



管理ガイド

Replication Agent™ 15.7.1

Linux、Microsoft Windows、および UNIX 版

ドキュメント ID：DC01883-01-1571-01

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 について	5
複製システムの基本概念	5
トランザクションの複製	5
複製システムのコンポーネント	6
Replication Agent の機能	8
Replication Agent インスタンス	8
Replication Agent の通信方法	9
Replication Agent コンポーネント	10
複数の Replication Agent	11
設定	13
Replication Agent インスタンスの作成	13
Replication Agent のインスタンス・ディレクト リ	14
Replication Agent ユーティリティ	14
ユーティリティを使用するための準備	15
コマンド・ライン・インタフェース	16
Administrator GUI ユーティリティ	37
Replication Agent の起動	40
起動の準備	40
文字設定	41
ra ユーティリティを使用した Replication Agent の起動	43
Administrator GUI を使用した Replication Agent インスタンスの起動	44
RUN スクリプトを使用した Replication Agent インスタンスの起動	46

agt_service を使用した Replication Agent の起 動	47
Windows サービス・ツールからの Replication Agent の起動	48
Replication Agent 管理ポート	48
Interfaces ファイルのエントリ	48
isql を使用した Replication Agent インスタンス へのログイン	49
Replication Agent の管理者ログインの作成また は変更	50
Replication Agent コネクティビティの設定	51
Replication Agent のプライマリ・データベース のユーザ・ログインの作成	51
Replication Agent への Replication Server ユー ザ・ログインの作成	52
RSSD ユーザのログイン名	53
コネクション設定パラメータの設定	55
ネットワーク接続テスト	60
Replication Agent のコネクションの確認	61
Replication Agent の初期化	61
オブジェクト名プレフィクス	61
Replication Agent の初期化	62
プライマリ・データベース・オブジェクトのマーク 付け	63
プライマリ・データベースのテーブルへの マーク付け	64
プライマリ・データベースのストアド・プロ シージャへのマーク付け	66
プライマリ・データベースの LOB カラムの複 写の有効化	67
DDL 複写	68
Replication Agent の設定要件	68

管理	71
Replication Agent インスタンスのステータスの判断 ...	71
Replication Agent の状態	71
Replication Agent のステータスの変更	74
Replication Agent の統計	75
Replication Agent インスタンスの停止	75
Replication Agent インスタンスの停止	76
複製の開始	76
Replication Agent での複製の停止	77
Replication Agent インスタンスのクワイス	78
Replication Agent インスタンスのサスペンド	79
複製システム・コンポーネントの移動	80
同じプラットフォームへの Replication Agent インスタンスの移動	80
Replication Agent の管理	82
Replication Agent インスタンスの初期化	83
Replication Agent インスタンスの初期化解除	86
トランザクション・ログのトランケーション	89
プライマリ・データベースのバックアップ	91
Replication Agent システム・データベース (RASD) の管理	91
RASD	91
RASD の更新	93
ログ・デバイス・レポジトリの更新	94
RASD のバックアップ	96
RASD のリストア	97
RASD のトランケーション	98
デフォルトのホストとポート番号の設定	99
複製済みトランザクションとプロシージャの特定	99
テーブルのマーク付けとマーク解除	100
DDL 複製の有効化と無効化	105

マーク付けされたテーブルの複製の有効化と無効化	106
LOB カラムの複製の有効化と無効化	108
ストアド・プロシージャのマーク付けとマーク解除	109
ストアド・プロシージャの複製の有効化と無効化	114
Oracle シーケンスのマーク付けとマーク解除 ..	115
シーケンスの複製の有効化と無効化	118
Replication Agent のチューニング	119
カスタマイズにチューニング	120
パラメータと変数	121
最適なパフォーマンスのチューニング	123
エンドツーエンドのチューニング	125
トラブルシューティング	127
コマンド・エラー	127
Replication Agent のエラー	128
プライマリ・データベース・オブジェクトに、複製のマークがついているかの確認	128
Replication Agent の現在のステータスの確認 ..	129
プライマリ・データベース・ログへのアクセス	131
Replication Agent ログ	132
Replication Agent オペレーションとパフォーマンスを確認するための ra_statistics の使用 ..	133
使用可能なメモリ	134
LTL のデバッグ	137
省略された DDL コマンド	138
JVM スレッド・ダンプ	139
Replication Server のトラブルシューティングの問題	140
複製定義とサブスクリプションの確認	141

Replication Server のステータスとオペレー ションのチェック	141
rs_username ログインに適切なパーミッショ ンが付与されたかの確認	142
ステーブル・キューの確認	143
パフォーマンスのモニタリング	143
プライマリ・データへのサブスクリプションのマテリア ライズ	145
マテリアライゼーション	145
バルク・マテリアライゼーション	145
データのアンロードとロード	146
アトミック・バルク・マテリアライゼーション	147
アトミック・バルク・マテリアライゼーショ ンの準備	147
アトミック・バルク・マテリアライゼーショ ンの実行	147
ノンアトミック・バルク・マテリアライゼーション	149
ノンアトミック・バルク・マテリアライゼー ションの準備	150
ノンアトミック・バルク・マテリアライゼー ションの実行	150
追加の説明や情報の入手	153
サポート・センタ	153
Sybase EBF と Maintenance レポートのダウンロー ド	153
Sybase 製品およびコンポーネントの動作確認	154
MySybase プロファイルの作成	154
アクセスビリティ機能	155
用語解説	157
索引	167

目次

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 について

Replication Agent により、Sybase 以外 (異機種) のデータ・サーバを Sybase 複写テクノロジーに基づいた複写システムのプライマリ・データ・サーバとして使用できるようになるため、Replication Server® の機能を拡張することができます。

複写システムの基本概念

トランザクションの複写を使用して、レプリケート・データベースと呼ばれる独立したデータベースにデータを維持します。レプリケート・データベースには、プライマリ・データベースのデータの正確な最新のコピーまたはサブセットが格納されます。

プライマリ・データベース内の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、そのテーブルのデータを変更するトランザクションが複写用に取り出されます。プライマリ・データベースでトランザクションが処理され、そのトランザクション (すべてのオペレーションを含む) のコピーがトランザクション・ログに格納されます。

ストアド・プロシージャが複写するようにマーク付けされている場合は、そのストアド・プロシージャがプライマリ・データベースで呼び出されると、そのプロシージャの呼び出しで指定されたすべてのパラメータ値が取り出され、トランザクション・ログに記録されます。マーク付けされたストアド・プロシージャによってプライマリ・データベース内のマーク付けされたテーブルのデータに影響を与えるトランザクションが生成された場合、そのストアド・プロシージャによって生成されたトランザクションは無視されるため、プロシージャの呼び出しだけが複写されます。

トランザクションの複写

Sybase 複写システムで複写用に取り出されたイベントは、プライマリ・データベースの実際のトランザクションと直接対応していても、すべて「トランザクション」とみなされます。たとえば、マークが付いたテーブルとマーク解除テーブルの両方に影響があるトランザクションの場合、マークが付いたテーブルに影響を与えるオペレーションのみが複写用に取り出されます。マーク解除テーブルのオペレーションは無視されます。

複写用に取り出されたすべてのデータ変更オペレーションは、トランザクションのコンテキスト内に存在します。すなわち、コミットされたトランザクション・オペレーションのみが複写され、ロールバックされたトランザクション・オペレーションは複写されません。

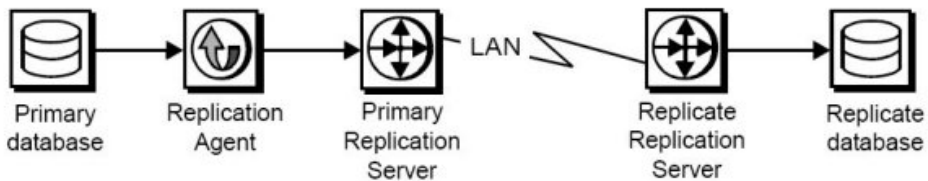
Replication Agent について

Sybase 複写システムによって複写されるデータ変更イベントは実際にはオペレーションですが、それらのオペレーションはアトミック・コレクションにグループ化され、プライマリ・データベースでコミットされたトランザクションの結果を表します。

複写システムのコンポーネント

複写システムの基本コンポーネントは、プライマリ・データベース、レプリケート・データベース、および Replication Server です。

図 1 : 標準の Sybase 複写システム



プライマリ・データベース

プライマリ・データベースは、レプリケート・データベースのデータを変更するトランザクションのソースです。トランザクションは、テーブルまたはプロシージャごとに複写されます。

プライマリ・データベースに複写するようにマーク付けされたテーブルは、プライマリ・テーブルと呼ばれます。プライマリ・テーブルには、このテーブルのデータに影響を与えるトランザクションを Replication Agent が特定して複写できるように、複写のマークを付ける必要があります。

プライマリ・テーブル内のラージ・オブジェクト (LOB) カラムは、プライマリ・テーブルとは別に複写を有効にする必要があります。プライマリ・テーブル内の LOB カラムは選択的に複写できます。

ストアド・プロシージャの呼び出しを複写するには、Replication Agent がプライマリ・データベースでそのプロシージャを特定して複写できるように、そのプロシージャに複写のマークを付ける必要があります。

Replication Agent

Replication Agent は、Sybase 複写システムのコンポーネントで、プライマリ・データベース内の複写トランザクションを収集し、レプリケート・データベースに分配する目的で Replication Server に転送します。

Replication Agent は、プライマリ・データベースのトランザクション・ログを読み取って、ログ転送言語 (LTL) 出力を生成します。LTL は、Replication Server が複写

トランザクションを処理し、複製システム全体に分散するために使用する言語です。

より洗練された複製機能を利用して、より効率的な LTL を生成するには、プライマリ Replication Server の Replication Server システム・データベース (RSSD) に格納されている情報を使用するように Replication Agent を設定します。

Replication Agent は、プライマリ・データ・サーバで保持されているネイティブ・トランザクション・ログから、トランザクションの複製に必要な情報を取得します。

Replication Agent は、以下のプライマリ・データベースの種類のプライマリ・データベースに対してログベースのソリューションを使用します。

- Oracle
- Microsoft SQL Server
- Linux、UNIX、および Windows 向け IBM DB2

注意： IBM DB2 では、プロシージャ複製と DDL 複製は使用できません。

Replication Server

プライマリ・データベースから (すなわち、Replication Agent から直接) 複製トランザクションを受信する Replication Server は、プライマリ Replication Server と呼ばれます。複製トランザクションをレプリケート・データベースに送信する Replication Server は、レプリケート Replication Server と呼ばれます。

注意： 単純な複製システムでは、1 つの Replication Server がプライマリ Replication Server とレプリケート Replication Server の両方として機能します。

Replication Agent から LTL を受信すると、プライマリ Replication Server は、直接またはレプリケート Replication Server を介して複製トランザクションをレプリケート・データベースに送信します。レプリケート Replication Server は、複製トランザクションを受信時の LTL からレプリケート・データベースのネイティブ言語に変換してから、処理目的でレプリケート・データ・サーバに送信します。レプリケート・データベースが複製トランザクションを正常に処理すると、レプリケート・データベースはプライマリ・データベースと同期されます。

各 Replication Server は、トランザクション・オペレーションをステーブル・キューに保持して、できるだけ早く他の Replication Server またはレプリケート・データベースに配布します。この結果、Replication Server によって、Replication Agent から正常に受信したすべてのトランザクションは、サブスクリプションを作成するレプリケート・データベースに適切に配布されるようになります。

各 Replication Server では、Replication Server システム・データベース (RSSD) と呼ばれるデータベースを使用して、複製システムのデータとメタデータを格納しま

す。Replication Agent は、RSSD に格納されている情報の一部を使用して、高度な複製機能を提供します。

Replication Agent の機能

Replication Agent は、Replication Server を使用したプライマリ・データベースからのトランザクションの複製をサポートしています。

注意： Replication Agent でサポートされている Oracle、Microsoft SQL Server、および IBM DB2 の特定のバージョンについては、『Replication Agent リリース・ノート』を参照してください。

Replication Agent は、プライマリ・データ・サーバ、プライマリ Replication Server、およびその他の複製システムのコンポーネントとは独立したスタンドアロンのアプリケーションとして実行されます。

