL] TIME ZONE データ型に対する以下の制約があります。

- TIMESTAMP WITH TIME ZONE データ型が複写される場合は、タイム・ゾーン情報を使用してタイムスタンプ値を「ローカル」のタイム・ゾーンに解決してから、解決された値が複写されます (タイム・ゾーン情報自体は複写されません)。

- たとえば、TIMESTAMP WITH TIME ZONE データ型が Oracle に “01-JAN-05 09:00:00.000000 AM -8:00” として記録され、「ローカル」のタイム・ゾーンが -6:00 である場合は、“01-JAN-05 11:00:00.000000” という値が複写されます。記録のタイムゾーンである -8:00 とローカルのタイム・ゾーンである -6:00 の間の差異に基づいてタイムスタンプ値が調整されてから、調整後の値が複写されません。

Oracle ANYDATA データ型の制限事項

Replication Agent は、ANYDATA カラムに格納されたデータの複写を一部の状況においてサポートしていません。

ANYDATA の複写は以下の場合にサポートされません。

- レプリケート・データベースのテーブル・カラムの型が ANYDATA ではない場合。ANYDATA カラムに格納されているデータを ANYDATA 以外の型のカラムに複写しようとする、Replication Server データ・サーバ・インタフェース (DSI) のスレッドが失敗します。
- ANYDATA カラムに格納されているデータのサイズが、Replication Server opaque データ型の最大サイズである 16K を超えている場合。
- Replication Agent は、ANYDATA 型のカラムに格納された次の Oracle データ型または構造のデータを複製しません。
 - BFILE
 - ネストされたテーブル
 - REF
 - UROWID
 - VARRAY

pdb_ignore_unsupported_anydata 設定パラメータは、タイプ ANYDATA のカラムに格納されているサポート対象外のデータ型のデータが Replication Agent によってどのように処理されるかを決定します。

ANYDATA と文字セット変換

Replication Agent の **rs_charset** 設定パラメータに Replication Server の **RS_charset** 設定パラメータと異なる値が設定されている場合、Replication Server は ANYDATA カラムに格納されている文字データの文字セット変換を実行しません。

複写定義と **create replication definition** 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。

Oracle 提供のデータ型については、Oracle SQL のリファレンス・ガイドの一覧を参照してください。

pdb_ignore_unsupported_anydata については、『Replication Agent リファレンス・マニュアル』を参照してください。

Oracle 10g および 11g の XMLTYPE に関する制限事項

Replication Agent は、CLOB または XML データとして格納されている XMLTYPE カラムおよびテーブルの複写をサポートしていますが、オブジェクト関係 XML 記憶域またはバイナリ XML 記憶域に格納されている XMLTYPE データの複写はサポートしていません。

Replication Agent では、Oracle プライマリ・データベースから Oracle または Adaptive Server Enterprise レプリケート・データベースに XMLTYPE カラムを複写できますが、これ以外のプラットフォームはサポートされていません。Replication Agent が XMLTYPE テーブルの複写をサポートするのは、Oracle プライマリ・データベースから Oracle レプリケート・データベースへの複写のみです。

注意： XMLTYPE データ型がサポートされるのは、複写先が Oracle 10g レプリケート・データベースであり、複写元が Oracle 10g および 11g プライマリ・データベースである場合に限ります。

参照：

- Oracle XMLTYPE データ型の互換性 (33 ページ)
- XMLTYPE データ・レプリケーション (72 ページ)

Oracle ROWID データ型の制限事項

Replication Agent が ROWID データを複写する場合、複写される値はプライマリ・データベースのテーブルに格納されている値を常に表し、レプリケート・データベース内の ROWID 値とは関係しません。ROWID データをレプリケート・データベース内のデータと一致させるための変換や調整は一切試みられません。

Oracle ラージ・オブジェクト (LOB) のサポート

Oracle 内に存在する Oracle LOB データにはいくつかのフォーマットがあります。

Oracle に存在する LOB データ型は次のとおりです。

- 文字：
 - LONG
 - CLOB
 - NCLOB
- バイナリ：
 - LONG RAW
 - BLOB
 - BFILE—Oracle データベース以外の場所に格納されているファイル・コンテンツを参照

データベースに格納される型 (BFILE 以外のすべての型) の場合、Oracle は LOB の内容を REDO ログに記録します。Replication Agent は LOB データを REDO ログから読み取って、複写用に送信します。

BFILE 型のデータはデータベース以外の場所に格納されるため、BFILE の内容は REDO ログには記録されません。BFILE の内容を複写するため、Replication Agent はプライマリ Oracle データベースに接続して、BFILE からデータを選択するためのクエリを発行します。BFILE データを REDO ログの他のデータから分離して選択すると、BFILE の内容が 2 回以上変更された場合に、一時的な非同期状態が生じることがあります。

参照：

- LOB カラムの複写 (37 ページ)

LOB カラムの複写

Oracle は、BFILE データ型を除くすべての LOB データを Oracle REDO ログに記録します。これにより、Replication Agent は LOB の変更を 1 つずつ適用することができます。ただし、BFILE データの場合も、これと同じ手法が用いられ、同じ制限事項が適用されます。つまり、BFILE データはログに記録されませんが、トランザクションの残りの部分が処理されるときにデータベースから読み取られます。

LOB カラムについて複写の有効または無効を指定する方法については、『Replication Agent 管理ガイド』を参照してください。

トランザクションの整合性と LOB データ

トランザクションの複写の際に Replication Agent によって LOB カラムが処理される方法のために、トランザクションの整合性が損なわれる可能性があります。たとえば、2 つのトランザクションが LOB カラムのデータを変更し、Log Reader が 2 番目のトランザクションがコミットされるまで最初のトランザクションを処理しない場合、LOB データがプライマリ・データベースから読み込まれると、そのデータの値は 2 番目のトランザクションの結果となります。この場合、最初のトランザクションの LOB データはレプリケート・データベースにまったく送信されません。Log Reader によって 2 番目のトランザクションが処理されると、プライマリ・データベースとレプリケート・データベースが再び同期されますが、最初のトランザクションを処理してから 2 番目のトランザクションが処理されるまでの間は、レプリケート・データベースには元のトランザクションと一致するデータが含まれません。

この問題が発生するのは、一連のトランザクションによって LOB カラムが複数回変更された場合のみです。この問題が存在する期間は、複写システム全体の速度が遅い場合や、複写システムのコンポーネントに障害が発生した場合は、長くなることがあります。LOB カラムを変更する最後のトランザクションがレプリケート・サイトで処理されると同時に、問題は解消されます。

ラージ・オブジェクトの複写に関する制限事項

Replication Agent は、LOB カラムに対する部分更新の複写をサポートしていません。

たとえば、指定されたオフセットからの LOB データを更新する Oracle **DBMS_LOB.WRITE()** 関数の使用は複写されません。

ロー外ラージ・オブジェクトの特別な処理

ロー外に格納されている LOB の Replication Agent による処理方法について説明します。

Oracle データベース内に格納される LOB 型 (BLOB、CLOB、および NCLOB) は、特定の記憶域特性を使用して定義される場合があります。これらの特性の 1 つである「ロー内記憶域を無効にする」は、LOB に関するデータをその LOB が所属するローの残りのデータとは別に記録する必要があることを指定するものです。このロー外記憶域では、これらの LOB 値に対する更新の複写を特別に処理することが必要となります。

ロー外の LOB 値が更新された場合、REDO ログに記録される変更は、LOB データを格納しているインデックスに対する変更です。つまり、LOB が所属するローは変更されません。その結果、テーブル内のどのローに LOB が所属しているかを識別する情報は REDO ログに記録されません。

たとえば、テーブル内の非 LOB カラムが更新された場合は、変更された値と参照カラムを識別するカラム・データが記録されます。コマンド `updated myTable set col2 = 2 where col1 = 1` を実行すると、“col2” と “col1” の両方の値が REDO ログに記録されます。

一方、**disable storage in row** 句を使用して定義された LOB しか更新しないコマンドは、LOB データに対する変更のみをインデックスに記録し、LOB が格納されているテーブルは記録しません。そのため、コマンド `updated myTable set ClobColumn = 'more data' where col1 = 1` を実行すると、変更された値のみが記録され、“col1” の値は記録されません。

where 句で指定されているカラムの値はその更新でログに記録されないため、レプリケート・サイトにデータを適用する際に使用される、正しい **where** 句を構築するための十分な情報がありません。この問題を解決するため、Replication Agent for Oracle では、**disable storage in row** を使用して定義された LOB カラムを更新するたびに、LOB が所属するテーブル内の同じローに **insert** または **update** をただちに適用することが必要となります。

Replication Agent は、関連するオペレーションからの追加のカラム・データを使用して、複写をサポートするうえで必要となる **where** 句を正しく構築します。

たとえば、以下のトランザクション・シーケンスは、**disable storage in row** 句を使用して定義されている LOB カラム “ClobColumn” に対する更新の複写をサポートしています。

```
begin
insert into myTable (col1, col2, ClobColumn, updated)
values (1,1,empty_clob(), sysdate);
update myTable set ClobColumn = 'more data' where col1 = 1;
commit

begin
update myTable set updated = sysdate() where col1 = 1;
update myTable set ClobColumn = 'more data' where col1 = 1;
commit

begin
update myTable set ClobColumn = 'more data' where col1 = 1;
update myTable set updated = sysdate() where col1 = 1;
commit
```

注意： 複写目的のため、**empty_clob** または **empty_lob** 関数を使用して値が挿入される LOB オブジェクトは NULL 値として複写されます。そのため、LOB カラムの複写定義にはカラム定義の一部として “null” キーワードを含める必要があります。

disable storage in row 句を使用して定義された LOB カラムの場合、以下のトランザクション・シーケンスはサポートされておらず、レプリケート・サイトへの LOB データの提供エラーの原因となります。

- 同じローに対する付随的な変更が欠落している場合：

```
begin
update myTable set ClobColumn = 'more data' where col1 = 1;
commit
```

- 同じローに対する付随的な変更が LOB 変更と隣接していない場合：

```
begin
update myTable set updated = sysdate where col1 = 1;
update myTable set col2 = 5 where col1 = 5;
update myTable set ClobColumn = 'more data' where col1 = 1;
commit
```

この制限事項は、**disable storage in row** 句を使用して定義されている LOB カラムに対してのみ適用されます。

次のクエリを Oracle データベースに対して使用すると、この制約のある LOB カラムをデータベース内で特定できます。

```
select owner, table_name, column_name from dba_lobs
where in_row = 'NO';
```

CLOB および NCLOB データ型の複写

Oracle NCLOB (各国語キャラクタ・ラージ・オブジェクト) は、マルチバイトの各国語文字セットを使用してラージ・キャラクタ・データを格納するデータ型です。

同様に、CLOB データ型も、Oracle データベースがダブルバイト文字セットまたは可変幅文字セットで定義されている場合に、マルチバイトの各国語文字セットを使用して文字データを格納する場合があります。

NCLOB データ型 (および、データベースがダブルバイト文字セットまたは可変幅文字セットで定義されている場合は CLOB) に格納されているマルチバイト文字のバイト・オーダーは、ビッグエンディアン・バイト・オーダーへの複写時にデフォルトで変換されます。これにより、一般的なネットワーク・バイト・オーダーであるビッグエンディアン・オーダーを使用して、データをネットワーク上で送信できるようになります。

NCLOB または CLOB の複写定義に含めるデータ型は `unitext` にする必要があります。こうすることで、そのデータに対する文字セット変換が Replication Server によって試みられるのを防ぐことができます。Replication Server のバージョンが `unitext` をサポートしていない場合は、`image` データ型を使用してください。

この NCLOB または CLOB データを受け取るターゲット・データベースがリトルエンディアン・プラットフォームにインストールされている場合は、複写されたデータのビッグエンディアン・オーダーからリトルエンディアン・オーダーへの変換がデータベースによって自動的に行われられない可能性があります。ビッグエンディアン (ネットワーク・オーダー) からリトルエンディアンへの必要な変換を行わないデータベース・サーバへの NCLOB または CLOB データの複写をサポートするには、`lr_nctxt_byte_order` パラメータを使用して `big` (ビッグエンディアンの場合) または `little` (リトルエンディアンの場合) の値を設定して、Replication Agent によるバイト・オーダーの送信を強制します。

`lr_nctxt_byte_order` パラメータは Microsoft SQL Server と Oracle で使用でき、それぞれ異なるプラットフォームに配置された 2 つのデータベース間で複写を行う際に重要となります。たとえば、Oracle と Microsoft SQL Server の間の複写の場合、プライマリ・データベースはビッグエンディアン・バイト・オーダーでデータを格納しますが、レプリケート・データベースはリトルエンディアン・オーダーでデータを格納します。これは、Microsoft SQL Server が Windows 上でのみ動作するからです。したがって、`lr_nctxt_byte_order` パラメータを `little` に設定して、リトルエンディアン (SQL Server に必要なフォーマット) へのデータ変換を Replication Agent に強制する必要があります。ただし、レプリケート・データベースが Microsoft SQL Server ではない場合は、バイト・オーダーを判断し、それに応じて `lr_nctxt_byte_order` パラメータを設定します。

注意： Replication Agent for Oracle のデフォルトの動作では、`ltl_big_endian_unitext` 設定パラメータの定義に従って、Unicode データが強制的にビッグエンディアン・オーダーになります。また、`lr_nctxt_byte_order` 設定パラメータで Oracle バイト・オーダーを正常に上書きするには、`lr_nctxt_byte_order` パラメータを使用するたびに `ltl_big_endian_unitext` 設定パラメータを `false` に設定する必要があります。

ltl_big_endian_unitext パラメータは、LTL を Replication Server に送信する前に、**unitext** データをリトルエンディアンからビッグエンディアンに変換するかどうかを指定します。有効な値は、**true** および **false** です。このパラメータを設定する場合は、**lr_nxtxt_byte_order** の設定方法を理解している必要があります。

lr_nxtxt_byte_order がレプリケート・データベースの正しいバイト・オーダーを送信するように設定されている場合は、バイト・オーダーが変更されないように

ltl_big_endian_unitext パラメータを **false** に設定してください。

ltl_big_endian_unitext はデフォルトで **true** に設定されています。

ltl_big_endian_unitext 設定パラメータと **lr_nxtxt_byte_order** 設定パラメータには、以下の違いがあります。

- **ltl_big_endian_unitext** が **true** の場合、Replication Agent for Oracle はすべての Unicode データをビッグエンディアン・オーダーで送信します。
- **ltl_big_endian_unitext** が **false** の場合、Replication Agent for Oracle では、データがトランザクション・ログ・ファイルに格納されているときに使用されるバイト・オーダーで Unicode データを送信できます。

lr_nxtxt_byte_order を使用すると、トランザクション・ログから読み込んだ Unicode データの結果は、トランザクション・ログ・ファイル内で通常どのような順序かに関係なく、正しいバイト・オーダーになります。

Oracle ユーザ定義データ型

ユーザ定義型 (UDD) は、アプリケーション内のデータの構造と動作をモデリングするビルディング・ブロックとして、Oracle のビルトイン・データ型とその他のユーザ定義データ型を使用します。

Replication Agent for Oracle は、ユーザ定義オブジェクト・タイプの複写をサポートしています。オブジェクト・タイプとは、発注書など、アプリケーション・プログラムによって処理される現実のエンティティを抽象化したものです。オブジェクト・タイプは、次の 3 種類のコンポーネントから成るスキーマ・オブジェクトです。

- 名前。そのスキーマ内でオブジェクト・タイプを一意に識別します。
- 属性。ビルトイン・タイプとその他のユーザ定義タイプのいずれかです。属性はオブジェクトの構造をモデリングします。
- メソッド。PL/SQL で記述されてデータベースに格納されるか、C や Java などの言語で記述されて外部に格納される関数またはプロシージャです。メソッドは、アプリケーションがオブジェクトに対して実行できるオペレーションを実装します。

Replication Server におけるデータ型定義の作成

ユーザ定義データ型の定義を作成します。

前提条件

Replication Server の管理者権限またはパーミッションが必要です。また、Replication Server 15.1 以前のバージョンを使用している場合は、「Replication Server と RSSD のスクリプト」を先に参照してください。

手順

Oracle でユーザ定義データ型を複製するには、複製定義で指定されているデータ型が `rs_char_raw` でなければなりません。

1. RSSD にログインします。
2. 次の例を参考にして、**rs_datatype** テーブルにローを追加します。

```

/* rs_oracle_udd raw - char with no delimiters */
insert into rs_datatype values(
0, /* prsid */
0x0000000001000008, /* classid */
'rs_oracle_udd', /* name */
0x0000000000010210, /* dtid */
0, /* base_coltype */
255, /* length */
0, /* status */
1, /* length_err_act */
'CHAR', /* mask */
0, /* scale */
0, /* default_len */
'', /* default_val */
0, /*-delim_pre_len-*/
'', /* delim_pre */
0, /*-delim_post_len-*/
'', /* delim_post */
0, /* min_boundary_len */
'', /* min_boundary */
3, /* min_boundary_err_act */
0, /* max_boundary_len */
'', /* max_boundary_err_act */
0 /* rowtype */
)
go

```

3. Replication Server を再起動します。
4. Replication Server で新しいタイプをテストします。

```

admin translate, 'The quick brown fox jumped over the lazy
dog.', 'char(255)', 'rs_oracle_udd'

go

```

Delimiter Prefix	Translated	Value	Delimiter Postfix
-----			-----
NULL	The quick brown fox jumped over the lazy dog.		NULL

この文章が正しく翻訳される場合は、新しいタイプが正しく定義されています。

参照：

- Replication Server と RSSD のスクリプト (25 ページ)

例：複写定義の作成

この例では、Replication Server で定義されている `rs_char_raw` タイプを使用して複写定義を作成する方法を示します。

この例では、以下の Oracle テーブルおよびタイプの定義が使用されています。

- Oracle UDD オブジェクト・タイプ名：NAME_T
- Oracle テーブル名：USE_NAME_T
- Oracle テーブル・カラム：PKEY INT、PNAME NAME_T

```
create replication definition use_name_t_repdef
with primary at ra_source_db.ra_source_ds
with all tables named 'USE_NAME_T'
(
  PKEY int,
  PNAME rs_rs_char_raw
)
primary key (PKEY)
searchable columns (PKEY)
go
```

注意： この例の場合、`lcl_character_case` は `upper` でなければなりません。

オブジェクト・タイプ属性の複写

ユーザ定義オブジェクト・タイプ属性に更新を複写するには、Replication Agent でテーブルレベルのサブリメンタル・ロギングを有効にしておく必要があります。テーブルレベルのサブリメンタル・ロギングは手動で有効にすることができます。

Replication Agent は、ユーザ定義オブジェクト・タイプが含まれているテーブルにマーク付けするときにも、このロギングを有効にするよう試みます。ただし、Replication Agent がこれらのテーブルにマーク付けするには、`pds_username` パラメータによって指定され、テーブルに対する ALTER パーミッションを付与された Oracle ユーザがすでに存在している必要があります。

ユーザ定義オブジェクト・タイプが含まれたテーブルに対してテーブルレベルのサブリメンタル・ロギングが有効にならなく、Replication Agent が Oracle ログ

で更新ログ・レコードを検出した場合は、Replication Agent のステータスが「Replicating」から「Admin」に変更され、次のエラーが返されます。

```
There is insufficient column data in the log to support Oracle UDD
update command processing. Please make sure table-level supplemental
logging is enabled.
```

この場合は、**pdb_skip_op** を使用してこのログ・レコードをスキップします。『Replication Agent リファレンス・マニュアル』を参照してください。

シーケンスのマーク付けとマーク解除

Oracle シーケンスの複製は、複製先が Oracle の場合にのみサポートされます。Oracle 以外のレプリケート・データベースへのシーケンス値の複製はサポートされていません。

Replication Agent は、プライマリ・データベースでのシーケンスの複製をサポートしています。プライマリ・データベースで呼び出されるシーケンスを複製するには、そのシーケンスに複製のマーク付けがされ、そのシーケンスの複製が有効になっている必要があります。テーブルにマークを付けることとテーブルの複製を有効にすることは類似しています。

注意：シーケンスに複製のマークを付けることと、そのシーケンスで複製を有効にすることは別のことです。**pdb_dflt_object_repl** パラメータの値が true である場合、シーケンスにマークが付けられると複製が自動的に有効になります。

シーケンスが増えるごとに Oracle が情報を記録することはありません。

Replication Agent がシーケンスのキャッシュが更新される時に生じるシステム・テーブルの更新を取得すると、シーケンスの複製が発生します。したがって、シーケンスに複製のマークが付けられるときに複製されるシーケンスの値は「次の」シーケンスの値となり、現在のキャッシュの期限が切れたときに使用されません。結果として、シーケンスの個々の増加は複製されませんが、レプリケート・サイトにはプライマリ・サイトで現在使用できるキャッシュ済みの値よりも大きい値が必ずあります。

マーク付けされたシーケンスの複製を無効にすると、そのシーケンスの複製を一時的にサスペンドすることができます。

参照：

- シーケンスの複製の有効化と無効化 (47 ページ)

シーケンスの複写をサポートするための Replication Server に対する変更

Replication Server では、Oracle シーケンス・オブジェクトの複写がデフォルトでサポートされていません。Oracle シーケンスを複写するには、Replication Server とレプリケート Oracle データベースに変更を加えておく必要があります。

Replication Server に対して、シーケンスの複写をサポートするためのストアド・プロシージャを定義する複写定義を作成する必要があります。\$SYBASE/RAX-15_5/scripts/oracle/oracle_create_rs_sequence_repdef.sql スクリプトの値 *{pds}* および *{pdb}* をプライマリ Replication Server コネクションの名前で置換するよう編集を加えた後で、プライマリ Replication Server に対してスクリプトを実行します。これらの値は、Replication Agent の設定プロパティである **rs_source_ds** と **rs_source_db** で確認できます。

注意： 複写定義は、データベース複写定義が存在することを前提としています。データベース複写定義が存在しない場合は、定義を変更する必要性が生じる可能性があります。詳細については、oracle_create_rs_sequence_repdef.sql スクリプトのコメントを参照してください。

レプリケート Oracle データベースでは、シーケンスの複写をサポートするためのストアド・プロシージャを作成する必要があります。レプリケート・データベースとの Replication Server コネクションで定義されているメンテナンス・ユーザとして、レプリケート Oracle データベースにログインします。\$SYBASE/RAX-15_5/scripts/oracle/oracle_create_replicate_sequence_proc.sql スクリプトを実行して、必要なストアド・プロシージャを作成します。

注意： レプリケート・データベースとの Replication Server コネクションで定義されているメンテナンス・ユーザには、Oracle DBMS_SQL パッケージでファンクションを実行するための十分な権限がなければなりません。また、このメンテナンス・ユーザは、複写されたシーケンスを更新するための権限をレプリケート Oracle データベースで持っている必要もあります。

複写対象シーケンスのマーク付け

複写対象シーケンスにマークを付けます。

1. 管理者ログインを使用して Replication Agent インスタンスにログインします。
2. シーケンスがプライマリ・データベースでマーク付けされているかどうかを確認します。

```
pdb_setrepseq pdb_seq
```

`pdb_seq` は、複写のマークを付けるシーケンスの名前です。

- 指定されたシーケンスにマーク付けされた情報を `pdb_setrepseq` が返す場合は、この手順を続ける必要はありません。
 - 指定されたシーケンスにマークが付けられていない情報を `pdb_setrepseq` が返す場合は、この手順を続けてシーケンスに複写のマークを付けます。
3. 複写対象シーケンスにマークを付けます。

`pdb_setrepseq` を使用すると、複写されるプライマリ・シーケンスにマークを付け、レプリケート・データベースで使用する別のシーケンス名を指定できます。

- レプリケート・サイトで増分する必要があるシーケンス名がプライマリ・サイトにおける名前と同じである場合は、次のコマンドを使用して複写対象シーケンスにマークを付けます。

```
pdb_setrepseq pdb_seq, mark
```

注意： 指定した名前と異なる名前のシーケンスを複写するとその他のマーキング・コマンドとの一貫性が保たれますが、これは一般的な設定ではありません。

- 異なるシーケンス名を使用してシーケンスに複写のマークを付けるには、次のコマンドを使用します。

```
pdb_setrepseq pdb_seq, rep_seq, mark
```

`rep_seq` には、レプリケート・データベースで増分する必要があるシーケンスの名前を指定します。

注意： レプリケート・サイトで特定のシーケンスの値を名前が異なるシーケンスに複写する場合、そのレプリケート・サイトのシーケンスにはプライマリ・サイトのシーケンスと同じ属性および開始値があるものと見なされます。

- `pdb_dflt_object_repl` の値が `true` の場合、`pdb_setrepseq` を使用して複写対象マークを付けたシーケンスは、`pdb_setrepseq` の呼び出しが成功した後で複写できるようになります。
 - `pdb_dflt_object_repl` の値が `true` (デフォルト値) の場合は、手順4を省略します。
 - `pdb_dflt_object_repl` の値が `false` の場合に複写を行うには、シーケンスの複写を有効にしておく必要があります。
4. シーケンスの複写を有効にします。

```
pdb_setrepseq pdb_seq, enable
```

シーケンスで複写が有効になると、プライマリ・データベースでのシーケンスの呼び出しの複写を開始できます。

注意： シーケンスを複写するには、`$SYBASE/RAX-15_5/scripts/oracle` ディレクトリの `oracle_create_replicate_sequence_proc.sql` スクリプト

トをレプリケート・サイトで実行して、**rs_update_sequence** という名前のプロシージャを作成することも必要となります。

シーケンスのマーク解除

シーケンスのマークを解除します。

1. 管理者ログインを使用して Replication Agent インスタンスにログインします。
2. シーケンスがプライマリ・データベースでマーク付けされているかどうかを確認します。

```
pdb_setrepseq pdb_seq
```

pdb_seq には、マーク解除するシーケンスの名前を指定します。

- 指定されたシーケンスにマーク付けされた情報を **pdb_setrepseq** が返す場合は、この手順を続けてシーケンスのマークを解除します。
- 指定されたシーケンスにマーク付けされた情報を **pdb_setrepseq** が返さない場合は、この手順を続ける必要はありません。

3. シーケンスの複写を無効にします。

```
pdb_setrepseq pdb_seq, disable
```

4. シーケンスから複写マークを削除します。

```
pdb_setrepseq pdb_seq, unmark
```

強制的にマークを解除するには、次のコマンドを使用します。

```
pdb_setrepseq pdb_seq, unmark, force
```

5. シーケンスに複写対象マークが付けられていないことを確認します。

```
pdb_setrepseq pdb_seq
```

シーケンスの複写の有効化と無効化

シーケンスの複写を一時的にサスペンドするには、**pdb_setrepseq** を使用して、マーク付けされたシーケンスの複写を無効にします。マーク付けされているシーケンスの複写を再開する準備が整ったら、**pdb_setrepseq** コマンドを使用して複写を再度有効にします。

注意： デフォルトでは、どのシーケンスにも複写のマーク付けはされていません。

プライマリ・データベースにあるシーケンスの更新を複写するには、そのシーケンスに複写のマーク付けがされ、そのシーケンスの複写が有効になっている必要があります。

シーケンスに複写のマークを付けることと、そのシーケンスで複写を有効にすることは別のことです。

参照：

- 複写対象シーケンスのマーク付け (45 ページ)

マーク付けされたシーケンスの複写の有効化

マーク付けされたシーケンスの複写を有効にします。

1. 管理者ログインを使用して Replication Agent インスタンスにログインします。
2. シーケンスの複写が有効になっているかどうかを確認します。

```
pdb_setrepseq pdb_seq
```

pdb_seq には、複写を有効にするシーケンスの名前を指定します。

指定されたシーケンスにマーク付けされた情報を **pdb_setrepseq** が返し、複写が無効化されている場合は、この手順を続けてシーケンスの複写を有効にします。

注意：シーケンスの複写を有効または無効にする前に、シーケンスに複写のマークを付けてください。

3. シーケンスの複写を有効にします。

```
pdb_setrepseq pdb_seq, enable
```

シーケンスの複写を有効にすると、そのシーケンスの呼び出しが複写されます。

4. **pdb_setrepseq** を再び使用して、シーケンスの複写が有効になったことを確認します。

```
pdb_setrepseq pdb_seq
```

マーク付けされたシーケンスの複写の無効化

マーク付けされたシーケンスの複写を無効にします。

1. 管理者ログインを使用して Replication Agent インスタンスにログインします。
2. シーケンスの複写が有効になっているかどうかを確認します。

```
pdb_setrepseq pdb_seq
```

pdb_seq には、複写を無効にするシーケンスの名前を指定します。

指定されたシーケンスにマーク付けされた情報を **pdb_setrepseq** が返し、複写が有効化されている場合は、この手順を続けてシーケンスの複写を無効にします。

注意：シーケンスの複写を有効または無効にする前に、シーケンスに複写のマークを付けてください。

3. シーケンスの複写を無効にします。

```
pdb_setrepseq pdb_seq, disable
```

シーケンスの複製を無効にすると、複製を再度有効にするまで、そのシーケンスの呼び出しが複製用に取り出されることはありません。

4. **pdb_setrepseq** を再び使用して、シーケンスの複製が無効になったことを確認します。

```
pdb_setrepseq pdb_seq
```

さまざまなマシン上での Replication Agent と Oracle の設定

Replication Agent とプライマリ・データ・サーバを異なるコンピュータで実行します。

1. プライマリ・データ・サーバが実行されているコンピュータと同じハードウェアおよびオペレーティング・システムのマシンに Replication Agent をインストールします。
2. Replication Agent と同じコンピュータに JDBC ドライバをインストールします。
3. `timezone.dat` ファイルに両方のマシンからアクセスできない場合は、`$ORACLE_HOME/oracle/timezone.dat` ファイルを Replication Agent マシンにコピーします。

注意： Replication Agent が複製元に指定している Oracle サーバの `timezone.dat` ファイルを必ずコピーしてください。

4. Replication Agent の **pdb_timezone_file** 設定パラメータを `timezone.dat` ファイルのフル・パス名に設定します。
5. Oracle アrchive ログを直接トランケートするよう Replication Agent for Oracle が設定されている場合は、Oracle アrchive ログに両方のマシンからアクセスできることを確認します。Replication Agent がログ・ファイルを指すためには、`ra_devicepath` コマンドを使用します。

Real Application Clusters (RAC)

Replication Agent for Oracle は、Oracle 10g および 11g の RAC 環境をサポートしています。Replication Agent for Oracle インスタンスが初期化されると、クラスタによってサポートされるノード数を決定するため、Oracle データベースは問い合わせの対象になります。この情報を基に、Replication Agent はすべてのノードから redo ログ情報を処理するように自動的に設定されます。

Oracle RAC クラスタ内にあるすべてのノードの redo ログ・データを処理するためには、Oracle ノードが各自の redo データの格納に使用する共有領域にアクセスできるロケーションから、Replication Agent を実行する必要があります。

1 つの Oracle インスタンスに接続するように Replication Agent を設定するには、必要なホスト、ポート、Oracle SID の値を **pds_host_name**、**pds_port_number**、

pds_database_name の各設定パラメータに指定します。ただし、Oracle RAC 環境では、ノードで障害が発生したり、他の理由で使用できない場合に備えて、Replication Agent はクラスタ内のすべてのノードに接続できる必要があります。複数ノード・ロケーションを設定するため、Replication Agent は、1つの指定したエントリに対して Oracle tnsnames.ora ファイルから必要な情報を取得して、必要なすべての RAC ノードへの接続をサポートします。これにより、すべてのノードに対して個々のホスト名、ポート名、インスタンス名を設定するのではなく、tnsnames.ora ファイルの場所と使用する TNS コネクションの名前のみが必要になります。