Replication Agent は、プライマリ・データベースまたはその他の複製システムのコンポーネントと同じホスト・マシン上、あるいはプライマリ・データベースへのネットワーク・アクセスが存在するその他の複製システムのコンポーネントとは別のマシン上のどちらに存在していてもかまいません。Replication Agents は、プライマリ・データベースのトランザクション・ログにアクセス可能なマシン上で実行される必要があります。UDB 向け Replication Agent の場合は、IBM DB2 クライアントを使用することでこの条件を満たすことができます。

Replication Agent は Replication Manager (RM) と互換性があります。Replication Agent インスタンスは、RM で設定、管理、および監視できます。また、Sybase Tabular Data Stream™ (TDS) プロトコル (**isqli** など) と通信可能な Open Client™ アプリケーションを使用して、Replication Agent インスタンスを完全に設定、管理、および監視できます。

Replication Agent インスタンス

トランザクションを複製するプライマリ・データベースごとに Replication Agent のインスタンスを作成する必要があります。各 Replication Agent インスタンスは、独自の設定およびログ・ファイルと管理ポートを持ち、プライマリ・データベースおよびプライマリ Replication Server とのコネクションが可能な独立したアプリケーションです。

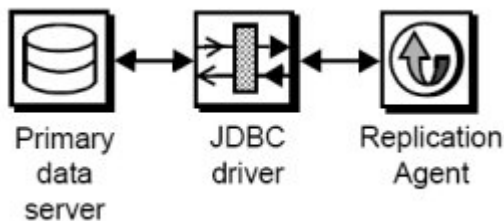
特定のプライマリ・データベース用に作成された Replication Agent インスタンスは、このマニュアルでは次のように記載されます。

- Oracle Database Server – Replication Agent for Oracle
- Microsoft SQL Server – Replication Agent for Microsoft SQL Server
- Linux、UNIX、および Windows 用の IBM DB2 – Replication Agent for UDB

Replication Agent の通信方法

Replication Agent は、すべての通信に JDBC™ (Java Database Connectivity) プロトコルを使用します。ただし、サポートされている一部のデータベースでは、ODBC (Open Database Connectivity) プロトコルが必要です。Replication Agent では、プライマリ・データベースと接続するときに、データベース・ベンダによって提供される JDBC ドライバまたは JDBC/ODBC ブリッジに接続します。

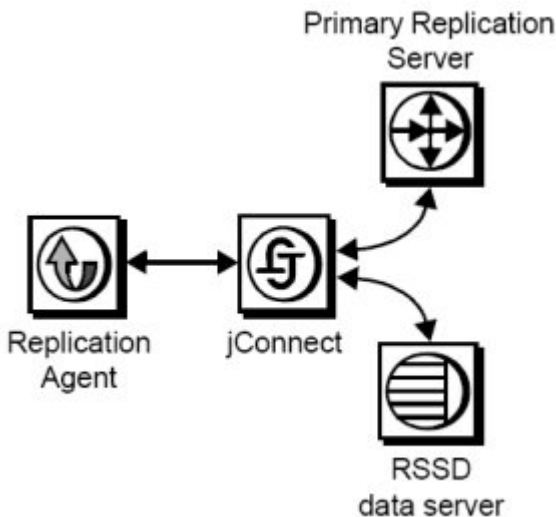
図 2 : JDBC ドライバを使用した Replication Agent プライマリ・データベースの通信



Replication Agent は、Adaptive Server Enterprise や Replication Server などの Open Client および Open Server™ アプリケーションと通信するために Sybase JDBC ドライバ (jConnect™ for JDBC) を使用します。各 Replication Agent インスタンスでは、jConnect for JDBC の 1 つのインスタンスを使用します。

トランザクションの複製中、Replication Agent はプライマリ・データベースとプライマリ Replication Server との通信を維持しますが、プライマリ Replication Server の RSSD とも接続して、複製定義データを取得する場合があります。

図 3 : Replication Server およびその RSSD との Replication Agent の通信



Replication Agent コンポーネント

Replication Agent は、複製のためにプライマリ・データベースからトランザクションを送信するために必要なすべてのオペレーションを連携して実行するコンポーネントのセットで構成されます。

以下は主な Replication Agent コンポーネントです。

- Log Reader – 複製のためのトランザクションを取得するためにプライマリ・データベースのトランザクション・ログを読み込みます。
- Log Transfer Interface (LTI) – ログ転送言語 (LTL) を生成し、プライマリ Replication Server に送信します。
- Log Administrator – Replication Agent トランザクション・ログおよびトランザクション・ログ・オブジェクトを管理します。
- Log Transfer Manager (LTM) – 他のすべてのコンポーネントを管理し、そのオペレーションと操作を調整します。

このプロセスでは次のようになります。

1. Log Reader コンポーネントは、プライマリ・データベースのトランザクション・ログからトランザクション・データを取得します。
2. Log Reader は、変更セット・データを生成し、変更セットを LTI に渡します。
3. LTI コンポーネントは、Log Reader から渡された変更セット・データを処理し、LTL を生成して、プライマリ Replication Server に送信します。

LTI コンポーネントでも、プライマリ Replication Server からメッセージを受信します。

LTM コンポーネントは、プライマリ・データベースとプライマリ Replication Server 間のデータのフローに含まれていませんが、その他の Replication Agent コンポーネントのアクティビティを調整し、これらのコンポーネントから発生するすべてのエラーを処理します。

管理ポート

Replication Agent の管理ユーザ・インタフェースでは、管理ポートを使用します。

この管理ポートを使用することで、Replication Agent が Open Server アプリケーションである場合と同様に、Open Client アプリケーションは Replication Agent インスタンスにログインできます。ログイン後、管理クライアントは、Replication Agent インスタンスを制御、管理、および監視するためのコマンドを発行できます。

管理ポートはクライアントとの通信に Sybase JDBC ドライバ (jConnect for JDBC) を使用します。

管理ポートでは、管理クライアントから Replication Agent コンポーネントにコマンドが渡されます。また、Replication Agent からのメッセージが処理され、これらのメッセージがクライアントに渡されます。

Java 要件

Replication Agent には Java Runtime Environment (JRE) が含まれているため、Replication Agent ホスト・マシンとして機能するコンピュータは JRE 要件を満たす必要があります。

JRE のインストールおよび設定の詳細については、以下を参照してください。

- 『Replication Agent インストール・ガイド』
- 『Replication Agent リリース・ノート』

複数の Replication Agent

Replication Agent for Oracle では、Replication Server Multi-Path™ Replication を使用して複数の Replication Agent インスタンスを使用できます。

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 Agent プライマリ・データベース・ガイド』を参照してください。

設定

インストール後に Replication Agent を設定し、複製システムでデータを複製する準備が整ったことを確認して複製を開始します。

注意：ここで説明する手順では、『Replication Agent インストール・ガイド』、および使用しているプラットフォーム用の『Replication Server インストール・ガイド』と『Replication Server 設定ガイド』の説明に従って、Replication Agent ソフトウェアと Replication Server ソフトウェアがインストールされていることを前提としています。

Replication Agent インスタンスの作成

Replication Agent ソフトウェアをインストールした場合、トランザクションを複製するプライマリ・データベースのそれぞれに Replication Agent のインスタンスを1つ作成する必要があります。

各 Replication Agent インスタンスは独立したプロセスで、インスタンス・ディレクトリには、設定ファイル、システム・ログ・ファイル、およびスクリプト・ファイルが含まれています。また、各 Replication Agent インスタンスによって、プライマリ・データベースに複数のテーブルとストアド・プロシージャが作成されます。また、Oracle および Microsoft SQL Server 用の Replication Agent によって、Replication Agent システム・データベース (RASD) にオブジェクトが作成されます。各 Replication Agent インスタンスによって、プライマリ・データ・サーバ、プライマリ Replication Server、および RSSD とのコネクションがそれぞれ管理されます。

Replication Agent インスタンスを作成する場合、以下を指定する必要があります。

- ユニークなインスタンス (サーバ) の名前
- ユニークな管理ポートのクライアント・ソケット・ポート番号
- インスタンスでサポートされているプライマリ・データベースの種類

1つのホスト・マシンで1つ以上の Replication Agent インスタンスを作成して実行できますが、各インスタンスの名前とポート番号はユニークである必要があります。

参照：

- ra_admin (19 ページ)
- コマンド・ラインを使用した Replication Agent インスタンスの作成 (32 ページ)

- リソース・ファイルを使用したインスタンスの作成 (27 ページ)

Replication Agent のインスタンス・ディレクトリ

Replication Agent のインスタンス・ディレクトリは、Replication Agent インスタンスを作成すると、Replication Agent ベース・ディレクトリ内に作成されます。

Replication Agent ベース・ディレクトリ (RAX-15_5) とインストール・ディレクトリ (sybase) は、Replication Agent ソフトウェアをインストールすると作成されます。

注意： 単一のインストール (単一のホスト・マシン) で、複数の Replication Agent インスタンスがサポートされます。各インスタンス・ディレクトリは、Replication Agent ソフトウェアをインストールしたときに作成される Replication Agent ベース・ディレクトリ内にあります。

Replication Agent ユーティリティ

Replication Agent で提供されるユーティリティは、`agt_service`、`ra`、および `ra_admin` です。

- **agt_service** – Replication Agent インスタンスを Windows サービスとして実行します。
- **ra** – Replication Agent インスタンスを起動するか、Replication Agent ソフトウェア・バージョン番号を返します。
- **ra_admin** – Replication Agent インスタンスを作成、コピー、確認、および削除するか、マシンにインストールされたすべての確認可能な Replication Agent インスタンスをリストします。

Replication Agent ユーティリティは、Windows プラットフォームの場合はバッチ・ファイルとして、UNIX プラットフォームの場合はシェル・スクリプトとして提供されます。ユーティリティ・ファイルは、Replication Agent ベース・ディレクトリ内の `bin` サブディレクトリにあります。

注意： Windows プラットフォームの場合、`run` スクリプトを実行するときに、**ra-i my_ra** 拡張子を省略できます。UNIX の場合は、**ra.sh -i my_ra** 拡張子を常に含める必要があります。

ユーティリティの情報を取得するには、**-help** オプションと **ra_admin** または **ra** を使用します。

参照：

- コマンド・ライン・インタフェース (16 ページ)

ユーティリティを使用するための準備

Replication Agent ユーティリティを使用するには、以下の作業を実行します。

前提条件

- Replication Agent ホスト・マシンのオペレーティング・システムにログインするには、Replication Agent インストール・ディレクトリおよびすべてのサブディレクトリの **execute** パーミッションを持っているユーザ・ログイン (“sybase” ユーザなど) を使用します。
- SYBASE 環境を設定してから、任意のユーティリティを起動します。

SYBASE 環境スクリプトは、Microsoft Windows プラットフォームの場合はバッチ・ファイル (SYBASE.bat)、UNIX プラットフォームの場合はシェル・スクリプト (SYBASE.sh または SYBASE.csh) として提供されます。

手順

1. 適切なパーミッションを持つユーザ・ログインを使用して、Replication Agent ホスト・マシンのオペレーティング・システムにログインします。
2. オペレーティング・システムのコマンド・ウィンドウを開きます。
3. オペレーティング・システムのプロンプトで、Replication Agent インストール・ディレクトリに移動します。
 - Microsoft Windows の場合：

```
cd c:¥sybase
```

c:¥sybase は Replication Agent インストール・ディレクトリのパスです。
 - UNIX の場合：

```
cd /opt/sybase
```

/opt/sybase は Replication Agent インストール・ディレクトリのパスです。
4. Replication Agent インストール・ディレクトリで、**SYBASE** 環境スクリプトを起動します。
 - Microsoft Windows の場合：

```
SYBASE
```
 - UNIX で、Bourne および Korn シェルの場合：

```
. SYBASE.sh
```
 - UNIX で、C シェルの場合：

```
source SYBASE.csh
```

注意： UNIX プラットフォームでは、Replication Agent 管理者 (または “sybase” ユーザ) の `.login` ファイルにソース・コマンド・ラインを挿入して、Replication Agent ホスト・マシンにログインしたときに、SYBASE 環境が自動的に設定されるように指定できます。

コマンド・ライン・インタフェース

Windows または UNIX コマンド・ラインから Replication Agent を管理するには、コマンド・ライン・インタフェースを使用します。

agt_service

Replication Agent を Windows サービスとして実行します。

構文

```
agt_service[.cmd]
[-install inst_name [-startup state] [-charset charset] [-maxmem mem] ] |
[-remove inst_name ] |
[-start inst_name ] |
[-stop inst_name ] |
[-h]
```

パラメータ

- **-install *inst_name*** – Windows サービスを作成する Replication Agent インスタンスを指定します。*inst_name* は、既存の Replication Agent インスタンスの名前です。
- **-startup *state*** – (オプション) Replication Agent インスタンスの起動時の状態を指定します。

有効な **-startup *state*** の値は次のとおりです。

- **-admin** – (デフォルト) Replication Agent インスタンスを Admin 状態で起動します。
- **-replicate** – Replication Agent インスタンスを複製状態で起動します。
- **-charset *charset*** – (オプション) JVM で使用される Java 文字。このオプションは `RA_JAVA_DFLT_CHARSET` 環境変数に対応します。
- **-maxmem *mem*** – (オプション) この Replication Agent インスタンスで使用される最大メモリ量。このオプションは `RA_JAVA_MAX_MEM` 環境変数に対応します。

- **-remove *inst_name*** – Windows サービスを削除する Replication Agent インスタンスを指定します。 *inst_name* は、既存の Replication Agent インスタンスの名前です。
- **-start *inst_name*** – Windows サービスを起動する Replication Agent インスタンスを指定します。 *inst_name* は、既存の Replication Agent インスタンスの名前です。
- **-stop *inst_name*** – Windows サービスを停止する Replication Agent インスタンスを指定します。 *inst_name* は、既存の Replication Agent インスタンスの名前です。
- **-h *inst_name*** – コマンドの使用方法の情報を返します。

例

- **例 1** – "my_ra" という名前の Replication Agent インスタンスの Windows サービスを複写状態で作成します。

```
agt_service -install my_ra -startup replicate
```
- **例 2** – "my_ra" という名前の Replication Agent インスタンスの Windows サービスを削除します。

```
agt_service -remove my_ra
```
- **例 3** – "my_ra" という名前の Replication Agent インスタンスの Windows サービスを起動します。

```
agt_service -start my_ra
```
- **例 4** – "my_ra" という名前の Replication Agent インスタンスの Windows サービスを停止します。

```
agt_service -stop my_ra
```

使用法

agt_service の機能は次のとおりです。

- 指定された Replication Agent インスタンスを実行するための Windows サービスを作成する。
- Replication Agent インスタンスを実行している Windows サービス・インスタンスを削除する。
- Windows サービスを起動することによって、対応する Replication Agent インスタンスを起動する。
- Windows サービスを停止することによって、対応する Replication Agent インスタンスを停止する。

agt_service を実行するには、オペレーティング・システムのプロンプトでコマンドとして呼び出します。

参照：

- Replication Agent インスタンスの停止 (75 ページ)
- Replication Agent の起動 (40 ページ)

ra

Replication Agent を起動します。

構文

```
ra [-help | -i inst_name [-state] | -v]
```

パラメータ

- **-help** – コマンドの使用方法の情報を返します。

注意： オプションを指定しないで **ra** を呼び出すと、コマンドの使用方法の情報を返すことができます。

- **-i inst_name** – 起動する Replication Agent インスタンスを指定します。 *inst_name* は、既存の Replication Agent インスタンスの名前です。
- **-state** – Replication Agent インスタンスの起動時の状態を指定するキーワードです。
有効な **-state** の値は次のとおりです。
 - **-admin** – (デフォルト) Replication Agent インスタンスを Admin 状態で起動します。
 - **-replicate** – Replication Agent インスタンスを複製状態で起動します。
- **-v** – Replication Agent ソフトウェアのバージョン番号を返します。

例

- **例** – "my_ra" という名前の Replication Agent インスタンスを複製状態で起動します。

```
ra -i my_ra -replicate
```

使用法

Replication Agent の **ra** ユーティリティの機能は次のとおりです。

- 指定された Replication Agent インスタンスを起動する。
- Replication Agent ソフトウェアのバージョン文字列を返す。

ra を実行するには、オペレーティング・システムのプロンプトでコマンドとして呼び出します。

参照：

- Replication Agent の起動 (40 ページ)
- ra_admin (19 ページ)

起動エラー

Replication Agent では、起動時にエラーが発生する場合があります。

- Microsoft Windows プラットフォームの場合、起動エラーはオペレーティング・システムのコマンド・ウィンドウに表示されます。
- UNIX プラットフォームの場合、起動エラーはオペレーティング・システムのコマンド・ウィンドウに表示され、Replication Agent のシステム・ログに記録されます。

参照：

- トラブルシューティング (127 ページ)

ra_admin

Replication Agent インスタンスを管理します。

構文

```
ra_admin [option [create options]] [inst_name]
```

パラメータ

- **-b** – Replication Agent インストール・ディレクトリの完全なパスを返します。
- **-c *inst_name*** – 指定された名前 (*inst_name*) を使用して新しい Replication Agent インスタンスを作成します。

inst_name 文字列は有効なサーバ名であり、ホスト・マシン上でユニークでなければなりません。

-c オプションを使用するときは、次のいずれかのオプションが必要です。

- **-p** と **-t**
- **-p** と **-f**

既存の Replication Agent 設定をコピーするときに **-f** オプションを指定する場合は、**-t** オプションを指定する必要がありません。**-f** オプションを指定すると、既存の Replication Agent インスタンスに指定されたプライマリ・データベースのタイプが新しい Replication Agent インスタンスの設定にコピーされます。

- **-h** – コマンドの使用方法の情報を返します。
- **-p *port_num*** – Replication Agent インスタンスの管理ポートのクライアント・ソケット・ポート番号を指定します。

port_num は有効なポート番号であり、Replication Agent ホスト・マシン上でユニークでなければなりません。Oracle および Microsoft SQL Server の場合、Replication Agent では 2 番目のポート *port_num+1* が RASD 用に使用可能である必要があります。

- **-t database** – プライマリ・データベースが格納されているデータ・サーバのタイプを識別します。

database 文字列は次のいずれかです。

- **oracle** – Oracle データベース・サーバ
- **mssql** – Microsoft SQL Server
- **ibmudb** – Linux、UNIX、Windows 向け IBM DB2

注意： *database* の値の大文字と小文字は区別されません。

-c を使用すると、既存の Replication Agent インスタンスの設定ファイルに基づいて新しい Replication Agent インスタンスの設定が行われるように指定できます。そのためには、**-f** オプションを使用します。

- **-f old_inst** – 新しい Replication Agent インスタンス用に既存の Replication Agent インスタンスの設定をコピーします。

old_inst 文字列は、新しい Replication Agent インスタンス用にコピーする設定の既存の Replication Agent インスタンスの名前です。

既存の Replication Agent 設定をコピーするときに **-f** オプションを指定する場合は、**-t** オプションを指定する必要がありません。**-f** オプションを指定すると、既存の Replication Agent インスタンスに指定されたプライマリ・データベースのタイプが新しい Replication Agent インスタンスの設定にコピーされます。

注意： **-f** オプションを使用すると、一部の設定パラメータがデフォルト値に設定されます。

- **-d inst_name** – 指定された Replication Agent インスタンスを削除します。

inst_name 文字列は、既存の Replication Agent インスタンスの名前にしてください。

ra_admin に **-d** オプションを指定して呼び出すと、Replication Agent インストール・ディレクトリから、指定されたインスタンスに関連したすべてのサブディレクトリが削除されます。

注意： Windows プラットフォームで、インスタンスの削除時に Replication Agent のインスタンスに関連したファイルまたはディレクトリにアクセスしているアプリケーションがあると、開いているファイルまたはディレクトリは削除されません。削除されていないファイルまたはディレクトリについて通知するエラー・メッセージが表示されます。

Microsoft Windows プラットフォームでファイルまたはディレクトリのアクセス競合が発生した後に Replication Agent のインスタンスの削除を完了するには、次の操作を実行する必要があります。

- ファイルまたはディレクトリがどのアプリケーションでも開いていないことを確認します。
- ファイルまたはディレクトリを手動で削除します。
- **-l (小文字の L)** – 確認可能なすべての Replication Agent インスタンスをリストします。
- **-v *inst_name*** – 指定された Replication Agent インスタンスの完全なディレクトリ構造を確認します。 *inst_name* 文字列は、既存の Replication Agent インスタンスの名前にしてください。
- **-vr *res_file*** – 指定されたリソース・ファイル (*res_file*) を検証しますが、Replication Agent インスタンスの作成または環境への変更を行いません。
- **-r *res_file*** – 指定されたリソース・ファイル (*res_file*) の内容に基づいて Replication Agent インスタンスを作成します。
- **-u *upgrade_option*** – 指定されたアップグレード・オプション (*upgrade_option*) に基づいて Replication Agent インスタンスをアップグレードします。
 - *source_installation_dir* – 現在の製品インストール・ディレクトリから、指定されたソース・インストール・ディレクトリにあるすべての Replication Agent インスタンスをアップグレードします。
 - *source_instance_dir* – 現在の製品インストール・ディレクトリから、指定されたソース・インスタンス・ディレクトリにある 1 つの Replication Agent インスタンスをアップグレードします。現在の製品インストール・ディレクトリ内でインスタンスをコピーしてから更新します。
 - **all** – 現在の製品インストール・ディレクトリにあるすべての Replication Agent インスタンスをアップグレードします。必要に応じて、アップグレードの前に、エラーのリカバリ用に設定ファイルがバックアップされます。エラーが発生した場合は、アップグレードがロールバックされます。
all オプションに必要な領域は比較的少なくなります。これはアップグレードが、コピーに対してではなく、現在の製品インストール・ディレクトリ内にあるインスタンスに対して直接実行されるためです。ただし、同じ理由のため、アップグレードを元に戻すことがさらに難しくなります。
- ***instance_name* | *instance= instance_name*** – 現在の製品インストール・ディレクトリにある指定されたインスタンスだけをアップグレードします。アップグレードはコピーに対してではなく、指定されたインスタンスに対して直接実行されます。必要に応じて、アップグレードの前に、エラーのリカバリ用に設定ファイルがバックアップされます。エラーが発生した場合は、アップグレードがロールバックされます。

使用法

注意：SYBASE 環境を設定してから、Replication Agent の **ra_admin** ユーティリティを起動します。

Replication Agent の **ra_admin** ユーティリティの機能は次のとおりです。

- Replication Agent インスタンスの作成、コピー、削除、検証を行う。
- Replication Agent ホスト・マシン上にあるすべての有効な Replication Agent インスタンスをリストする。
- Replication Agent インストール・ディレクトリのパスを返す。
- \$SYBASE/RAX-15_5/admin_logs ディレクトリに出力ファイルを作成する。ファイル名のフォーマットは adminmmdyyy_hhmmss.log です。mmdyyy は現在の日付、hhmmss は現在の時刻です。
- Replication Agent インスタンスをアップグレードする。

ra_admin を実行するには、オペレーティング・システムのプロンプトでコマンドとして呼び出します。

注意：オプションを指定しないで **ra_admin** を呼び出すと、コマンドの使用方法の情報を返すことができます。

参照：

- Replication Agent インスタンスの作成 (13 ページ)
- Administrator GUI ユーティリティ (37 ページ)
- リソース・ファイルの作成 (24 ページ)
- リソース・ファイルの編集 (25 ページ)
- リソース・ファイルの検証 (25 ページ)
- リソース・ファイルを使用したインスタンスの作成 (27 ページ)
- ユーティリティを使用するための準備 (15 ページ)

Replication Agent インスタンスの作成

Replication Agent ソフトウェアのインストール後に Replication Agent インスタンスを作成します。

Replication Agent ソフトウェアをインストールした後はいつでも、Administrator GUI ユーティリティを使用するか、または **ra_admin** に **-c** オプションを指定して呼び出します。

```
ra_admin -c new_inst -p port_num {-t database|-f old_inst}
```