クラスタ内のすべてのノードのアドレスが含まれている tnsnames.ora エントリを Replication Agent で参照することをおすすめします。

たとえば、このエントリが 3 ノード・クラスタの tnsnames.ora ファイルに含まれている場合は、tnsnames.ora ファイルのロケーションを **pds_tns_filename** 設定プロパティに指定し、**RAC10G** を **pds_tns_connection** 設定プロパティの値として指定して、このエントリを使用するよう Replication Agent に指示します。

```
RAC10G =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (LOAD_BALANCE = yes)
      (FAILOVER = ON)
      (ADDRESS = (PROTOCOL = TCP) (HOST = www.xxx.yyy.zz1)
        (PORT = 1521))
      (ADDRESS = (PROTOCOL = TCP) (HOST = www.xxx.yyy.zz2)
        (PORT = 1521))
      (ADDRESS = (PROTOCOL = TCP) (HOST = www.xxx.yyy.zz3)
        (PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = rac10g)
    )
  )
```

tnsnames.ora ファイルには、クラスタ内の各ノードに関する connect 記述子が含まれている必要もあります。

```
NODE1-VIP =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP) (HOST = www.xxx.yyy.zz1)
      (PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = rac10g)
      (INSTANCE_NAME = node1-vip)
    )
  )
NODE2-VIP =
```

```

(DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP) (HOST = www.xxx.yyy.zz2)
  (PORT = 1521))
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = rac10g)
    (INSTANCE_NAME = node2-vip)
  )
)
)
NODE3-VIP =
(DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP) (HOST = www.xxx.yyy.zz3)
  (PORT = 1521))
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = rac10g)
    (INSTANCE_NAME = node3-vip)
  )
)
)

```

tnsnames.ora ファイルに各ノードの connect 記述子が含まれていない場合、Replication Agent はコミット・レコードを正しく生成できません。

pds_tns_filename と **pds_tns_connection** の詳細については、『Replication Agent リファレンス・マニュアル』を参照してください。

注意： Replication Agent に tnsnames.ora ファイルの読み取りアクセス権があることが必要です。

pdb_archive_path

pdb_archive_path 設定パラメータは、アーカイブされた Oracle REDO ログ・ファイルの Replication Agent による検索をどこで行うかを識別します。

Oracle RAC 環境では、Oracle インスタンスごとにアーカイブ・ログの保存先を 1 つまたは複数参照するように設定できます。複写をサポートするには、Oracle RAC クラスタ内のすべてのインスタンスが、アーカイブ・ログ・ファイルのコピーを共有ロケーションに配置する必要があります。Replication Agent for Oracle は、この共有ロケーションを使用して、アーカイブされたすべての REDO ログにアクセスできるようになります。すべての Oracle インスタンスがアーカイブ済みログ・データの書き込み先とするロケーションを参照するよう、**pdb_archive_path** を設定します。Replication Agent は、このディレクトリおよびディレクトリ内のすべてのアーカイブ済み REDO ログへの読み取りアクセス権を持っている必要があります。

注意： アーカイブされた REDO ログは ASM 内に格納することもできます。

pdb_archive_remove 設定パラメータを使用すると、**pdb_archive_path** によって指定されたロケーションからアーカイブ済みログを削除するよう Replication Agent を設

定できます。これにより、複写をサポートするうえで必要となくなったアーカイブ済みログ・ファイルを Replication Agent が削除できるようになります。

pdb_archive_remove が true に設定されている場合、Replication Agent はアーカイブ・ログ・ディレクトリに対する更新権限と個々のアーカイブ・ログ・ファイルに対する削除権限を持っていなければなりません。

注意： **rman_enabled** パラメータを設定すると、Replication Agent が Oracle **RMAN** ユーティリティを使用して古いアーカイブ・ログ・ファイルをトランケートできるようにになります。『Replication Agent リファレンス・マニュアル』を参照してください。

参照：

- Automatic Storage Management (52 ページ)
- 複写対象シーケンスのマーク付け (45 ページ)

Oracle インスタンス・フェールオーバー

Replication Agent が接続している Oracle インスタンスが何らかの理由で失敗した場合、Replication Agent は、`tnsnames.ora` ファイル・エントリで定義されたインスタンスをリストから選択することによって、残りの稼働中のインスタンスとの再接続を試みます。

手動による介入や設定は一切必要となりません。選択可能なインスタンスがない場合、Replication Agent はエラーを報告し、REDO ログ・ファイルの情報にアクセスできる限り処理を継続します。

Automatic Storage Management

Replication Agent for Oracle では、Oracle Automatic Storage Management (ASM) 機能の使用がサポートされています。ASM は、Oracle データベース環境に対してファイル・システム管理とボリューム管理をサポートします。ASM は、Real Application Cluster (RAC) 環境と RAC 以外の環境の両方で使用できます。

アーカイブ・ログの削除と設定

ASM によって管理されているアーカイブ・ログが Replication Agent for Oracle で不要になった場合は、ASM から削除できます。

pdb_archive_remove が true に設定され、アーカイブ・ログが ASM によって管理されている場合は、アーカイブ・ログが格納されている ASM ディスク・グループの名前に **pdb_archive_path** を設定する必要があります。ディスク・グループ名の先頭には、パスが ASM パスであることを示すプラス記号 (+) を付ける必要があります。例：

```
pdb_archive_remove=true  
pdb_archive_path=+DISK_GROUP1
```

ASM によって格納および管理されるアーカイブ・ログは、対応するユニークな Oracle データベース名が所有します。Oracle データベース名がグローバルにユニークなデータベース名と異なる場合は、**pdb_archive_path** を ASM ディスク・グループの名前とアーカイブ・ログが格納されているデータベースのグローバルにユニークなデータベース名の両方に設定する必要があります。

```
pdb_archive_path=+DISK_GROUP1/database_name
```

ASM からのアーカイブ・ログの自動削除に加え、**pdb_truncate_xlog** を使用した手動削除もサポートされています。手動で削除するアーカイブ・ログの **pdb_archive_path** を ASM ディスク・グループ名に設定し、プラス (+) 記号を先頭に付ける必要があります。

注意： **rman_enabled** では、Replication Agent で Oracle **RMAN** ユーティリティを使用して古いアーカイブ・ログ・ファイルをトランケートできます。『Replication Agent リファレンス・マニュアル』を参照してください。

設定パラメータ

ASM でログ・ファイルを管理する場合は、一部の設定パラメータを設定する必要があります。

それらの設定パラメータは次のとおりです。

- **asm_password**
- **asm_tns_connection**
- **asm_tns_filename**
- **asm_username**

asm_username の ASM ユーザ ID には **sysdba** のパーミッションが必要です。Oracle 10g または 11g の場合は、**asm_username** を次のように設定します。

```
asm_username="sys as sysdba"
```

また Oracle 11g では、**asm_username** を次のように設定できます。

```
asm_username="sys as sysasm"
```

『Replication Agent リファレンス・マニュアル』を参照してください。

Replication Serverset autocorrection コマンド

Replication Server の set autocorrection コマンドは、複写テーブルで欠けているか、重複しているローによって発生する障害を回避します。

set autocorrection コマンドは、各 update または insert オペレーションを delete、さらに insert に変換することによってマテリアライゼーション中に発生することのある不一致を修正します。

オートコレクションを設定するには次のようにします。

Replication Agent for Oracle

- Replication Agent からプライマリ・データベース内の 1 つまたはすべてのテーブルについてオートコレクションを設定する場合は、Replication Agent の **ra_set_autocorrection** コマンドを使用します (『Replication Agent リファレンス・マニュアル』を参照)。
- Replication Server から複写定義で **set autocorrection** コマンドを使用します。これは、必ず Replication Server から行ってください。ただし、Replication Agent から複写定義のオートコレクション設定を変更できません。

Replication Agent for Oracle では、ラージ・オブジェクト (LOB)、LONG、LONG RAW、またはユーザ定義データ型のオートコレクション機能の使用はサポートされていません。さらに、次のコマンドを実行するには **pds_username** に **ALTER ANY TABLE** 権限が必要です。

- ALTER TABLE *tablename* ADD SUPPLEMENTAL LOG DATA (ALL) COLUMNS;
- ALTER TABLE *tablename* DROP SUPPLEMENTAL LOG DATA (ALL) COLUMNS;

分割されたテーブル

Replication Agent では Oracle の分割機能がサポートされています。

分割により、テーブル、インデックス、またはインデックス構成テーブルをさらに小さい部分に分割できます。これらの部分はそれぞれ、そのようなデータベース・オブジェクトのパーティションと呼ばれます。各パーティションには独自の名前があり、オプションで独自の記憶域特性を持つこともできます。LONG または LONG RAW データ型のカラムを含むテーブルを除き、どのテーブルも多数の個別のパーティションに分割できます。

データベース内の LOB カラムに格納された非構造化データ (画像やドキュメント) も分割できます。テーブルを分割すると、独自のテーブル領域内に格納できる LOB カラムを除き、すべてのカラムはそのパーティションのテーブル領域内に存在します。Oracle の分割機能の詳細については、『Oracle Database VLDB and Partitioning Guide』 (http://download.oracle.com/docs/cd/B28359_01/server.111/b32024/toc.htm) を参照してください。

truncate partition コマンドの複写

Replication Agent では **truncate partition** コマンドの複写がサポートされています。

truncate partition コマンドを次のいずれかによって複写します。

- **lr_send_trunc_partition_ddl** を使用する。
- ストアド・プロシージャ内で **truncate partition** をラッピングし、プロシージャを複写する。

lr_send_trunc_partition_ddl の使用

Replication Agent の設定パラメータ **lr_send_trunc_partition_ddl** を使用して、**truncate partition** コマンドがレプリケート・データベースに DDL として送信されたか DML として送信されたかを確認します。次のような設定が考えられます。

- true (デフォルト t) – truncate partition コマンドは DDL コマンド (**alter table**) として送信されます。この設定を使用して Oracle に複写します。
- false – truncate partition が DML オペレーションとして送信されます。**truncate partition** コマンドを DML (たとえば Microsoft SQL Server) として扱うデータベースに複写する場合は、この設定を使用します。

Replication Agent の設定プロパティの詳細については、『Replication Agent リファレンス・マニュアル』を参照してください。

truncate partition コマンドのラッピング

ストアド・プロシージャ定義内で **truncate partition** コマンドをラッピングし、プロシージャを複写できます。

たとえば、Oracle プライマリから Adaptive Server Enterprise に **truncate partition** コマンドを複写するには、プライマリ・データベースで次のストアド・プロシージャを作成します。

```
create procedure sp_truncate_partition
as
begin
execute immediate 'ALTER TABLE myTable TRUNCATE PARTITION part1';
end;
```

レプリケート・データベースで対応するストアド・プロシージャを作成します。

```
create proc sp_truncate_partition as
truncate table myTable part1
```

sp_truncate_partition プロシージャを複写するようにマーク付けします。

sp_truncate_partition がプライマリ・データベースで実行されると、**truncate partition** コマンドはレプリケート・データベースに複写されます。

マテリアライズド・ビュー

マテリアライズド・ビューは、保存されたビュー・クエリ結果です。

ビューの定義に使用するデータをマスタ・テーブルと呼びます。マテリアライズド・ビューは独自のテーブルに格納され、マスタ・テーブルに加えた変更に基づいて更新されます。マテリアライズド・ビューはローカルの場合(マスタ・テーブルと同じデータベース上で定義されている場合)も、リモートの場合(マスタ・テーブルとは別のデータベース上で定義されている場合)もあります。

マテリアライズド・ビューは、保存されたビュー・クエリ結果です。ビューの定義に使用するデータをマスタ・テーブルと呼びます。マテリアライズド・ビューは独自のテーブルに格納され、マスタ・テーブルに加えた変更に基づいて更新されます。マテリアライズド・ビューはローカルの場合(マスタ・テーブルと同じデータベース上で定義されている場合)も、リモートの場合(マスタ・テーブルとは別のデータベース上で定義されている場合)もあります。

Oracle は、次のタイプのマテリアライズド・ビューをサポートしています。

- 読み取り専用 – マテリアライズド・ビューの内容が対応するマスタ・テーブルから派生しているため、ビューの内容を変更できません。
- 書き込み可能 – マテリアライズド・ビューの内容を一時的に変更できますが、対応するマスタ・テーブルへの変更に基づきマテリアライズド・ビューを含むテーブルが更新されると上書きされます。
- 更新可能 – マテリアライズド・ビューに適用した更新は、マテリアライズド・ビューの更新時に対応するマスタ・テーブルに書き込まれます。

マテリアライズド・ビューの詳細については、Oracle のマニュアルを参照してください。

複写とマテリアライズド・ビュー

Oracle のマテリアライズド・ビューでは、ベース・クエリの結果セットを格納するための領域を割り当てます。Replication Agent では、マテリアライズド・ビューの定義に使用したデータおよびマテリアライズド・ビュー自体のデータを含むトランザクションを複写できます。

マテリアライズド・ビュー DDL

デフォルトでは、Replication Agent はマテリアライズド・ビューに使用する

CREATE MATERIALIZED VIEW、ALTER MATERIALIZED VIEW、DROP MATERIALIZED

VIEW などの Oracle DDL コマンドを複写しません。マテリアライズド・ビュー

DDL コマンドでは、**pdb_setrepddl** コマンドを使用して別に指定されていない限り、複写は無効です。『Replication Agent リファレンス・マニュアル』の「コマン

ド・リファレンス」の「**pdb_setrepddl**」を参照してください。

プライマリ・データベースとレプリケート・データベースでのマテリアライズド・ビュー

マテリアライズド・ビューがプライマリ・データベースとレプリケート・データベースの両方に存在する場合があります。そのような状況は、**pdb_setrepddl** コマンドによってマテリアライズド・ビュー DDL が複写のために有効になっている場合や、レプリケート・データベースがプライマリ・データベース・ダンプからマテリアライズされている場合に発生します。

マテリアライズド・ビューの定義に使用したマスタ・テーブルがプライマリ・データベースにある場合、Replication Agent はこのマスタ・テーブルを複写します。

レプリケート・データベースにあるマテリアライズド・ビューは、複写済みのマスタ・テーブルの内容に従って更新されます。どのような状況にあっても、Replication Agent によってプライマリ・データベースでマテリアライズド・ビューが格納されていたテーブルが複写されることはなく、そのテーブルの複写は試みないでください。

マテリアライズド・ビューがリモートの場合、つまりマテリアライズド・ビューの定義に使用したマスタ・テーブルがプライマリ・データベースに存在しない場合、レプリケート・データベースのマテリアライズド・ビューは、マスタ・テーブルのあるデータベースを指すようにリダイレクトされる必要があります。レプリケート・データベースをリダイレクトしないと、マテリアライズド・ビューの更新はレプリケート・データベースで失敗します。レプリケート・データベースをリダイレクトする場合、マスタ・テーブルを格納するデータベースへの接続にレプリケート・データベースが使用する Oracle データベースのリンクを再度作成します。

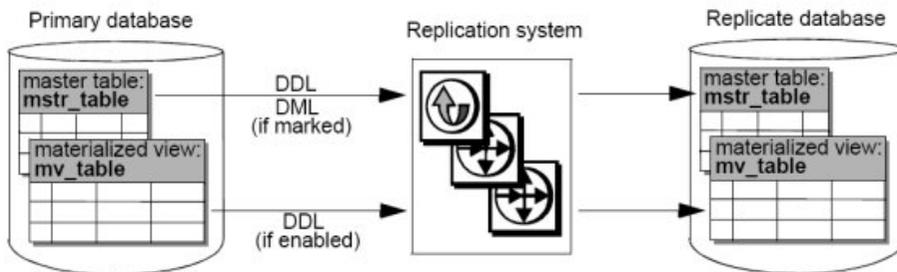
書き込み可能なマテリアライズド・ビューと更新可能なマテリアライズド・ビュー

マスタ・テーブルが複写するようにマーク付けされている場合は、マテリアライズド・ビューを格納するテーブルに変更内容を複写する代わりに、Replication Agent は変更内容をマスタ・テーブルに複写します。したがって、Replication Agent は書き込み可能なマテリアライズド・ビューに加えた変更内容を複写しません。ただし、更新可能なマテリアライズド・ビューに加えた変更内容は、マテリアライズド・ビューの更新時に対応するマスタ・テーブルに書き込まれるため、Replication Agent はプライマリ・データベース上に更新可能なマテリアライズド・ビューに加えた変更を、レプリケート・データベース上の対応するマスタ・テーブルに複写します。レプリケート・データベース上の更新可能なマテリアライズド・ビューに加えた変更は、双方向複写が有効になっていない限り、ローカルのマスタ・テーブルのみに影響します。

マテリアライズド・ビューの複写のシナリオ

この図では、マテリアライズド・ビュー、および対応するマスタ・テーブルはプライマリ・データベースとレプリケート・データベースの両方に存在します。

図 2 : プライマリ・データベース上のマスタ・テーブルとマテリアライズド・ビュー



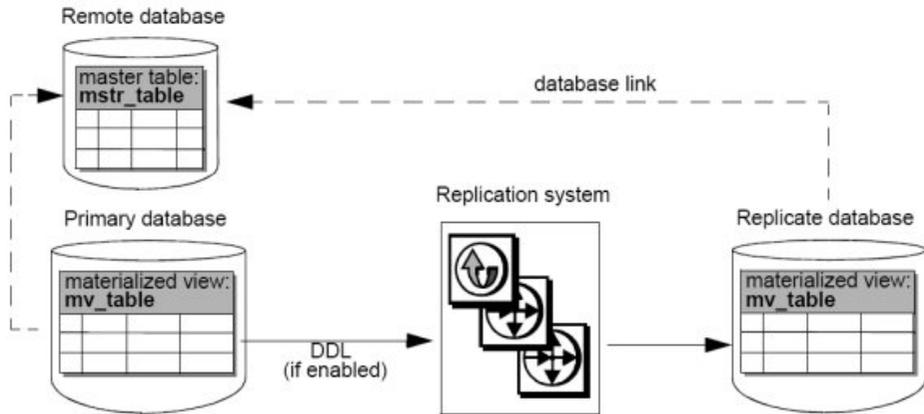
この場合、マスタ・テーブルに影響する DDL コマンドに加え、複製するようにマーク付けされた DML の影響を受けるオブジェクトも複製できます。

マテリアライズド・ビューに影響する DDL コマンドは、**pbid_setrepddl** コマンドによって DDL が有効化されていない限り、複製されません。マテリアライズド・ビューは、レプリケート・データベースにも存在するため、マテリアライズド・ビューの定義に使用したすべてのマスタ・テーブルも複製する必要があります。これを行わないと、レプリケート・データベース上のマテリアライズド・ビューが無効になる場合があります。

プライマリ・データベース上のマテリアライズド・ビューが更新可能な場合、このビューに加えた変更内容は対応するマスタ・テーブルに書き込まれ、マスタ・テーブルが複製するようにマーク付けされている場合は、レプリケート・データベースに複製されます。レプリケート・データベース上のマテリアライズド・ビューが更新可能な場合、このビューに加えた変更内容は対応するレプリケート・データベース上のマスタ・テーブルに書き込まれますが、双方向複製が有効になっていない限りプライマリ・データベース上のマスタ・テーブルは変更されません。

この図では、マテリアライズド・ビューが定義されているプライマリ・データベースは、別の (リモート) データベース上に存在します。

図 3: リモート・データベース上のマスタ・テーブル、プライマリ・データベース上のマテリアライズド・ビュー

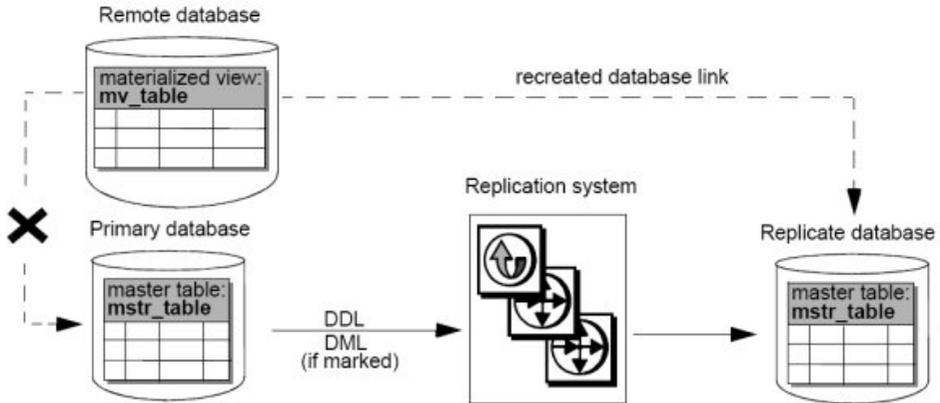


この場合、マスタ・テーブルに影響する DML も DDL も複製されません。マテリアライズド・ビューに影響する DDL コマンドは、**pbdd_setrepddl** コマンドによって DDL が有効化されていない限り、複製されません。マテリアライズド・ビューもレプリケート・データベース上に存在するため、それがマテリアライズド・ビューが定義されたマスタ・テーブルを含むデータベースを指すように、データベース・リンクを作成する必要があります。

プライマリ・データベースおよびレプリケート・データベース上のマテリアライズド・ビューの両方が更新され、リモート・データベースのマスタ・テーブルに正しくリンクされていると、それらのビューのいずれかに加えた変更は、マスタ・テーブルに書き込まれ、更新と同時に両方のマテリアライズド・ビューで変更が反映されます。

下の図では、マテリアライズド・ビューはリモート・データベース上にあり、マテリアライズド・ビューが定義されたマスタ・テーブルはプライマリ・データベース上に存在します。このマスタ・テーブルのコピーは、レプリケート・データベースにも存在し、リモート・データベースとプライマリ・データベースの間のデータベース・リンクはその後切断されます。

図 4: プライマリ・データベース上のマスタ・テーブル、リモート・データベース上のマテリアライズド・ビュー

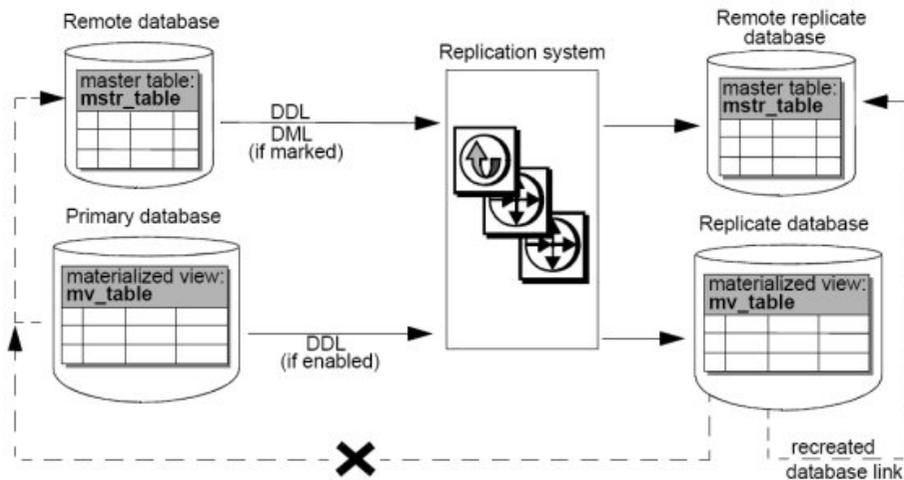


この場合、プライマリ・データベースのマスタ・テーブルに影響する DDL コマンドは、複製するようにマーク付けされた DML コマンドとともに複製できます。マテリアライズド・ビューに影響する DDL コマンドは、レプリケート・データベースに対応するマテリアライズド・ビューがないため、複製できません。リモート・データベースとプライマリ・データベース間のデータベース・リンクが壊れている場合、更新が発生する前にリモート・データベースはレプリケート・データベースへのリンクを作成する必要があります。

リモート・データベース上のマテリアライズド・ビューが更新可能な場合、このビューに加えた変更内容は、リモート・データベースが現在リンクされているデータベース上のマスタ・テーブルに書き込まれます。

下の図では、マスタ・テーブルは 2 つの異なるリモート・データベースに存在し、その 1 つはレプリケート・データベースです。マテリアライズド・ビューはプライマリ・データベースとレプリケート・データベースに存在します。レプリケート・データベースのマテリアライズド・ビューは、最初はリモート・データベース上のマスタ・テーブルによって定義されますが、そのデータベース・リンクは切断されたため、レプリケート・データベースは代わりにリモート・レプリケート・データベースへのリンクを再度作成します。

図 5: リモート・データベース上のマスタ・テーブル、プライマリ・データベースとレプリケート・データベース上のマテリアライズド・ビュー



レプリケート・データベース上のマテリアライズド・ビューが更新可能な場合、このビューに加えた変更内容は、レプリケート・データベースが現在リンクされているデータベース上のマスタ・テーブルに書き込まれます。レプリケート・データベースとリモート・データベース間のデータベース・リンクが壊れる前に、レプリケート・データベース上のマテリアライズド・ビューに適用した更新は、リモート・データベース上のマスタ・テーブルにのみ書き込まれます。レプリケート・データベースとリモート・レプリケート・データベース間にリンクが作成されると、レプリケート・データベース上のマテリアライズド・ビューに適用した更新は、リモート・レプリケート・データベース上のマスタ・テーブルにのみ書き込まれます。

インデックス構成テーブル

Replication Agent では、Oracle インデックス構成テーブル (IOT) の DML を複写できます。

Oracle 10g および 11g データベースの場合、次のタイプのインデックス構成テーブルの DML を複写できます。

- 単純な IOT
- **including** 句と **overflow** 句を含む IOT
- 複合パーティションを含む IOT
- マッピング・テーブルを含む IOT
- インデックス圧縮 IOT

- ローの依存性のある IOT
- ラージ・オブジェクトを含む IOT
- セカンダリ・インデックスを含む IOT

Replication Agent は、ネストされたテーブルと VARRAY 型のカラムを含む IOT は複写できません。

レプリケート・データベースでのトリガ実行の制御

複写システム内のトリガ実行は、トリガの起動の原因となったトランザクションと、トリガの結果発生したトランザクションの両方が複写された場合に、問題発生の原因となることがあります。

監査テーブルで実行された 1 つのオペレーションのデータがトリガによって 2 回記録される場合、データの重複につながることがあります。また、トリガの結果 DML コマンドが 2 回(1 度目はプライマリ・データベースでのトリガの起動、2 度目はトリガによって変更されたデータがすでに複写済みのレプリケート・データベースでのトリガの起動によって)実行される場合、データの不整合にもつながります。データの重複と不整合を防ぐには、複写システムでトリガの実行を制御することが重要です。しかし、Oracle はトリガの実行を無効にするセッション・レベルのコマンドは提供していません。

Replication Server では、セッション・レベルまたはコネクション・レベルでトリガの実行を無効にすることができます。トリガの起動は、PL/SQL コマンドがレプリケート・データベースに対して実行されるたびに制御することができます。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、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データの重複と不整合を排除できます。

Replication Server の `rs_triggers_reset` 関数の詳細については、『Replication Server リファレンス・マニュアル』を参照してください。レプリケート・データベースでのトリガ実行の制御方法の詳細については、『Replication Server 異機種間複写ガイド』の「Oracle レプリケート・データ・サーバの問題」を参照してください。

プライマリ・データ・サーバからの複写定義の変更

プライマリ・データ・サーバから複写定義を変更することができます。

複写定義の変更前に複写システムをクワイース状態にする必要がないように、プライマリ・データ・サーバから Replication Server の `alter replication definition` コマンドを発行し、同時にプライマリ・データベース・オブジェクトへのスキーマ変更を加えることができます。複写定義への変更の伝播は、複写プロセスを停止しなくても、データ複写とともに自動的に調整されます。

Replication Server の `alter replication definition` コマンドをプライマリ・データ・サーバから発行するには、`rs_send_repserver_cmd` というストアド・プロシージャをプ

ライマリ Oracle データベースに作成します。このプロシージャを作成するための SQL は、Replication Server 上の適切なコネクション・プロファイルに含まれていません。コネクション・プロファイルのリストについては、Replication Server の `admin show_connection_profiles` コマンドを使用してください。

`rs_send_repserver_cmd` および `alter replication definition` Replication Server 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。

セキュリティの考慮事項

`rs_send_repserver_cmd` プロシージャをプライマリ・データ・サーバで呼び出すと、Replication Agent は対応する複製コマンド言語 (RCL) を直接 Replication Server に渡します。そのため、`rs_send_repserver_cmd` プロシージャについて誰に実行権限を割り当てるかを慎重に考慮し、環境とセキュリティ・ポリシーに適した権限の割り当てを行う必要があります。

制限事項

特定のデータ型のカラムを含むテーブルの複製定義を変更する場合、`rs_send_repserver_cmd` プロシージャを使用することはできません。

次のタイプがあります。

- BINARY ROWID
- BINARY UROWID
- DATE
- INTERVAL YEAR TO MONTH
- INTERVAL DAY TO SECOND
- MLSLABEL
- RAW
- REF
- TIMESTAMP
- TIMESTAMP WITH LOCAL TZ
- TIMESTAMP WITH TZ

注意： Replication Server でテーブル・レベルの複製定義を手動で変更した場合、Replication Agent で複製をサスペンドしてから開始し、Replication Agent によってキャッシュがクリアされ再表示されることを確認する必要があります。

Oracle Data Guard

Data Guard は障害保護のアーキテクチャで、プライマリ Oracle データベースと、プライマリ・データベースがデータをコピーする 1 つ以上のスタンバイ Oracle データベースで構成されます。これらのスタンバイ・コピーは、プライマリ Oracle データベースで障害が発生した場合に使用できます。Replication Agent は、

Data Guard を使用する Oracle データベース・システムからのデータの複写をサポートしています。

推奨される設定

Replication Agent では、Data Guard のプライマリ・データベースまたは Data Guard スタンバイ・データベースのどちらからもデータを複製できますが、Data Guard スタンバイ・データベースのトランザクション・ログから読み取るように Replication Agent を設定することをおすすめします。このようにすると、プライマリ Data Guard データベースがスタンバイ Data Guard データベースにフェールオーバーした場合でも、Replication Agent は稼働中の Oracle データベースに接続しているため、複写は中断しません。

Data Guard スタンバイ・データベースのトランザクション・ログから読み取るように Replication Agent を設定する場合、**pds_username** には **alter database** パーミッションが必要です。

データベース再同期化

レプリケーション・システムがデータベースの再同期を行うように設定されている場合、Replication Agent for Oracle を初期化する際にプライマリ・データベースのクワイスを回避することができます。

データベースの再同期の設定の詳細については、『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」を参照してください。

Oracle トランザクションおよびオペレーションのトラブルシューティング

ra_dumptran コマンドおよび **ra_helpop** コマンドは、それぞれ指定の Oracle データベース・トランザクションまたはデータベース・オペレーションのトラブルシューティングで使用する情報を返します。

ra_dumptran コマンドおよび **ra_helpop** コマンドでは、Oracle LogMiner が収集した Replication Agent for Oracle のトラブルシューティングに役立つ情報を使用します。Oracle LogMiner は、Oracle redo ログからデータベース・アクティビティに関する詳細情報を取得するための Oracle プロシージャとビューで構成されます。

ra_dumptran および **ra_helpop** を使用するには、Oracle LogMiner をインストールする必要があります。これを行わないとこれらのコマンドでエラーが返されます。

ra_dumptran と ra_helpop を使用するための Replication Agent と Oracle の設定

ra_dumptran と **ra_helpop** を使用したトラブルシューティング

1. \$ORACLE_HOME/rdbms/admin へ移動します。

2. “sys as sysdba” ユーザとしてログインします。
3. Oracle LogMiner のインストール・スクリプトを実行します。

```
@dbmslm.sql
```

4. LogMiner がインストールされたら、次のように入力してパブリック・シノニムを作成します。こうすることで、所有者としてログインしなくても LogMiner 関数を実行できるようになります。

```
CREATE PUBLIC SYNONYM DBMS_LOGMNR FOR
SYS.DBMS_LOGMNR;
```

注意： この手順は Oracle 10g を使用している場合に必要となります。

5. 次の権限を **pds_username** に付与します。

- EXECUTE_CATALOG_ROLE
- SELECT ON V_\$LOGMNR_CONTENTS
- SELECT ON V_\$LOGMNR_LOGS
- SELECT ANY TRANSACTION

注意： **ra_migrate** コマンドは、これらの権限が **pds_username** に付与されたことを確認します。**ra_migrate** を呼び出すときにこれらの権限が付与されていない場合は、警告メッセージが返され、Replication Agent ログ・ファイルに記録されます。