- *new_inst* は、作成する新しい Replication Agent インスタンスの名前です。

- *port_num* は、新しい Replication Agent インスタンスの管理ポートのクライアント・ソケット・ポート番号です。Oracle および Microsoft SQL Server の場合は、2 番目のポート *port_num+1* が RASD 用に使用できることを確認してください。
- *database* は、プライマリ・データベースを格納するデータ・サーバのタイプです。
- *old_inst* は、新しい Replication Agent インスタンスに設定をコピーする既存の Replication Agent インスタンスの名前です。

構文の説明は次のとおりです。

参照：

- Administrator GUI ユーティリティ (37 ページ)

リソース・ファイルを使用した Replication Agent インスタンスの作成

リソース・ファイルを使用して Replication Agent インスタンスを作成します。リソース・ファイルによる Replication Agent インスタンスの作成および検証をサポートするコマンド・ライン・パラメータを指定して、**ra_admin** を呼び出します。

```
ra_admin {-vr res_file | -r res_file}
```

- **-vr res_file**
指定されたリソース・ファイル (*res_file*) を検証しますが、Replication Agent インスタンスの作成または環境への変更を行いません。
- **-r res_file**
指定されたリソース・ファイル (*res_file*) の内容に基づいて Replication Agent インスタンスを作成します。

構文の説明は次のとおりです。

リソース・ファイルは、**ra_admin** によって作成される Replication Agent インスタンスの設定情報が格納された ASCII テキスト・ファイルです。

リソース・ファイルの **ra_admin** パラメータでは、Replication Agent インスタンスを作成する他に、次のオプションを指定できます。

- プライマリ・データ・サーバにインスタンスのユーザ・ログインを作成し、必要なすべてのパーミッションを付与する。
- 新しいインスタンスの作成後に、そのインスタンスを起動する。
- 新しいインスタンスの起動後に、そのインスタンスを初期化する。

注意： **ra_admin -vr** を使用してリソース・ファイルを検証する場合は、他のアクションは何も実行されず、Replication Agent インスタンスも作成されません。

リソース・ファイルの作成

指定されたテンプレートのいずれかを使用して、リソース・ファイルを作成します。

前提条件

リソース・ファイルのテンプレートである `mssql.rs` (Microsoft SQL Server)、`oracle.rs` (Oracle)、および `ibmudb.rs` (Linux、UNIX、および Windows 用の IBM DB2) を Replication Agent インストール・ディレクトリの `init` サブディレクトリで検索します。例：

```
C:\sybase\RAX-15_5\init\oracle.rs
```

または

```
C:\sybase\RAX-15_5\init\mssql.rs
```

または

```
C:\sybase\RAX-15_5\init\ibmudb.rs
```

リソース・ファイル・テンプレートには、各設定パラメータとその値に関するコメントが含まれています。

注意： リソース・ファイルを使用して Replication Agent インスタンスを作成する前に、各リソース・ファイルを検証することをおすすめします。

手順

1. 目的のリソース・ファイル・テンプレートを編集対象の別のファイルにコピーして、新しいリソース・ファイルを作成します。例：

```
cp oracle.rs pubs2.rs
```

`pubs2.rs` は、作成する新しいリソース・ファイルの名前です。

既存のリソース・ファイルがある場合は、テンプレートをコピーせずに、そのファイルをコピーして、新しいリソース・ファイルを作成します。

2. ご使用のテキスト・エディタで、作成したリソース・ファイルのコピーを編集します。

新しいリソース・ファイルを作成したら、そのファイルを検証します。

参照：

- リソース・ファイルの検証 (25 ページ)

リソース・ファイルの編集

ra_admin リソース・ファイルは ASCII テキスト・ファイルで、標準的なテキスト・エディタを使用して編集できます。

- Replication Agent と **ra_admin** の両方の設定パラメータで、以下のフォーマットを使用します。

```
param=value
```

構文の説明は次のとおりです。

- *param* は設定パラメータの名前です。
- *value* は設定パラメータの値です。

注意： = 記号の前後または *value* 文字列内にスペースを入れることはできません。

- 各 *param=value* 文は、別の行に配置する必要があります。
- 設定パラメータのデフォルト値が存在する場合、以下を使用します。

```
param=USE_DEFAULT
```

param は設定パラメータの名前です。

- これらの **ra_admin** 設定パラメータには *yes* または *no* の値を含める必要があります。
 - **create_pds_username**
 - **start_instance**
 - **initialize_instance**

yes 値は、大文字と小文字が区別されます。{y|Y}{e|E}{s|S} 以外の文字列は、*no* として解釈されます。

注意： 空白行やシャープ記号 (#) で始まる行は、リソース・ファイルでは無視されます。

リソース・ファイルの検証

リソース・ファイルを検証するには、**ra_admin** を使用します。

ra_admin を、**-vr** オプションを指定して起動すると、ユーティリティによって、指定したリソース・ファイルが検証され、検証プロセスの情報が返されます。

ra_admin は以下を検証します。

- Replication Agent の管理ポート番号とインスタンス名のユニーク性
- プライマリ・データ・サーバ、Replication Server、および RSSD へのアクセス
- 各サーバのホスト名、ポート番号、データベース名、ユーザ・ログイン、およびパスワード

設定

- プライマリ・データベースの Replication Server データベース・コネクション
- **pds_username** ユーザにプライマリ・データベースで必要なすべてのパーミッションが付与されていること

検証に失敗した場合、**ra_admin** からエラー・メッセージとそのエラーに関する情報が返されます。

検証プロセスは、必要な回数だけ繰り返すことができます。このプロセスで、エンティティは変更または作成されません。

注意： 新しいリソース・ファイルを使用して Replication Agent インスタンスを作成する前に、このファイルを検証することをおすすめします。

1. **ra_admin** を起動し、**-vr** オプションとリソース・ファイルの名前を指定します。

```
ra_admin -vr res_file
```

res_file は、検証するリソース・ファイルの名前です。

たとえば、リソース・ファイルに `pubs2.rs` という名前が付けられている場合、コマンド・プロンプトで、この名前を以下のように入力します。

```
ra_admin -vr pubs2.rs
```

次のいずれかの検証結果が返されます。

- Response-file processing completed.
または
- Response-file processing completed with errors.

検証が正常に完了した場合は、手順 2 を省略し、リソース・ファイルを使用して新しい Replication Agent インスタンスを作成します。

検証中にエラーが発生した場合は、手順 2 を続行します。

2. 以下の手順を実行して、検証エラーを修正します。
 - a) エラー・メッセージを確認して、問題の原因を判別します。
 - b) リソース・ファイルを編集して、該当する値を修正します。
 - c) 再度 **ra_admin -vr** を起動し、リソース・ファイルの名前を指定します。

リソース・ファイルが正常に検証されるまで、この手順を繰り返します。

参照：

- リソース・ファイルを使用したインスタンスの作成 (27 ページ)

リソース・ファイルを使用したインスタンスの作成

Replication Agent インスタンスを作成して、同時にリソース・ファイルを指定します。

ra_admin を **-r** オプションを使用して呼び出すと、ユーティリティは最初に以下を除く指定のリソース・ファイルを検証します。

- Replication Agent for Microsoft SQL Server または Replication Agent for Oracle のプライマリ・データベースのユーザ・ログインがプライマリ・データ・サーバに存在しない場合、リソース・ファイル (`create_pds_username=yes`) で指定があれば、ユーティリティによって作成されます。ユーザ・ログインがプライマリ・データ・サーバに存在しても、必要な権限の一部が含まれていない場合は、**create** を **yes** に設定して、必要なすべてのパーミッションがユーティリティに付与されるようにします。

Replication Agent for Microsoft SQL Server または Replication Agent for Oracle のプライマリ・データベースのユーザ・ログインがプライマリ・データ・サーバに存在せず、必要なすべての権限が含まれ、リソース・ファイルでユーザ・ログインの作成が指定されている場合、エラー・メッセージが返され、インスタンスは作成されません。

- リソース・ファイルで新しい Replication Agent インスタンスの初期化が指定されている場合 (`initialize_instance=yes`):
 - Replication Agent プライマリ・データベースのユーザ・ログインがプライマリ・データ・サーバに存在するか、**ra_admin** (`create_pds_username=yes`) を指定して作成する必要があります。
 - リソース・ファイルで、Replication Agent インスタンスの起動を指定する必要があります (`start_instance=yes`)。それ以外の場合、エラー・メッセージが返され、インスタンスは作成されません。

リソース・ファイルが正常に検証されると、**ra_admin** によって以下の処理が実行されます。

- 指定されたリソース・ファイルの内容に基づいて、Replication Agent インスタンスを作成および設定します。
- リソース・ファイルで指定されている場合、インスタンス・ユーザに必要なすべての権限を作成または付与します。
- リソース・ファイルで指定されている場合、新しい Replication Agent インスタンスを起動します。
- リソース・ファイルで指定されている場合、新しい Replication Agent インスタンスを初期化します。

作成されたインスタンスとその結果に関する情報も返されます。

インスタンスの作成が失敗した場合、`ra_admin` からエラー・メッセージとそのエラーに関する情報が返されます。

注意： 新しいリソース・ファイルを使用して Replication Agent インスタンスを作成する前に、このファイルを検証することをおすすめします。

`ra_admin` を起動して、`-r` オプションとリソース・ファイルの名前を指定します。

```
ra_admin -r res_file
```

`res_file` はリソース・ファイルの名前です。

たとえば、リソース・ファイルに `pubs2.rs` という名前が付けられている場合、コマンド・プロンプトで、この名前を以下のように入力します。

```
ra_admin -r pubs2.rs
```

次のいずれかの結果が返されます。

- Response-file processing completed.
または
- Response-file processing completed with errors.

インスタンスが正常に作成されたら、新しい Replication Agent インスタンスの使用を開始できます。

インスタンスの作成に失敗した場合は、以下の手順を実行する必要があります。

- インスタンス・ディレクトリ内のすべてのファイルとサブディレクトリを削除し、Replication Agent インストール・ディレクトリからインスタンス・ディレクトリを削除します。
- リソース・ファイルを編集して、該当する値を修正します。

注意： インスタンスの作成に失敗した場合は、エラー・リカバリ手順を実行してから、再度インスタンスの作成を試行するようにしてください。

参照：

- リソース・ファイルの検証 (25 ページ)

インスタンスの作成エラーからのリカバリ

インスタンスの作成中にエラーが発生したら、リカバリする必要がある場合があります。

1. リソース・ファイルで、インスタンス・ユーザのログインがプライマリ・データ・サーバで作成されるように指定されていない場合、以下の手順を省略して、手順 2 に進んでください。

リソース・ファイルで、インスタンス・ユーザのログインがプライマリ・データ・サーバに作成されるように指定されている場合 (すなわち、

create_pds_username=yes が指定されている場合)、以下の手順を実行します。

- a) インスタンス・ユーザが追加されたかどうかプライマリ・データベースを確認します。
- b) **pds_sa_username** にプライマリ・データベースでインスタンスのログインを作成するのに十分な権限が付与されていることを確認します。
- c) リソース・ファイルを編集して、インスタンス・ユーザのログインがプライマリ・データ・サーバに作成されないように指定します (create_pds_username=no)。

注意： インスタンスの作成が失敗する前に Replication Agent のプライマリ・データベースが正常に作成された場合、以下のいずれかの手順を実行します。

- リソース・ファイルを編集して、**create_pds_username** パラメータの値を no に設定します。
 - プライマリ・データ・サーバにログインして、インスタンス・ログインを削除します。
-

2. 新しいインスタンス・ディレクトリが作成されているかどうかについて、Replication Agent ホストの Replication Agent ベース・ディレクトリを確認します。Replication Agent ベース・ディレクトリは次のとおりです。

```
%SYBASE%\%RAX-15_5
```

%SYBASE% は Replication Agent インストール・ディレクトリです。

Replication Agent ベース・ディレクトリに新しいインスタンス・ディレクトリが作成されていない場合は、手順 4 に進みます。

Replication Agent ベース・ディレクトリに新しいインスタンス・ディレクトリが作成されている場合は、手順 3 に進みます。

3. 新しいインスタンス・ディレクトリを削除するには、以下の 2 つのオプションのいずれかを指定します。

- 以下のように、**ra_admin** を使用してインスタンスを削除します。

```
ra_admin -d inst_name
```

inst_name は、削除するインスタンスの名前です。

- オペレーティング・システムのコマンドを使用して、新しいインスタンス・ディレクトリ内のすべてのファイルとサブディレクトリを削除してから、新しいインスタンス・ディレクトリを削除します。