6. **ra_dumptran** および **ra_helpop** は、『Replication Agent リファレンス・マニュアル』に記載された手順に従って使用してください。

BOOLEAN 引数を持つストアド・プロシージャの複写

boolean 型の引数を持つ Oracle ストアド・プロシージャを複写する方法について説明します。

BOOLEAN 型の引数を持つ Oracle ストアド・プロシージャを複写するには、Replication Agent は BOOLEAN 引数を整数として Replication Server に送ります。次に、ストアド・プロシージャをレプリケート・データベースで実行できるように、Replication Server はファンクション文字列を使用してその引数を BOOLEAN 値に変換し直します。Replication Server では、BOOLEAN 型の引数を持つ各 Oracle ストアド・プロシージャに対してこのファンクション文字列を手動で作成する必要があります。

一部のレプリケート・データベースでは、BOOLEAN ストアド・プロシージャ引数はサポートされていません。これらの場合、Oracle ストアド・プロシージャの BOOLEAN 引数を、レプリケート・データベースの対応するストアド・プロシージャ内の整数引数にマップする必要があります。これでファンクション文字列が不要になります。

次の例は、BOOLEAN 型の引数を持つ Oracle ストアド・プロシージャを、Oracle レプリケート・データベースおよび Oracle 以外のレプリケート・データベースに複写する方法を示しています。

例：Oracle レプリケート・データベースへの複写

BOOLEAN 型の引数を持つストアド・プロシージャを Oracle レプリケート・データベースに複写します。

次の PL/SQL 文によって定義されたストアド・プロシージャを複写するには、次のようにします。

```
CREATE PROCEDURE boolproc (a IN BOOLEAN, b INT) AS
BEGIN
    IF (a = true) THEN
        DBMS_OUTPUT.PUT_LINE('True');
    ELSE
        DBMS_OUTPUT.PUT_LINE('False or NULL');
    ENDIF;
END;
```

1. 次の RCL コマンドを使用して Replication Server に複写定義を手動で作成します。

```
create function replication definition ra$xxx_boolproc
with primary at myprimary.pdb
with all functions named boolproc (
@a" rs_oracle_decimal
@b" rs_oracle_decimal )
searchable parameters(@a", @b")
send standby all parameters
```

注意： Replication Agent の `pdb_auto_create_repdefs` 設定パラメータが `true` に設定されている場合、複写定義は自動的に作成されます。

2. 以下の場合、ストアド・プロシージャに複写のマークを付けます。

```
pdb_setrepproc boolproc, mark
```

3. Replication Server に次のようにファンクション文字列を作成します。

```
create function string ra$xxx_boolproc.boolproc
for rs_oracle_function_class
output language
'begin execute immediate "begin
ra_user.boolproc
(?a!param?=1,?b!param?);;end;;";;end;;'
go
```

4. Replication Server に次のように複写定義のサブスクリプションを作成します。

```
create subscription sub_intproc
for ra$xxx_boolproc
with replicate at myreplicate.rdb
go
```

ストアド・プロシージャは、プライマリ・データベースで実行されたときに複写されます。

```
EXECUTE boolproc(true,1);
```

例：Oracle 以外のデータベースへの複写

BOOLEAN 型の引数を持つストアド・プロシージャを Oracle 以外のレプリケート・データベースに複写します。

次の PL/SQL 文によって定義されたストアド・プロシージャを複写するには、次のようにします。

```
CREATE PROCEDURE boolproc (a IN BOOLEAN, b INT) AS
BEGIN
    IF (a = true) THEN
        DBMS_OUTPUT.PUT_LINE('True');
    ELSE
        DBMS_OUTPUT.PUT_LINE('False or NULL');
    ENDIF;
END;
```

1. 次の RCL コマンドを使用して Replication Server に複写定義を手動で作成します。

```
create function replication definition ra$xxx boolproc
with primary at myprimary.pdb
with all functions named boolproc (
@a" rs_oracle_decimal
@b" rs_oracle_decimal )
searchable parameters(@a", @b")
send standby all parameters
```

注意： Replication Agent の `pdb_auto_create_repdefs` 設定パラメータが `true` に設定されている場合、複写定義は自動的に作成されます。

2. 以下の場合は、ストアド・プロシージャに複写のマークを付けます。

```
pdb_setrepproc boolproc, mark
```

3. Adaptive Server Enterprise では BOOLEAN ストアド・プロシージャ引数がサポートされていないため、Oracle ストアド・プロシージャの BOOLEAN 引数を、レプリケート・データベースの対応するストアド・プロシージャの整数引数にマップする必要があります。
4. レプリケート・データベースで次の Transact-SQL® 文によって定義されるストアド・プロシージャを作成します。

```
create proc boolproc (@a int, @b int) as
if @a = 1
    print 'True'
else
    print 'False or NULL'
go
```

5. Replication Server に次のように複製定義のサブスクリプションを作成します。

```
create subscription sub_intproc
    for ra$xxx_boolproc
    with replicate at myreplicate.rdb
go
```

ストアド・プロシージャは、プライマリ・データベースで実行されたときに複製されます。

```
EXECUTE boolproc(true,1);
```

ただし、レプリケート Adaptive Server Enterprise の **boolproc** プロシージャは、BOOLEAN 引数ではなく整数値によって呼び出されます。

```
boolproc 1, 1
go
```

Oracle ウォーム・スタンバイ

Replication Server を使用すると、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。

スタンバイ・モードでは、Replication Agent は次のように動作します。

- トランザクション・ログをスキャンし、Replication Agent システム・データベース (RASD) を最新の状態に維持します。
- LTL を Replication Server に送信しません。
- ログ・トランケーションの実行を継続します。

アクティブ・データベースで障害が発生した場合またはアクティブ・データベースでメンテナンスを実行する場合、スタンバイ・データベースに切り替えることができます。アクティブ・データベースとスタンバイ・データベースを切り替える手順の詳細については、『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。

ウォーム・スタンバイ・モードで機能するための条件は次のとおりです。

- Replication Agent がプライマリ側とスタンバイ側の両方にインストールされ、正常に初期化されている必要があります。スタンバイ側の Replication Agent は、**ra_standby** パラメータを true に設定したスタンバイ・モードで実行されている必要があります。
- Replication Agent で **rs_source_ds** パラメータと **rs_source_db** パラメータが Replication Server への物理コネクションとして設定されている必要があります。
- Replication Agent で **pdb_setrep_ddl** コマンドを使用して、DDL 文の複製が必要に応じて有効または無効にされている必要があります。

- Replication Agent で **pdb_auto_create_repdefs**、**pdb_dflt_column_repl**、**pdb_dflt_object_repl**、および **pdb_automark_tables** の各パラメータが true に設定されている必要があります。

これらのコマンドと設定パラメータの使用の詳細については、『Replication Agent リファレンス・マニュアル』を参照してください。Oracle のウォーム・スタンバイの作成と管理に必要な手順の詳細については、『Replication Agent 異機種間複写ガイド』を参照してください。

Oracle Flashback

Replication Agent は、テーブル・レベルおよびトランザクション・レベルで実行した Oracle Flashback オペレーションを複写できます。

- Replication Agent は、Flashback Table コマンドを複写して System Change Number (SCN)、タイムスタンプ、または Oracle **UNDO_RETENTION** パラメータで指定されたしきい値内のリストア・ポイントまでデータベースをリストアできます。テーブルが複写するようにマーク付けされている場合、Replication Agent は Oracle **FLASHBACK_TRANSACTION_QUERY** ビューの **UNDO_SQL** カラムの内容の実行から発生したどのような DML 変更も複写できます。
- Replication Agent は、次の Oracle DDL コマンドを複写します。
 - **DROP TABLE** *table* (Recycle Bin が有効な場合)
 - **FLASHBACK TABLE** *table* **TO BEFORE DROP**
 - **PURGE TABLE** *table*
 - **PURGE INDEX** *index*
 - **PURGE TABLESPACE** *tablespace*
 - **PURGE RECYCLEBIN**
 - **PURGE DBA_RECYCLEBIN**

注意：レプリケート・データベースはプライマリ・データベースの正確なコピーではない可能性があるため、これらの DDL コマンドがレプリケート・データベースで正しく実行されないことも、レプリケート・データベースで異なる結果になることもあります。たとえば、レプリケート・データベースの Recycle Bin がプライマリ・データベースの Recycle Bin よりも多くのオブジェクトを格納する場合、**PURGE DBA_RECYCLEBIN** コマンドはレプリケート・データベースでより多くのオブジェクトをパージします。

- Replication Agent は、**pdb_archive_path** 設定パラメータが Oracle Flash Recovery Area に設定されている場合、その場所から読み取ることができます。

Oracle Flashback の要件

Replication Agent で Oracle Flashback を使用する場合は、以下の要件を確認します。

Replication Agent で Oracle Flashback を使用するには、次の条件があります。

- **pds_username** のユーザ ID には **SYS.RECYCLEBIN\$** に対する **select** パーミッションが必要です。
- Replication Agent で Oracle **PURGE DBA_RECYCLEBIN** コマンドを複製するには、**ddl_username** のユーザ ID に **sysdba** パーミッションが必要で、"as sysdba" というサフィックスを付ける必要があります。例：

```
ra_config ddl_username, "myuser as sysdba"
go
```
- Replication Agent は、Oracle 10g and 11g データベースからは Flashback オペレーションを複製しますが、前のバージョンの Oracle からは複製しません。

Oracle Flashback の制限事項

Oracle Flashback コマンドの複製にはいくつかの制限事項があります。

- レプリケート・データベースで Recycle Bin が有効になっていない場合、**DROP TABLE** コマンドが正常に複製された場合でも、Flashback コマンド、**PURGE** コマンド、および Recycle Bin オブジェクトにアクセスするコマンドはすべて失敗します。
- Replicate Agent は異なるタイプのプライマリ・データベースとレプリケート・データベース間での DDL コマンドの変換をサポートしていません。そのため、Oracle プライマリ・データベースから Oracle 以外のレプリケート・データベースへの複製では DDL 複製を無効にする必要があります。そしてこの場合、Flashback DDL コマンドも複製できません。
- Oracle **FLASHBACK TABLE** コマンドが **RENAME TO** 句付きで発行された場合、Replication Agent では複製定義を新しいテーブル名で自動的に更新しません。次の処理を手動で実行する必要があります。
- Replication Agent は、オブジェクトの Recycle Bin 名ではなく、元のオブジェクト名に基づき Flashback コマンドを再構築します。Recycle Bin にオブジェクトの複数のバージョンがある場合、Replication Agent はレプリケート・データベースの Recycle Bin に存在するオブジェクトの最新バージョンを使用するように Flashback コマンドを再構築します。その結果、Recycle Bin に影響するその後の Oracle コマンドは、プライマリ・データベースとレプリケート・データベース間の不整合の原因となる可能性があります。
たとえば、プライマリ Oracle データベースにはテーブル TAB1 の次の削除されたバージョンが含まれます。

```
SQL> SELECT object_name as recycle_name, original_name,
          FROM recyclebin;
```

RECYCLE_NAME	ORIGINAL_NAME	TYPE
BIN\$zyxwvutsrqponmlkjihgfedcba\$1	TAB1	TABLE
BIN\$zyxwvutsrqponmlkjihgfedcba\$2	TAB1	TABLE
BIN\$zyxwvutsrqponmlkjihgfedcba\$3	TAB1	TABLE

レプリケート Oracle データベースには、テーブル TAB1 の次の削除されたバージョンが含まれます。

```
SQL> SELECT object_name as recycle_name, original_name, type
        FROM recyclebin;
```

RECYCLE_NAME	ORIGINAL_NAME	TYPE
BIN\$abcdefghijklmnopqrstuvwxyz\$1	TAB1	TABLE
BIN\$abcdefghijklmnopqrstuvwxyz\$2	TAB1	TABLE
BIN\$abcdefghijklmnopqrstuvwxyz\$3	TAB1	TABLE

Flashback コマンドはプライマリ Oracle データベースで次のように実行されます。

```
FLASHBACK TABLE "BIN$zyxwvutsrqponmlkjihgfedcba$2" TO BEFORE
DROP;
```

Replication Agent は元のオブジェクト名に基づき Flashback コマンドを再構築し、削除されたオブジェクトの最新バージョンを Flashback コマンドで使用するため、次のコマンドがレプリケート Oracle データベースで実行されます。

```
FLASHBACK TABLE "BIN$abcdefghijklmnopqrstuvwxyz$3" TO BEFORE
DROP;
```

BIN\$zyxwvutsrqponmlkjihgfedcba\$2 の内容が BIN \$abcdefghijklmnopqrstuvwxyz\$3 と異なる場合、プライマリ・データベースとレプリケート・データベースが不整合になっています。

Oracle Recycle Bin の無効化

レプリケート Oracle データベースで Recycle Bin を使用しない場合は、それを手動で無効化できます。

前提条件

Recycle Bin を無効化するには **sysdba** 権限が必要です。

手順

次のコマンドを入力してから、レプリケート Oracle データベースを起動します。

```
ALTER SYSTEM SET RECYCLEBIN=OFF SCOPE=SPFILE;
```

また、RECYCLEBIN パラメータのない Oracle のバージョンを使用している場合は、次のように入力します。

```
ALTER SYSTEM SET "_recyclebin"=FALSE SCOPE=BOTH;
```

注意： Oracle RAC を使用している場合は、クラスタ内のインスタンスごとに Recycle Bin を無効にします。

削除されたオブジェクトとアーティクルのステータス

プライマリ・データベースで Recycle Bin が有効なときにマーク付けしたテーブルが削除されると、**ra_helparticle** コマンドは対応するアーティクルのステータスを

依然として Current として報告します。**ra_helpparticle** は、削除されたテーブルがブライマリ・データベースの Recycle Bin からパージされたあとにのみ、対応するアートのステータスを Dropped として報告します。

Recycle Bin が無効な状態での Flashback 複写の無効化

Recycle Bin が無効なときに Oracle Flashback DDL コマンドの複写を無効にします。

レプリケート Oracle データベースで Recycle Bin が有効になっていない場合、レプリケート・データベースで Recycle Bin オブジェクトにアクセスするコマンドはすべて失敗します。そのため、Oracle Flashback DDL コマンドの複写を無効にする必要があります。

Flashback DDL コマンドの複写は次のいずれかの方法で無効にします。

- **pdb_setrepddl** コマンドを使用して Flashback DDL コマンドが複写されないようにします。『Replication Agent リファレンス・マニュアル』を参照してください。
- レプリケート Recycle Bin が無効な状態で複写を続行できるように、Flashback DDL 実行失敗の “warning” エラー・アクションを Replication Server に追加します。RSSD に対して次の Replication Server スクリプトを実行します。

```
$SYBASE/RAX-15_5/scripts/oracle/  
hds_oracle_setup_flashback_errors.sql
```

このスクリプトの実行後、Replication Server を再起動する必要があります。

注意：このスクリプトは、テンプレートとして **rs_oracle_error_class** デフォルト・エラー・クラスを使用します。カスタム・エラー・クラスを使用していて、Replication Server が中断せずに複写を続行するには、Replication Server がエラー・ログに警告メッセージ 38305 と 38307 を表示するように指定する必要があります。

```
assign action warn for your_error_class to 38305, 38307
```

your_error_class は、カスタム・エラー・クラスの名前です。

Flashback DDL 実行失敗の “warning” エラー・アクションを Replication Server から削除するには、RSSD に対して Replication Server スクリプトを実行します。

```
$SYBASE/RAX-15_5/scripts/oracle/  
hds_oracle_remove_flashback_errors.sql
```

XMLTYPE データ・レプリケーション

XMLTYPE カラムは含めるが XML スキーマの指定なしで Oracle テーブルを作成すると、非表示の CLOB カラムが自動的に作成され XML データが保存されます。

XMLTYPE カラムは非表示の CLOB カラムの仮想カラムになります。対応する

Oracle ベース・テーブルでは、非表示のカラムはそれが表す XMLTYPE カラムのすぐ後に配置され、SYS_NC##### という名前が付けられます。##### は、ベース・テーブル内の非表示のカラムの位置を表します。

たとえば、次の DDL コマンドにより Oracle データベースにテーブルが作成されま

```
CREATE TABLE sampletable
( col1 INT,
, col2 INT,
, xml1 XMLTYPE
, xml2 XMLTYPE);
```

Oracle データベースは、SYS_NC00004\$ および SYS_NC00006\$ という名前の非表示のカラムを作成します。これらはそれぞれ xml1 カラムと xml2 カラムに対応します。非表示の CLOB カラムには直接アクセスできません。ただし、Oracle データ辞書の col\$ ベース・テーブルと obj\$ ベース・テーブルをクエリすることによってこれらのカラムを表示することはできます。Oracle のマニュアルを参照してください。

参照：

- Oracle XMLTYPE データ型の互換性 (33 ページ)
- Oracle 10g および 11g の XMLTYPE に関する制限事項 (36 ページ)

例：Oracle から Oracle への XMLTYPE カラム・データの複写

Oracle プライマリ・データベースから Oracle レプリケート・データベースに XMLTYPE カラムのデータを複写します。

DDL 文で定義されたテーブルを複写するには、次のようにします。

```
CREATE TABLE sampletable
( col1 INT,
col2 INT,
xml1 XMLTYPE,
xml2 XMLTYPE);
```

1. 次の RCL コマンドを使用して Replication Server に複写定義を手動で作成します。

```
create replication definition ra$xxx_sampletable
with primary at myprimary.pdb
with all tables named sampletable (
col1 int,
col2 int,
xml1 as SYS_NC00004$ text,
xml2 as SYS_NC00006$ text )
primary key (col1, col2)
go
```

注意： Replication Agent の `pdb_auto_create_repdefs` 設定パラメータが `true` に設定されている場合、複写定義は自動的に作成されます。

2. テーブルを複写するようマーク付けします。

```
pdb_setreptable sampletable, mark
```

3. 非表示の CLOB カラムがあるため、`pdb_setrepcol` を使用してテーブルの複写を有効にする必要があります。

```
pdb_setrepcol sampletable, enable
```

4. レプリケート Oracle データベースで対応するテーブルを作成します。

```
CREATE TABLE sampletable
( col1 INT,
  col2 INT,
  xml1 XMLTYPE,
  xml2 XMLTYPE);
```

注意： DDL のレプリケーションが有効になっている場合は、手動でレプリケート・テーブルを作成します。

例：Oracle から Adaptive Server Enterprise への XMLTYPE カラム・データの複写

Oracle プライマリ・データベースから Adaptive Server Enterprise レプリケート・データベースに XMLTYPE カラムのデータを複写します。

DDL 文で定義されたテーブルを複写するには、次のようにします。

```
CREATE TABLE sampletable
( col1 INT,
  col2 INT,
  xml1 XMLTYPE,
  xml2 XMLTYPE);
```

1. 次の RCL コマンドを使用して Replication Server に複写定義を手動で作成します。

```
create replication definition ra$xxx_sampletable
with primary at myprimary.pdb
with all tables named sampletable (
col1 int,
col2 int,
xml1 as SYS_NC00004$ text,
xml2 as SYS_NC00006$ text )
primary key (col1, col2)
go
```

注意： Replication Agent の `pdb_auto_create_repdefs` 設定パラメータが `true` に設定されている場合、複写定義は自動的に作成されます。

2. テーブルを複写するようマーク付けします。

```
pdb_setreptable sampletable, mark
```

3. 非表示の CLOB カラムがあるため、**pdb_setrepcol** を使用してテーブルの複写を有効にする必要があります。

```
pdb_setrepcol sampletable, enable
```

4. プライマリ・データベース・テーブルから非表示のカラム名を使用して、レプリケート Adaptive Server Enterprise データベースに対応するテーブルを作成します。

```
create table sampletable
( coll int,
  col2 int,
  SYS_NC00004$ text,
  SYS_NC00006$ text)
go
```

注意： Oracle プライマリ・データベース・テーブルの XMLTYPE カラムは、Adaptive Server Enterprise レプリケート・データベース・テーブルの text カラムにマップされます。

例：Oracle から Oracle への XMLTYPE テーブルの複写

Oracle プライマリ・データベースから Oracle レプリケート・データベースに XMLTYPE テーブルを複写します。

この文では、1つの暗黙的な CLOB カラムを持ち、XMLDATA というデフォルトの疑似カラムからアクセスできる、単純な XMLTYPE テーブルが作成されます。

```
CREATE TABLE sampletable OF XMLTYPE;
```

DDL 文で定義されたテーブルを複写するには、次のようにします。

1. この RCL コマンドと非表示のカラム SYS_NC_OID\$ を使用して、手動で Replication Server に複写定義を作成します。これには sampletable のオブジェクト ID が含まれています。

```
create replication definition ra$xxx_sampletable
with primary at myprimary.pdb
with all tables named sampletable (
SYS_NC_OID$ rs_oracle_binary,
XMLDATA text )
primary key (SYS_NC_OID$)
go
```

Oracle RAW データ型の Replication Server の名前は、rs_oracle_binary です。

注意： Replication Agent の pdb_auto_create_repdefs 設定パラメータが true に設定されている場合、複写定義は自動的に作成されます。

2. テーブルを複写するようマーク付けします。

```
pdb_setreptable samplatable, mark
```

3. レプリケート Oracle データベースで対応するテーブルを作成します。

```
CREATE TABLE samplatable OF XMLTYPE;
```

Oracle 11g Release 2

Oracle Database 11g Release 1 について Replication Agent でサポートされている機能はすべて、Replication Agent for Oracle Database 11g Release 2 でもサポートされています。また、Oracle Database 11g Release 2 で導入された機能の一部もサポートされています。

Replication Agent では、Oracle Database 11g Release 1 および Oracle Database 11g Release 2 に共通の次の機能をサポートしています。

- Automatic Storage Management (ASM) および Real Application Clusters (RAC) を使用する、または使用しないシステムでの Oracle DDL および DML 複写
- Oracle Recovery Manager (RMAN) ユーティリティを使用した古いアーカイブ・ログ・ファイルのトランケート
- Oracle Recycle Bin の使用と Oracle Flashback オペレーションの複写
- Oracle Data Guard

また、Replication Agent は、Oracle Database 11g Release 2 から導入された新機能の一部もサポートしています。

- 型の依存性を持つ型に対する **FORCE** オプションと **CREATE OR REPLACE TYPE** 文の使用
- Flashback Data Archive について有効にされたテーブルの DDL 文
- バージョン 11.2 のタイム・ゾーン・ファイルおよび新規タイム・ゾーン動作

ネットワーク設定ファイルの場所と構造

tnsnames.ora ファイルは ORACLE_HOME¥network¥admin にあります。ASM または RAC を実行中の Oracle Database 11g Release 2 インスタンスでは、tnsnames.ora ファイルは、デフォルトで Grid_home¥network¥admin のグリッド・インフラストラクチャ・ホーム・ディレクトリから読み込まれます。

ASM または RAC を実行中の Oracle Database 11g Release 2 インスタンスを使用する場合、Replication Agent の **asm_tns_filename** パラメータを Grid_home¥network¥admin¥tnsnames.ora に設定します。

デフォルトでは、Grid_home¥network¥admin にある tnsnames.ora ファイルには、DESCRIPTION フィールドおよび SERVICE_NAME フィールドの情報が欠けた不完全な ASM エントリが含まれています。ASM を実行中の Oracle Database 11g Release 2 インスタンスを使用する場合、Replication Agent の **asm_tns_connection** パラメータをこの不完全な ASM エントリで指定された ASM コネクション名に設定

します。Replication Agent によって DESCRIPTION フィールドおよび SERVICE_NAME フィールドの処理が完了すると、Grid_home¥network¥admin にある tnsnames.ora ファイルを使用して、ASM インスタンス・サーバに接続できます。

タイム・ゾーン・ファイル

デフォルトでは、Oracle Database 11g Release 2 は、大容量のタイム・ゾーン・ファイル timezone_11.dat を使用します。このファイルには、データベースで定義されたすべてのタイム・ゾーンが含まれています。

Oracle Database 11g Release 2 を使用する場合、Replication Agent の **pdb_timezone_file** パラメータを timezone_11.dat ファイルの場所に設定します。

```
ra_config pdb_timezone_file, $ORACLE_HOME/oracore/zoneinfo/
timezone_11.dat
```

ユーザ定義型の依存性

既存のユーザ定義型の定義を変更するには、**CREATE OR REPLACE TYPE** コマンドを使用します。ただし、このコマンドを実行すると、参照された型がテーブルまたは型の依存性を持つ場合、エラーが返されます。

Oracle Database 11g Release 2 では、**FORCE** および **CREATE OR REPLACE TYPE** コマンドを使用して、型依存性を持つ型を置換できます。

```
CREATE TYPE mytype1 AS OBJECT (a number) NOT FINAL;
CREATE TYPE mytype2 UNDER mytype1 (b varchar(10));
CREATE OR REPLACE TYPE mytype1 FORCE AS OBJECT (c varchar(20));
```

Oracle Database 11g Release 2 では、**FORCE** および **CREATE OR REPLACE TYPE** コマンドを使用して、テーブル依存性を持つ型を置換できません。

```
CREATE TABLE mytable1 (colA mytype1);
CREATE OR REPLACE TYPE mytype1 FORCE AS OBJECT (d number);
```

最後のコマンドを実行すると、mytype1 が mytable1 に対するテーブル依存性を持つため、次のようなエラーが発生します。

```
ERROR at line 1:
ORA-22866: cannot replace a type with table dependents
```

Replication Agent では、Oracle Database 11g Release 2 の **FORCE** オプションと **CREATE OR REPLACE TYPE** コマンドを使用して型依存性を持つ型を置換できますが、テーブル依存性を持つ型は置換できません。

DDL コマンドの Flashback Data Archive のサポート

Oracle Flashback を使用すると、データベース管理者およびユーザは、データベース・オブジェクトの過去の状態を表示して、データベース・オブジェクトをポイ

ントインタイム・リカバリを使用せずに以前の状態にリストアできます。ユーザは、過去のデータを問い合わせ、メタデータを問い合わせさせて変更の履歴の詳細を作成し、データを以前の特定の時点に戻して、データベースがオンライン中にトランザクションをロールバックできます。

Replication Agent は、Oracle Database 11g Release 2 の Flashback Data Archive で追跡されるテーブルへの DDL コマンドの複写をサポートしています。このような DDL コマンドには、次のものがあります。

- **Add、Drop、Rename、Modify Column**
- **Drop、Truncate Partition**
- **Rename、Truncate Table**
- **Add、Drop、Rename、Modify Constraint**

Oracle 9i

Replication Agent は、Oracle 10g または 11g が 9i 互換モードで実行されている場合、Oracle 9i をサポートします。

Oracle 9i の制限事項

Oracle 9i では一部の Replication Agent 機能を使用できません。

これらの Replication Agent 機能は Oracle 9i と一緒に使用できません。

- Oracle インデックス構成テーブルと ANYDATA データ型
- オートコレクション
- XMLTYPE データのレプリケーション
- Flashback
- RAC
- ASM

Oracle プライマリ・データベースでの Replication Agent オブジェクト

Replication Agent は、複写作業用のオブジェクトをプライマリ・データベースで作成します。

Replication Agent オブジェクトを作成するには、init キーワードを指定して ra_admin コマンドを呼び出します。このコマンドを呼び出すと、Replication Agent ではプライマリ・データベースで作成または変更されたオブジェクトの SQL 文を含む SQL スクリプトが生成されます。このスクリプトは、RAX-15_5¥inst_name ¥scripts¥xlog ディレクトリの partinit.sql ファイルに保存されます。これらの

オブジェクトは、複製のマークをプライマリ・データベース・オブジェクトに付ける前に作成する必要があります。

注意： 生成されたスクリプトは、参照時に使われるだけです。これらのスクリプトを手動で実行しても、プライマリ・データベースまたは Replication Agent を初期化することはできません。これは **pdb_setreproc** を使用したときに、生成されるスクリプトにマークを付ける、またはマークを解除する手順にも当てはまります。**pdb_setreptable** でテーブルにマークを付ける、またはマークを解除すると、スクリプトは生成されなくなります。

『Replication Agent 管理ガイド』を参照してください。

Replication Agent オブジェクト名

Replication Agent は、複製作業用のオブジェクトをプライマリ・データベースで作成します。

Replication Agent のデータベース・オブジェクト名には、次の2つの変数があります。

- **prefix** – **ra_admin_instance_prefix** パラメータ (Replication Agent for Microsoft SQL Server または Replication Agent for UDB に使用) または **ra_admin_prefix** (Replication Agent for Oracle に使用) パラメータの 1～3 の文字列値を表します。デフォルトは **ra_** です。
- **xxx** – 英数字のカウンタを表します。この文字列は、データベース内でユニークな名前にするためにデータベース・オブジェクト名に追加されます。

ra_admin_instance_prefix パラメータの値は、すべての Replication Agent システム・オブジェクト名で使用されるプレフィクス文字列です。

ra_admin_prefix_chars パラメータの値は、**ra_admin_instance_prefix** で指定するプレフィクス文字列で使用できる非英数字のリストです。使用できる文字のリストはデータベース固有です。たとえば、Oracle では、データベース・オブジェクト名に使用できる非英数字は \$、#、_ のみです。

プライマリ・データベース内の Replication Agent トランザクション・ログ・コンポーネントの名前を表示するには、**ra_admin** コマンドを使用します。

オブジェクト名を設定する方法の詳細については、『Replication Agent 管理ガイド』を参照してください。

作成されたオブジェクトの名前を見つける

プライマリ Oracle データベースに作成された Replication Agent オブジェクトの名前を見つけてみます。

Replication Agent 管理ポートで、キーワードを指定せずに **ra_admin** を呼び出します。

```
ra_admin
```

ra_admin は、すべての Replication Agent オブジェクトのリストを返します。

テーブル・オブジェクト

Replication Agent は、Oracle プライマリ・データベースにテーブル・オブジェクトを作成します。

これらのテーブルは、Replication Agent オブジェクトとみなされます。

表 4 : Replication Agent のテーブル

テーブル	データベース名
Proc-Active テーブル	<prefix>PROCACTIVE

複数 Replication Agent インスタンスのテーブル

Replication Agent グループ内の各 Replication Agent インスタンスに関する情報を維持します。

複数 Replication Agent インスタンスのテーブルは、名前付き *ra_admin_prefix*AGENT です。ただし、*ra_admin_prefix* は Replication Agent システム・オブジェクト名に使用されるプレフィクス文字列です。

表 5 : 複数 Replication Agent インスタンスのテーブル

カラム名	型	内容
prefix	VARCHAR2(3)	この Replication Agent インスタンスのインスタンス・プレフィクス。このプレフィクスは、 ra_admin_instance_prefix パラメータの設定で決まります。
locator	VARCHAR2(128)	この Replication Agent インスタンスのトランケーション・ポイント。
inittime	TIMESTAMP	この Replication Agent インスタンスが初期化された日時。
version	VARCHAR2(8)	この Replication Agent インスタンスのバージョン。
rasd_export_file	VARCHAR(255)	RASD データのエクスポート先ファイル。
upg_from_ver	VARCHAR(16)	Replication Agent のアップグレード元バージョン。
upg_fin_ts	TIMESTAMP	アップグレードが完了した日時。

複数 Replication Agent マーク付けテーブルのテーブル

Replication Agent グループ内の複写するようマーク付けされたテーブルに関する情報を維持します。

複数 Replication Agent マーク付けテーブルのテーブルは、名前付き *ra_admin_prefix*TABLE です。ただし、*ra_admin_prefix* は Replication Agent システム・オブジェクト名に使用されるプレフィクス文字列です。

表 6 : 複数 Replication Agent マーク付けテーブルのテーブル

カラム名	型	内容
prefix	VARCHAR2(3)	この Replication Agent インスタンスのインスタンス・プレフィクス。このプレフィクスは、 ra_admin_instance_prefix パラメータの設定で決まります。
tableid	NUMBER	このマーク付けテーブルのオブジェクト ID。
autocorrection	NUMBER(38)	マーク付けテーブルのオートコレクションが有効になっているかどうかを示します。オートコレクションが有効の場合、このカラムの値は 1 です。そうでない場合、カラムの値は 0 (デフォルト) です。
ruletype	NUMBER(38)	未使用。
rulevalue	VARCHAR2(30)	未使用。