4. エラー・メッセージを確認して、インスタンスの作成に失敗した原因を見つけ、リソース・ファイルを編集して、該当する値を修正します。

リソース・ファイルを編集したら、リソース・ファイルを検証します。

```
ra_admin -vr res_file
```

`res_file` はリソース・ファイルの名前です。

リカバリ手順が完了したら、Replication Agent インスタンスの作成を再試行します。

参照：

- リソース・ファイルの検証 (25 ページ)

Replication Agent の設定のコピー

`ra_admin` を使用して Replication Agent の設定をコピーできます。

新しい Replication Agent インスタンスを作成するときは、`ra_admin` に `-c` オプションおよび `-f` オプションを指定して呼び出すことによって、既存のインスタンスの設定をコピーします。

```
ra_admin -c new_inst -p port_num -f old_inst
```

構文の説明は次のとおりです。

- `new_inst` は、作成する新しい Replication Agent インスタンスの名前です。
- `port_num` は、新しい Replication Agent インスタンスの管理ポートのクライアント・ソケット・ポート番号です。Oracle および Microsoft SQL Server の場合は、2 番目のポート `port_num+1` が RASD 用に使用できることを確認してください。
- `old_inst` は、新しい Replication Agent インスタンスに設定をコピーする既存の Replication Agent インスタンスの名前です。

参照：

- Replication Agent インスタンスの作成 (13 ページ)

新しいインスタンスに既存の Replication Agent インスタンスの設定をコピーする既存のインスタンスの設定に基づいて、新しい Replication Agent インスタンスを作成します。

- Replication Agent ホスト・マシンで、オペレーティング・システムのコマンド・ウィンドウを開きます。
- Replication Agent の bin ディレクトリに移動します。

- Windows プラットフォームの場合：

```
cd %SYBASE%\RAX-15_5\bin
```

`%SYBASE%` は、Replication Agent インストール・ディレクトリのパスです。

- UNIX プラットフォームの場合

```
cd $SYBASE/RAX-15_5/bin
```

`$$SYBASE` は、Replication Agent インストール・ディレクトリのパスです。

3. Replication Agent の `bin` ディレクトリで、`ra_admin` を呼び出し、その設定が既存のインスタンスの設定に基づいた新しい Replication Agent インスタンスを作成します。

```
ra_admin -c new_inst -p port_num -f old_inst
```

構文の説明は次のとおりです。

- `new_inst` は、新しい Replication Agent インスタンスの名前です。
- `port_num` は、新しいインスタンスの管理ポートのクライアント・ソケット・ポート番号です。
- `old_inst` は、新しいインスタンスにコピーする設定の既存の Replication Agent インスタンスです。

`ra_admin` を呼び出した後、新しい Replication Agent インスタンスが作成されると、オペレーティング・システムのプロンプトに戻ります。

4. 次の方法のいずれかを使用して、Replication Agent のインスタンスが正しく作成されたことを確認します。

- `ra_admin` に `-v` オプションを指定して呼び出し、Replication Agent インスタンスの名前を指定します。

```
ra_admin -v new_inst
```

`new_inst` は、Replication Agent インスタンスの名前です。

Replication Agent インスタンスを確認するときに `-v` オプションを使用すると、Replication Agent ベース・ディレクトリの下にある指定されたインスタンス名のインスタンス・ディレクトリと、Replication Agent インスタンス・ディレクトリの下にあるすべてのサブディレクトリを確認することによって、インスタンスが確認されます。

- `ra_admin` に `-l` オプションを指定して呼び出します。

```
ra_admin -l
```

`-l` オプションを指定すると、確認可能な Replication Agent インスタンスがすべてリストされ、作成したばかりの新しいインスタンスも含まれます。

- `ra_admin` を使用する代わりに、オペレーティング・システムのコマンドを使用して、Replication Agent インスタンス・ディレクトリが正しく作成されたことを確認できます。

注意：新しい Replication Agent インスタンスを作成して既存のインスタンスの設定をコピーすると、一部の設定パラメータはデフォルト値に設定され、既存の設定からコピーされません。

次の設定パラメータの値は、既存の設定からコピーされません。

設定

- **admin_port**
- **log_directory**
- **pds_database_name**
- **pds_datasource_name**
- **pds_host_name**
- **pds_password**
- **pds_port_number**
- **pds_retry_count**
- **pds_retry_timeout**
- **pds_server_name**
- **pds_username**
- **rs_source_db**
- **rs_source_ds**

注意：これらのパラメータは、Replication Agent for Oracle と Replication Agent for Microsoft SQL Server にのみ有効です。

- **rasd_backup_dir**
- **rasd_database**
- **rasd_trace_log_dir**
- **rasd_tran_log**
- **asa_port**

Replication Agent の設定パラメータの詳細については、『Replication Agent リファレンス・マニュアル』を参照してください。

Replication Agent のインスタンスを作成したら、**ra** を使用してインスタンスを開始し、管理と設定を行うことができます。

注意：Replication Agent のインスタンスを作成したら、直ちにユーザ・ログイン名とパスワードを作成してデフォルトの sa ログインを置き換え、管理ポートへのアクセスをセキュリティ保護することをお勧めします。

参照：

- ユーティリティを使用するための準備 (15 ページ)
- Replication Agent の管理者ログインの作成または変更 (50 ページ)

コマンド・ラインを使用した Replication Agent インスタンスの作成

Windows または UNIX のコマンド・ラインを使用して Replication Agent インスタンスを作成します。

前提条件

SYBASE 環境を設定してから、**ra_admin** を起動します。

手順

1. Replication Agent ホスト・マシンで、オペレーティング・システムのコマンド・ウィンドウを開きます。
2. Replication Agent の bin ディレクトリに移動します。

- Windows プラットフォームの場合：

```
cd %SYBASE%\RAX-15_5\bin
```

%SYBASE% は、Replication Agent インストール・ディレクトリのパスです。

- UNIX プラットフォームの場合

```
cd $SYBASE/RAX-15_5/bin
```

\$SYBASE は、Replication Agent インストール・ディレクトリのパスです。

3. Replication Agent の bin ディレクトリで、`ra_admin` を呼び出して、新しい Replication Agent インスタンスを作成します。

```
ra_admin -c new_inst -p port_num -t database
```

- `new_inst` は、新しい Replication Agent インスタンスの名前です。
- `port_num` は、新しいインスタンスの管理ポートのクライアント・ソケット・ポート番号です。Oracle および Microsoft SQL Server の場合は、2 番目のポート `port_num+1` が RASD 用に使用できることを確認してください。
- `database` は、プライマリ・データベースが格納されているデータ・サーバのタイプを識別します。
 - **oracle** – Oracle データベース・サーバ
 - **mssql** – Microsoft SQL Server (Microsoft Windows プラットフォームでのみ有効です)
 - **ibmudb** – Linux、UNIX、Windows 向け IBM DB2

構文の説明は次のとおりです。

`ra_admin` を呼び出した後、新しい Replication Agent インスタンスが作成されると、オペレーティング・システムのプロンプトに戻ります。

4. 次の方法のいずれかを使用して、Replication Agent のインスタンスが正しく作成されたことを確認します。

- `ra_admin` に `-v` オプションを指定して呼び出し、Replication Agent インスタンスの名前を指定します。

```
ra_admin -v new_inst
```

`new_inst` は、Replication Agent インスタンスの名前です。

Replication Agent インスタンスを確認するときに `-v` オプションを使用すると、Replication Agent ベース・ディレクトリの下にある指定されたインスタンス名のインスタンス・ディレクトリと、Replication Agent インスタンス・

ディレクトリの下にあるすべてのサブディレクトリを確認することによって、インスタンスが確認されます。

- `ra_admin` に `-l` オプションを指定して呼び出します。

```
ra_admin -l
```

`-l` オプションを指定すると、確認可能な Replication Agent インスタンスがすべてリストされ、作成したばかりの新しいインスタンスも含まれます。

- `ra_admin` を使用する代わりに、オペレーティング・システムのコマンドを使用して、Replication Agent インスタンス・ディレクトリが正しく作成されたことを確認できます。

Replication Agent のインスタンスを作成したら、`ra` を使用してインスタンスを開始し、管理と設定を行うことができます。

注意： Replication Agent インスタンスを作成したら、直ちにユーザ・ログイン名とパスワードを作成してデフォルトの“sa” ログインを置き換え、管理ポートへのアクセスをセキュリティ保護することをおすすめします。

参照：

- Replication Agent の起動 (40 ページ)
- ユーティリティを使用するための準備 (15 ページ)
- Replication Agent の管理者ログインの作成または変更 (50 ページ)

Replication Agent のインスタンスの削除

Replication Agent のインスタンスの削除には、`ra_admin` を使用します。

前提条件

- Replication Agent インスタンスを実行している場合は停止します。
- Replication Server ソフトウェアが Microsoft Windows プラットフォームにインストールされている場合は、インスタンスのサブディレクトリの中のファイルがいずれも開いていないこと、このインスタンスのサブディレクトリにアクセスしているアプリケーションまたはウィンドウがないことを確認します。

注意： SYBASE 環境を設定してから、`ra_admin` ユーティリティを起動します。

注意： Replication Agent のインスタンスを削除しても、複製するようマーク付けされているプライマリ・データベース・オブジェクトのマーク付けは解除されず、トランザクション・ログ・オブジェクトが削除されることもありません。

Replication Agent のインスタンスを停止して削除する前に、プライマリ・データベース・オブジェクトのマーク付けを解除し、初期化を解除して、プライマリ・データベースに作成されたオブジェクトが削除されるようにしてください。

手順

1. Replication Agent ホスト・マシンで、オペレーティング・システムのコマンド・ウィンドウを開きます。
2. Replication Agent の bin ディレクトリに移動します。

- Windows プラットフォームの場合：

```
cd %SYBASE%\RAX-15_5\bin
```

%SYBASE% は、Replication Agent インストール・ディレクトリのパスです。

- UNIX プラットフォームの場合

```
cd $SYBASE/RAX-15_5/bin
```

\$SYBASE は、Replication Agent インストール・ディレクトリのパスです。

3. Replication Agent の bin ディレクトリで、**-d** オプションを使用して **ra_admin** を呼び出して、Replication Agent のインスタンスを削除します。

```
ra_admin -d inst_name
```

inst_name は、削除する Replication Agent インスタンスの名前です。

-d オプションを使用して **ra_admin** を呼び出すと、次のように表示されます。

```
Are you sure you want to delete the Replication Agent instance  
inst_name? [y/n]
```

4. **y** と入力して、Replication Agent のインスタンスを削除します。

インスタンスが削除されると、オペレーティング・システムのプロンプトが再度表示されます。

-d オプションを使用して **ra_admin** を呼び出したときにインスタンスが実行中である場合、ユーティリティによって次のように返されます。

```
Cannot delete Replication Agent instance 'inst_name' because it is  
currently running.
```

Replication Agent のインスタンスを停止するには、管理ポートにログインし、**shutdown** コマンドを使用します。

5. 次の方法のいずれかを使用して、Replication Agent のインスタンスが正しく削除されたことを確認します。

- **ra_admin** に **-v** オプションを指定して呼び出し、Replication Agent インスタンスの名前を指定します。

```
ra_admin -v new_inst
```

new_inst は、Replication Agent インスタンスの名前です。

Replication Agent インスタンスを確認するときに **-v** オプションを使用すると、Replication Agent ベース・ディレクトリの下にある指定されたインスタ

ンス名のインスタンス・ディレクトリと、Replication Agent インスタンス・ディレクトリの下にあるすべてのサブディレクトリを確認することによって、インスタンスが確認されます。

- ra_admin に -l オプションを指定して呼び出します。

```
ra_admin -l
```

-l オプションを指定すると、確認可能な Replication Agent インスタンスがすべてリストされ、削除したばかりの新しいインスタンスも含まれます。

- ra_admin を使用する代わりに、オペレーティング・システムのコマンドを使用して、Replication Agent インスタンス・ディレクトリが正しく削除されたことを確認できます。