複数 Replication Agent マーク付けプロシージャのテーブル

Replication Agent グループ内の複写するようマーク付けされたプロシージャに関する情報を維持します。

複数 Replication Agent マーク付けプロシージャのテーブルは、名前付き *ra_admin_prefix*PROCEDURE です。ただし、*ra_admin_prefix* は Replication Agent システム・オブジェクト名に使用されるプレフィクス文字列です。

表 7 : 複数 Replication Agent マーク付けプロシージャのテーブル

カラム名	型	内容
prefix	VARCHAR2(3)	この Replication Agent インスタンスのインスタンス・プレフィクス。このプレフィクスは、 ra_admin_instance_prefix パラメータの設定で決まります。
procid	NUMBER	このマーク付けプロシージャのオブジェクト ID。

カラム名	型	内容
shadowtable	VARCHAR2(30)	このプロシージャに対応する隠しテーブルのオブジェクト ID。
ruletype	NUMBER(38)	未使用。
rulevalue	VARCHAR2(30)	未使用。

マーカ・オブジェクト

Replication Agent は、プライマリ・データベースにマーカ・オブジェクトを作成します。

これらの Replication Agent オブジェクトは、Replication Server マーカと関係があります。これらのオブジェクトの作成時に付与されるパーミッションはありません。

表 8 : Replication Agent のマーカ・オブジェクト

オブジェクト	名前
トランザクション・ログ・マーカ・プロシージャ	RS_MARKER
ダンプ・マーカ・プロシージャ	RS_DUMP
トランザクション・ログ・マーカ隠しテーブル	<prefix>MARKERSH[xxx]
ダンプ・マーカ隠しテーブル	<prefix>DUMPSH[xxx]

シーケンス

Replication Agent は、Oracle プライマリ・データベースにシーケンスを作成します。

これらの Oracle シーケンスは、Replication Agent のオブジェクトとみなされます。

表 9 : Replication Agent シーケンス

シーケンス	データベース名
プロシージャ・コールの割り当て	<prefix>PCALL_[xxx]

マーク付きのプロシージャ

Replication Agent は、複写対象としてマーク付けされているプライマリ・プロシージャごとに Oracle プライマリ・データベースにオブジェクトを作成します。

これらの Replication Agent オブジェクトは、複写対象としてマーク付けされたプライマリ・プロシージャごとに作成されます。これらのオブジェクトは、プロシージャが複写用にマーク付けされている場合にのみ作成されます。

表 10: マーク付けされたプロシージャごとに作成される Replication Agent オブジェクト

オブジェクト	名前
隠しテーブル	<prefix><procedure_name>SH

トランザクション・ログのトランケーション

Replication Agent では、自動と手動の両方のログ・トランケーションがサポートされています。

Replication Agent には自動トランザクション・ログ・トランケーションについてのオプションが 2 つあります。

- 指定した時間間隔に基づく定期的トランケーション
- Replication Agent がプライマリ Replication Server から新しい LTM ロケータ値を受信するたびに自動でトランケートする自動ログ・トランケーション機能をオフにすることもできます。デフォルトでは、自動ログ・トランケーションは有効になっており、Replication Agent がプライマリ Replication Server から新しい LTM ロケータ値を受信するごとにログをトランケートするように設定されています。

Replication Agent のログ・トランケーションを設定するには、次のガイドラインに従ってください。

- **pdb_include_archives** がデフォルトの true に設定されており、**pdb_archive_remove** が false に設定されている場合、Replication Agent ではオンラインの、またはアーカイブされたトランザクション・ログのトランケーションを行いません。**pdb_include_archives** がデフォルトの true に設定されており、**pdb_archive_remove** が true に設定されていると、Replication Agent は **pdb_archive_path** の場所から処理済みのアーカイブ redo ログを削除します。Replication Agent はオンラインのトランザクション・ログのアーカイブは行いません。

注意： 追加のアーカイブ・ログ・ディレクトリを使用した場合にのみアーカイブ・ログ・ファイルを削除するように Replication Agent を設定することをおすすめします。

- 設定パラメータ **pdb_include_archives** を false に設定すると、Replication Agent はアーカイブ・ログ・シーケンスのキーワードを使用して **alter system** コマンドを発行し、オンラインの redo ログのトランケーションを (定期的、または手動で) 行います。このコマンドは、Replication Agent によってコンテンツが処理され、アーカイブの準備ができた redo ログ・ファイルのログ・シーケンス番号を使用します。

注意： Oracle の **alter system** コマンド構文を使用すると、コマンドで指定された 1 つのログ・シーケンス以外にも redo ログ・ファイルをアーカイブすること

ができます。誤ってアーカイブするのを防ぐため、Replication Agent は、ステータスが現在になっている redo ログ・ファイルを処理している場合にのみこのコマンドを発行します。

- ra_config を使用して truncation_type 設定パラメータの値を設定することで、自動トランケーション・オプション(「なし」を含む)を指定できます。
時間間隔に基づきトランザクション・ログを自動的にトランケートするには、ra_config を使用して truncation_interval 設定パラメータの値を設定します。
- Replication Agent トランザクション・ログは、Replication Agent 管理ポートで pdb_truncate_xlog を呼び出すことによっていつでも手動でトランケートできます。

これらのプロパティの詳細については、『Replication Agent リファレンス・マニュアル』を参照してください。トランケート方法の詳細については、『Replication Agent 管理ガイド』の「管理」を参照してください。

Replication Agent for Microsoft SQL Server

Replication Agent for Microsoft SQL Server に特有の Replication Agent の特性を確認します。

「Replication Agent for Microsoft SQL Server」という用語は、Microsoft SQL Server データ・サーバに配置されているプライマリ・データベース用にインストールおよび設定された Replication Agent ソフトウェアのインスタンスを指します。

注意： Replication Agent の基本的な機能とオペレーションの詳細については、『Replication Agent 管理ガイド』と『Replication Agent リファレンス・マニュアル』を参照してください。

Microsoft SQL Server 固有の考慮事項

以下の全般的な問題と考慮事項は、Replication Agent と Microsoft SQL Server データ・サーバの併用に固有です。

Replication Agent for Microsoft SQL Server は、Microsoft SQL Server プライマリ・データベースのログを読み取ります。データベース・ログを読み取るには、ログ・ファイルに直接アクセスできる場所に Replication Agent をインストールする必要があります。Replication Agent がインストールされているマシンがプライマリ・データベースが配置されているマシンと同じハードウェアおよびオペレーティング・システムでなければならないため、Replication Agent for Microsoft SQL Server は Windows プラットフォームでのみ使用可能です。「Windows」という単語は、サポート対象のすべての Microsoft Windows プラットフォームを示します。サポート対象のプラットフォームの完全なリストについては、『Replication Agent インストール・ガイド』を参照してください。

Microsoft SQL Server の要件

Microsoft SQL Server 用の以下の要件に従ってください。

- Replication Agent は、Microsoft SQL Server 2005 Service Pack 2 をサポートしているため、データベース互換性レベルを「SQL Server 2005 (90)」に設定する必要があります。Replication Agent は Microsoft SQL Server 2008 もサポートしていますが、サポートは Microsoft SQL Server 2005 でサポートされているのと同じ機能に制限されます。
- Microsoft レプリケーションと Replication Agent を同時に同じ Microsoft SQL Server データベースで使用することはできません。Replication Agent for

Microsoft SQL Server を使用する前に、Microsoft レプリケーションを無効にしてください。

- Replication Agent for Microsoft SQL Server が実行されているプライマリ・データベースに Microsoft SQL Server パブリケーションを作成することはできません。
- Microsoft SQL Server TCP/IP プロトコルを有効にする必要があります。

Microsoft SQL Server の制限

Microsoft SQL Server を Replication Agent と一緒にプライマリ・データベースとして使用した場合、次の制限が適用されます。

- 複製対象にマークされているテーブルでの **TRUNCATE TABLE** コマンドの使用は禁止されています。
- 複製対象のマークが付けられているストアド・プロシージャの削除は禁じられています。
- Replication Agent では、Microsoft SQL Server 2008 の一部の機能がサポートされていません。

サポート対象外のソフトウェア機能

以下の機能は Sybase 複製ではサポートされていません。

- Microsoft SQL Server クラスター
- Microsoft SQL Server の仮想計算カラム
- Replication Server 並列 DSI (Adaptive Server Enterprise 以外のデータベース用)
- Replication Server ウォーム・スタンバイ (Adaptive Server Enterprise 以外のデータベース用)
- Replication Server **rs_init** ユーティリティ (Adaptive Server Enterprise 以外のデータベース用)
- Replication Server **rs_subcomp** ユーティリティ (Adaptive Server Enterprise 以外のデータベース用)
- Replication Server 自動マテリアライゼーション (Adaptive Server Enterprise 以外のデータベース用)
- 他のベンダが複製している環境で複製を行う際の Replication Server (Adaptive Server Enterprise 以外のデータベース用)
- 一部の Microsoft SQL Server 2008 機能

Replication Agent では、Microsoft SQL Server 2008 の以下の機能はサポートされていません。

- カラム・セット
- **MERGE SQL** 文

- テーブル値パラメータを持つプロシージャ
- スパース・カラム
- 透過的なデータ暗号化 (TDE)

これらの機能を使用するテーブルまたはストアド・プロシージャは、**pdb_setreptable** と **force** キーワードを使用してもマークできません。

プライマリ・キーでの遅延更新の複写

ユニークなカラム・インデックスを持つテーブルでの更新は、従来の複写ではサポートされていないため、Replication Server はエラーを報告します。

ユニークなカラム・インデックスを持つテーブルの更新の複写は、サポートされていないため、Replication Server はエラーを報告します。たとえば、テーブル **t** はカラム **c** 上にユニークなインデックスを持ち、値は 1、2、3、4、および 5 です。このテーブルには単一の **update** 文が適用されます。

```
update t set c = c+1
```

従来の複写を使用すると、この文は以下ようになります。

```
update t set c = 2 where c = 1
update t set c = 3 where c = 2
update t set c = 4 where c = 3
update t set c = 5 where c = 4
update t set c = 6 where c = 5
```

最初の更新で、値 **c=2** をテーブルに挿入しようとする。ところが、この値はテーブルに既に存在します。Replication Server は、エラー 2601 (重複キーの挿入の試み) を表示します。

ASE はこの問題を SQL 文機能で解決します。この機能により、更新を遅延させ逆順に実行することで、重複キー・エラーを回避します。ただし、Sybase 以外のデータ・サーバではこの機能は提供されていないので、ユニークなカラム・インデックスを持つ Sybase 以外のテーブルへの更新を複写しようとする、Replication Server DSI がクラッシュします。この問題を回避するには、ユニークなインデックス定義を拡充します。

サポート対象外のデータ型

以下のデータ型は Sybase 複写ではサポートされていません。

- cursor
- table
- xml
- 一部の Microsoft SQL Server 2008 データ型

Replication Agent は以下の Microsoft SQL Server 2008 データ型をサポートしていません。

- date
- datetime2
- datetimeoffset
- filestream
- geography
- geometry
- hierarchyid
- time
- Large ユーザ定義データ型

これらのデータ型のカラムが含まれるテーブルは、**pdb_setreptable** と **force** キーワードを使用してもマークできません。

Microsoft SQL Server パッチの適用

Microsoft SQL Server にパッチを適用します。

1. すべてのデータがレプリケート・サイトに複製されていることを確認してください。

注意： この手順の前にすべてのアクティビティを停止し、**pds_username** を除くすべてのユーザがプライマリ・データベースからログオフする必要があります。

既存の各 Replication Agent for Microsoft SQL Server インスタンスごとに、インスタンスが *Replicating* 状態にあることを確認し、複製を完了させます。複製が完了したことを確認するには、**quiesce** コマンドで Replication Agent のインスタンスをクワイース状態にします。

注意： Replication Agent はログ・ファイルからすべてのデータを読み取り、Replication Server に送信するため、コマンドが返されるまでにしばらく時間がかかる場合があります。

2. データベース・トリガを無効にします。

pdb_automark_tables 設定パラメータが true に設定されている場合は、プライマリ・データベースにログオンし、次のコマンドを実行して自動マーク付けトリガを無効にします。

```
DISABLE TRIGGER ra_createtable_trig_ ON DATABASE
```

ここで **ra_createtable_trig_** は Replication Agent によって作成された自動マーク付けトリガの名前です。

3. Microsoft のマニュアルの指示に従って、サービス・パッチを適用します。
4. Microsoft SQL Server システム・リソース・データベースでオブジェクトを再生成します。

- 新しいコマンド・ウィンドウを開き、次のコマンドを実行して、シングル・ユーザ・モードで Microsoft SQL Server を再起動します。

```
"C:¥Program Files¥Microsoft SQL
Server¥MSSQL.1¥MSSQL¥Binn¥sqlservr.exe" -m
-sserverName¥instanceName
```

ここで *instanceName* は Microsoft SQL Server インスタンスの名前です。

- Replication Agent インスタンスにログインします。

```
isql -U username -P password -S instanceName
```

- Microsoft SQL Server を再初期化します。

```
server_xlog remove, force
go
server_xlog init
go
```

- Microsoft SQL Server をマルチユーザ・モードで再起動します。

5. パッチを適用する前に **pdb_automark_tables** 設定パラメータが true に設定されていた場合は、次のようにします。

- プライマリ・データベースにログインし、次のコマンドを実行して自動マーク付けトリガを有効にします。

```
ENABLE TRIGGER ra_createtable_trig_ ON DATABASE
```

ここで **ra_createtable_trig_** は Replication Agent によって作成された自動マーク付けトリガの名前です。

- プライマリ・データベースにログインし、次のコマンドを実行して DDL トリガを有効にします。

```
ENABLE TRIGGER ra_ddl_trig_ ON DATABASE
```

ここで **ra_ddl_trig_** は、Replication Agent によって作成された DDL トリガの名前です。

6. LTM ロケータの値をゼロに設定し、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。

- RSSD にログインし、次のコマンドを実行して、LTM ロケータをゼロに設定します。

```
rs_zeroltm < ra_instance > , < pdb_name > "
```

- Replication Agent にログインし、次のコマンドを実行して、トランケーション・ポイントをログの末尾に移動します。

```
ra_locator move_truncpt
```

7. Replication Agent で複製またはその他の操作をレジュームします。

DDL 複写

データ定義言語 (DDL) コマンドの複写はサポートされていますが、Microsoft SQL Server データベースのみに限られます。

注意： Replication Agent による DDL コマンドの変換または調整は提供されていません。したがって、DDL コマンドは、他の Microsoft SQL Server データベースにのみ複写してください。

Replication Agent では `pdb_setrep_ddl` コマンドを使用して、DDL コマンドの複写を有効または無効にします。Replication Server はプライマリ・データベースで DDL コマンドを実行した同じユーザとして、`ddl_username` パラメータを使用してレプリケート・データベースで DDL コマンドを実行します。

`pdb_setrepddl` および `ddl_username` の使用の詳細については、『Replication Agent リファレンス・マニュアル』の「コマンド・リファレンス」の「`pdb_setrepddl`」、および『Replication Agent リファレンス・マニュアル』の「設定パラメータ」の「`ddl_username`」を参照してください。

DDL パラメータ

Microsoft SQL Server で DDL を複写するには、`pdb_setrepddl` の値を `enable` に設定する以外に、Replication Agent の `ddl_username` パラメータと `ddl_password` パラメータを設定します。

`ddl_username` パラメータは、DDL コマンドをレプリケートまたはターゲット・データベースに複写するために LTL に含まれているレプリケート・データベースのユーザ名です。

パーミッション

レプリケート・データベースで複写されたすべての DDL コマンドを実行するパーミッションに加えて、`ddl_username` には DDL コマンドがレプリケート・データベースに複写される可能性があるすべてのユーザに対する `impersonate` パーミッションが付与されている必要があります。この `impersonate` パーミッションは、DDL コマンドの実行時にレプリケート・データベースでセッション・コンテキストを切り替える際に必要です。このユーザはコンテキストを切り替えて、プライマリ・データベースで DDL コマンドを実行したユーザと同じ権限とデフォルトのスキーマ設定を使用して、DDL コマンドを適用します。このコンテキストの切り替えを行うには、`ddl_username` ユーザに、プライマリ・データベースから複写される DDL コマンドを実行する可能性があるすべてのユーザに対して `execute as user` Microsoft SQL Server コマンドを実行するパーミッションが必要になります。

たとえば、`user1` のデフォルトのスキーマが `schema1` の場合、プライマリ・データベースでは次の DDL コマンドが実行されます。

```
create table tab1 (id int)
```

この結果、プライマリ・データベースに `schema1.tab1` というテーブルが作成されます。レプリケート・データベースでは、`user2` のデフォルトのスキーマが `schema2` である場合、`schema2.tab1` というテーブルが生成されるため、すぐにこの DDL を実行できません。したがって、**ddl_username** 設定パラメータによって名前が指定された `user2` は、レプリケート・データベースで次のコマンドを実行して、`user1` になり代わる必要があります。

```
execute as user = 'user1'
```

こうすれば、レプリケート・データベースで `user2` が正しいスキーマを使用して DDL を実行し、`schema1.tab1` というテーブルを生成することができます。

『Replication Agent リファレンス・マニュアル』を参照してください。

Impersonate パーミッション

`impersonate` パーミッションを **ddl_username** ユーザに付与するには、2つの方法があります。

- **ddl_username** ユーザにデータベース所有者のパーミッションを付与することができます。この場合、`impersonate` パーミッションを暗黙的に付与することになります。
- または、明示的に `impersonate` パーミッションを付与することもできます。

```
GRANT IMPERSONATE ON USER::user1 TO ddl_user
```

ここで `user1` はレプリケート・データベースに複製される DDL を持っていることされるユーザで、`ddl_user` は **ddl_username** ユーザになります。

注意： この `grant` コマンドは、**ddl_username** と定義されたユーザが DDL コマンドを実行するレプリケート・データベースで実行する必要があります。

Microsoft SQL Server で DDL を複製する場合は、Microsoft SQL Server をレプリケート・データベースに使用してください。Microsoft SQL Server から Microsoft SQL Server 以外のレプリケート・データベースに DDL コマンドを複製することはできません。

注意： DDL を複製するには、Replication Server にデータベースレベルの複製定義が必要であり、その定義で **replicate DDL** が設定されていなければなりません。

『Replication Server リファレンス・マニュアル』を参照してください。

複写からフィルタされる DDL コマンドおよびオブジェクト

次のデータベーススコープの DDL コマンドは複写されません。

- ALTER_APPLICATION_ROLE
- ALTER_ASSEMBLY
- ALTER_AUTHORIZATION_DATABASE
- ALTER_CERTIFICATE
- CREATE_APPLICATION_ROLE
- CREATE_ASSEMBLY
- CREATE_CERTIFICATE
- CREATE_EVENT_NOTIFICATION
- DROP_EVENT_NOTIFICATION

次のサーバスコープの DDL コマンドは複写されません。

- ALTER_AUTHORIZATION_SERVER
- ALTER_DATABASE
- ALTER_LOGIN
- CREATE_DATABASE
- CREATE_ENDPOINT
- CREATE_LOGIN
- DENY_SERVER
- DROP_DATABASE
- DROP_ENDPOINT
- DROP_LOGIN
- GRANT_SERVER
- REVOKE_SERVER

複写されないユーザのリストで定義されたユーザが所有するオブジェクトは、どれも複写されません。このリストは、pdb_ownerfilter コマンドを使用して変更できます。さらに、複写されないオブジェクトの所有者のデフォルト・リストが提供されています。複写されないオブジェクト所有者のリストの返却、追加、または削除には、pdb_ownerfilter コマンドを使用します。『Replication Agent リファレンス・マニュアル』を参照してください。

Replication Agent コネクティビティ

Replication Agent for Microsoft SQL Server では、すべての複写システム・コンポーネントとの通信に JDBC™ を使用します。

Microsoft SQL Server JDBC ドライバは Replication Agent ホスト・マシンにインストールされ、このドライバのインストール先ディレクトリは CLASSPATH 環境変数に含まれる必要があります。

インストールする特定の JDBC ドライバおよびバージョンの詳細については、『Replication Agent インストール・ガイド』を参照してください。

Replication Agent のパーミッションと役割

Replication Agent for Microsoft SQL Server では、複写作業用のデータベース・オブジェクトをプライマリ・データベースで作成する必要があります。

次のパーミッションは、**pds_username** パラメータで指定されたユーザに付与する必要があります。

- **create table** – プライマリ・データベースにテーブルを作成するために必要です。
- **create trigger** – プライマリ・データベースに DDL トリガを作成するために必要です。
- **create procedure** – プライマリ・データベースにプロシージャを作成するために必要です。

pds_username パラメータで指定されたユーザをプライマリ・データベースの次の役割に追加する必要があります。

- **db_owner** – Replication Agent がプライマリ・データベースで **sp_repltrans** と **sp_repldone** を実行できるようにするために必要です。このロールはプライマリ・データベースの初期化にも必要になる。
- **sysadmin** – Microsoft SQL Server データ・サーバの初期化と初期化解除に必要です (それぞれ **ra_admin init** および **ra_admin deinit** を使用します)。

sybfilter ドライバ

Replication Agent は Microsoft SQL Server のログ・ファイルを読み込むことができる必要があります。ただし、Microsoft SQL Server プロセスは排他的な読み込みパーミッションを使用してこれらのログ・ファイルを開くので、Replication Agent を含む他のプロセスからこのファイルを読み込むことはできません。Replication Agent でデータを複写するには、sybfilter ドライバを使用してログ・ファイルを読み込み可能にする必要があります。

参照：

- sybfilter ドライバのリファレンス (151 ページ)

プライマリ・データ・サーバと Replication Agent の初期化

Microsoft SQL Server を初期化するため、Replication Agent for Microsoft SQL Server はデータ・サーバ・レベルとデータベース・レベルの両方にオブジェクトをインストールします。

データ・サーバ・レベルの変更は 1 回だけ必要です。ただし、サーバ・レベルの変更を行うには、追加のパーミッションが必要になるため、**pds_dac_port_number**

パラメータを使用し、プライマリ・データベースをスタンドアロン・モードにする必要があります。その後の **ra_admin init** の実行では、サーバは変更されず、追加のパーミッションや設定も必要ありません。

初めての初期化

プライマリ Microsoft SQL Server を初期化して、Replication Agent が複写対象にマークされたテーブルまたはプロシージャのサプリメンタル・ログを開けるようにする必要があります。この作業はプライマリ・データ・サーバごとに 1 回だけ行います。

初めてプライマリ・データ・サーバと Replication Agent を初期化する際には、次の点に注意してください。

1. Microsoft SQL Server Analysis Service を停止します。コントロールパネル管理ツールサービスで、SQL Server (SERVER) というサービス名を探します。ここで、SERVER は Microsoft SQL Server データ・サーバの名前です。このサービスを停止します。
2. Microsoft SQL Server でリモート専用管理接続 (DAC) が許可されていることを確認します。

```
sp_configure 'remote admin connections', 1
GO
RECONFIGURE
GO
```

設定オプションを変更するパラメータおよび **RECONFIGURE** 文を実行するパラメータの両方と一緒に **sp_configure** を実行するには、**ALTER SETTINGS** サーバレベルのパーミッションが付与されている必要があります。このパーミッションは、**sysadmin** および **serveradmin** の固定サーバ役割によって暗黙的に保持されています。

3. プライマリ Microsoft SQL Server DAC ポート番号を判断します。
 - a) ERRORLOG ファイルをテキスト・エディタで開きます。このファイルは、Microsoft SQL Server の log ディレクトリ内にあります。例：

```
C:¥Program Files¥Microsoft SQL Server¥MSSQL.1¥MSSQL¥LOG
¥ERRORLOG
```
 - b) 次のようなエントリを見つけるために "Dedicated admin" という文字列を検索します。

```
2007-11-09 13:40:02.40 Server Dedicated admin
connection support was established for listening
locally on port 1348.
```
 - c) このエントリで指定したポート番号を記録します。
4. Replication Agent にログインし、**pds_dac_port_number** 設定パラメータを設定します。

```
ra_config pds_dac_port_number, port
```

ここで *port* は、記録した DAC ポート番号です。

5. また、Microsoft SQL Server プライマリ・データベースの Replication Agent コネクティビティ・パラメータを設定します。

- **pds_server_name**
- **pds_database_name**
- **pds_username**
- **pds_password**
- **pds_port_number**

これらの設定パラメータの詳細については、『Replication Agent インストール・ガイド』および『Replication Agent リファレンス・マニュアル』を参照してください。

6. Microsoft SQL Server サービスを停止します。
 - a) コントロールパネル管理ツールサービスで、SQL Server (SERVER) というサービス名を探します。ここで、SERVER は Microsoft SQL Server データ・サーバの名前です。
 - b) このサービスを停止します。
7. コマンド・ウィンドウを開き、シングル・ユーザ・モードで Microsoft SQL Server を再起動します。

```
"C:¥Program Files¥Microsoft SQL
Server¥MSSQL.1¥MSSQL¥Binn¥sqlservr.exe" -m -s
instanceName
```

ここで *instanceName* は Microsoft SQL Server インスタンスの名前です。

8. プライマリ・データベースに他のコネクションがないこと、また Replication Agent がプライマリ・データベースに接続できることを確認します。
 - a) Replication Agent インスタンスのインスタンス名とポート番号で *sql.ini* ファイルを更新します。
 - b) Replication Agent インスタンスにログインします。

```
isql -U username -P password -S instanceName
```

ここで *username*、*password*、および *instanceName* は、ユーザ ID、パスワード、および Replication Agent のインスタンス名です。

- c) 次のコマンドを実行します。

```
test_connection PDS
```

9. Microsoft SQL Server データ・サーバと Replication Agent を初期化します。

```
server_xlog init
ra_admin init
```

プライマリ・データベースで、Replication Agent はテーブル、プロシージャ、トリガを作成します。sp_SybSetLogforReplTable、sp_SybSetLogforReplProc、および sp_SybSetLogforLOBCol プロシージャが、execute パーミッションが PUBLIC に付与された状態で mssqlsystemresource データベースに作成されます。

10. Microsoft SQL Server サービスを再度停止します。

- a) コントロールパネル管理ツールサービス で、SQL Server (SERVER) というサービス名を探します。ここで、SERVER は Microsoft SQL Server データ・サーバの名前です。
- b) このサービスを停止します。

11. Microsoft SQL Server をマルチユーザ・モードで再起動します (通常の起動)。

- a) コントロールパネル管理ツールサービス で、SQL Server (SERVER) というサービス名を探します。ここで、SERVER は Microsoft SQL Server データ・サーバの名前です。
- b) このサービスを開始します。

Microsoft SQL Server Agent サービスや Microsoft SQL Server Analysis Service などの他の Microsoft SQL Server サービスを開始します。

参照：

- Microsoft SQL Server プライマリ・データベースの Replication Agent オブジェクト (107 ページ)

初回以降の初期化

初めて Replication Agent を初期化し、続けて ra_admin deinit を使用して Replication Agent の初期化を解除したら、同じプライマリ・データ・サーバ内の別のデータベースでもこの Replication Agent インスタンスまたは別の Replication Agent インスタンスを再初期化します。

初めての初期化に続いて Replication Agent インスタンスを初期化する際は、次の点に注意してください。

1. プライマリ Microsoft SQL Server DAC ポート番号を特定し、Microsoft SQL Server でリモート DAC が許可されていることを確認します。

```
sp_configure 'remote admin connections', 1
GO
RECONFIGURE
GO
```

設定オプションを変更するパラメータおよび RECONFIGURE 文を実行するパラメータの両方と一緒に sp_configure を実行するには、ALTER SETTINGS サーバレベルのパーミッションが付与されている必要があります。ALTER SETTINGS

パーミッションは、**sysadmin** および **serveradmin** 固定サーバ役割によって暗黙的に保持されています。

2. Replication Agent にログインし、**pds_dac_port_number** 設定パラメータを設定します。
3. Microsoft SQL Server プライマリ・データベースの Replication Agent コネクティビティ・パラメータを設定します。
 - **pds_server_name**
 - **pds_database_name**
 - **pds_username**
 - **pds_password**

これらの設定パラメータの詳細については、『Replication Agent インストール・ガイド』および『Replication Agent リファレンス・マニュアル』を参照してください。

4. Replication Agent がプライマリ・データベースに接続できることを確認します。

```
test_connection PDS
```

5. Microsoft SQL Server データ・サーバと Replication Agent を初期化します。

```
ra_admin init
```

最終のクリーンアップ

ra_admin init を実行した各データベースで **ra_admin deinit** を実行して所定のプライマリ・データ・サーバのすべてのデータベースからすべての Replication Agent オブジェクトを削除したら、Replication Agent の残りのオブジェクトをすべて削除して、プライマリ・データ・サーバを完全にクリーンアップすることができます。

プライマリ・データ・サーバから残りのすべての Replication Agent のオブジェクトをクリーンアップするには、次の手順に従います。

1. Microsoft SQL Server サービスを停止します。
 - a) コントロールパネル管理ツールサービス で、SQL Server (SERVER) というサービス名を探します。ここで、SERVER は Microsoft SQL Server データ・サーバの名前です。
 - b) このサービスを停止します。
2. コマンド・ウィンドウを開き、シングル・ユーザ・モードで Microsoft SQL Server を再起動します。

```
"C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Binn\sqlservr.exe" -m -s instanceName
```

ここで *instanceName* は Microsoft SQL Server インスタンスの名前です。

3. Microsoft SQL Server SQL Browser サービスが実行中であることを確認し、**sqlcmd** ユーティリティと **-A** オプション、または Management Studio を使用し

て、データ・サーバに接続します。サーバ名を `Admin:servername` と指定します。ここで `servername` はデータ・サーバの名前です。

4. Replication Agent 用に `pds_username` ユーザが作成されていた場合は削除します。

```
drop user pds_username
```

5. `mssqlsystemresource` データベースから特別なプロシージャのマーク付けを削除します。

```
drop procedure sp_SybSetLogforReplTable;  
drop procedure sp_SybSetLogforReplProc;  
drop procedure sp_SybSetLogforLOBCol;
```

6. Windows サービスを停止するか、**shutdown** コマンドと **sqlcmd** ユーティリティを実行して、シングル・ユーザ・モードで Microsoft SQL Server を停止します。
7. 各ログ・デバイスの `sybfilter` ドライバの作用を元に戻すため、設定ファイルを編集するか、`sybfilter` マネージャ・コンソールを使用して、ログ・パスのエントリを削除します。
8. Microsoft SQL Server をマルチユーザ・モードで再起動します (通常の起動)。
 - a) コントロールパネル管理ツールサービスで、SQL Server (SERVER) というサービス名を探します。ここで、SERVER は Microsoft SQL Server データ・サーバの名前です。
 - b) このサービスを開始します。

参照：

- `sybfilter` ドライバのリファレンス (151 ページ)

Microsoft sqlcmd ツール

Microsoft **sqlcmd** (または互換ツール) を使用して、Microsoft SQL Server データベースにアクセスし、テスト・スクリプトを実行します。

データベース・オブジェクト名の大文字と小文字

データベース・オブジェクト名は、複写定義に指定したものと同一フォーマットでプライマリ Replication Server に送信する必要があります。そうしないと、複写に失敗します。たとえば、複写定義ではテーブル名がすべて小文字で指定されている場合、Replication Agent がプライマリ Replication Server にこのテーブル名を送信する際は、すべて小文字で表示されている必要があります。

Replication Agent がプライマリ Replication Server にデータベース・オブジェクト名を送信するときの大文字と小文字の区別を制御するには、`lml_character_case` 設定パラメータを次のいずれかの値に設定します。

- **asis** – (デフォルト) データベース・オブジェクト名は、プライマリ・データ・サーバに格納されているものと同じフォーマットで Replication Server に渡されます。
- **lower** – データベース・オブジェクト名は、プライマリ・データ・サーバにどのようなに格納されているかにかかわらず、すべて小文字で Replication Server に渡されます。
- **upper** – データベース・オブジェクト名は、プライマリ・データ・サーバにどのようなに格納されているかにかかわらず、すべて大文字で Replication Server に渡されます。

Microsoft SQL Server では、データベース・オブジェクト名は入力された大文字・小文字と同じ文字 (大文字または小文字) で格納されます。このため、**asis** オプションを使用し、Microsoft SQL Server に格納されているのと同じ大文字・小文字を使用して、プライマリ Replication Server にデータベース・オブジェクト名を送信する必要があります。

オリジン・キュー ID のフォーマット

トランザクション・ログの各レコードは、64 の 16 進文字 (32 バイト) で構成されるオリジン・キュー ID で識別されます。オリジン・キュー ID のフォーマットは Replication Agent インスタンスによって決定され、プライマリ・データベースの種類によって異なります。

表 11 : Replication Agent for Microsoft SQL Server オリジン・キュー ID のフォーマット

文字	バイト	説明
0-3	2	データベース世代 ID
4-11	4	仮想ファイルのシーケンス番号。
12-19	4	ページ開始オフセット
20-23	2	オペレーション番号
24-31	4	ユニーク性を指定するために使用可能
32-39	4	最も古いアクティブなトランザクション：仮想ファイルのシーケンス番号
40-47	4	最も古いアクティブなトランザクション：ページ開始オフセット
48-51	2	最も古いアクティブなトランザクション：オペレーション番号
52-59	4	最新のコミットされたトランザクション：ページ開始オフセット

文字	バイト	説明
60-63	2	最新のコミットされたトランザクション：オペレーション番号

Microsoft SQL Server とのデータ型互換性

Replication Agent は、Microsoft SQL Server トランザクションを処理し、トランザクション情報をプライマリ Replication Server に渡します。プライマリ Replication Server は、複写定義に指定されているデータ型のフォーマットを使用して、Replication Agent からデータを受信します。

表 12 : Microsoft SQL Server から Replication Server のデフォルトのデータ型マッピング

Microsoft SQL Server データ型	Microsoft SQL Server の長さ／範囲	Sybase データ型	Sybase の長さ／範囲	注意
bigint	-2^{63} から $2^{63}-1$	bigint	-2^{63} から $2^{63}-1$	
binary	8000 バイトまでの固定長	binary	32K	
bit	0 または 1 の整数値	bit	0 または 1 の整数値	
char	8000 文字までの固定長	char	32K	
datetime	01/01/1753 から 12/31/9999 までの日時	datetime	01/01/1753 から 12/31/9999 までの日時	
decimal	-10^{38} から $10^{38}-1$ の数値	decimal	-10^{38} から $10^{38}-1$ の数値	
float	$-1.79E+308$ から $1.79E+308$ の浮動小数点精度	float	$-1.79E+308$ から $1.79E+308$ の浮動小数点精度	Sybase の結果は、機種によって異なります。
image	$2^{31}-1$ バイトまでの可変長	image	2GB	

Microsoft SQL Server データ型	Microsoft SQL Server の長さ / 範囲	Sybase データ型	Sybase の長さ / 範囲	注意
int	-2^{31} から $2^{31} - 1$	int	-2^{31} から $2^{31} - 1$	
money	-2^{63} から $2^{63} - 1$ の通貨	money	-2^{63} から $2^{63} - 1$ の通貨	
nchar	4000 文字までの固定長 Unicode	unichar または char	32K	実際の最大長は @ncharsize * 文字数になります。
ntext	$2^{30} - 1$ 文字までの可変長 Unicode	unitext または image	2GB	Replication Server 15.0 以降では、ntext は unitext にマップします。以前のバージョンの Replication Server では、ntext は image にマップします。
nvarchar	4000 文字までの可変長 Unicode	univarchar または varchar	32K	実際の最大長は @ncharsize * 文字数になります。

Microsoft SQL Server データ型	Microsoft SQL Server の長さ／範囲	Sybase データ型	Sybase の長さ／範囲	注意
nvarchar (max)	2 ³⁰ - 1 文字までの可変長 Unicode	unitext または image	2GB	nvarchar (max) データ型は、Microsoft SQL Server 以外のデータ・サーバに複写することはできません。 Replication Server 15.0 以降では、nvarchar (max) は unitext にマップします。以前のバージョンの Replication Server では、nvarchar (max) は image にマップします。
numeric	decimal データ型の同意語	numeric	decimal データ型の同意語	
real	-3.40E + 38 から 3.40E + 38 の浮動小数点精度	real	-3.40E + 38 から 3.40E + 38 の浮動小数点精度	Sybase の結果は、機種によって異なります。
smalldatetime	01/01/1900 から 06/06/2079 までの日時	datetime	01/01/1900 から 06/06/2079 までの日時	
smallint	-2 ¹⁵ から 2 ¹⁵ - 1 の整数値	smallint	-2 ¹⁵ から 2 ¹⁵ - 1 の整数値	
smallmoney	-214,748.3648 から 214,748.3647 の通貨	smallmoney	-214,748.3648 から 214,748.3647 の通貨	

Microsoft SQL Server データ型	Microsoft SQL Server の長さ／範囲	Sybase データ型	Sybase の長さ／範囲	注意
sql_variant	text、ntext、timestamp、および sql_variant 以外の 8000 バイトまでの任意のデータ型	varchar または opaque	32K	Replication Server 15.0 以前のバージョンへの複製では、Sybase データ型に varchar を使用します。Replication Server 15.1 以降のバージョンでは、Sybase データ型に opaque を使用します。
text	2 ³¹ - 1 文字までの可変長	text	2GB	
timestamp	データベース全体のユニークな番号	timestamp または varbinary	データベース全体のユニークな番号	Replication Server 15.0 以前のバージョンへの複製では、Sybase データ型に varbinary (8) を使用します。Replication Server 15.1 以降のバージョンへの複製では、Sybase データ型に timestamp を使用します。
tinyint	0 から 255 の整数値	tinyint	0 から 255 の整数値	
uniqueidentifier	Globally Unique Identifier	char	Globally Unique Identifier	Sybase の同等機能はない。char (38) にマップ。
varbinary	8000 バイトまでの可変長	varbinary	32K	

Microsoft SQL Server データ型	Microsoft SQL Server の長さ / 範囲	Sybase データ型	Sybase の長さ / 範囲	注意
varbinary(max)	2 ³¹ - 1 バイトまでの可変長	image	2GB	varbinary(max) データ型は、Microsoft SQL Server 以外のデータ・サーバに複写することはできません。
varchar	8000 文字までの可変長	varchar	32K	
varchar(max)	2 ³¹ - 1 文字までの可変長	text	2GB	varchar(max) データ型は、Microsoft SQL Server 以外のデータ・サーバに複写することはできません。

Replication Server 15.0 の符号なしデータ型のマッピング

Replication Server バージョン 15.0 以降では、符号なしデータ型がサポートされており、複写定義で指定することができます。

表 13 : 符号なし整数の複写定義データ型のマッピング

RepServer 15.0 符号なしデータ型	複写定義データ型
unsigned bigint	numeric (20)
unsigned int	numeric (10)
unsigned smallint	int
unsigned tinyint	tinyint

ntext データ型複写

Microsoft SQL Server は、リトル・エンディアンのバイト順序でダブルバイトの ntext データ型値を格納します。デフォルトで、ntext データのバイト順序は複写中にビッグ・エンディアンに変換されるため、一般的な「ネットワーク」バイト順序であるビッグ・エンディアンを使用してネットワークを介してデータを転送できます。

ターゲット・データベースも Microsoft SQL Server である場合、Microsoft SQL Server は送信したビッグ・エンディアン順序から Microsoft SQL Server で必要なリトル・エンディアン順序に複製したデータを自動的に変換しません。ntext データの Microsoft SQL Server (または、必要な変換を提供できない他のレプリケート・サーバ) への複製をサポートするには、**lr_ntext_byte_order** プロパティを使用してバイト順序を「強制的に」送信します。それには、レプリケート・データベースの要求を満たすため、必要に応じて、**big** (ビッグ・エンディアンの場合) または **little** (リトル・エンディアンの場合) を指定します。

lr_ntext_byte_order パラメータは、Microsoft SQL Server と Oracle に使用でき、2つの異なるデータベース型間および異なるプラットフォーム上に常駐するデータベース間での複製にとって特に重要です。たとえば、2つの Microsoft SQL Server データベース間で複製する場合、プライマリ・データベースとレプリケート・データベースのどちらもデータをリトル・エンディアンのバイト順序で格納します。これは、Microsoft SQL Server は Windows 上でのみ動作するからです。したがって、**lr_ntext_byte_order** パラメータは **little** に設定する必要があります。ただし、レプリケート・データベースが Microsoft SQL Server ではない場合は、バイト・オーダーを判断し、それに従って **lr_ntext_byte_order** パラメータを設定します。

注意： Microsoft SQL Server 向け Replication Agent のデフォルトの動作では、**ltl_big_endian_uni** 設定プロパティの定義に従って、Unicode データが強制的にビッグ・エンディアン順序になります。また、**lr_ntext_byte_order** 設定プロパティで Microsoft SQL Server バイト順序を正常に上書きできるようにするには、**ltl_big_endian_uni** 設定プロパティを **false** に設定してください (**lr_next_byte_order** プロパティを使用するたびに)。

ltl_big_endian_uni パラメータは、LTL を Replication Server に送信する前に、**unintext** データをリトル・エンディアンからビッグ・エンディアンに変換するかどうかを指定します。有効な値は、**true** および **false** です。このパラメータを設定する場合は、**lr_ntext_byte_order** パラメータの設定方法を理解する必要があります。**lr_ntext_byte_order** パラメータがレプリケート・データベースの正しいバイト順序を送信するように設定されている場合は、バイト順序が変更されないように **ltl_big_endian_uni** パラメータを **false** に設定してください。

ltl_big_endian_uni 設定プロパティと **lr_ntext_byte_order** 設定プロパティの間には、重要な違いがあります。デフォルトでは、**ltl_big_endian_uni** プロパティは **true** です。**ltl_big_endian_uni** プロパティが **true** のとき、Microsoft SQL Server 向け Replication Agent ではすべての Unicode データが確実にビッグ・エンディアン順序で送信されます。**ltl_big_endian_uni** プロパティが **false** のときは、Microsoft SQL Server 向け Replication Agent ではトランザクション・ログ・ファイルへのデータ格納に使用されているどのバイト順序でも Unicode データを送信できます。

lr_ntext_byte_order プロパティを使用すると、トランザクション・ログから読み込んだ Unicode データの結果は、トランザクション・ログ・ファイル内で通常どのような順序かに関係なく、要求したバイト順序になります。

プライマリ・データ・サーバからの複写定義の変更

プライマリ・データ・サーバから複写定義を変更することができます。

複写定義の変更前に複写システムをクワイース状態にする必要がないように、プライマリ・データ・サーバから Replication Server の alter replication definition コマンドを発行し、同時にプライマリ・データベース・オブジェクトへのスキーマ変更を加えることができます。複写定義への変更の伝播は、複写プロセスを停止しなくても、データ複写とともに自動的に調整されます。

Replication Server の alter replication definition コマンドをプライマリ・データ・サーバから発行するには、rs_send_repserver_cmd というストアド・プロシージャをプライマリ Microsoft SQL Server データベースに作成します。このプロシージャを作成するための SQL は、Replication Server 上の適切なコネクション・プロファイルに含まれています。コネクション・プロファイルのリストについては、Replication Server の admin show_connection_profiles コマンドを使用してください。

rs_send_repserver_cmd および alter replication definition Replication Server 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。

セキュリティの考慮事項

rs_send_repserver_cmd プロシージャをプライマリ・データ・サーバで呼び出すと、Replication Agent は対応する複写コマンド言語 (RCL) を直接 Replication Server に渡します。そのため、**rs_send_repserver_cmd** プロシージャについて誰に実行権限を割り当てるかを慎重に考慮し、環境とセキュリティ・ポリシーに適した権限の割り当てを行う必要があります。

制限事項

特定のデータ型のカラムを含むテーブルの複写定義を変更する場合、rs_send_repserver_cmd プロシージャを使用することはできません。

次のタイプがあります。

- nvarchar(max)
- varbinary(max)
- varchar(max)

注意： Replication Server でテーブル・レベルの複写定義を手動で変更した場合、Replication Agent で複写をサスペンドしてから開始し、Replication Agent によってキャッシュがクリアされ再表示されることを確認する必要があります。

Replication Serverset autocorrection コマンド

Replication Server の set autocorrection コマンドは、複写テーブルで欠けているか、重複しているローによって発生する障害を回避します。

set autocorrection コマンドは、各 update または insert オペレーションを delete、さらに insert に変換することによってマテリアライゼーション中に発生することのある不一致を修正します。

ra_set_autocorrection コマンドを使用することで、Replication Agent からプライマリ・データベース内の 1 つまたはすべてのマーク付けされたテーブルについて、オートコレクションを設定できます (『Replication Agent リファレンス・マニュアル』で説明)。Replication Server からオートコレクションを設定するには、複写定義で **set autocorrection** コマンドを使用します。これは、必ず Replication Server から行ってください。Replication Agent からでは複写定義のオートコレクション設定を変更できません。『Replication Server 管理ガイド』を参照してください。

計算カラム

あるテーブルに物理的に格納されている計算カラム (Microsoft SQL Server で PERSISTED とマーク付けされているカラム) を含んでいるテーブルは、複写するようマーク付けすることができ、これらのカラムは複写されます。

仮想計算カラム (あるテーブルに物理的に格納されていない計算カラム) を含んでいるテーブルは、複写するようマーク付けすることができますが、これらのカラムは複写されません。仮想計算カラムを含んでいるマーク付けされたテーブルについて、プライマリ・データベースとレプリケート・データベースとの間の整合性を保つには、仮想計算カラムを定義している式がプライマリ・データベースとレプリケート・データベースの両方で同じであることを確認します。

Microsoft SQL Server プライマリ・データベースの Replication Agent オブジェクト

Replication Agent は、複写作業用のオブジェクトをプライマリ・データベースで作成します。

Replication Agent オブジェクトを作成するには、init キーワードを指定して ra_admin コマンドを呼び出します。このコマンドを呼び出すと、Replication Agent ではプライマリ・データベースで作成または変更されたオブジェクトの SQL 文を含む SQL スクリプトが生成されます。このスクリプトは、RAX-15¥inst_name ¥scripts¥xlog ディレクトリの partinit.sql ファイルに保存されます。これらの

オブジェクトは、複製のマークをプライマリ・データベース・オブジェクトに付ける前に作成する必要があります。

注意：生成されたスクリプトは、参照時に使われるだけです。これらのスクリプトを手動で実行しても、プライマリ・データベースまたは Replication Agent を初期化することはできません。

『Replication Agent 管理ガイド』を参照してください。

Replication Agent オブジェクト名

Replication Agent は、複写作業用のオブジェクトをプライマリ・データベースで作成します。

Replication Agent のデータベース・オブジェクト名には、次の2つの変数があります。

- `prefix - ra_admin_instance_prefix` パラメータ (Replication Agent for Microsoft SQL Server または Replication Agent for UDB に使用) または `ra_admin_prefix` (Replication Agent for Oracle に使用) パラメータの 1～3 の文字列値を表します。デフォルトは `ra_` です。
- `xxx` - 英数字のカウンタを表します。この文字列は、データベース内でユニークな名前にするためにデータベース・オブジェクト名に追加されます。

`ra_admin_instance_prefix` パラメータの値は、すべての Replication Agent システム・オブジェクト名で使用されるプレフィクス文字列です。

`ra_admin_prefix_chars` パラメータの値は、`ra_admin_instance_prefix` で指定するプレフィクス文字列で使用できる非英数字のリストです。使用できる文字のリストはデータベース固有です。たとえば、Oracle では、データベース・オブジェクト名に使用できる非英数字は `$`、`#`、`_` のみです。

プライマリ・データベース内の Replication Agent トランザクション・ログ・コンポーネントの名前を表示するには、`ra_admin` コマンドを使用します。

オブジェクト名を設定する方法の詳細については、『Replication Agent 管理ガイド』を参照してください。

テーブル・オブジェクト

Replication Agent は、Microsoft SQL Server プライマリ・データベースにテーブル・オブジェクトを作成します。

これらのテーブルは、Replication Agent オブジェクトとみなされます。insert パーミッションと delete パーミッションは、DDL 隠しテーブルでのみ PUBLIC に付与されます。他のテーブルに対するパーミッションは与えられません。

表 14 : Replication Agent テーブル・オブジェクト

オブジェクト	名前
DDL 隠しテーブル	<i>prefixddl_trig_XXX</i>
インスタンス・テーブル	<i>prefixinstance_XXX</i>
オブジェクト・マーク付けテーブル	<i>prefixmarkObject_XXX</i>
オブジェクト確認テーブル	<i>prefixcheckObject_XXX</i>

Microsoft SQL Server システム・テーブル

Replication Agent は初期化時に、Microsoft SQL Server **sp_replicationdboption** ストアド・プロシージャを使用して、プライマリ・データベースにシステム・テーブルをいくつか作成します。

これらのテーブルは、**sp_replicationdboption** でも削除されます。その際は、Replication Agent **ra_admin deinit** コマンドが使用されます。これらのテーブルは直接変更しないでください。「**sp_replicationdboption**」の詳細については、Microsoft SQL Server のマニュアルを参照してください。

プロシージャ・オブジェクト

Replication Agent は、Microsoft SQL Server プライマリ・データベースにプロシージャ・オブジェクトを作成します。

次の表は、Replication Agent オブジェクトとみなされるプロシージャ・オブジェクトのリストです **sp_SybSetLogforReplTable** プロシージャ、

sp_SybSetLogforReplProc プロシージャ、および **sp_SybSetLogforLOBCol** プロシージャは、Microsoft SQL Server の **mssqlsystemresource** システム・データベースに作成されます。これらのプロシージャの **execute** パーミッションは **PUBLIC** に付与されますが、Replication Agent の **pds_username** ユーザだけがプロシージャを正常に実行できます。これは、**pds_username** ユーザだけが **select** パーミッションを **sys.syssschobjs** テーブルに対して付与されているためです。他のプロシージャの作成時に付与されるパーミッションはありません。

注意： 複写コンテキスト以外で実行された場合、ここに示したストアド・プロシージャに効力はありません。

表 15 : Replication Agent のプロシージャ・オブジェクト

オブジェクト	名前
オブジェクトのマーク付けまたはマーク付け解除	<i>prefixmark_XXX</i>
オブジェクトの確認	<i>prefixcheck_XXX</i>

オブジェクト	名前
最後にコミットされたトランザクションの ID の取得	<i>prefixlct_sql_XXX</i>
テーブルのマーク付けまたはマーク付け解除	sp_SybSetLogforReplTable
プロシージャのマーク付けまたはマーク付け解除	sp_SybSetLogforReplProc
LOB カラムのマーク付けまたはマーク付け解除	sp_SybSetLogforLOBCol

マーカ・オブジェクト

Replication Agent は、プライマリ・データベースにマーカ・オブジェクトを作成します。

以下のマーカ・プロシージャとマーカ隠しテーブルは、Replication Agent オブジェクトとみなされます。これらのオブジェクトの作成時に付与されるパーミッションはありません。

表 16 : Replication Agent のマーカ・オブジェクト

オブジェクト	名前
トランザクション・ログ・マーカ・プロシージャ	rs_marker_XXX
ダンプ・マーカ・プロシージャ	rs_dump_XXX
トランザクション・ログ・マーカ隠しテーブル	<i>prefixmarkersh_XXX</i>
ダンプ・マーカ隠しテーブル	<i>prefixdumpsh_XXX</i>

トリガ・オブジェクト

Replication Agent は、Microsoft SQL Server プライマリ・データベースにトリガ・オブジェクトを作成します。

表 17 : Replication Agent トリガ・オブジェクト

オブジェクト	名前
DDL コマンドの取得	<i>prefixddl_trig_XXX</i>
取得 create_table DDL コマンド	<i>prefixcreatetable_trig_XXX</i>

トランザクション・ログ管理

必要な唯一のトランザクション・ログ管理作業は、トランザクション・ログのバックアップとトランケーションです。

トランザクション・ログのバックアップとリストア

Replication Agent では、トランザクション・ログの自動バックアップとリストアをサポートしていません。その代わりに、お使いの Microsoft SQL Server ソフトウェア

に付属のデータベース・バックアップ・ユーティリティを使用して、トランザクション・ログを定期的にバックアップすることをおすすめします。

注意： Replication Agent では、リストアされたログからのトランザクションの再生をサポートしていません。

トランザクション・ログのトランケーション

Replication Agent では、自動と手動の両方のログ・トランケーションがサポートされています。

Replication Agent には自動トランザクション・ログ・トランケーションについてのオプションが2つあります。

- 指定した時間間隔に基づく定期的トランケーション
- Replication Agent がプライマリ Replication Server から新しい LTM ロケータ値を受信するたびに自動でトランケートする自動ログ・トランケーション機能をオフにすることもできます。デフォルトでは、自動ログ・トランケーションはオフです。

Replication Agent のログ・トランケーションを設定するには、次のガイドラインに従ってください。

- `ra_config` を使用して `truncation_type` 設定パラメータの値を設定することで、自動トランケーション・オプション(「なし」を含む)を指定できます。
時間間隔に基づきトランザクション・ログを自動的にトランケートするには、`ra_config` を使用して `truncation_interval` 設定パラメータの値を設定します。
- Replication Agent トランザクション・ログは、Replication Agent 管理ポートで `pdb_truncate_xlog` を呼び出すことによっていつでも手動でトランケートできます。
トランザクション・ログを特定の時点でトランケートするには、スケジューラユーティリティを使用して `pdb_truncate_xlog` コマンドを自動的に実行します。
- Microsoft SQL Server 向け Replication Agent は、プライマリ・データベース・ログをトランザクション単位でトランケートします。Microsoft SQL Server 向け Replication Agent は、Replication Server から LTM ロケータを受信した後、プライマリ・データベースに問い合わせ、トランケートできる最新のトランザクションの ID を取得します。そして、最新トランザクションの前のトランザクション・ログ領域を再使用可能としてマーク付けします。すると、Microsoft SQL Server はログ・レコードを再使用可能領域に書き込めるようになります。
- Microsoft SQL Server 内でのログ・トランケーションを制御するため、Replication Agent によって Microsoft SQL Server の `sp_repltrans` コマンドと `sp_repldone` コマンドが発行されます。これらのコマンドでは、Replication Agent ユーザに `db_owner` ロール・パーミッションがあることが必要とされます。

注意：Microsoft SQL Server では、`sp_repltrans` コマンドと `sp_repldone` コマンドによるログ・トランケーションの制御に 1 つのセッションのみ許可されます。Replication Agent がログ・トランケーション処理を制御している間は、これらのコマンドを使用しないでください。

これらのプロパティの詳細については、『Replication Agent リファレンス・マニュアル』を参照してください。トランケート方法の詳細については、『Replication Agent 管理ガイド』の「管理」を参照してください。

Microsoft SQL Server での Windows 認証の使用

Windows プラットフォーム上で Microsoft SQL Server 向け Replication Agent を実行するときは、ユーザを認証するための Windows 資格情報を使用して Microsoft SQL Server に接続するよう設定することもできます。

Replication Agent で Windows 認証を使用するよう設定するには、以下の手順に従います。

1. プライマリ Microsoft SQL Server で、Replication Agent を起動するユーザ `<ra_user>` を Windows 認証ユーザとして追加します (該当するユーザ・ドメインも含む)。`<ra_user>` をプライマリ・データベースに追加し、適切なパーミッションを付与します。詳細については、Microsoft SQL Server のマニュアルを参照してください。
2. Microsoft SQL Server 向け Replication Agent が実行されているマシンでは、`<domain>\<ra_user>` を Windows ユーザ・アカウントに追加します。ドメインが存在しない場合、`<ra_user>` だけを Windows ユーザ・アカウントに追加します
3. 同じマシンで、`sqljdbc_auth.dll` ファイルを Microsoft SQL Server JDBC ドライバの保管場所から Windows システム・パス上のディレクトリにコピーします。Microsoft SQL Server JDBC ドライバをインストールした場合、`sqljdbc_auth.dll` ファイルは次の場所にインストールされます。

```
<install_dir>\sqljdbc_<version>\<language>\auth\
```

注意：32 ビット・プロセッサでは、x86 フォルダにある `sqljdbc_auth.dll` ファイルを使用します。64 ビット・プロセッサでは、x64 フォルダにある `sqljdbc_auth.dll` ファイルを使用します。

4. 同じマシンで、`<ra_user>` としてログインし、Microsoft SQL Server 向け Replication Agent インスタンスを起動します。
5. Replication Agent にログインし、プライマリ Microsoft SQL Server に適切な値を使用して以下のパラメータを設定します。

```
ra_config pds_server_name, <server>  
ra_config pds_port_number, <port>
```

```
ra_config pds_database_name, <database>  
ra_config pds_username, <ra_user>  
ra_config pds_integrated_security, true
```

6. Replication Agent のマニュアルに従って、Replication Agent の設定と使用を続行します。

さまざまなマシン上での Replication Agent と Microsoft SQL Server のセットアップ

Replication Agent とプライマリ・データ・サーバを異なるコンピュータで実行します。

1. sybfilter ドライバをプライマリ Microsoft SQL Server と同じマシンにインストールし、このドライバを使用して Replication Agent がトランザクション・ログを読み込めるようにします。
2. プライマリ Microsoft SQL Server が実行されているマシンで、トランザクション・ログ・ファイルを格納しているドライブを共有して、それらのドライブが Replication Agent のインストール先マシンにマウントできるようにします。
3. プライマリ・データ・サーバが実行されているコンピュータと同じハードウェアおよびオペレーティング・システムのマシンに Replication Agent をインストールします。
4. Replication Agent と同じコンピュータに JDBC ドライバをインストールします。
5. Replication Agent マシンで、プライマリ Microsoft SQL Server のデータベース・トランザクション・ログ・ファイルを格納しているネットワーク・ドライブをマッピングします。Replication Agent がログ・ファイルを指すためには、`ra_devicepath` コマンドを使用します。

Replication Agent for UDB

Replication Agent for UDB に特有の Replication Agent の特性を確認します。

「Replication Agent for UDB」という用語は、Linux、Unix、および Windows サーバ用の IBM DB2 に配置されているプライマリ・データベース用にインストールおよび設定された Replication Agent ソフトウェアのインスタンスを指します。

注意： Replication Agent の基本的な機能とオペレーションの詳細については、『Replication Agent 管理ガイド』と『Replication Agent リファレンス・マニュアル』を参照してください。

IBM DB2 固有の考慮事項

以下の全般的な問題と考慮事項は、Replication Agent と Linux、Unix、および Windows サーバ用 IBM DB2 の併用に固有です。

サポート対象外のソフトウェア機能

以下の機能は Sybase 複写ではサポートされていません。

- DB2 データ定義言語 (DDL) コマンド
- DB2 ストアド・プロシージャ
- DB2 のロー圧縮
- Replication Server 並列 DSI
- Replication Server の `rs_init` ユーティリティ
- Replication Server `rs_subcomp` ユーティリティ
- Replication Server の自動マテリアライゼーション
- 他のベンダが複写している環境で複写を行う際の Replication Server

プライマリ・キーでの遅延更新の複写

ユニークなカラム・インデックスを持つテーブルでの更新は、従来の複写ではサポートされていないため、Replication Server はエラーを報告します。

ユニークなカラム・インデックスを持つテーブルの更新の複写は、サポートされていないため、Replication Server はエラーを報告します。たとえば、テーブル `t` はカラム `c` 上にユニークなインデックスを持ち、値は 1、2、3、4、および 5 です。このテーブルには単一の `update` 文が適用されます。

```
update t set c = c+1
```

従来の複写を使用すると、この文は以下のようになります。

Replication Agent for UDB

```
update t set c = 2 where c = 1
update t set c = 3 where c = 2
update t set c = 4 where c = 3
update t set c = 5 where c = 4
update t set c = 6 where c = 5
```

最初の更新で、値 c=2 をテーブルに挿入しようとしています。ところが、この値はテーブルに既に存在します。Replication Server は、エラー 2601 (重複キーの挿入の試み) を表示します。

ASEはこの問題を SQL 文機能で解決します。この機能により、更新を遅延させ逆順に実行することで、重複キー・エラーを回避します。ただし、Sybase 以外のデータ・サーバではこの機能は提供されていないので、ユニークなカラム・インデックスを持つ Sybase 以外のテーブルへの更新を複写しようとする、Replication Server DSI がクラッシュします。この問題を回避するには、ユニークなインデックス定義を拡充します。

サポート対象外のデータ型

以下のデータ型は Sybase 複写ではサポートされていません。

- ROWID
- XML
- ユーザ定義データ型

以下のデータ型は、レプリケート・データベースが IBM DB2 のときはサポートされません。

- BLOB
- CLOB
- DBCLOB
- LONG VARCHAR
- LONG VARGRAPHIC

Replication Agent for UDB での機能差

以下の Replication Agent 機能は、Replication Agent for UDB でユニークな動作をします。

Replication Agent の初期化

Replication Agent for UDB では、Replication Agent の他の実装と同じ Replication Agent の初期化機能とプライマリ・データベースのオブジェクト作成機能が用意されています。Replication Agent for UDB では、システム情報を格納するためのテーブルがプライマリ・データベースに少し作成されるにすぎません。

Replication Agent for UDB では、プライマリ・データベース内にストアド・プロシージャおよびトリガを作成しません。

Replication Agent for UDB では、UDB トランザクション・ログにアクセスする必要があるため、Replication Agent がプライマリ・データベースにアクセスするために使用するユーザ ID には、データベースでの **SYSADM** 権限または **DBADM** 権限のいずれかが必要です。これがない場合、**ra_admin init** コマンドがエラーを返します。このユーザ ID は、Replication Agent の **pds_username** 設定パラメータに格納されています。

テーブルを複製するようマーク付けする方法

Replication Agent for UDB では、Replication Agent の他の実装と同じテーブルの複製マーク付け機能とその解除機能が用意されています。ただし、Replication Agent for UDB では、プライマリ・データベース内にストアド・プロシージャおよびトリガを作成しません。

テーブルを複製するようマーク付けすると、Replication Agent for UDB は UDB **DATA CAPTURE** 属性を **DATA CAPTURE CHANGES** に設定して、テーブルを変更します。テーブルのマーク付けが解除されると、テーブルは元の **DATA CAPTURE** 属性に戻るよう変更されます。

注意： Replication Agent for UDB によって複製するようマーク付けされたテーブルの **DATA CAPTURE** 属性を手動で変更しないでください。そうすると、複製結果が悪影響を受ける可能性があります。

すべてのテーブルの同時マーク付けおよびマーク付け解除は不可

pdb_setreptable all, mark または **pdb_setreptable all, unmark** を使用してプライマリ・データベース内のすべてのテーブルから一度にマークを付けたり外したりすることは、Replication Agent for UDB ではサポートされていません。テーブルごとに個別にマークを付けたり外したりする必要があります。

使用できない機能

以下の Replication Agent 機能は、Replication Agent for UDB と併用できません。

- ストアド・プロシージャ複製 - ストアド・プロシージャの複製は、Replication Agent for UDB と併用できません。したがって、**pdb_setrepproc** コマンドはサポートされません。
- DDL 複製 - プライマリ・データベース内で実行されるシステム・プロシージャとデータ定義言語 (DDL) コマンドの複製は、サポートされていません。
- プライマリ・データ・サーバからの複製定義の変更 - これにはストアド・プロシージャ複製が関係しており、ストアド・プロシージャ複製はサポートされていません。
- 複製定義の自動作成 - これには Replication Agent の **rs_create_repdef** コマンドを使用する必要がありますが、このコマンドは Replication Agent for UDB で使用できません。

注意：これらの機能に関する Replication Agent コマンドを呼び出すと、エラーが表示されます。

参照：

- DB2 プライマリ・データベースの Replication Agent オブジェクト (133 ページ)