注意： Microsoft Windows プラットフォームで、インスタンスの削除時に Replication Agent のインスタンスに関連したファイルまたはディレクトリにアクセスしているアプリケーションがあると、開いているファイルまたはディレクトリは削除されません。削除されていないファイルまたはディレクトリについて通知するエラー・メッセージが表示されます。

Microsoft Windows プラットフォームでファイルまたはディレクトリのアクセス競合が発生した後に Replication Agent のインスタンスの削除を完了するには、次の操作を実行する必要があります。

- ファイルまたはディレクトリがどのアプリケーションでも開いていないことを確認します。
 - ファイルまたはディレクトリを手動で削除します。
6. 次の方法のいずれかを使用して、Replication Agent のインスタンスが正しく削除されたことを確認します。

参照：

- Replication Agent インスタンスの停止 (75 ページ)
- ユーティリティを使用するための準備 (15 ページ)

Replication Agent インスタンスのアップグレード

Replication Agent インスタンスをコマンド・ライン・インタフェースからアップグレードできます。

同じリリース・レベルでのインスタンスのアップグレード

ra_admin -u source_installation_dirを使用して、指定のディレクトリに入っているバージョン 15.5 以前の Replication Agent インスタンスのバージョンを現在のインストール・ディレクトリからアップグレードします。

バージョン 15.5 以降の Replication Agent のインスタンスについては、**ra_admin -u all** または **ra_admin -u instance_name |instance= instance_name** を使用して、インスタンスをアップグレードします。

参照：

- ra (18 ページ)

Replication Agent のインスタンスのダウングレード

ダウングレード API を使用して、Replication Agent のインスタンスをダウングレードします。

ra_downgrade、**ra_migrate**、**resume purge** コマンドを使用して、Replication Agent のインスタンスをダウングレードします。これらのコマンドの使用方法については、『Replication Agent プライマリ・データベース・ガイド』の Replication Agent のアップグレードとダウングレードの項を参照してください。

注意： **ra_downgrade_prepare** コマンドと **ra_downgrade_accept** コマンドは Replication Agent 15.7.1 で廃止されましたが、それ以前のバージョンにダウングレードするために必要になることがあります。

Administrator GUI ユーティリティ

グラフィカル・インタフェースを使用して Replication Agent を管理するには、Administrator GUI ユーティリティを使用します。

SYBASE 環境を設定してから、**administrator** を起動します。

Administrator GUI の起動

Administrator GUI を起動します。

Administrator GUI を起動するには、次のいずれかを実行します。

- オペレーティング・システムのプロンプトに対して次のコマンドを入力します。

```
administrator
```
- Windows で、[ファイル マネージャ] または [エクスプローラ] 内のファイル名 `administrator.bat` をダブルクリックします。

このファイルは、インストール・ディレクトリの RAX-15_5¥bin サブディレクトリ内にあります。

Administrator GUI により、\$SYBASE/RAX-15_5/admin_logs ディレクトリに出力ファイルが作成されます。ファイル名のフォーマットは adminmddyymm_hhmmss.log です。mmdyyy は現在の日付、hhmmss は現在の時刻です。

Administrator GUI ウィンドウでは、Replication Agent のインスタンスごとに次の情報が表示されます。

- 名前 — インスタンスの名前。
- 種類 — インスタンスの種類。
- ポート — インスタンスに割り当てられたポート番号。
- 稼働中 — インスタンスのステータス (稼働中か否か)。

参照：

- ユーティリティを使用するための準備 (15 ページ)

Administrator を使用した Replication Agent インスタンスの作成

Administrator GUI を使用して、Replication Agent のインスタンスを作成します。

1. インスタンスの種類を選択します。

- oracle
- mssql
- ibmudb

2. ユニークなインスタンス名を入力します。

デフォルトのインスタンス名は **repagent** です。

3. 新しい Replication Agent インスタンスの管理ポートとして、マシン上の他のアプリケーションに割り当てられていないクライアント・ソケット・ポート番号を入力します。Oracle および Microsoft SQL Server の場合は、2 番目のポート *port_num+1* が RASD 用に使用できることを確認してください。

デフォルトのクライアント・ソケットのポート番号は **10000** です。

4. [作成] をクリックします。

インスタンスのリストにインスタンスが表示されます。「稼働中」の下にあるステータスは **no** になります。

5. [完了] をクリックして、Administrator GUI ウィンドウを終了します。

参照：

- ユーティリティを使用するための準備 (15 ページ)

Administrator を使用した Replication Agent インスタンスのコピー

Administrator を使用して Replication Agent のインスタンスをコピーします。

1. コピーする Replication Agent のインスタンスを選択します。
2. ユニークなインスタンス名を入力します。
3. 新しい Replication Agent インスタンスの管理ポートとして、マシン上の他のアプリケーションに割り当てられていないクライアント・ソケット・ポート番号を入力します。

別のアプリケーションで使用されているポート番号を入力すると、エラーが返されます。

4. [コピー] をクリックします。

インスタンスのリストにインスタンスが表示されます。

注意： Replication Agent のインスタンスをコピーしても、プライマリ・データベース・サーバのパラメータとポート番号は複製されません。

参照：

- ユーティリティを使用するための準備 (15 ページ)

Administrator を使用した Replication Agent インスタンスの削除

Administrator GUI を使用して、Replication Agent のインスタンスを削除します。

前提条件

注意： Replication Agent のインスタンスを削除しても、複写するようマーク付けされているプライマリ・データベース・オブジェクトのマーク付けは解除されず、トランザクション・ログ・オブジェクトが削除されることもありません。

Replication Agent のインスタンスを停止して削除する前に、プライマリ・データベース・オブジェクトのマーク付けを解除し、初期化を解除して、プライマリ・データベースに作成されたオブジェクトが削除されるようにしてください。

手順

1. 削除する Replication Agent のインスタンスを選択します。
2. [削除] をクリックし、続けて [はい] をクリックして確定します。

インスタンスの一覧からインスタンスが削除されます。ただし、選択したインスタンスが実行中の場合は、エラーが返されます。

実行中のインスタンスを削除するには、最初に管理ポートにログインし、**shutdown** コマンドを使用して、インスタンスを停止する必要があります。その後インスタンスを削除します。

参照：

- Replication Agent インスタンスの停止 (75 ページ)
- ユーティリティを使用するための準備 (15 ページ)

Replication Agent の起動

Replication Agent インスタンスを起動するには、Replication Agent インストール・ディレクトリおよびすべてのサブディレクトリの **execute** パーミッションを持っているユーザ名 (“sybase” ユーザなど) を使用して Replication Agent ホスト・マシンにログインする必要があります。

注意： Windows Vista では、コマンド・ウィンドウを管理者として実行する必要があります。そのためには、[スタート]>[すべてのプログラム]>[アクセサリ] をクリックし、[コマンド プロンプト] を右クリックして、[管理者として実行] を選択します。

Replication Agent インスタンスは次の方法で起動できます。

- **ra** を起動して、起動するインスタンスを指定します。
- **administrator** GUI ユーティリティを起動して、起動するインスタンスを指定します。
- 起動するインスタンスの RUN スクリプトを起動します。
- Replication Agent が Windows システムにインストールされている場合、起動するインスタンスの Windows サービスを起動します。

ra ユーティリティ、**administrator** GUI ユーティリティ、および RUN スクリプトは、Microsoft Windows ではバッチ・ファイルで、UNIX ではシェル・スクリプトです。

起動の準備

Replication Agent インスタンスを起動してプライマリ・データ・サーバに接続する前に、すべての必要な変数を設定してください。

1. プライマリ・データベースの JDBC ドライバのロケーションを CLASSPATH 環境変数に追加します。

プライマリ・データベースの JDBC ドライバのインストールと設定、および Replication Agent 接続の設定の詳細については、『Replication Agent プライマリ・データベース・ガイド』を参照してください。

2. Replication Agent の文字セットがプライマリ・データベースと異なる場合は、プライマリ・データベースと同じ文字セットになるように RA_JAVA_DFLT_CHARSET 環境変数を設定する必要があります。

次のステップ

プライマリ・データベース固有の接続要件の詳細については、『Replication Agent プライマリ・データベース・ガイド』を参照してください。

文字設定

プライマリ・データ・サーバとレプリケート・データ・サーバの種類が異なる異機種間複写システムでは、データ・サーバが同じ文字セットをサポートしていない場合があります。そのような場合、複写システム・コンポーネントは、(プライマリ・データ・サーバの文字セットからレプリケート・データ・サーバの文字セットへの) 文字セットの変換を、少なくとも 1 回は実行する必要があります。

プライマリ・データ・サーバとレプリケート・データ・サーバの種類が同じである同機種間複写システムでも、複写システム・コンポーネントが複数の種類のプラットフォームに存在している場合は、文字セットの変換が必要になることがあります。

文字セットの問題は、プライマリ・データベースとレプリケート・データベース間のデータの不整合を引き起こす場合があります。この問題を回避するには、次のいずれかを行います。

- 複写システムのすべてのサーバおよびプラットフォームで同じ文字セットを使用する。
- 複写システムのすべてのサーバおよびプラットフォームで、互換性のある文字セットを使用し、適切な文字セットの変換を実行するように、複写システム・コンポーネントを設定する。

文字セット変換を行うと、パフォーマンスは低下します。

注意： Replication Agent システムのすべてのサーバとプラットフォームで同じ文字セットを使用することをお勧めします。

環境文字セットの設定

デフォルトでは、Replication Agent のインスタンスが動作する Java 仮想マシン (JVM) が、使用しているシステムのデフォルトの文字セットを見つけ出します。Replication Agent が処理する文字データの種類は、文字セット (エンコーディングとも呼ばれる) によって決まります。JVM がシステムから検出するデフォルトの文字セットを上書きする場合を除いて、文字セットに関連する環境変数を明示的に設定する必要はありません。

デフォルトの文字セットの上書きをサポートするために、/bin ディレクトリにあるすべての実行可能スクリプト (バッチ・ファイル) は、RA_JAVA_DFLT_CHARSET という名前の環境変数を参照しています。この環境変数を設定して、目的の文字セットを使用できます。指定する文字セットは、プライマリ・データベースで設定されている文字セットと同じである必要があります。有効な Java 6 文字セットのリストについては、<http://docs.oracle.com/javase/6/docs/technotes/guides/intl/encoding.doc.html> で「Internationalization」ページの「Supported Encodings」を参照してください。

すべての Replication Agent インスタンスの RUN スクリプトも、RA_JAVA_DFLT_CHARSET 環境変数を参照します。

注意： を使用して、多数の異なる種類の文字セットを複製している場合は、文字セットを UTF8 に設定する必要があります。

次のいずれかの方法で、システムのデフォルトの文字セットを上書きできます。

- 使用している環境の RA_JAVA_DFLT_CHARSET という名前のシステム変数の値を設定し、**ra** を使用して、Replication Agent インスタンスを起動する。
- Replication Agent インスタンスの RUN スクリプトの変数 RA_JAVA_DFLT_CHARSET の値を設定し、RUN スクリプトを使用して、Replication Agent インスタンスを起動する。

ra を呼び出して Replication Agent のインスタンスを起動する場合、環境で RA_JAVA_DFLT_CHARSET システム変数の値を上書きして、文字セットを指定することができます。

インスタンスの RUN スクリプト (またはバッチ・ファイル) を呼び出して Replication Agent インスタンスを起動する場合、インスタンスの RUN スクリプトを編集し、RA_JAVA_DFLT_CHARSET のデフォルト値を指定して、使用する文字セットを指定できます。

すべての Replication Agent インスタンスのデフォルト文字セットの上書き

Replication Agent のすべてのインスタンスのデフォルト文字セットを上書きします。

1. **ra** スクリプトに文字セットの値を入力します。

- Windows では、%SYBASE%\RAX-15_5\bin\ra.bat ファイルを編集します。
- UNIX では、\$SYBASE/RAX-15_5/bin/ra.sh ファイルを編集します。

```
RA_JAVA_DFLT_CHARSET=charset
```

ここで *charset* は Java でサポートされているエンコードになります。

たとえば、ISO-1 では ISO8859_1 または Cp1252 (別名 Latin-1)、ヘブライ語では ISO8859_8 または Cp1255 になります。

注意： UNIX では、等号の前後にスペースを入れることはできません。

2. 次のコード行のコメントを解除します。

- Windows の場合：

```
set RA_JAVA_DFLT_CHARSET=charset
```

- UNIX の場合：

```
RA_JAVA_DFLT_CHARSET=charset
export RA_JAVA_DFLT_CHARSET
```

指定された Replication Agent インスタンスのデフォルト文字セットの上書き

指定された 1 つだけの Replication Agent インスタンスのデフォルト文字セットを上書きします。

RUN スクリプトに文字セットの値を入力します。

- Windows では、%SYBASE%\RAX-15_5\<instance>\%RUN_<instance>.bat スクリプトを編集します。
- UNIX では、\$SYBASE/RAX-15_5/<instance>/RUN_<instance>.sh バッチ・ファイルを編集します。

ここで *charset* は Java でサポートされているエンコードになります。

たとえば、ISO-1 では ISO8859_1 または Cp1252 (別名 Latin-1)、ヘブライ語では ISO8859_8 または Cp1255 がサポートされています。

注意： UNIX では、等号の前後にスペースを入れることはできません。

ra ユーティリティを使用した Replication Agent の起動

ra を使用して Replication Agent を起動するときに、インスタンスの起動時の状態を指定することができます。**ra** を起動したときに起動時の状態を指定しないと、Replication Agent のインスタンスはデフォルトの Admin 状態で開始されます。

前提条件

SYBASE 環境を設定してから、**ra** を起動します。

手順

1. Replication Agent ホスト・マシンで、オペレーティング・システムのコマンド・ウィンドウを開きます。
2. Replication Agent の bin ディレクトリに移動します。

設定

- Windows プラットフォームの場合：

```
cd %SYBASE%\RAX-15_5\bin
```

%SYBASE% は、Replication Agent インストール・ディレクトリのパスです。

- UNIX プラットフォームの場合

```
cd $SYBASE/RAX-15_5/bin
```

\$SYBASE は、Replication Agent インストール・ディレクトリのパスです。

3. Replication Agent の bin ディレクトリで、**ra** を呼び出して Replication Agent のインスタンスを開始します。

```
ra -i inst_name
```

または

```
ra -i inst_name -state
```

ここで、

- *inst_name* は Replication Agent のインスタンスのサーバ名になります。
- *state* は起動時状態のオプションのキーワードです。
 - **admin** – Replication Agent インスタンスを Admin 状態で起動します。
 - **replicate** – Replication Agent インスタンスを複製状態で起動します。

注意：状態オプションを指定しないと、Replication Agent は Admin 状態で開始されます。

たとえば、"my_ra" という名前の Replication Agent インスタンスを複製状態で起動する場合は、次のように入力します。

```
ra -i my_ra -replicate
```

Replication Agent のインスタンスを開始したら、別のオペレーティング・システム・コマンド・ウィンドウを開いて管理ポートにログインする必要があります。

参照：

- ra (18 ページ)
- ユーティリティを使用するための準備 (15 ページ)

Administrator GUI を使用した Replication Agent インスタンスの起動

Administrator GUI を使用して、Replication Agent のインスタンスを起動できます。

前提条件

- Replication Agent のインスタンスを開始する前に、**administrator** が実行中である必要があります。

- SYBASE 環境を設定してから、**administrator** を起動します。

手順

1. 起動する Replication Agent のインスタンスを選択します。[開始] をクリックします。

[はい] をクリックして、CLASSPATH 環境変数で適切な JDBC ドライバが指定されていることを確認します。

注意：適切な JDBC ドライバが CLASSPATH 環境変数の一覧に表示されない場合は、Replication Agent のインスタンスを開始できますが、プライマリ・データベース・サーバへのコネクションを確立することはできません。

2. [開始] をクリックします。[更新] をクリックして、Replication Agent のインスタンスが実行中かどうかを確認しなければならないことがあります。

Replication Agent ウィンドウに、選択したインスタンスが実行中であることが示されます。

- Windows では、開始することを選択した Replication Agent のインスタンスごとにコンソール・ウィンドウが開きます。
- UNIX では、選択した Replication Agent のインスタンスがバックグラウンドで開始されます。

いずれのプラットフォームでも、Replication Agent は administrator スクリプトを使用して、Admin 状態で開始されます。このスクリプトは RAX-15_5/bin ディレクトリにあります。起動時の標準エラー出力は error.log ファイルにリダイレクトされます。このファイルは Replication Agent インスタンスの log ディレクトリにあります。起動後、標準エラーおよび標準出力は <instance>/log ディレクトリの <instance>.log ファイルに送信されます。

注意：すでに実行中のインスタンスと同じ管理ポート番号を使用して Replication Agent のインスタンスを開始しようとすると、Replication Agent のプロセスがアボートされ、エラー・メッセージがシステム・ログに記録されます。

起動しようとしているインスタンスが実行されない場合は、*instance* ログを調べてエラーが発生したかどうかを確認します。

3. [完了] をクリックして、Administrator GUI ウィンドウを終了します。

参照：

- Replication Agent ログ (132 ページ)
- Administrator GUI の起動 (37 ページ)

- ユーティリティを使用するための準備 (15 ページ)

RUN スクリプトを使用した Replication Agent インスタンスの起動

RUN スクリプトを使用して、Replication Agent のインスタンスを起動することができます。

前提条件

注意： RUN スクリプトを呼び出す前に SYBASE 環境変数を設定する必要はありません。これは、Replication Agent のインスタンスが開始される前に、RUN スクリプトによって SYBASE 環境変数が設定されるためです。

手順

1. Replication Agent ホスト・マシンで、オペレーティング・システムのコマンド・ウィンドウを開きます。
2. オペレーティング・システムのプロンプトで、Replication Agent インスタンス・ディレクトリに移動し、次のように入力します。

- Windows の場合

```
cd %SYBASE%\RAX-15_5\inst_name
```

構文の説明は次のとおりです。

- %SYBASE% は Replication Agent インストール・ディレクトリのパスです。
 - inst_name は、Replication Agent インスタンスの名前です。
- UNIX の場合：

```
cd $SYBASE/RAX-15_5/inst_name
```

構文の説明は次のとおりです。

- \$SYBASE は Replication Agent インストール・ディレクトリのパスです。
 - inst_name は、Replication Agent インスタンスの名前です。
3. Replication Agent インスタンスのディレクトリで、RUN スクリプトを呼び出して、Replication Agent のインスタンスを開始します。

```
RUN_inst_name
```

ここで、inst_name は Replication Agent インスタンスのサーバ名です。

たとえば、"my_ra" という名前の Replication Agent インスタンスを起動する場合は、次のように入力します。

```
RUN_my_ra
```

Replication Agent のインスタンスが作成されると、RUN スクリプトが自動的に作成されます。RUN スクリプトは適切なパラメータ値を使用して ra を呼び出

し、Replication Agent のインスタンスを開始します。RUN スクリプトを編集して、起動時の状態を指定することができます。

注意： UNIX バージョンの RUN スクリプトには、`.sh` の拡張子がありません。

Replication Agent のインスタンスを開始したら、別のオペレーティング・システム・コマンド・ウィンドウを開いて管理ポートにログインする必要があります。

agt_service を使用した Replication Agent の起動

agt_service を使用して、Replication Agent のインスタンスを開始することができます。

前提条件

SYBASE 環境を設定してから、**agt_service** を起動します。

手順

Replication Agent が Windows システムにインストールされている場合、**agt_service** を使用して、Replication Agent を Windows サービスとして実行することができます。Replication Agent のインスタンスは、Windows サービスを作成したときに指定した起動時の状態で開始されます。起動時の状態を指定しないと、Replication Agent のインスタンスはデフォルトの Admin 状態で開始されます。

1. Replication Agent ホスト・マシンで、オペレーティング・システムのコマンド・ウィンドウを開きます。
2. Replication Agent の bin ディレクトリに移動します。

```
cd %SYBASE%\RAX-15_5\bin
```

`%SYBASE%` は、Replication Agent インストール・ディレクトリのパスです。

3. Replication Agent の bin ディレクトリで、**agt_service** を呼び出して Replication Agent インスタンスの Windows サービスを開始します。

```
agt_service -start inst_name
```

ここで `inst_name` は、作成時に Windows サービスに割り当てられた Replication Agent インスタンスの名前です。これは Windows サービスの名前でもあります。たとえば、Windows サービス名が `my_ra` という名前の `my_ra` という名前の Replication Agent インスタンスを起動する場合は、次のように入力します。

```
agt_service -start my_ra
```

参照：

- `agt_service` (16 ページ)
- ユーティリティを使用するための準備 (15 ページ)

Windows サービス・ツールからの Replication Agent の起動

Windows サービス管理ツールを使用して、自動的に、または手動で Replication Agent のインスタンスを開始することができます。

1. Windows の管理者権限を持つアカウントを使用して、Windows にログインします。
2. [スタート]>[設定]>[コントロール パネル]>[管理ツール]>[サービス] を選択します。
3. 利用可能なサービスのリストをスクロールして、設定を行う Sybase サーバを探します。

サーバ名は次のようになります。

“Sybase Replication Agent *inst_name*”

ここで *inst_name* は、Replication Agent インスタンスの名前です。

4. サーバ名を右クリックし、[開始] を選択します。
5. [クローズ] をクリックします。

Sybase Central™ を使用するか、または Windows サービス・ツールの [状態] カラムを調べると、インスタンスのステータスを確認できます。

Replication Agent 管理ポート

Replication Agent のインスタンスを作成する場合、管理ポートのクライアント・ソケットのポート番号を指定します。クライアント・アプリケーションは、このポートを使用して Replication Agent に接続します。

管理ポートを使用すると、Open Client (または Open Client 互換の) アプリケーションでログインし、Replication Agent コマンドを実行することができます。任意の Sybase Open Client インタフェース・ユーティリティ (**isql** または SQL Advantage® など) を使用して、Replication Agent 管理ポートに接続することができます。

注意：クライアント・アプリケーションは、Replication Agent ソフトウェアに付属していません。**isql** は Replication Server ソフトウェアに付属しており、**isql** および SQL Advantage は Adaptive Server Enterprise ソフトウェアに付属しています。

Interfaces ファイルのエントリ

一般に Open Client アプリケーション (**isql** など) では、使用可能なサーバ、ホスト・マシン、クライアント・ポートを識別するのに Interface ファイルが必要です。

Windows では、Interfaces ファイルの名前は `sql.ini` で、UNIX では、Interfaces ファイルの名前は `interfaces` となります。

他の Open Server アプリケーションに接続するのと同じように Open Client アプリケーションを Replication Agent 管理ポートに接続するには、Open Client アプリケーションのホスト・マシンの Interface ファイルに Replication Agent のサーバ・エントリを作成します。

Interface ファイルにおける Replication Agent 管理ポートのサーバ・エントリは、次のように表示されます。

```
[inst_name]
query=protocol,host_name,port_num
```

構文の説明は次のとおりです。

- `inst_name` は、Replication Agent インスタンスの名前です。
- `protocol` は、コネクションに使用されるネットワーク・プロトコルです。
- `host_name` は Replication Agent ホスト・マシンの名前です。
- `port_num` は Replication Agent インスタンスのクライアント・ソケットのポート番号です。

たとえば、クライアント・ソケットのポート番号が 10002 の “my_host” というホストで、Windows ソケット・プロトコルを使用して “my_ra” という名前の Replication Agent インスタンスの Interface ファイル・エントリを指定するには、次の行を Interfaces ファイルに追加します。

```
[my_ra]
query=NLWNSCK,my_host,10002
```

Interface ファイルが TLI 形式でなければならないシステムもあります。お使いのシステムで TLI 形式が必要な場合は、Interfaces ファイルを編集するユーティリティ (`sybtli` や `dsedit` など) を使用して結果を TLI 互換の形式で保存する必要があります。