IBM DB2 要件

Linux、Unix、および Windows 用の IBM DB2 の以下の要件に従ってください。

- データベースのバージョンは、9.1、9.5、または 9.7 でなければなりません。
-
- UDB クライアント・インスタンスと UDB サーバ・インスタンスが別々のマシン上にある場合、そのクライアントとサーバの USB バージョンは同じでなければなりません。
- データベースには有効な JDK パスワードが設定されていなければなりません。JDK_PATH 設定パラメータに bin ディレクトリの 1 つ上のディレクトリの完全パスが含まれていなければなりません。bin ディレクトリには、java 実行可能ファイルが入っています。データベース・マネージャの JDK_PATH 設定を判断するには、次の DB2 コマンドを使用します。

```
get dbm cfg
```

注意：64 ビット IBM DB2 インスタンスには 64 ビット JDK、32 ビット DB2 インスタンスには 32 ビット JDK が必要です。

- Replication Agent が Linux または UNIX ホストにインストールされている場合、クライアントまたはサーバの 64 ビット DB2 インスタンスを設定してください。
- データベース LOGARCHMETH1 設定パラメータを LOGRETAIN または DISK:<path> に設定します。ここで、<path> は、ログのアーカイブ先のディレクトリです。これにより、循環ロギングの代わりにアーカイブ・ロギングが有効になります。LOGARCHMETH1 設定を判断するには、次の DB2 コマンドを使用します。

```
get db cfg for <db-alias>
```

- Windows システムで、DB2 接続 **autocommit** パラメータを有効にしてください (**autocommit=1**)。autocommit パラメータは、プライマリ・データベースの DB2 呼び出しレベル・インタフェース (CLI) 設定ファイルで指定されます。autocommit パラメータが有効でない場合、デッド・ロック問題が発生する可能性があります。CLI 設定ファイルのパスは、以下のとおりです。

```
%DB2DIR%¥sqllib¥db2cli.ini
```

ここで、%DB2DIR% は DB2 クライアントのインストール・パスです。

- Replication Agent をエラーなく初期化するため、データベースには以下の特性で作成されたテーブル領域が必要です。

- テーブル領域は、ユーザ・テンポラリ・テーブル領域でなければならない。デフォルトでは、データベース作成時にユーザ・テンポラリ・テーブル領域は作成されません。
- テーブル領域は、システム管理領域 (SMS) でなければならない。
- **PAGESIZE** パラメータは、8,192 (8 KB) 以上に設定されていなければならない。
- **pds_username** ユーザとして指定するユーザ ID には、プライマリ・データベース・トランザクション・ログにアクセスするための SYSADM または DBADM 権限のいずれかが必要です。
- DB2 環境変数をすべて設定してから、Replication Agent を起動してください。Replication Agent では、DB2 CLI ドライバを使用してプライマリ DB2 データベースに接続します。UNIX の場合、ドライバはオペレーティング・システムに応じて libdb2.so、libdb2.sl、または libdb2.a にあります。Windows の場合、DB2 ドライバは db2cli.dll にあります。Replication Agent は、トランザクション・ログの読み込みに DB2 API ライブラリも使用します。したがって、Replication Agent が実行時に正しいドライバと API ライブラリをロードできるように、ライブラリ・パス環境変数を設定してください。

UNIX と Linux の場合、64 ビット・バージョンのライブラリは \$HOME/sql1lib/lib64 ディレクトリにあります。ただし、\$HOME は DB2 インスタンス所有者のホーム・ディレクトリです。Replication Agent が Linux または UNIX 上にインストールされている場合、ライブラリ・パス環境変数は 64 ビット・ライブラリをポイントする必要があります。Windows の場合、ライブラリ・パス環境変数は 32 ビット・ライブラリをポイントする必要があります。

ライブラリ・パス環境変数の正確な名前は、オペレーティング・システムによって異なります。Linux の場合、ライブラリ・パス変数は名前付き LD_LIBRARY_PATH です。Windows の場合、ライブラリ・パス変数は名前付き PATH です。

Windows 上では、DB2 サーバまたはクライアント・インストール環境が、必要な環境変数をすべて設定します。UNIX または Linux 上では、DB2 db2cshrc (C シェルの場合) または db2profile (Bourne と Korn シェルの場合) スクリプトを取得してから、Replication Agent を起動してください。これらのスクリプトは、\$HOME/sql1lib にあります。ただし、\$HOME は DB2 インスタンス所有者のホーム・ディレクトリです (DB2 クライアントまたはサーバ・インスタンスの場合)。

さまざまなマシン上での Replication Agent と DB2 Server

Replication Agent for UDB ソフトウェアが、DB2 サーバと異なるホスト・マシンにインストールされている場合、Replication Agent と同じホスト・マシンに DB2 Administration Client をインストールしてください。

Replication Agent for UDB ソフトウェアが DB2 サーバと同じホスト・マシンにインストールされている場合、個別の DB2 Administration Client は必要ありません。

Replication Agent for UDB ソフトウェアが、Linux または UNIX 上にインストールされている場合、64 ビット DB2 クライアント・インスタンスを設定してください。Windows 上では、32 ビット DB2 クライアント・インスタンスを設定します。

DB2 接続

Windows システムでは、Replication Agent for UDB のコネクティビティを設定するときは、DB2 Administration Client で DB2 Universal Database JDBC データ・ソースを設定してから、その DB2 Universal Database JDBC データ・ソースについて指定されているデータベース名とデータベース・エイリアスを使用してください。

UNIX システムでは、ODBC を使用する代わりに、DB2 のノードとプライマリ・データベースをカタログするだけです。Replication Agent の `pds_datasource_name` パラメータをデータベース・エイリアスに設定します。また、`pds_host_name` と `pds_host_number` も設定します。

DB2 クライアントからのリモート TCP/IP ノードのカタログ作成

リモート DB2 クライアント・ノードのカタログを作成します。

1. DB2 インスタンス所有者としてログインします。

環境スクリプトを実行することで、ログインにより DB2 環境変数が設定されます。これらのスクリプトを以下のように手動で実行することもできます。

Korn シェルで、`db2profile` ファイルを取得します。

```
. $HOME/sqlllib/db2profile
```

C シェルで、`db2cshrc` ファイルを取得します。

```
source $HOME/sqlllib/db2cshrc
```

ここで、`$HOME` は DB2 インスタンス所有者のホーム・ディレクトリです。

2. `db2` コマンドを入力して、DB2 コマンドライン・プロセッサを起動します。
3. DB2 プロンプトでこのコマンドを使用して、リモート TCP/IP ノードのカタログを作成します。

```
catalog tcpip node MYNODE remote MYHOST server XXXX
```

ただし、*MYNODE*はノード名、*MYHOST*はデータ・サーバのホスト名またはIPアドレス、および *XXXX*はデータ・サーバのポート番号です。

4. カタログ・エントリを確認します。

```
list node directory
```

DB2 から次のような結果が返されます。

```
Node 1 entry:
Node name           = MYNODE
Comment             =
Directory entry type = LOCAL
Protocol            = TCPIP
Hostname            = MYHOST
Service name        = XXXX
```

DB2 クライアントからのプライマリ・データベースのカタログ作成 プライマリ・データベースのカタログを作成します。

1. DB2 プロンプトでこのコマンドを使用して、プライマリ・データベースのカタログを作成します。

```
catalog database MYDB as MYDB_ALIAS at node MYNODE
```

ただし、*MYDB*はデータベース名、*MYDB_ALIAS*はデータベースのエイリアス、および *MYNODE*はノード名 (**catalog tcpip node** コマンドで使用) です。

2. カタログ・エントリを確認します。

```
list database directory
```

DB2 から次のような結果が返されます。

```
System Database Directory

Number of entries in the directory = 1

Database 1 entry:

Database alias      = MYDB_ALIAS
Database name       = MYDB
Node name           = MYNODE
Database release level = b.00
Comment             =
Directory entry type = Remote
```

pds_datasource_name の設定

Replication Agent の **pds_datasource_name** パラメータを設定します。

1. Replication Agent で、**pds_datasource_name** をデータベース・エイリアスに設定します。

```
ra_config pds_datasource_name, MYDB_ALIAS
```

ただし、*MYDB_ALIAS*はプライマリ・データベースのカatalog作成時に使用されたデータベース・エイリアスです。

2. また、以下の Replication Agent パラメータも設定します。

- **pds_database_name**
- **pds_username**
- **pds_password**
- **pds_host_name**
- **pds_port_number**

『Replication Agent リファレンス・マニュアル』を参照してください。

Replication Agent for UDB 接続パラメータ

Replication Agent for UDB と DB2 サーバとの間のコネクションを設定するには、以下の Replication Agent 設定パラメータが必要です。

- **pds_username** – DBADM 権限が必要。たとえば、repuser。
- **pds_password** – ユーザ ID (**pds_username**, で指定) 用。たとえば、repuser_pwd。
- **pds_database_name** – DB2 データベース名。たとえば、TEST_DB1。
- **pds_datasource_name** – DB2 データ・ソース名。たとえば、TEST_DB1_DS。
- **pds_host_name** – プライマリ DB2 データ・サーバがあるホストの名前。
- **pds_port_number** – プライマリ DB2 データ・サーバのポート番号。

ログでの再配置

Replication Agent は、プライマリ Replication Server から受信した LTM ロケータの値を使用して、DB2 トランザクション・ログ内に Replication Server に送信するトランザクションがないかどうかを調べる作業の開始点を判断します。

Replication Agent for UDB では、LTM ロケータ値を以下のように使用します。

- Replication Server から受信した LTM ロケータの値および Replication Agent によって格納された LTM ロケータの値が両方ともゼロ (0) である場合、Replication Agent は Log Reader コンポーネントを DB2 トランザクション・ログの末尾に配置します。

警告！ 両方の LTM ロケータ値がゼロである場合、2つの特定の条件によりデータ損失が発生することがあります。

- Replication Agent の Log Reader コンポーネントが、*Replicating* ステータスに入るときは、非同期的に入ります。 **resume** コマンドを呼び出した後にプロンプトが表示された場合、Log Reader コンポーネントは *Replicating* ステータスに入ることと、自己をログの末尾に配置することをまだ完了していない可能性があります。 **resume** コマンドからプロンプトが返された直後にテーブルをマーク付けすると、そのマーク情報を格納したレコードがログに書

き込まれてから、Log Reader コンポーネントが自己の配置を完了する可能性があります。その場合、Log Reader コンポーネントは、そのレコードを失うため、それ以降はそのテーブルのデータを複製しなくなります。この問題を回避するには、**resume** コマンドを呼び出してからしばらく待ってから、テーブルを複製のためにマーク付けします。

- テーブルを複製のためにマーク付けし、データをテーブルに挿入してから、複製を再開する場合、Replication Agent と Replication Server の LTM ロケータの値がゼロ (複製の開始時と同様に) だと、データは複製されません。この問題が発生するのは、両方の LTM ロケータの値がゼロのときに複製を再開すると、Log Reader コンポーネントがログの末尾に再配置されて、それ以前のトランザクションがスキップされてしまうからです。Replication Agent と Replication Server の LTM ロケータの値がゼロのときにこの問題を回避するには、**resume** コマンドを発行してから、テーブルを複製のためにマーク付けします。
-
- Replication Server から受信した LTM ロケータの値および Replication Agent によって格納された LTM ロケータの値が両方ともゼロ (0) ではない場合、Replication Agent は Replication Server から受信した LTM ロケータの値を使用して最も古いオープン・トランザクションの開始位置を判断し、Log Reader コンポーネントを DB2 トランザクション・ログ内のその位置に配置します。
 - Replication Server から受信した LTM ロケータの値が 0 (ゼロ) で、Replication Agent によって格納された LTM ロケータの値がゼロではない場合、Replication Agent は自己が格納した LTM ロケータの値を使用して最も古いオープン・トランザクションの開始位置を判断し、Log Reader コンポーネントを DB2 トランザクション・ログ内のその位置に配置します。

Replication Agent for UDB の動作

Replication Agent の以下の問題は、Replication Agent for UDB に特有のものです。

LTM ロケータがゼロのときに、再開直後にテーブルをマーク付け

Replication Agent インスタンスが *Replicating* ステータスに入ると、Log Reader コンポーネントはプライマリ・データベース・トランザクション・ログを読み込み、オリジン・キュー ID の値を使用して読み込みを開始するログ内の位置を判断します。LTM ロケータの値が 0 (ゼロ) の場合、Log Reader はログの末尾から読み込みを開始します。

Log Reader オペレーションは非同期的なので、Replication Agent インスタンスは **resume** コマンドの後、Log Reader が起動プロセスを完了する前に、オペレーティング・システムのプロンプトに戻る可能性があります。**pdb_setreptable** コマンドをすぐに呼び出して複製するようテーブルをマーク付けする (**resume** コマンドが返した後に) と、Log Reader がログの末尾を見つける前に、マーク・オブジェクト

ト・エントリがトランザクション・ログ内に配置される可能性があります。この場合、Log Reader はマーク・テーブル・エントリを逃すので、テーブルのマーク付けに失敗します。

この問題を回避するには、**resume** コマンドを呼び出した後 5 ～ 10 秒待ってから **pdb_sreptable** を呼び出して、テーブルを複写のためにマーク付けします。

DB2 FORCE APPLICATION コマンド

DB2 FORCE APPLICATION コマンドは、データ・サーバにアプリケーションとのコネクションを削除させます。**FORCE APPLICATION ALL** は、データ・サーバにすべてのアプリケーションとのコネクションを削除させます。

FORCE APPLICATION が呼び出されて、Replication Agent アプリケーション・ハンドルまたは **ALL** キーワードのいずれかが指定されると、データ・サーバは Replication Agent インスタンスとのコネクションを削除します。この場合、Replication Agent は DB2 エラー・コード -30081 を受け取り、リカバリできないので、Replication Agent インスタンスは自己を停止します。

この状況を回避するには、Replication Agent の **shutdown** コマンドを呼び出してから、**FORCE APPLICATION** を使用します。

バッファ・サイズの読み込み

Replication Agent for UDB の Log Reader コンポーネントは、**lr_read_buffer_size** パラメータの値を使用して、各スキャン中にトランザクション・ログから読み込まれる最大バイト数を決定します。Log Reader は、バイトを読み込むので、読み込まれるバイトを格納するためのバッファが必要です。

常にうまくいく最小バッファ・サイズを突き止めるのは、簡単ではありません。**lr_read_buffer_size** の値の範囲は、10,000 ～ 2,147,483,647 です。

読み込みバッファ・サイズが小さすぎて 1 つのオペレーションを読み込めない場合、Replication Agent は *Admin* ステータスに入り、Log Reader コンポーネントは停止し、DB2 -2650 エラーをレポートします。ただし、バッファ・サイズが不十分な場合だけでなく、一般的な通信エラーもこのエラー・メッセージの対象に含まれます。

LOB 複写

LOB カラムに対して複写が有効に設定されている場合、Replication Agent は **prefixblob_columns_** テーブルにエントリを作成して、そのカラムの複写をサポートします。

Replication Agent が LOB カラムに影響を及ぼすトランザクションを処理するとき、LOB データはその取り得るサイズのために、トランザクション・ログに格納されない可能性があります。その代わりに、Replication Agent の Log Reader コンポーネン

トは、トランザクション処理時に LOB データをプライマリ・データベースから直接読み込みます。

GRAPHIC カラムまたは VARGRAPHIC カラムも含む、マーク付けされたテーブルの大規模オブジェクト・カラムを複製するには、Replication Agent for UDB はそのテーブルに GRAPHIC カラムも VARGRAPHIC カラムも含まないプライマリ・キーがあることを必要とします。たとえば、GRAPHIC データまたは VARGRAPHIC データを MYTABLE から複製するには、COL1 をプライマリ・キーとして定義してください。

```
CREATE TABLE MYTABLE
(COL1 SMALLINT NOT NULL,
 COL2 GRAPHIC (127),
 COL3 VARGRAPHIC (250),
 PRIMARY KEY (COL1))
IN DATABASE MYDB;
```

LOB カラムについて複製の有効または無効を指定する方法については、『Replication Agent 管理ガイド』を参照してください。

トランザクションの整合性と LOB データ

トランザクションの複製の際に Replication Agent によって LOB カラムが処理される方法のために、トランザクションの整合性が損なわれる可能性があります。たとえば、2つのトランザクションが LOB カラムのデータを変更し、Log Reader が 2 番目のトランザクションがコミットされるまで最初のトランザクションを処理しない場合、LOB データがプライマリ・データベースから読み込まれると、そのデータの値は 2 番目のトランザクションの結果となります。この場合、最初のトランザクションの LOB データはレプリケート・データベースにまったく送信されません。Log Reader によって 2 番目のトランザクションが処理されると、プライマリ・データベースとレプリケート・データベースが再び同期されますが、最初のトランザクションを処理してから 2 番目のトランザクションが処理されるまでの間は、レプリケート・データベースには元のトランザクションと一致するデータが含まれません。

この問題が発生するのは、一連のトランザクションによって LOB カラムが複数回変更された場合のみです。この問題が存在する期間は、複製システム全体の速度が遅い場合や、複製システムのコンポーネントに障害が発生した場合は、長くなることがあります。LOB カラムを変更する最後のトランザクションがレプリケート・サイトで処理されると同時に、問題は解消されます。

データベース・オブジェクト名の大文字と小文字

データベース・オブジェクト名は、複製定義に指定したものと同一フォーマットでプライマリ Replication Server に送信する必要があります。そうしないと、複製に失敗します。たとえば、複製定義ではテーブル名がすべて小文字で指定されて

いる場合、Replication Agent がプライマリ Replication Server にこのテーブル名を送信する際は、すべて小文字で表示されている必要があります。

Replication Agent がプライマリ Replication Server にデータベース・オブジェクト名を送信するときの大文字と小文字の区別を制御するには、`lcl_character_case` 設定パラメータを次のいずれかの値に設定します。

- `asis` – (デフォルト) データベース・オブジェクト名は、プライマリ・データ・サーバに格納されているものと同じフォーマットで Replication Server に渡されます。
- `lower` – データベース・オブジェクト名は、プライマリ・データ・サーバにどのように格納されているかにかかわらず、すべて小文字で Replication Server に渡されます。
- `upper` – データベース・オブジェクト名は、プライマリ・データ・サーバにどのように格納されているかにかかわらず、すべて大文字で Replication Server に渡されます。

DB2 サーバでは、データベース・オブジェクト名はすべて大文字で格納されます。

オリジン・キュー ID のフォーマット

トランザクション・ログの各レコードは、64 の 16 進文字 (32 バイト) で構成されるオリジン・キュー ID で識別されます。オリジン・キュー ID のフォーマットは Replication Agent インスタンスによって決定され、プライマリ・データベースの種類によって異なります。

表 18 : Replication Agent for DB2 のオリジン・キュー ID のフォーマット

文字	バイト	説明
0-3	2	データベース世代 ID
4-19	8	オペレーション・シーケンス番号
20-35	8	トランザクション ID
36-51	8	最も古いアクティブなトランザクションの最初のオペレーション・シーケンス番号
52-55	2	オペレーション・タイプ (begin = 0、data/LOB = 1、commit/rollback = 7FFF)
56-59	2	LOB シーケンス ID
60-63	2	未使用

DB2 データ型の互換性

Replication Agent for UDB は、トランザクションを処理し、データをプライマリ Replication Server に渡します。プライマリ Replication Server は、複写定義に指定されているデータ型のフォーマットを使用して、Replication Agent for UDB からデータを受信します。

次の表に、DB2 データ型から Sybase データ型へのデフォルトの変換を示します。この表内の各データ型に対し、2 番目のカラムの長さは次のように記述されています。

- 文字データ型 – 最大バイト数。
- グラフィック・データ型 – 最大文字数。
- 数値データ型 – 最小値から最大値までの範囲。
- 時間データ型 – 最も古い時間から最新の時間までの範囲。

表 19 : DB2 から Sybase へのデフォルトのデータ型マッピング

DB2 データ型	DB2 の長さ／範囲	Sybase データ型	Sybase の長さ／範囲	注意
BIGINT	-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807	bigint	10 ⁻³⁸ ~ 10 ³⁸ 、38 の有効桁数	
BLOB	可変長、2GB、バイナリ・データ	image	2GB	
CHAR	254 バイト	char	32K	
CHAR FOR BIT DATA	254 バイト、バイナリ・データ	binary	32K	
CLOB	可変長、2GB、文字データ	text	2GB	

DB2 データ型	DB2 の長さ／範囲	Sybase データ型	Sybase の長さ／範囲	注意
DATE	0001-01-01 から 9999-12-31 まで	char、 date、または datetime	32K (char)	データ型を交換するには、 Replication Server の異機種データ型サポート (HDS) 機能を使用してください。
DBCLOB	可変長、2GB、ダブルバイト文字データ	unitext または image	2GB	Replication Server 15.0 以降では、 DBCLOB は unitext に マップします。以前のバージョンの Replication Server では、 DBCLOB は image にマップ します。
DECFLOAT (16)	8 バイト、 -9.9999999999999999 $9999 \times 10^{384} \sim -1.0 \times$ 10^{-383} および $1.0 \times$ $10^{-383} \sim$ 9.9999999999999999 $\times 10^{384}$	float	精度と範囲は C の double データ型、約 16 有効桁数 に対応	精度が低下する 場合があります。

DB2 データ型	DB2 の長さ／範囲	Sybase データ型	Sybase の長さ／範囲	注意
LONG VAR- GRAPHIC	16,350 文字、ダブルバイト文字データ	unitext または image	2GB	Replication Server 15.0 以降では、LONG VARGRAPHIC は unitext にマップします。以前のバージョンの Replication Server では、LONG VAR-GRAPHIC は image にマップします。
NUMERIC (DECIMAL の同意語)				「DECIMAL」を参照してください。
REAL	$-3.402^{38} \sim 3.402^{38}$	decimal	$10^{-38} \sim 10^{38}$ 、 38 の有効桁数	
SMALLINT	-32,768 ~ 32,767	smallint	-32,768 ~ 32,767	
TIME	00:00:00 ~ 24:00:00	char、 time、または datetime	32K (char)	
TIMESTAMP	0001-01- 01-00.00.00.000000 ~ 9999-12- 31-24.00.00.000000	char または datetime	32K (char)	データ型を交換するには、Replication Server の異機種データ型サポート (HDS) 機能を使用してください。
VARCHAR	32,672 バイト	varchar	32K	

DB2 データ型	DB2 の長さ／範囲	Sybase データ型	Sybase の長さ／範囲	注意
VARCHAR FOR BIT DATA	32,672 バイト、バイナリ・データ	varbinary	32K	
VARGRAPHIC	16,336 文字、ダブルバイト文字データ	univarchar	32K	

Replication Server 15.0 の符号なしデータ型のマッピング

Replication Server バージョン 15.0 以降では、符号なしデータ型がサポートされており、複写定義で指定することができます。

表 20 : 符号なし整数の複写定義データ型のマッピング

RepServer 15.0 符号なしデータ型	複写定義データ型
unsigned bigint	numeric (20)
unsigned int	numeric (10)
unsigned smallint	int
unsigned tinyint	tinyint

DECFLOAT データ型の互換性

Replication Agent for UDB は、DECFLOAT データ型の複写をサポートしています。精度は 16 桁と 34 桁の両方がサポートされています。

Replication Agent for UDB が DECFLOAT カラムを DB2 プライマリ・データベースから DECFLOAT 型または同等の型をサポートしていないレプリケート・データベースに複写する場合、Replication Agent は DECFLOAT 型を FLOAT 型にマップします。その結果、精度が低下する場合があります。

数値のほかに、DECFLOAT カラムには Replication Agent がサポートしていない特別値が含まれている場合があります。これらの値には正と負の INFINITY、NAN、SNAN があります。これらの値の複写はサポートされていません。このような場合、Replication Agent ではカラムが NULL 入力可能であれば NULL、カラムが NULL 入力不可であれば 0.0 に特別値を複写します。

XML データ型の互換性

Replication Agent for UDB では XML データ型の複写はサポートしていません。

XML カラムを含むテーブルをマークしようとする、Replication Agent によってエラーが報告されます。XML カラムを含むテーブルは **force** オプション (**pdb_setreptable** コマンド) を使用してマークできますが、このカラムは複写されません。

Replication Server set autocorrection コマンド

Replication Server の **set autocorrection** コマンドは、複写テーブルで欠けているか、重複しているローによって発生する障害を回避します。

set autocorrection コマンドは、各 update または insert オペレーションを delete、さらに insert に変換することによってマテリアライゼーション中に発生することのある不一致を修正します。

ra_set_autocorrection コマンドを使用することで、Replication Agent からプライマリ・データベース内の 1 つまたはすべてのマーク付けされたテーブルについて、オートコレクションを設定できます (『Replication Agent リファレンス・マニュアル』で説明)。Replication Server からオートコレクションを設定するには、複写定義で **set autocorrection** コマンドを使用します。これは、必ず Replication Server から行ってください。Replication Agent からでは複写定義のオートコレクション設定を変更できません。『Replication Server 管理ガイド』を参照してください。

長い識別子

Replication Agent for UDB は、UDB 9.5 の長い識別子、つまり最高 128 バイトのオートソライゼーション ID、カラム、スキーマ名をサポートしています。

UDB 9.5 の長い識別子をサポートするには、Replication Agent インスタンスをバージョン 15.0、15.1、または 15.2 から 15.5 以降に移行して、Replication Agent のシステム・テーブルの変更を反映させる必要があります。また、長い識別子を含むテーブルを複写するには、DB2 を以前のバージョンから 9.5 にアップグレードするときに Replication Agent インスタンスも移行する必要があります。

参照：

- Replication Agent for UDB のアップグレードおよび移行手順 (143 ページ)

圧縮

Replication Agent for UDB は、値の圧縮、つまり **VALUE COMPRESSION** 句を使用して作成したテーブルと、ローの圧縮をサポートしています。

DB2 プライマリ・データベースの Replication Agent オブジェクト

Replication Agent は、複写作業用のオブジェクトをプライマリ・データベースで作成します。Replication Agent では、DB2 サーバが管理しているネイティブ・データベースのトランザクション・ログも使用して、プライマリ・データベース内のトランザクションを複写のために取り込みます。

Replication Agent オブジェクトを作成するには、`init` キーワードを指定して `ra_admin` コマンドを呼び出します。このコマンドを呼び出すと、Replication Agent ではプライマリ・データベースで作成または変更されたオブジェクトの SQL 文を含む SQL スクリプトが生成されます。このスクリプトは、`RAX-15_¥inst_name ¥scripts¥xlog` ディレクトリの `create.sql` ファイルに保存されます。これらのオブジェクトは、複写のマークをプライマリ・データベース・オブジェクトに付ける前に作成する必要があります。

注意： JAR ファイルは `ra_admin init` コマンドの実行時にインストールされます。`ra_admin deinit` コマンドは、プライマリ・データベースから JAR ファイルを削除します。`ra_admin deinit` コマンドは、Replication Agent の再初期化前に発行する必要があります。

『Replication Agent 管理ガイド』を参照してください。

参照：

- Java プロシージャ・オブジェクト (135 ページ)

Replication Agent オブジェクト名

Replication Agent は、複写作業用のオブジェクトをプライマリ・データベースで作成します。

Replication Agent のデータベース・オブジェクト名には、次の 2 つの変数があります。

- `prefix-ra_admin_instance_prefix` パラメータの 1～3 の文字列値を表します (デフォルトは `ra_`)。
- `xxx` - 英数字のカウンタを表します。この文字列は、データベース内でユニークな名前にするためにテーブル名に追加されます。

`ra_admin_instance_prefix` の値は、すべての Replication Agent システム・オブジェクト名で使用されるプレフィクス文字列です。

この値がプライマリ・データベースの既存のデータベース・オブジェクト名と競合する場合は、**ra_admin_instance_prefix** の値を **ra_config** コマンドを使用して変更できます。

注意： Replication Agent は、**ra_admin_instance_prefix** の値を使用して、プライマリ・データベース内のオブジェクトを検出します。Replication Agent オブジェクトを作成した後で **ra_admin_instance_prefix** の値を変更した場合、Replication Agent インスタンスでは古いプレフィクスを使用するオブジェクトを見つけられません。

ra_admin コマンドを使用すると、プライマリ・データベース内の Replication Agent オブジェクトの名前を表示できます。

複製オブジェクト名を設定する方法の詳細については、『Replication Agent 管理ガイド』を参照してください。

テーブル・オブジェクト

Replication Agent は、DB2 プライマリ・データベースにテーブル・オブジェクトを作成します。

これらのテーブルは、Replication Agent オブジェクトとみなされます。これらのテーブルの作成時に付与されるパーミッションはありません。これらのすべてのテーブルには 1 つ以上のインデックスがあり、複数のインデックスを持つものもあります。

表 21 : Replication Agent のテーブル

テーブル	データベース名
アークティクル・テーブル	<i>prefixvarticles_XXX</i>
LOB カラム・テーブル	<i>prefixvblob_columns_XXX</i>
rs_dump 隠しテーブル	<i>prefixdumpsh_XXX</i>
Force Record テーブル	<i>prefixforce_record_XXX</i>
マーク付けされたオブジェクトのテーブル	<i>prefixvmarked_objs_XXX</i>
rs_marker 隠しテーブル	<i>prefixmarkersh_XXX</i>
Proc-Active テーブル	<i>prefixproactive_XXX</i>
Log Admin ワーク・テーブル	<i>prefixrawork_XXX</i>
システム・テーブル	<i>prefixxlog_system_XXX</i>

Java プロシージャ・オブジェクト

Replication Agent は、Java プロシージャ・オブジェクトを DB2 プライマリ・データベースに作成します。

Replication Agent for UDB により、SYBRAUJAR.jar と SYBTRUNCJAR.jar が次のディレクトリにインストールされます。

- Windows では、ファイルは \$DB2DIR/SQLLIB/FUNCTION/jar/ *pds_username* にインストールされます。ここで、\$DB2DIR は DB2 インストールのパスで、*pds_username* は **pds_username** の値です。
- UNIX では、ファイルは \$HOME/sqlllib/function/jar/ *pds_username* にインストールされます。ここで、\$HOME は DB2 インスタンス所有者のホーム・ディレクトリで、*pds_username* は **pds_username** の値です。

注意： 複数の Replication Agent インスタンスが 1 つの DB2 サーバに設定されている場合、つまりデータベースごとに 1 つの Replication Agent インスタンスが存在する場合は、ユニークなプライマリ・データベース・ユーザ名を各 Replication Agent インスタンスの **pds_username** 設定パラメータに指定する必要があります。これは、これらの JAR ファイルのインストールとアンインストールに必要になります。

これらの JAR ファイルは複数の Java プロシージャを UDB プライマリ・データベースに実装します。これらのファイルはログ・トランケーションにおいて作成および使用されます。

表 22：トランケーション用の Java プロシージャ

プロシージャ	データベース名
現在の LSN が格納されたログ・ファイルの名前を取得する	<i>prefixget_log_name_</i>
get_log_name Java クラスのバージョンを取得する	<i>prefixget_version_str_</i>
データベース・ログ・ファイルまたはアーカイブ・ログ・ディレクトリのファイルをトランケートする	<i>prefixtrunc_log_files_</i>
trunc_log_files Java クラスのバージョンを取得する	<i>prefixget_trunc_ver_str_</i>

Replication Agent オブジェクトの名前の検索

Replication Agent インスタンスは、そのデータベース・オブジェクトの名前を生成します。これらのオブジェクトの実際の名前を見つけるには、**ra_admin** コマンドを使用します。

プライマリ DB2 データベース内の Replication Agent オブジェクトの生成名を取得するには、次の操作を実行します。

Replication Agent 管理ポートで、キーワードを指定せずに **ra_admin** コマンドを呼び出します。

```
ra_admin
```

ra_admin コマンドによって、プライマリ・データベース内のオブジェクトのリストが返されます。

マーク付けされたオブジェクトのテーブル

Replication Agent オブジェクトの 1 つに *marked objects table* があります。マーク付けされたオブジェクト・テーブルには、プライマリ・データベース内でマーク付けされている各テーブルのエントリが含まれています。

マーク付けされた各テーブル・エントリには、以下が含まれています。

- マーク付けされたプライマリ・オブジェクト (テーブル) の名前
- プライマリ・オブジェクトの複写済みの名前
- プライマリ・オブジェクトのタイプ (テーブルのみ、Replication Agent for UDB)
- プライマリ・オブジェクトの“複写有効”フラグ
- プライマリ・オブジェクトの所有者
- “所有者送信”フラグ
- プライマリ・オブジェクトのテーブルスペース ID
- プライマリ・オブジェクトのテーブル ID
- “日時変換”フラグ
- テーブルの **DATA CAPTURE** 属性の元の値
- オートコレクション・フラグ

トランザクション・ログのトランケーション

Replication Agent では、自動と手動の両方のログ・トランケーションがサポートされています。

Replication Agent には自動トランザクション・ログ・トランケーションについてのオプションが 2 つあります。

- 指定した時間間隔に基づく定期的トランケーション
- Replication Agent がプライマリ Replication Server から新しい LTM ロケータ値を受信するたびに自動でトランケートする

Replication Agent のログ・トランケーションを設定するには、次のガイドラインに従ってください。

- DB2 のトランザクション・ログはすべてデータ・サーバを經由して管理されます。Replication Agent for UDB は、アクティブなログ・ディレクトリまたはアーカイブ・ログ・ディレクトリのトランザクション・ログをトランケートするよ

うに設定できます。**LOGARCHMETH1**を使用したDB2のアーカイブ処理を有効にした場合は、**LOGARCHMETH2** DB2 設定パラメータを設定して2番目のアーカイブ・ロケーションを設定することもできます。DB2は2つのディレクトリにログをアーカイブするようになります。その後、これらのディレクトリのどちらかの処理済みアーカイブを自動的にトランケートするように Replication Agent を設定できます。

pdb_archive_path を使用すると、**LOGARCHMETH1** または **LOGARCHMETH2** によって指定されているロケーションを指し示すことができます。

警告！ **pdb_archive_path** を設定せずにトランケーションを有効にした場合、Replication Agent は DB2 の **prune** コマンドを使用して、不要になったプライマリ・データベースのログ・ファイルをアクティブ・ログ・ディレクトリから削除します。アクティブ・ディレクトリは DB2 のリカバリに使用されるので、アクティブ・ディレクトリを指すように **pdb_archive_path** を設定することや、最初に **pdb_archive_path** を設定せずにトランケーションを有効にすることは避けることをおすすめします。

- 不要になったアーカイブを Replication Agent が削除するように設定するには、**pdb_archive_remove** を true に設定します。

注意： デフォルトでは、**pdb_archive_remove** は false に設定されます。

pdb_archive_path を設定してから、**pdb_archive_remove** を true に設定してください。

- 自動トランケーションを有効にするには、**truncation_type** を interval に設定し、**truncation_interval** を 0 (ゼロ) より大きい値に設定します。それによって、指定した間隔でログ・ファイルが削除されます。あるいは、**truncation_type** を locator_update に設定します。それによって、Replication Agent がプライマリ Replication Server から新しい LTM ロケータ値を受け取るたびに、トランケーションが行われます。
- Replication Agent トランザクション・ログは、Replication Agent 管理ポートで **pdb_truncate_xlog** を呼び出すことによっていつでも手動でトランケートできます。
- DB2 のトランケーションが実行されると、Replication Agent がコミット/ロールバックを処理していない最も古い LSN (最も古いアクティブな LSN) が取得され、その LSN を含むアーカイブ・ログ・ファイルが特定されます。最も古いアクティブな LSN を含むファイルの直前までのアーカイブ・ログ・ファイルがすべて削除されます。

これらのプロパティの詳細については、『Replication Agent リファレンス・マニュアル』を参照してください。トランケート方法の詳細については、『Replication Agent 管理ガイド』の「管理」を参照してください。

参照：

- Java プロシージャ・オブジェクト (135 ページ)

Replication Agent のアップグレードとダウングレード

Replication Agent のアップグレードとダウングレードの手順を見直してください。

警告！ 新しい機能を使用した後に Replication Agent をダウングレードすることはできません。新しい機能を使用する前に、前のバージョンの Replication Agent と同じ機能レベルで Replication Agent の新しいバージョンをテストしてください。この既存の機能レベルで新しいバージョンの Replication Agent に満足した場合にのみ、新しい機能を使用したアップグレードを行ってください。また、新しい機能を使用する前に、アップグレード済みの Replication Agent バージョンのインスタンスのバックアップ・コピーを作成してください。

Replication Agent for Oracle のアップグレードおよび移行手順

Replication Agent for Oracle 15.7.1 は、プライマリ Oracle サーバを実行しているホストにインストールする必要があります。

ここで説明するいずれかのアップグレード手順を使用した場合、新しい Replication Agent for Oracle 15.7.1 インスタンスの設定は、インスタンス名、管理ユーザ ID およびパスワード、管理ポート番号を含め、以前の既存のインスタンスと同じ設定になります。

Replication Agent for Oracle 15.7.1 をプライマリ Oracle データ・サーバと同じマシンにインストールする必要はありません。ただし、次の点に注意してください。

- Replication Agent for Oracle は、Oracle LogMiner にアクセスできるホストにインストールする必要があります。
- Replication Agent for Oracle が Oracle ログを自動的にトランケートするように設定されている場合は、Oracle ログに直接アクセスできるマシンにインストールする必要があります。

注意： 共通のリリース・レベル内でアップグレードする場合は、特定のバージョンの Replication Agent に適用される ESD の場合と同様に、Replication Agent の特定のインスタンスまたは Replication Agent のすべてのインスタンスに適用される `ra_admin -u` オプションを使用します。『Replication Agent 管理ガイド』を参照してください。

Replication Agent for Oracle の 15.7.1 へのアップグレード

Replication Agent for Oracle をバージョン 15.7.1 にアップグレードします。

注意： Replication Agent 15.7.1 は、プライマリ Oracle サーバを実行しているホストにインストールしてください。

1. 既存の有効なすべての Replication Agent インスタンスの 15.7.1 バージョンを作成します。

注意： この手順では、既存のインスタンスが Oracle 用であるか、Microsoft SQL Server 用であるか、または IBM DB2 用であるかにかかわらず、以前のバージョンの Replication Agent の有効な既存の全インスタンスに対して新しい Replication Agent 15.7 インスタンスを作成します。Microsoft SQL Server または DB2 のインスタンスのアップグレードを完了するには、適切な手順を参照してください。新しく作成するインスタンスをこのホストで実行しない場合は、新しいインスタンス・ディレクトリを削除してください。

- a) UNIX では Replication Agent 15.7.1 がインストールされている SYBASE ディレクトリに変更し、source コマンドで SYBASE スクリプトを実行することにより、SYBASE 環境変数を設定します。

- C シェルでは、次のように入力します。source SYBASE.csh
- Bourne または Korn シェルでは次のように入力します。 .SYBASE.sh

- b) Replication Agent bin ディレクトリに変更します。

- UNIX の場合：

```
cd $SYBASE/RAX-15_5/bin
```
- Windows の場合：

```
cd %SYBASE%\RAX-15_5\bin
```

注意： リリース 15.5 以降の Replication Agent ディレクトリ名は RAX-15_5 です。

- c) 既存の有効なすべてのインスタンスの新しいバージョンを次のように作成します。

```
ra_admin init
```

2. Replication Agent 15.7.1 インスタンスを起動し、ログインします。ra_migrate コマンドを発行することにより Replication Agent のメタデータを移行します。

Oracle 10g を 11g にアップグレードする際の Replication Agent for Oracle 15.7.1 の移行

Oracle 10g も 11g にアップグレードする場合は、Replication Agent for Oracle 15.7.1 を移行します。

Oracle 10g から Oracle 11g へのアップグレードをサポートするための Replication Agent for Oracle の移行は、Replication Agent for Oracle 15.1 または 15.2 から Replication Agent for Oracle 15.7.1 と似ています。

注意： Oracle 10g を Oracle 11g にアップグレードする前に、Replication Agent をクワイースしてください。Oracle をアップグレードする前に、複写環境ですべてのトランザクションの処理が完了している必要があります。Replication Agent では Replication Agent の移行中にトランケーション・ポイントをログの末尾に移動するためです。

1. Oracle 10g から Oracle 11g にアップグレードするための Oracle のマニュアルの手順に従います。
2. Oracle をアップグレードしたら、Replication Agent を再起動し、**ra_migrate** コマンドを発行します。
3. ログベースの Replication Agent のアップグレード・プロセスと同様に、Oracle の設定によっては、アーカイブ・ログを読み取るにあたり、Replication Agent for Oracle インスタンスの再設定が必要になる場合があります。これは、Oracle のアップグレード後に変更できます。

ログベースの Replication Agent からのアップグレードと、Oracle 10g から Oracle 11g へのアップグレードを同時に行う場合は、Replication Agent 15.7.1 を一度だけ移行してください。

Replication Agent for Microsoft SQL Server のアップグレード手順

Replication Agent for Microsoft SQL Server は、プライマリ Microsoft SQL Server を実行している Windows ホストにインストールする必要があります。Replication Agent for Microsoft SQL Server を UNIX または Linux ホストにインストールすることはできません。アップグレード前に、前のバージョンの Replication Agent の既存のインスタンスとプライマリ・データ・サーバの現在のバージョンのインストール場所を確認してください。

この項で説明するいずれかのアップグレード手順を使用した場合、新しい Replication Agent for Microsoft SQL Server インスタンスの設定は、インスタンス名、

Replication Agent のアップグレードとダウングレード

管理ユーザ ID およびパスワード、管理ポート番号を含め、以前の既存のインスタンスと同じ設定になります。

Replication Agent for Microsoft SQL Server の 15.7.1 へのアップグレード

Replication Agent for Microsoft SQL Server を 15.7.1 にアップグレードします。

注意： Replication Agent 15.7.1 は、プライマリ Microsoft SQL Server を実行しているホストにインストールしてください。

1. 既存の有効なすべての Replication Agent インスタンスの 15.7.1 バージョンを作成します。

注意： この手順では、既存のインスタンスが Oracle 用であるか、Microsoft SQL Server 用であるか、または IBM DB2 用であるかにかかわらず、以前のバージョンの Replication Agent の有効な既存の全インスタンスに対して新しい Replication Agent 15.7.1 インスタンスを作成します。Oracle または DB2 のインスタンスのアップグレードを完了するには、適切な手順を参照してください。新しく作成するインスタンスをこのホストで実行しない場合は、新しいインスタンス・ディレクトリを削除してください。

- a) コマンド・ウィンドウを開きます。
- b) Replication Agent 15.7.1 がインストールされている SYBASE ディレクトリに変更して、SYBASE.bat スクリプトを実行して SYBASE 環境変数を設定します。
- c) 次のように変更します。

```
cd %SYBASE%\RAX-15_5\bin
```

注意： リリース 15.5 以降の Replication Agent ディレクトリ名は RAX-15_5 です。

- d) 既存の有効なすべてのインスタンスの新しいバージョンを次のように作成します。

```
ra_admin init
```

2. Replication Agent for Microsoft SQL Server 15.7.1 インスタンスのそれぞれを起動し、ログインします。
 - a) **rs_charset** 設定パラメータが Replication Server の文字セットと一致するように設定します (『Replication Agent リファレンス・マニュアル』を参照)。
 - b) **test_connection** コマンドを使用して Replication Agent が Microsoft SQL Server と Replication Server の両方に接続できることを確認します。
 - c) Replication Agent インスタンスを初期化し、**ra_migrate** コマンドを発行することにより、Replication Agent インスタンスのメタデータを移行します。

このコマンドが Replication Agent 15.7.1 の最初のインスタンスで実行されると、Microsoft SQL Server が初期化されます。その後の Replication Agent 15.7.1 インスタンスでは、そのインスタンスのみが初期化され、インスタンスのメタデータが移行されます。

- d) ra_migrate コマンドを発行することにより Replication Agent のメタデータを移行します。

Replication Agent for UDB のアップグレードおよび移行手順

IBMDB2 をバージョン 8.2 または 9.1 からバージョン 9.5 または 9.7 にアップグレードする場合、Replication Agent for UDB 15.7.1 は、Replication Agent for UDB のバージョン 15.0 以降のインスタンスに対して自動アップグレードを行い、また Replication Agent for UDB インスタンスに対して自動移行を行います。

この項で説明するいずれかのアップグレード手順を使用した場合、新しい Replication Agent for UDB 15.7.1 インスタンスの設定は、インスタンス名、管理ユーザ ID およびパスワード、管理ポート番号を含め、以前の既存のインスタンスと同じ設定になります。

Replication Agent for UDB 15.7.1 では以下をサポートしていません。

- Replication Agent for UDB のバージョン 12.6 またはそれ以前からバージョン 15.0 以降へのアップグレード
- UDB をバージョン 6 または 7 からバージョン 8 または 9 にアップグレードする際の Replication Agent for UDB 12.6 の移行

Replication Agent for UDB の 15.7.1 へのアップグレード

Replication Agent for UDB バージョン 15.5 以降を 15.7.1 にアップグレードします。

1. 既存の有効なすべての Replication Agent インスタンスの 15.7.1 バージョンを作成します。

注意：この手順では、既存のインスタンスが Oracle 用であるか、Microsoft SQL Server 用であるか、または IBM DB2 用であるかにかかわらず、以前のバージョンの Replication Agent の有効な既存の全インスタンスに対して新しい Replication Agent 15.7.1 インスタンスを作成します。Microsoft SQL Server または Oracle のインスタンスのアップグレードを完了するには、適切な項目を参照してください。新しく作成するインスタンスをこのホストで実行しない場合は、新しいインスタンス・ディレクトリを削除してください。

Replication Agent のアップグレードとダウングレード

- a) UNIX では Replication Agent 15.7.1 がインストールされている SYBASE ディレクトリに変更し、source コマンドで SYBASE スクリプトを実行することにより、SYBASE 環境変数を設定します。
 - C シェルでは、次のように入力します。source SYBASE.csh
 - Bourne または Korn シェルでは次のように入力します。 .SYBASE.sh
- b) Replication Agent bin ディレクトリに変更します。

- UNIX の場合：

```
cd $SYBASE/RAX-15_5/bin
```

- Windows の場合：

```
cd %SYBASE%\RAX-15_5\bin
```

注意：リリース 15.5 以降の Replication Agent ディレクトリ名は RAX-15_5 です。

- c) 既存の有効なすべてのインスタンスの新しいバージョンを次のように作成します。

```
ra_admin init
```

2. Replication Agent 15.7.1 インスタンスを起動し、ログインします。そして、**ra_migrate** コマンドを発行して、Replication Agent のメタデータを移行します。

DB2 をバージョン 8.2 または 9.1 からバージョン 9.5 または 9.7 にアップグレードする際の Replication Agent for UDB の移行

DB2 もアップグレードする場合は、Replication Agent for UDB を移行します。

1. 複製済みデータの消失を防ぐには、以前の既存の Replication Agent **pds_username** ユーザ以外のユーザに対して、プライマリ・データベースへのそれ以上のアクセスを拒否します。
2. Replication Agent 15.7.1 インスタンスにログインし、それが *Replicating* 状態にあることを確認し、複製を完了させます。複製が完了したことを確認するには、以下を実行します。
 - a) **ra_statistics** コマンドを定期的に発行して、次の統計がすべて 0 (ゼロ) になるまで監視します。
 - 入力キューのサイズ
 - 出力キューのサイズ
 - b) これらの値がすべてゼロになったら、最後の統計セットの Last QID Sent を記録します。
 - c) Replication Agent が Replication Server からトランケーション・ポイントを取得するように、**ra_locatorupdate** コマンドを発行します。

- d) 待ってから、再び **ra_locator** コマンドを発行し、表示されたロケータを Last QID Sent のロケータと比較します。差がある場合は、この手順を繰り返します。
 - e) **quiesce** コマンドを発行して Replication Agent インスタンスをクワイース状態にします。
 - f) **shutdown** コマンドを発行して、Replication インスタンスを停止します。
3. DB2 をアップグレードするための DB2 のマニュアルの手順に従います。
 4. すべてのプライマリ・データベース要件が満たされていることを確認します。

注意：移行前に、**use_rssd** 設定パラメータが true に設定されていた場合は、この手順を省略してください。

5. Replication Agent インスタンスを起動し、**use_rssd** 設定パラメータを true に設定します。

```
ra_config use_rssd, true
```

Replication Agent for UDB はこの設定を使用して RSSD に接続し、ロケータをゼロにリセットします。

6. **ra_migrate** コマンドを発行することにより Replication Agent のメタデータを移行します。

注意：移行前に、**use_rssd** 設定パラメータが true に設定されていた場合は、この手順を省略してください。

7. Replication Agent 15.7.1 インスタンスで、**resume** コマンドを発行することにより、複写を再開します。
8. すべてのユーザのプライマリ・データベースへのアクセスを許可します。

注意：Replication Agent のアップグレードと、UDB のバージョン 8.2 または 9.1 からバージョン 9.5 または 9.7 へのアップグレードを同時に行う場合は、Replication Agent を一度だけ移行する必要があります。

参照：

- IBM DB2 要件 (118 ページ)

Replication Agent for Oracle のダウングレード

Replication Agent for Oracle は 15.7.1 からバージョン 15.2 ESD #2 以降にダウングレードできます。

アップグレード・プロセスが失敗した場合、またはアップグレード後の複写が失敗した場合は、Replication Agent のダウングレードが必要な可能性があります。新

Replication Agent のアップグレードとダウングレード

機能が予想どおりに機能しない場合や、以下に変更が加えられた場合、複製が失敗することがあります。

- DDL および Replication Agent によるその処理方法
- Replication Agent システム・データベース (RASD) の内容または構造

Replication Agent 15.7.1 の新機能のいずれかを使用している場合は、ダウングレードできません。Replication Agent 15.7.1 の新機能のリストについては、『Replication Server Options 新機能ガイド』を参照してください。

Replication Agent 15.7.1 は、プライマリ Oracle サーバを実行しているプラットフォームにインストールしてください。

1. Replication Agent 15.7.1 bin ディレクトリに変更します。

- UNIX の場合：

```
cd $SYBASE/RAX-15_5/bin
```

- Windows の場合：

```
cd %SYBASE%\RAX-15_5\bin
```

リリース 15.5 以降の Replication Agent ディレクトリ名は RAX-15_5 です。

2. ダウングレードの対象となる (現在のバージョンの) Replication Agent インスタンスで `ra_downgrade` コマンドを実行します。

```
ra_downgrade
```

`ra_downgrade` コマンドは、Replication Agent システム・データベース (RASD) の内容を `timestamp.export` という名前のファイルに抽出します。ここで、`timestamp` は `ra_downgrade` の呼び出し時のタイムスタンプです。このファイルは、ダウングレード先 (前のバージョン) の Replication Agent インスタンスの `rasd_backup_dir` 設定パラメータで指定されたディレクトリにある `import` サブディレクトリにあります。`ra_downgrade` が正常に実行されると、このファイルの絶対パスが返されます。

注意： `ra_downgrade_prepare` コマンドと `ra_downgrade_accept` コマンドは、Replication Agent 15.7.1 で廃止されています。

3. ダウングレード先 (前のバージョン) の Replication Agent インスタンスで `ra_downgrade_accept` コマンドを実行し、ダウングレードを完了します。

```
ra_downgrade_accept timestamp.export
```

ここで、`timestamp.export` は `ra_downgrade` コマンドによる RASD の内容の抽出先ファイルです。

`ra_downgrade_accept` コマンドが正常に実行されると、Replication Agent は停止します。

4. ダウングレード先(前のバージョン)の Replication Agent インスタンスを起動し、複写を再開します。

```
resume purge
```

この Replication Agent の接続先コネクションの Replication Server インバウンド・キューからデータをパージするには、**purge** キーワードが必要です。パージにより、Replication Agent の前のバージョンと後のバージョン間での OQID フォーマットの変更の結果、Replication Server に重複したレコードが作成されることを防ぐことができます。

Replication Agent for Microsoft SQL Server のダウングレード

Replication Agent for Microsoft SQL Server は 15.7.1 からバージョン 15.2 ESD #2 以降にダウングレードできます。

アップグレード・プロセスが失敗した場合、またはアップグレード後の複写が失敗した場合は、Replication Agent のダウングレードが必要な可能性があります。新機能が予想どおりに機能しない場合や、以下に変更が加えられた場合、複写が失敗することがあります。

- DDL および Replication Agent によるその処理方法
- オリジン・キュー ID (OQID) のフォーマット
- Replication Agent システム・データベース (RASD) の内容または構造
- プライマリ・データベース内の Replication Agent システム・オブジェクト

Replication Agent 15.7.1 の新機能のいずれかを使用している場合は、ダウングレードできません。Replication Agent 15.7.1 の新機能のリストについては、『Replication Server Options 新機能ガイド』を参照してください。

1. Replication Agent 15.7.1bin ディレクトリに変更します。

```
cd %SYBASE%\RAX-15_5\bin
```

注意：リリース 15.5 以降の Replication Agent ディレクトリ名は RAX-15_5 です。

2. ダウングレードの対象となる(現在のバージョンの) Replication Agent インスタンスで ra_downgrade コマンドを実行します。

```
ra_downgrade
```

ra_downgrade コマンドは、Replication Agent システム・データベース (RASD) の内容を timestamp.export という名前のファイルに抽出します。ここで、timestamp は ra_downgrade の呼び出し時のタイムスタンプです。このファイル

Replication Agent のアップグレードとダウングレード

は、ダウングレード先 (前のバージョン) の Replication Agent インスタンスの `rasd_backup_dir` 設定パラメータで指定されたディレクトリにある `import` サブディレクトリにあります。`ra_downgrade` が正常に実行されると、このファイルの絶対パスが返されます。

注意： `ra_downgrade_prepare` コマンドと `ra_downgrade_accept` コマンドは、Replication Agent 15.7.1 で廃止されています。

3. ダウングレード先 (前のバージョン) の Replication Agent インスタンスで **`ra_downgrade_accept`** コマンドを実行し、ダウングレードを完了します。

```
ra_downgrade_accept timestamp.export
```

ここで、`timestamp.export` は **`ra_downgrade`** コマンドによる RASD の内容の抽出先ファイルです。

`ra_downgrade_accept` コマンドが正常に実行されると、Replication Agent は停止します。

4. ダウングレード先 (前のバージョン) の Replication Agent インスタンスを起動し、複写を再開します。

```
resume purge
```

この Replication Agent の接続先接続の Replication Server インバウンド・キューからデータをパージするには、**`purge`** キーワードが必要です。パージにより、Replication Agent の前のバージョンと後のバージョン間での OQID フォーマットの変更の結果、Replication Server に重複したレコードが作成されることを防ぐことができます。

Replication Agent for UDB のダウングレード

Replication Agent for UDB は 15.7.1 からバージョン 15.2 ESD #2 以降にダウングレードできます。

アップグレード・プロセスが失敗した場合、またはアップグレード後の複写が失敗した場合は、Replication Agent のダウングレードが必要な可能性があります。

Replication Agent 15.7.1 の新機能のいずれかを使用している場合は、ダウングレードできません。Replication Agent 15.7.1 の新機能のリストについては、『Replication Server Options 新機能ガイド』を参照してください。

1. Replication Agent 15.7.1 bin ディレクトリに変更します。

- UNIX の場合：

```
cd $SYBASE/RAX-15_5/bin
```

- Windows の場合：

```
cd %SYBASE%\RAX-15_5\bin
```

リリース 15.5 以降の Replication Agent ディレクトリ名は RAX-15_5 です。

2. Replication Agent for UDB 15.7 インスタンスにログインします。ダウングレードの対象となる (現在のバージョンの) Replication Agent インスタンスで `ra_downgrade` コマンドを実行します。

```
ra_downgrade
```

`ra_downgrade` コマンドは、Replication Agent システム・データベース (RASD) の内容を `timestamp.export` という名前のファイルに抽出します。ここで、`timestamp` は `ra_downgrade` の呼び出し時のタイムスタンプです。このファイルは、ダウングレード先 (前のバージョン) の Replication Agent インスタンスの `rasd_backup_dir` 設定パラメータで指定されたディレクトリにある `import` サブディレクトリにあります。`ra_downgrade` が正常に実行されると、このファイルの絶対パスが返されます。

注意： `ra_downgrade_prepare` コマンドと `ra_downgrade_accept` コマンドは、Replication Agent 15.7.1 で廃止されています。

3. `shutdown` コマンドを発行して、Replication インスタンスを停止します。
4. ダウングレード先 (前のバージョン) の Replication Agent インスタンスを起動しログインします。そして、`ra_migrate` を実行します。

```
ra_migrate
```

5. 複写をレジュームします。

```
resume
```


sybfilter ドライバのリファレンス

ここでは sybfilter ドライバのインストール、設定、使用、トラブルシューティングの方法について説明します。

Replication Agent は Microsoft SQL Server のログ・ファイルを直接読み込むことができる必要があります。ただし、Microsoft SQL Server プロセスは排他的な読み込みパーミッションを使用してこれらのログ・ファイルを開くので、Replication Agent を含む他のプロセスからこれらのファイルを読み込むことはできません。

Replication Agent でデータを複製するには、sybfilter ドライバを使用してログ・ファイルを読み込み可能にする必要があります。

Microsoft Filter Manager Library のバージョンの確認

sybfilter ドライバが正しく機能するためには、Microsoft Filter Manager Library のバージョンが 5.1.2600.2978 以降である必要があります。

ライブラリのバージョンを調べるには次のようにします。

1. Windows エクスプローラで `c:\windows\system32\fltlib.dll` を右クリックします。
2. [プロパティ] を選択し、[プロパティ] ダイアログの [バージョン] タブをクリックします。
3. バージョンが 5.1.2600.2978 より前の場合は、Microsoft Web サイト (<http://windowsupdate.microsoft.com>) で Windows システムを更新します。

sybfilter ドライバのインストールと設定

sybfilter ドライバをインストールし、設定します。

注意： Windows Vista で sybfilter ドライバをインストール、設定、実行するには、管理者としてログインする必要があります。

1. Windows エクスプローラで、sybfilter ドライバのインストール・ディレクトリに移動します。Windows では、このディレクトリは `%SYBASE%\RAX-15_5\system\<platform>` にあります。

ここで、`<platform>` は次のいずれかになります。

- winx86-オペレーティング・システムが 32 ビット・バージョンの Windows Server 2003、Windows Server 2008、Windows Vista、または Windows XP の場合
 - winx64-オペレーティング・システムが 64 ビット・バージョンの Windows Server 2003 または Windows XP の場合
 - winvistax64-オペレーティング・システムが 64 ビット・バージョンの Windows Server 2008 または Windows Vista の場合
2. sybfilter.inf を右クリックして sybfilter ドライバをインストールします。

注意： Windows マシンには 1 つのみの sybfilter ドライバをインストールできます。ドライバがインストールされると、同じマシンで実行されているすべての Replication Agent for Microsoft SQL Server のインスタンスに対して機能するようになります。sybfilter ドライバは、プライマリ Microsoft SQL Server と同じマシンにインストールする必要があります。

3. 任意のディレクトリに、プライマリ・データベース用のすべてのログ・ファイル・パスを格納するための設定ファイルを作成します。設定ファイルには .cfg のサフィックスを含める必要があります。

たとえば、ディレクトリ %SYBASE%\RAX-15_5\system<platform> の下に、LogPath.cfg という名前のファイルを作成します。

4. *RACFGFilePath* という名前のシステム環境変数を追加し、その値を設定ファイルのパスに設定します。
 - a) [コントロールパネル] で、[システム] > [詳細設定] > [環境変数] を開きます。
 - b) [新規] をクリックし、新しいシステム変数を追加します。
 - c) 変数を *RACFGFilePath* と名付け、その値を設定ファイルのロケーションに設定します。
5. Windows エクスプローラで、%SYBASE%\RAX-15_5\bin に移動し、sybfiltermgr.exe をダブルクリックし、sybfilter ドライバ管理コンソールを起動します。
6. sybfilter ドライバを起動するには、管理コンソールで start と入力します。
7. ユーザ・マネージャを使用するか、設定ファイルを変更することで、ログ・ファイルのパスを sybfilter ドライバに追加します。プライマリ Microsoft SQL Server が認識できるディレクトリとドライブ名を使用してください。

- ユーザ・マネージャ - 管理コンソールで **add** コマンドを使用します。このコマンドの構文は次のとおりです。

```
add serverName dbName logFilePath
```

たとえば、pdb2_log.ldf というログ・ファイル (D:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data\) を *dbName* デー

データベース (*serverName* データ・サーバ上) に追加するには、次のように指定します。

```
add myserverName dbName D:¥Program
```

```
Files¥Microsoft SQL
```

```
Server¥MSSQL.1¥MSSQL¥Data¥pdb2_log.ldf
```

注意： ユーザ・マネージャを使用してログ・ファイルのパスを追加する場合、ログ・パスが設定ファイルに追加された後に、ユーザ・マネージャによって sybfilter ドライバへのすべてのログ・パスが自動的に更新されます。

- 設定ファイル – ログ・ファイルのパスを設定ファイルに直接追加するには、設定ファイルを開き、手動で編集します。これはログ・ファイル・パスのエントリ例です。

```
[myserver, pdb1]
```

```
log_file_path=D:¥Program Files¥Microsoft SQL Server¥MSSQL.1¥MSSQL¥Data¥pdb11_log.ldf
```

```
log_file_path=D:¥Program Files¥Microsoft SQL Server¥MSSQL.1¥MSSQL¥Data¥pdb12_log.ldf
```

```
[myserver, pdb2]
```

```
log_file_path=D:¥Program Files¥Microsoft SQL Server¥MSSQL.1¥MSSQL¥Data¥pdb2_log.ldf
```

注意： ログ・ファイルのパスを設定ファイルに追加した後、管理コンソールで **refresh** コマンドを使用します。

8. ログ・ファイルのパスを sybfilter ドライバに追加する前にプライマリ・データベースのログ・ファイルを追加した場合は、Microsoft SQL Server を再起動してログ・ファイルを読み取り可能にします。
9. 管理コンソールで、check と入力してログ・ファイルが読み取り可能であることを確認します。

ログ・ファイルが読み取り不可能な場合は、必要に応じて、ファイルが作成されており、Microsoft SQL Server が再起動されていることを確認してください。

トラブルシューティング

sybfilter ドライバをトラブルシューティングするときは、次の問題を考慮してください。

表 23 : sybfilter ドライバの既知の問題

問題の説明
<p>システム環境変数が設定されていない。</p> <p>問題：管理コンソールから次のようなエラーが報告される： ERROR: System environment variable RACFGFilePath has not been set. Please set its value before starting this manager. Fatal error occurs. Please press any key to quit.</p> <p>対処方法：RACFGFilePath 環境変数を設定します。</p>
<p>設定ファイルが存在しない。</p> <p>問題：list コマンドに対して管理コンソールが以下を報告する： ERROR: Cannot open config file.</p> <p>対処方法：設定ファイルを作成します。</p>
<p>設定ファイルが書き込み可能でない。</p> <p>問題：add コマンドに対して管理コンソールが以下を報告する： ERROR: Cannot open config file.</p> <p>対処方法：設定ファイルに書き込みパーミッションを追加します。</p>
<p>Microsoft SQL Server ログ・ファイルがロックされている。</p> <p>問題：Replication Agent for Microsoft SQL Server がインストールされているマシンを再起動した後、Microsoft SQL Server ログ・ファイルがロックされていて開くことができない。</p> <p>対処方法：sybfilter 管理コンソールを再起動します。stop コマンドに続き、start コマンドを発行して sybfilter ドライバを再起動します。プライマリ Microsoft SQL Server データ・サーバを再起動します。</p>

トレース・ログの使用

sybfilter トレース・ログ情報を使用して問題を診断し、トラブルシューティングします。

1. sybfilter 管理コンソールで **trace** コマンドと適切なトレース・フラグを使用して、トレースを有効にします。たとえば、再起動後に Microsoft SQL Server ログ・ファイルの読み取りができない理由を突き止めるには、Microsoft SQL Server を再起動する前に、次のように **T3** フラグを指定してトレースを有効にします。

```
trace T3
```

2. sybfilter トレース・ログ・ファイル `sybfilter.trc` を開き、記録されたメッセージを表示します。
3. sybfilter 管理コンソールでトレースを無効にします。

```
trace off
```

sybfilter コマンド・リファレンス

以下のコマンドは、sybfilter 管理コンソールで使用できます。コマンドのリストと説明を表示するには、sybfilter 管理コンソールで **help** コマンドを入力してください。

add

ログ・ファイル・パスを sybfilter ドライバと設定ファイルに追加します。

構文

```
add serverName dbName logFilePath
```

パラメータ

- **serverName** – Microsoft SQL Server の名前。
- **dbName** – 複写するデータベースの名前。
- **logFilePath** – データベース・ログのパス。

check

sybfilter ドライバが実行されているかどうかを確認します。設定ファイルと sybfilter ドライバのパス名の違いを確認します。sybfilter の設定ファイルが読み取り可能であるかどうかを確認し、読み取り可能でないファイルをリストします。

構文

```
check
```

exit

sybfilter 管理コンソールを終了します。

構文

```
exit
```

help

すべての sybfilter コマンドに関するヘルプ情報を出力します。

構文

```
help
```

list

設定済みのすべてのデータベース名と、設定ファイル内の対応するログ・ファイル・パスをリストします。

構文

```
list
```

refresh

sybfilter 設定ファイル内の内容を更新します。

構文

```
refresh
```

remove

ログ・ファイル・パスを sybfilter ドライバと設定ファイルから削除します。

構文

```
remove logFilePath
```

パラメータ

- **logFilePath** – データベース・ログのパス。

start

sybfilter ドライバを起動します。

構文

```
start
```

stop

sybfilter ドライバを停止します。

構文

```
stop
```

trace

sybfilter ドライバの実行をトレースします。

構文

```
trace [T1] [T2] [T3] [T4] | all | off
```

パラメータ

- **T1** – 定期的なトレース・メッセージを記録します。
- **T2** – オペレーション・ステータス情報メッセージを記録します。
- **T3** – 通常のメッセージを記録します。
- **T4** – エラー・メッセージを記録します。
- **all** – T1、T2、T3、T4 フラグに関するすべてのメッセージを記録します。
- **off** – トレースを無効にします。

追加の説明や情報の入手

Sybase Getting Started CD、製品マニュアル Web サイト、オンライン・ヘルプを利用すると、この製品リリースについて詳しく知ることができます。

- Getting Started CD (またはダウンロード) – PDF フォーマットのリリース・ノートとインストール・ガイド、その他のマニュアルや更新情報が収録されています。
- Sybase 製品マニュアル Web サイト (<http://sybooks.sybase.com/>) にある製品マニュアルは、Sybase マニュアルのオンライン版であり、標準の Web ブラウザを使用してアクセスできます。マニュアルはオンラインで参照することも PDF としてダウンロードすることもできます。この Web サイトには、製品マニュアルの他に、EBFs/Maintenance、Technical Documents、Case Management、Solved Cases、Community Forums/Newsgroups、その他のリソースへのリンクも用意されています。
- 製品のオンライン・ヘルプ (利用可能な場合)

PDF 形式のドキュメントを表示または印刷するには、Adobe の Web サイトから無償でダウンロードできる Adobe Acrobat Reader が必要です。

注意： 製品リリース後に追加された製品またはマニュアルについての重要な情報を記載したさらに新しいリリース・ノートを製品マニュアル Web サイトから入手できることがあります。

サポート・センタ

Sybase 製品に関するサポートを得ることができます。

組織でこの製品の保守契約を購入している場合は、サポート・センタとの連絡担当者が指定されています。マニュアルだけでは解決できない問題があった場合には、担当の方を通して Sybase 製品のサポート・センタまでご連絡ください。

Sybase EBF と Maintenance レポートのダウンロード

EBF と Maintenance レポートは、Sybase Web サイトからダウンロードしてください。

1. Web ブラウザで <http://www.sybase.com/support> を指定します。

追加の説明や情報の入手

2. メニュー・バーまたはスライド式メニューの [Support (サポート)] で [EBFs/Maintenance (EBF/メンテナンス)] を選択します。
3. ユーザ名とパスワードの入力が求められたら、MySybase のユーザ名とパスワードを入力します。
4. (オプション) [Display (表示)] ドロップダウン・リストからフィルタを指定し、期間を指定して、[Go (実行)] をクリックします。
5. 製品を選択します。

鍵のアイコンは、「Authorized Support Contact」として登録されていないため、一部の EBF/Maintenance リリースをダウンロードする権限がないことを示しています。未登録ではあるが、Sybase 担当者またはサポート・センタから有効な情報を得ている場合は、[My Account (マイ・アカウント)] をクリックして、「Technical Support Contact」役割を MySybase プロファイルに追加します。

6. EBF/Maintenance レポートを表示するには [Info] アイコンをクリックします。ソフトウェアをダウンロードするには製品の説明をクリックします。

Sybase 製品およびコンポーネントの動作確認

動作確認レポートは、特定のプラットフォームでの Sybase 製品のパフォーマンスを検証します。

動作確認に関する最新情報は次のページにあります。

- パートナー製品の動作確認については、http://www.sybase.com/detail_list?id=9784 にアクセスします。
- プラットフォームの動作確認については、<http://certification.sybase.com/ucr/search.do> にアクセスします。

MySybase プロファイルの作成

MySybase は無料サービスです。このサービスを使用すると、Sybase Web ページの表示方法を自分専用カスタマイズできます。

1. <http://www.sybase.com/mysybase> を開きます。
2. [Register Now (今すぐ登録)] をクリックします。

アクセシビリティ機能

アクセシビリティ機能を使用すると、身体障害者を含むすべてのユーザーが電子情報に確実にアクセスできます。

Sybase 製品のマニュアルには、アクセシビリティを重視した HTML 版もあります。

オンライン・マニュアルは、スクリーン・リーダーで読み上げる、または画面を拡大表示するなどの方法により、視覚障害を持つユーザがその内容を理解できるよう配慮されています。

Sybase の HTML マニュアルは、米国のリハビリテーション法第 508 条のアクセシビリティ規定に準拠していることがテストにより確認されています。第 508 条に準拠しているマニュアルは通常、World Wide Web Consortium (W3C) の Web サイト用ガイドラインなど、米国以外のアクセシビリティ・ガイドラインにも準拠しています。

注意：アクセシビリティ・ツールを効率的に使用するには、設定が必要な場合もあります。一部のスクリーン・リーダーは、テキストの大文字と小文字を区別して発音します。たとえば、すべて大文字のテキスト (ALL UPPERCASE TEXT など) はイニシャルで発音し、大文字と小文字の混在したテキスト (Mixed Case Text など) は単語として発音します。構文規則を発音するようにツールを設定すると便利かもしれませんが、詳細については、ツールのマニュアルを参照してください。

Sybase のアクセシビリティに対する取り組みについては、Sybase Accessibility サイト (<http://www.sybase.com/products/accessibility>) を参照してください。このサイトには、第 508 条と W3C 標準に関する情報へのリンクもあります。

製品マニュアルには、アクセシビリティ機能に関する追加情報も記載されています。

追加の説明や情報の入手

用語解説

この用語解説では Replication Server Options で使用されている用語について説明します。

- **Adaptive Server** – Sybase リレーショナル・データベース管理システム (RDBMS) ソフトウェア製品のブランド名。
 - Adaptive Server Enterprise は、高容量オンライン・トランザクション処理 (OLTP: Online Transaction Processing) システムとクライアント・アプリケーション向けの大規模なリレーショナル・データベースを複数管理します。
 - Sybase®IQ は、特殊なインデックス・アルゴリズムで大規模なリレーショナル・データベースを複数管理して、高速、高容量のビジネス・インテリジェンス、意思決定サポート、レポート・クライアント・アプリケーションなどをサポートします。
 - SQL Anywhere® (旧称 Adaptive Server Anywhere) は、埋め込みアプリケーションやモバイル・デバイス・アプリケーションに最適な小型 DBMS によってリレーショナル・データベースを管理します。

「DBMS」と「RDBMS」参照。

- **アトミック・マテリアライゼーション** – マテリアライゼーション・メソッドの1つで、サブスクリプション・データをプライマリ・データベースからレプリケート・データベースに1回のアトミック・オペレーションでコピーします。プライマリ・データベースでサブスクリプション・データが取り込まれるまで、プライマリ・データの変更はできません。「バルク・マテリアライゼーション」と「ノンアトミック・マテリアライゼーション」参照。
- **BCP ユーティリティ** – ターゲット・データベースのテーブルに複数ローのデータをロードできるバルク・コピー転送ユーティリティ。「バルク・コピー」参照。
- **バルク・コピー** – データベース・テーブルとプログラム変数との間の高速データ転送に使用される Open Client™ インタフェース。バルク・コピーは、SQL **insert** コマンドと **select** コマンドを使用するデータ転送の代替となります。
- **バルク・マテリアライゼーション** – マテリアライゼーション・メソッドの1つで、レプリケート・データベースのサブスクリプション・データを複写システム外で初期化します。バルク・マテリアライゼーションは、テーブル複写定義とファンクション複写定義のどちらのサブスクリプションにも使用できます。「アトミック・マテリアライゼーション」と「ノンアトミック・マテリアライゼーション」参照。
- **クライアント** – クライアント/サーバ・システムでは、サーバに要求を送信して、その結果を処理する部分。「クライアント・アプリケーション」参照。

- **クライアント・アプリケーション** – メニュー、データ入力画面、レポート形式などのユーザ・インタフェースを制御するソフトウェア。「クライアント」参照。
- **コミット** – トランザクションで要求された変更を永続化する DBMS への命令。「トランザクション」参照。「ロールバック」と対比。
- **データベース** – ユーザのためにデータの受け入れ、格納、提供などを行う特定の構造 (スキーマ) を備えたデータの集まり。「データ・サーバ」、「DBMS」、「RDBMS」参照。
- **データベース・コネクション** – Replication Server がデータベースを管理し、トランザクションをデータベースに分配するためのコネクション。複写システム内の各データベースに接続できるデータベース・コネクションは Replication Server で1つだけです。「Replication Server」と「ルート」参照。
- **データ・クライアント** – データ・サーバに接続してデータへのアクセスを提供するクライアント・アプリケーション。「クライアント」、「クライアント・アプリケーション」、「データ・サーバ」参照。
- **データの分配** – 1つのデータ・セットの個々の部分を複数のシステムまたは複数のサイトに配置する方法。データ複写システムをデータ分散の実装やサポートに使用することもできますが、データの分配はデータの複写とは異なります。「データの複写」と対比。
- **データの複写** – プライマリ・データをリモート・ロケーションにコピーし、コピーされたデータをプライマリ・データに同期するプロセス。データの複写はデータの分配とは異なります。複写データはリモート・サイトのシステム全体で保管されているデータのコピーで、必ずしも分配されたデータであるとは限りません。「データの分配」と対比。「トランザクションの複写」参照。
- **データ・サーバ** – データベース内のテーブルの物理表現を管理するのに必要な機能を提供するサーバ。通常、データ・サーバとデータベース・サーバは同じですが、データ・クライアントに必要なインタフェースと機能を備えたデータ・レポジトリの場合もあります。「クライアント」、「クライアント・アプリケーション」、「データ・クライアント」参照。
- **データ型** – コンピュータに保存されている情報の特性を識別するキーワード。一般的なデータ型は次のとおりです。*char*、*int*、*smallint*、*date*、*time*、*numeric*、*float*。データ・サーバによって、サポートしているデータ型が異なります。
- **DBMS** – データベース管理システム (Database Management System) の略語。データベースを定義、作成、操作、制御、管理、使用するためのコンピュータベースのシステム。DBMS は、データベースを使用するためのユーザ・インタフェースを含む場合や、スタンドアロンのデータ・サーバ・システムである場合があります。「RDBMS」と対比。

- **ERSSD** – Embedded Replication Server システム・データベース (Embedded Replication Server System Database) の略語。Replication Server の複製システム情報を管理します。「*Replication Server*」参照。
- **フェールバック** – フェールオーバー手続きによってプライマリ・データベースからレプリケート・データベースにアクセスが切り替わった後で、ユーザとクライアントが正常にプライマリ・データベースにアクセスできるようにリストアする手順。「*フェールオーバー*」参照。
- **フェールオーバー** – プライマリ・データベースでの操作やプライマリ・データベースへのアクセスが中断するようなエラーが発生した場合に、ユーザとクライアントのアクセスをプライマリ・データベースからレプリケート・データベースに切り替える手続き。フェールオーバーは、高可用性を必要とするシステムにとって重要なフォールト・トレランス機能です。「*フェールバック*」参照。
- **ファンクション** – 1 つまたは一連のオペレーションを表すデータ・サーバ・オブジェクト。Replication Server は、これらのオペレーションをファンクションとしてレプリケート・データベースに配信します。「*ストアド・プロシージャ*」参照。
- **ファンクション文字列** – ファンクションとそのパラメータをデータ・サーバの API にマップするときに Replication Server が使用する文字列。ファンクション文字列を使用すると、プライマリ・データベースとレプリケート・データベースの種類、SQL 拡張機能、コマンド機能などが異なる異機種間の複製を Replication Server がサポートできるようになります。「*ファンクション*」参照。
- **ゲートウェイ** – ネットワーク・アーキテクチャが異なる複数のコンピュータ・システム間の通信を可能にする接続ソフトウェア。
- **インバウンド・キュー** – Replication Agent から受け取ったメッセージをスプールするために Replication Server が管理するステープル・キュー。「*アウトバウンド・キュー*」と「*ステープル・キュー*」参照。
- **interfaces ファイル** – Sybase Open Client/Open Server™ アプリケーションが他の Open Client/Open Server アプリケーションと接続を確立するために必要な情報が含まれたファイル。「*Open Client*」と「*Open Server*」参照。
- **isql** – Sybase Open Server アプリケーション (Adaptive Server、Replication Agent、Replication Server など) に接続して通信できる Interactive SQL クライアント・アプリケーション。「*Open Client*」と「*Open Server*」参照。
- **Java** – Sun Microsystems が開発したオブジェクト指向プログラミング言語。プラットフォームに依存しない“write once, run anywhere (一度書けばどこでも動く)”プログラミング言語。
- **Java VM** – Java 仮想マシン。Java VM (または JVM) は、Java バイト・コードの解釈を実行する Java ランタイム環境 (JRE) の一部です。「*Java*」と「*JRE*」参照。

- **JDBC** – Java データベース・コネクティビティ (Java Database Connectivity) の略語。JDBC は Java クライアントとデータ・サーバを接続するための標準通信プロトコルです。「データ・サーバ」と「Java」参照。
- **JRE** – Java ランタイム環境 (Java Runtime Environment) の略語。JRE は Java 仮想マシン (Java VM または JVM)、Java コア・クラス、サポート・ファイルで構成されています。Replication Agent などの Java アプリケーションを実行するには、マシンに JRE がインストールされている必要があります。「Java VM」参照。
- **LAN** – ローカル・エリア・ネットワーク (Local Area Network) の略語。ユーザの構内にあり、限定された地域 (通常は 1 サイト) をカバーするコンピュータ・ネットワーク。ローカル・エリア・ネットワーク内の通信は外部規制を受けませんが、LAN 圏外の通信は一定の規制を受けます。「WAN」と対比。
- **遅延時間** – トランザクションの複製では、プライマリ・データベースからレプリケート・データベースへのトランザクションの複製にかかる時間。厳密に言うと、遅延時間とは、プライマリ・データベースで元のトランザクションをコミットしてから、レプリケート・データベースで複製トランザクションをコミットするまでの経過時間です。

ディスクの複製では、遅延時間は、プライマリ・デバイスでブロックやページを変更するディスク書き込み処理から、レプリケート・デバイスで複製されたブロックやページを変更するディスク書き込み処理までの経過時間です。

「トランザクションの複製」参照。

- **LOB** – ラージ・オブジェクト (Large Object) の略語。データベースに 1 つのエンティティとして格納されている大きいデータ・コレクション。
- **Log Reader** – プライマリ・データベースと対話して複製のためのトランザクションを取り込む Replication Agent の内部コンポーネント。「Log Transfer Interface」と「Log Transfer Manager」参照。
- **Log Transfer Interface (ログ転送インタフェース)** – Replication Server に配信するトランザクションを転送するために Replication Server と対話する Replication Agent の内部コンポーネント。「Log Reader」と「Log Transfer Manager」参照。
- **Log Transfer Language (ログ転送言語)** – プライマリ・データベースから Replication Server にデータを複製するために Replication Agent と Replication Server の間で使用される独自のプロトコル。「Log Reader」と「Log Transfer Interface」参照。
- **Log Transfer Manager** – Replication Agent のその他の内部コンポーネントと対話して Replication Agent のオペレーションを制御および調整する Replication Agent の内部コンポーネント。「Log Reader」と「Log Transfer Interface」参照。
- **メンテナンス・ユーザ** – Replication Server がデータベースに複製トランザクションを適用するとき使用するレプリケート・データベースの特別なユーザ・ログイン名。「レプリケート・データベース」と「Replication Server」参照。

- **マテリアライゼーション** – プライマリ・データベースからレプリケート・データベースにデータをコピーして、複写システムがトランザクションの複写を開始できるようにレプリケート・データベースを初期化するプロセス。「アトミック・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」、「ノンアトミック・マテリアライゼーション」参照。
- **Multi-Path Replication™** – 送信元データベースからターゲット・データベースへのデータの並列パスを有効にすることによってパフォーマンスを向上させる Replication Server の機能。これらの複数のパスではデータが個別に処理され、それらのパス間のトランザクションの一貫性を必要とせずにデータ・セットを並列処理できる場合に適用されます。
- **ノンアトミック・マテリアライゼーション** – マテリアライゼーション・メソッドの1つで、プライマリ・データベースをロックせずにサブスクリプション・データをコピーします。データの転送中もプライマリ・データを変更できるので、プライマリ・データベースとレプリケート・データベース間で一時的に不一致が生じる場合があります。「アトミック・マテリアライゼーション」と対比。「バルク・マテリアライゼーション」参照。
- **ODBC** – Open Database Connectivity の略語。クライアントがデータ・サーバに接続するための業界標準通信プロトコル。「クライアント」、「データ・サーバ」、「JDBC」参照。
- **Open Client** – カスタム・アプリケーション、サードパーティ製品、他の Sybase 製品が Open Server アプリケーションと通信するために必要なインタフェースを提供する Sybase 製品。「Open Server」参照。
- **Open Client アプリケーション** – Sybase Open Client ライブラリを使用して Open Client 通信プロトコルを実装するアプリケーション。「Open Client」と「Open Server」参照。
- **Open Server** – カスタム・サーバの作成に必要なツールとインタフェースを提供する Sybase 製品。「Open Client」参照。
- **Open Server アプリケーション** – Sybase Open Server ライブラリを使用して Open Server 通信プロトコルを実装するサーバ・アプリケーション。「Open Client」と「Open Server」参照。
- **アウトバウンド・キュー** – レプリケート・データベースにメッセージをスプールするために Replication Server が管理するステープル・キュー。「インバウンド・キュー」、「レプリケート・データベース」、「ステープル・キュー」参照。
- **プライマリ・データ** – 複写に使用されるデータ・ソース。プライマリ・データはプライマリ・データベースによって保存および管理されます。「プライマリ・データベース」参照。
- **プライマリ・データベース** – 複写システムを使用して別のデータベース(レプリケート・データベース)に複写するデータが格納されているデータベース。複写システムではプライマリ・データベースが複写データのソースです。アク

ティブ・データベースと呼ばれることもあります。「レプリケート・データベース」と対比。「プライマリ・データ」参照。

- **プライマリ・キー** – テーブル内の各ローをユニークに識別するカラムまたはカラムのセット。
- **プライマリ・サイト** – 通常の業務処理をサポートするためにプライマリ・データ・サーバとプライマリ・データベースが配備されている場所または施設。アクティブ・サイトまたはメイン・サイトと呼ばれることもあります。「プライマリ・データベース」と「レプリケート・サイト」参照。
- **プライマリ・テーブル** – 複写のソースとして使用されるテーブル。プライマリ・テーブルはプライマリ・データベース・スキーマで定義されます。「プライマリ・データ」と「プライマリ・データベース」参照。
- **プライマリ・トランザクション** – プライマリ・データベースでコミットされ、プライマリ・データベースのトランザクション・ログに記録されたトランザクション。「プライマリ・データベース」、「複写トランザクション」、「トランザクション・ログ」参照。
- **クワイス** – システムを、これ以上データ変更ができない状態に切り替えること。「クワイス状態」参照。
- **クワイス状態** – 複写システムでは、すべての更新がその送信先に反映された状態。Replication Agent と Replication Server の一部のコマンドでは、最初に複写システムをクワイスする必要があります。

データベースでは、トランザクションがデータを変更できないようにデータの更新がすべて停止し、データ・デバイスとログ・デバイスが静止している状態。

この用語は「クワイスされている」または「クワイス」と同義です。「クワイス」参照。

- **RASD** – Replication Agent システム・データベース (Replication Agent System Database) の略語。RASD 内の情報は、トランザクション・ログでデータベース構造やスキーマ・オブジェクトを認識するためにプライマリ・データベースによって使用されます。
- **RCL** – 複写コマンド言語 (Replication Command Language) の略語。Replication Server の管理に使用されるコマンド言語。「*Replication Server*」参照。
- **RDBMS** – リレーショナル・データベース管理システム (Relational Database Management System) の略語。リレーショナル・データベースを管理および制御するアプリケーション。「*DBMS*」と対比。「*リレーショナル・データベース*」参照。
- **リレーショナル・データベース** – カラム (データ項目) とロー (情報の単位) から成るテーブルにデータを格納して表示するデータの集まり。リレーショナル・データベースは SQL 要求によってアクセスできます。「データベース」と対比。「*SQL*」参照。

- **レプリケート・データ** - 複写システムによってプライマリ・データベースからレプリケート・データベースに複写されたデータ・セット。「プライマリ・データベース」、「複写システム」、「レプリケート・データベース」参照。
- **レプリケート・データベース** - 複写システムによって別のデータベース(プライマリ・データベース)から複写されたデータが格納されているデータベース。レプリケート・データベースは複写システムで複写されたデータを受け取るデータベースです。「プライマリ・データベース」と対比。「レプリケート・データ」、「複写トランザクション」、「複写システム」参照。
- **複写トランザクション** - トランザクション複写システムによってプライマリ・データベースからレプリケート・データベースに複写されたプライマリ・トランザクション。「プライマリ・データベース」、「プライマリ・トランザクション」、「レプリケート・データベース」、「トランザクション複写」参照。
- **レプリケート・サイト** - プライマリ・サイトでのスケジュールされているダウン時間中の通常の業務処理をサポートするためにレプリケート・データ・サーバとレプリケート・データベースが配備されている場所または施設。「プライマリ・サイト」と対比。「レプリケート・データベース」参照。
- **Replication Agent** - プライマリ・データベース・トランザクションのログを読み取ってプライマリ・データベースのデータ変更トランザクションに関する情報を取得し、ログ情報を処理してから、それをレプリケート・データベースに分配する目的で Replication Server に送信するアプリケーション。「プライマリ・データベース」と「Replication Server」参照。
- **複写定義** - サブスクリプションを作成できるプライマリ・データベースのテーブルまたはストアド・プロシージャの記述。Replication Server によって管理される複写定義には、複写されるカラムとプライマリ・テーブルまたはストアド・プロシージャの場所に関する情報が含まれています。「Replication Server」と「サブスクリプション」参照。
- **Replication Server** - トランザクション複写システムのインフラストラクチャを提供する Sybase ソフトウェア製品。「Replication Agent」参照。
- **複写システム** - データを別の場所に複写するデータ処理システム。データは1つのサイトの異なるシステム間、またはローカル・システムとリモート・システム間で複写できます。「トランザクションの複写」参照。
- **ロールバック** - 作業単位(すなわちトランザクション)で要求された変更を取り消すデータベースへの命令。「コミット」と対比。「トランザクション」参照。
- **ルート** - プライマリ Replication Server からレプリケート Replication Server への一方方向のメッセージ・ストリーム。ルートは、データ変更コマンド(RSSDのコマンドも含む)と、Replication Server 間で複写されたファンクション(データベース・プロシージャ)を転送します。「Replication Server」参照。

- **RSSD** – Replication Server システム・データベース (Replication Server System Database) の略語。Replication Server の複写システム情報を管理します。
「*Replication Server*」参照。
- **SQL** – 構造化問合せ言語 (Structured Query Language) の略語。リレーショナル・データベースのデータ処理に使用される非手続き型プログラミング言語。ANSI SQL は業界標準の1つです。「*トランザクション*」参照。
- **ステーブル・キュー** – Replication Server が管理するディスク・デバイスベースの蓄積転送キュー。ステーブル・キューに書き込まれたメッセージは、適切なプロセスまたはレプリケート・データベースに配信できる時まで、このキューに格納されます。Replication Server は受信メッセージ (インバウンド・キュー) と送信メッセージ (アウトバウンド・キュー) の両方にステーブル・キューを提供します。「*データベース・コネクション*」、「*Replication Server*」、「*ルート*」参照。
- **ストアド・プロシージャ** – 1つまたは一連のオペレーションを表すデータ・サーバ・オブジェクト。この用語は、「*ファンクション*」と同じ意味で使用される場合があります。
- **サブスクリプション** – 指定した場所のレプリケート・データベース内のテーブルの複写コピーやテーブルのローを Replication Server に管理させる要求。「*レプリケート・データベース*」、「*複写定義*」、「*Replication Server*」参照。
- **テーブル** – リレーショナル DBMS では、テーブルに固有のカラム・グループから成る順不同のローが特定数含まれている、2次元のデータ配列または名前付きデータ・オブジェクト。「*データベース*」参照。
- **トランザクション** – データベースの作業単位。0、1、または多数のオペレーション (*insert*、*update*、*delete* など) を含むことができ、全体として適用または拒否されます。データベースの設定によっては、データを変更する各 SQL ステートメントを別々のトランザクションとして処理できます。「*SQL*」参照。
- **トランザクションの一貫性** – プライマリ・データベースのすべてのトランザクションが、プライマリ・データベースと同じ順序でレプリケート・データベースに適用される状態。
- **トランザクション・ログ** – 通常は、データ・サーバが管理しているデータに影響するトランザクションのログ。Replication Agent はトランザクション・ログを読み取って、複写するトランザクションを識別し、プライマリ・データベースから取得します。「*Replication Agent*」、「*プライマリ・データベース*」、「*Replication Server*」参照。
- **トランザクションの複写** – データ複写方式の1つで、データ変更オペレーションをプライマリ・データベースからレプリケート・データベースにコピーします。「*データの複写*」参照。
- **UDB** – IBM DB2 Universal Database (旧称 IBM DB2 for Linux, UNIX, and Windows)

- **WAN**-広域ネットワーク (Wide Area Network) の略語。ローカル・エリア・ネットワーク (LAN) をデータ通信回線で接続しているシステム。「LAN」と対比。

索引

A

Administration Client 120

C

CLASSPATH 環境変数 8

D

DB2

 オリジン・キュー ID 126
 要件 118

F

FORCE APPLICATION コマンド 124

I

IBM DB2 Universal Database

 次を参照： UDB

J

Java スタアド・プロシージャ 135

JDBC ドライバ

 Oracle 8

L

Log Reader コンポーネント

 読み込みバッファ・サイズ 124
 非同期オペレーション 123

LTM ロケータ

 オリジン・キュー ID 24, 99, 126

M

Microsoft SQL Server

 Replication Agent ユーザ ID 93
 オリジン・キュー ID 99

 パーミッション 93

 プライマリ・データベース 85

 役割 93

Microsoft SQL Server 向け Replication Agent

 トランザクション・ログ 107

Microsoft Windows プラットフォーム 85

O

Oracle の分割されたテーブル 54

Oracle データベース・サーバ

 JDBC ドライバ 8

 TNS リスナ・サービス 8

 オリジン・キュー ID 24

 プライマリ・データベース 5

P

pdb_dflt_object_repl 設定パラメータ 46

pdb_setrepproc コマンド 45

pdb_setrepseq コマンド 49

pdb_setreptable 設定パラメータ

 Replication Agent for UDB のマーク付け|マ
 ーク解除でサポートされていない
 すべてのキーワード 116

R

Replication Agent

 Log Reader コンポーネント 123

 オリジン・キュー ID 24, 99, 126

 トランザクション・ログ 78, 107, 133

 プライマリ・データベース・ユーザ ID 93,
 116

 マーク付けされたオブジェクトのテーブル
 136

Replication Agent for Microsoft SQL Server 85

 データ型の互換性 100

 パーミッション 93

 プライマリ・データベースのユーザ ID
 93

 役割 93

索引

Replication Agent for Oracle 5
 JDBC ドライバ 8
 Oracle Server と Replication Agent を別々の
 マシン上で実行 49
 トランザクション・ログ 78
Replication Agent for UDB 115
 スキャン・バッファ・サイズ 124
 設定パラメータ 122
 データベース通信エラー (-30081) 124
 データ型の互換性 127
 トランザクション・ログの作成 116
 トランザクション・ログ 133
 プライマリ・データベース・ユーザ ID
 116
 マーク付けされたオブジェクトのテーブル
 136

T

TNS リスナ・サービス、Oracle 8
truncate partition コマンド
 複写 54

U

UDB
 DATA CAPTURE テーブル属性 116
 Replication Agent のユーザ ID 116
 データ型 127
 プライマリ・データベース 115
 プライマリ・テーブルのマーク付け 116
 マーク付けされたオブジェクトのテーブル
 136
 通信エラー (-30081) 124

W

Windows
 次を参照： Microsoft Windows プラットフォー
 ーム

お

オペレーティング・システム
 Microsoft Windows プラットフォーム 85
オリジン・キュー ID
 DB2 126

Microsoft SQL Server 99
 Oracle 24
オリジンのコミット時刻
 Oracle 25

か

隠しテーブル
 マーカ 82, 109

こ

コマンド
 pdb_setreproc 45
 pdb_setrepseq 49

さ

作成
 トランザクション・ログ 116

し

シーケンス 82
 マーク解除 44, 47
 マーク付け 44-46
シーケンスのマーク解除 47
シーケンスのマーク付け 45

せ

設定パラメータ
 pdb_dflt_object_repl 46

ち

遅延更新 5, 87, 115

つ

通信
 JDBC ドライバ 8

て

データ型
UDB 127

と

トランケーション
プロシージャ 135
トランザクション・ログ
Microsoft SQL Server 向け Replication Agent
107, 111
Replication Agent for Oracle 78
Replication Agent for UDB 133
隠しテーブル 82, 109
作成 116
トランケート 83
プライマリ・テーブル・オブジェクト
124
ベース・オブジェクト 134
マーク付けされたオブジェクト・テーブル
136

ひ

表記規則
スタイル 1
構文 1

ふ

複数 Replication Agent インスタンスのテーブル
80
複数 Replication Agent マーク付けテーブルのテ
ーブル 81
複数 Replication Agent マーク付けプロシージャ
のテーブル 81

プライマリ・データベース
Microsoft SQL Server 85
Oracle データベース・サーバ 5
Replication Agent ユーザ ID 93, 116
UDB 115
プライマリ・テーブル
UDB でのマーク付け 116
トランザクション・ログ・オブジェクト
124
プライマリ・テーブルのマーク付け
UDB 116
分割されたテーブル 54

へ

ベース・オブジェクト、トランザクション・
ログ 133, 134

ま

マカ隠しテーブル 82, 109
マーク付きのプロシージャ 82
マーク付けされたオブジェクトのテーブル
UDB 136

ゆ

ユーザ ID
プライマリ・データベース 93, 116

ろ

ログベースの Replication Agent
テーブルのマーク付け 116