Interfaces ファイルに Replication Agent のインスタンスのエントリを作成したら、Interfaces ファイルを使用する任意の Open Client アプリケーションを使用して、管理ポートに接続できます。

isql を使用した Replication Agent インスタンスへのログイン

`isql` 対話型 SQL ユーティリティを使用して、Replication Agent 管理ポートにログインすることができます。

前提条件

Open Client アプリケーション (`isql` など) を使用して Replication Agent 管理ポートにログインする前に、Interfaces ファイルに Replication Agent のインスタンスのサーバ・エントリを作成します。

注意：新しく作成した Replication Agent のインスタンスに初めてログインする場合は、パスワードなしでデフォルトの管理者ログイン“sa”を使用してください。

手順

1. オペレーティング・システムのコマンド・ウィンドウを開きます。
2. SYBASE.csh ファイルを source コマンドで実行するか (UNIX または Linux)、SYBASE.bat ファイルを実行して (Windows)、環境変数を設定します。
3. オペレーティング・システムのプロンプトで、次のように入力します。

```
isql -Uusername -Ppassword -Sinst_name
```

- *username* は Replication Agent の管理者ログインです。
- *password* は対応するパスワードです。
- *inst_name* は、Replication Agent インスタンスの名前です。

構文の説明は次のとおりです。

たとえば、“my_ra” という名前の新しい Replication Agent インスタンスにログインする場合は、次のように入力します。

```
isql -Usa -P -Smy_ra
```

管理ポートに正常にログインしたら、Replication Agent コマンドを使用して Replication Agent のインスタンスを管理できます。

参照：

- Interfaces ファイルのエントリ (48 ページ)

Replication Agent の管理者ログインの作成または変更

Replication Agent インスタンスには、それぞれ 1 つずつ管理者ログインが指定されます。デフォルトの管理者ログイン (パスワードなしの sa) は、Replication Agent インスタンスが作成されると、作成されます。

注意：Replication Agent のインスタンスを作成したら、直ちに新しい管理者ログインとパスワードを作成してデフォルトの“sa”ログインを置き換え、管理ポートへのアクセスをセキュリティ保護することをおすすめします。

Replication Agent インスタンスの管理者ログインを作成 (または変更) するには、**ra_set_login** を使用します。

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。
最初に Replication Agent インスタンスにログインするときは、デフォルトの管理者ログインを使用します。

2. ログインしたら、次のように入力します。

```
ra_set_login admin_user,admin_pw
```

構文の説明は次のとおりです。

- `admin_user`は、この Replication Agent インスタンスに使用する新しい管理者ログイン名です。
- `admin_pw`は新しい管理者ログインのパスワードです。

現在の管理者ログインが新しいログイン名に置き換えられます。次回、Replication Agent インスタンスにログインするときは、新しい管理者のログイン名とパスワードを使用する必要があります。

Replication Agent コネクティビティの設定

Replication Agent インスタンスと、プライマリ・データ・サーバ、Replication Server、および RSSD 間のコネクティビティを設定する必要があります。

プライマリ・データベースでは、Replication Agent とプライマリ・データベース間のコネクションを設定する前に、特定の設定作業を実行する必要があります。プライマリ・データベースに必要な設定作業が実行されているかどうかを確認するには、『Replication Agent プライマリ・データベース・ガイド』を参照してください。

注意：このマニュアルでは、「RSSD」という単語は、若干の違いがあるものの、RSSD と ERSSD の両方を示します。

Replication Agent のコネクティビティを設定するには、次の作業が必要です。

- Replication Agent について、プライマリ・データ・サーバとプライマリ・データベースの適切な権限を持つユーザ・ログイン名の作成
- Replication Agent について、Replication Server の **connect source** パーミッションと **create object** パーミッションを持つユーザ・ログイン名の作成
- Replication Agent について、RSSD データ・サーバと RSSD の適切な権限を持つユーザ・ログイン名の作成
- Replication Agent コネクション設定パラメータの値の設定

Replication Agent インスタンスごとにコネクション設定パラメータの値を記録するには、『Replication Agent インストール・ガイド』の「インストールおよび設定ワークシート」を使用します。

Replication Agent のプライマリ・データベースのユーザ・ログインの作成

Replication Agent では、プライマリ・データベースからデータベース・スキーマの情報を取得したり、プライマリ・データベースで Replication Agent オブジェクトの

読み取りと管理を行ったり、データベース・ログ・デバイス (Replication Agent for Oracle および Replication Agent for Microsoft SQL Server) についての情報を取得するためのクライアント・アクセスが必要です。

Replication Agent のインスタンス用にプライマリ・データ・サーバとプライマリ・データベースのユーザ・ログイン名を設定するには、次の手順に従います。

注意： これらの手順を実行するには、プライマリ・データ・サーバでのシステム管理者のユーザの役割が必要です。

1. システム管理者のユーザの役割でプライマリ・データ・サーバにログインします。
2. Replication Agent ログイン名をプライマリ・データ・サーバに追加し、必要に応じてプライマリ・データベースにも追加します。
 - 各プライマリ・データ・サーバおよびプライマリ・データベースに必要なパーミッションと権限の詳細については、『Replication Agent プライマリ・データベース・ガイド』を参照してください。
 - プライマリ・データ・サーバ(および必要に応じてプライマリ・データベース)に Replication Agent ログイン名を作成するために実行する必要がある特定のコマンドの詳細については、プライマリ・データ・サーバのベンダから提供されているマニュアルを参照してください。

プライマリ・データ・サーバに Replication Agent のユーザ・ログインを設定したら、新しいユーザ・ログイン名が有効であることを確認します(プライマリ・データ・サーバにログインでき、プライマリ・データベースにアクセスできることを確認します)。

Replication Agent への Replication Server ユーザ・ログインの作成

Replication Agent では、プライマリ Replication Server から複製トランザクションを送信するためにクライアント・アクセスが必要です。

前提条件

この手順を実行するには、Replication Server の **sa** パーミッションが必要です。

手順

Replication Agent インスタンスへの Replication Server ユーザ・ログインを設定するには、次の手順に従います。

1. “sa” パーミッションがあるログインを使用して Replication Server にログインします。
2. Replication Server で、Replication Agent のユーザ・ログイン名を作成します。

```
create user ra_rs_user set password ra_rs_pwd
```

構文の説明は次のとおりです。

- `ra_rs_user` は、Replication Agent のユーザ・ログイン名です。
- `ra_rs_pwd` はユーザ・ログイン名のパスワードです。

3. Replication Agent ログイン名に **connect source** パーミッションを付与します。

```
grant connect source to ra_rs_user
```

ここで `ra_rs_user` は、Replication Agent のユーザ・ログイン名です。

プライマリ Replication Server に Replication Agent のユーザ・ログインを設定したら、新しいユーザ・ログイン名が有効であることを確認します (Replication Server にログインできることを確認します)。

RSSD ユーザのログイン名

Replication Agent では、複写定義に関する情報を取得するために、ERSSD または RSSD へのクライアント・アクセスが必要です。

Replication Agent の ERSSD ユーザ・ログインの設定

SQL Anywhere® で管理される ERSSD の Replication Agent インスタンスへのユーザ・ログイン名を設定します。

前提条件

この手順を実行するには、ERSSD にプライマリ・ユーザの役割 (Replication Server の “sa” パーミッション) が必要です。

手順

デフォルトでは、Replication Server は Embedded RSSD を使用します。

1. プライマリ・ユーザとして ERSSD にログインします。
2. Replication Agent ログイン名を ERSSD に追加します。

```
grant connect to ra_rssd_user
identified by ra_rssd_pwd
```

構文の説明は次のとおりです。

- `ra_rssd_user` は Replication Agent のユーザ・ログイン名です。
- `ra_rssd_pwd` はユーザ・ログイン名のパスワードです。

3. Replication Agent ユーザのパーミッションを付与して、Replication Server のシステム・テーブルを読み取ります。

```
grant membership in group rs_systabgroup
to ra_rssd_user
```

ここで *ra_rssd_user* は、Replication Agent ユーザ・ログイン名です。

ERSSD に Replication Agent のユーザ・ログインを設定したら、新しいユーザ・ログイン名が有効であることを確認します (ERSSD データ・サーバにログインでき、Replication Server システム・テーブルにアクセスできることを確認します)。

Replication Agent の RSSD ユーザ・ログインの設定

Adaptive Server Enterprise で管理される RSSD の Replication Agent インスタンスへのユーザ・ログイン名を設定します。

前提条件

これらの手順を実行するには、RSSD を管理する Adaptive Server Enterprise でのシステム管理者のユーザの役割が必要です。

手順

外部の Adaptive Server Enterprise データベースを使用して RSSD 情報をホストするように Replication Server を設定することができます。デフォルトでは、Replication Server は Embedded RSSD を使用します。お使いの環境で Adaptive Server Enterprise を使用して RSSD をホストしなければならない場合、次の指示が適用されます。

1. RSSD を管理する Adaptive Server Enterprise にログインします。
2. Replication Agent ログイン名を RSSD データ・サーバに追加します。

```
use master
sp_addlogin ra_rssd_user, ra_rssd_pwd, rssid_db
```

構文の説明は次のとおりです。

- *ra_rssd_user* は Replication Agent のユーザ・ログイン名です。
- *ra_rssd_pwd* はユーザ・ログイン名のパスワードです。
- *rssd_db* は RSSD のデータベース名です。

3. Replication Agent ユーザ・ログイン名を RSSD に追加し、このログイン名を *rs_systabgroup* グループに追加します。

```
use rssid_db
sp_adduser ra_rssd_user
sp_changegroup rs_systabgroup, ra_rssd_user
```

構文の説明は次のとおりです。

- *rssd_db* は RSSD のデータベース名です。
- *ra_rssd_user* は Replication Agent のユーザ・ログイン名です。

RSSD に Replication Agent のユーザ・ログインを設定したら、新しいユーザ・ログイン名が有効であることを確認します (RSSD データ・サーバにログインでき、RSSD にアクセスできることを確認します)。

コネクション設定パラメータの設定

Replication Agent は、別の複製システム・コンポーネントに接続する場合、そのコンポーネントの設定パラメータに格納されている値を使用して、接続プロパティの最小セットを定義します。

次のプロパティがあります。

- サーバのホスト名
- ポート番号
- ユーザ・ログイン名
- ユーザ・ログイン・パスワード

注意：コネクション・パラメータの完全なセットは、データベースごとに異なります。各データベースで要求されるコネクション・パラメータの完全なセットについては、『Replication Agent プライマリ・データベース・ガイド』を参照してください。

Replication Agent は、Replication Server と接続する場合、2つの追加の設定パラメータ (**rs_source_db** および **rs_source_ds**) の値に依存して、**LTl connect source** コマンドで Replication Server プライマリ・データベース・コネクションを識別します。

コネクション・パラメータを設定するには、Replication Agent インスタンスは Admin 状態である必要があります。インスタンスが Admin 状態の場合、他の複製システム・コンポーネントとの接続は確立できませんが、管理コマンドを実行することはできます。

注意：**rs_source_db** パラメータおよび **rs_source_ds** パラメータの値は、Replication Server プライマリ・データベース・コネクションの **create connection** コマンドで指定されたデータベースおよびデータ・サーバ名と正確に一致する必要があります。値の大文字と小文字は区別されます。

『Replication Agent リファレンス・マニュアル』には、**rs_source_db** パラメータおよび **rs_source_ds** パラメータの詳細な説明が記載されています。

Replication Agent インスタンスごとにコネクション設定パラメータの値を記録するには、『Replication Agent インストール・ガイド』の「インストールおよび設定ワークシート」を使用します。

注意：Replication Agent インスタンスのコネクション設定パラメータ値を設定するには、Replication Agent インスタンスが実行中である必要があります。

参照：

- Replication Agent の状態 (71 ページ)
- Replication Agent の起動 (40 ページ)

プライマリ・データベースのコネクション・パラメータの設定

プライマリ・データベースへの接続を設定します。

1. Replication Agent 管理ポートにログインし、Replication Agent インスタンスが Admin 状態になっていることを確認します。

```
ra_status
```

インスタンスが Admin 状態になっていない場合は、Admin 状態に変更します。

```
suspend
```

- 2.

プライマリ・データ・サーバの種類	指定内容
Oracle	プライマリ・データ・サーバのホスト名: <pre>ra_config pds_hostname, pds_host</pre> ここで <i>pds_host</i> は、プライマリ・データ・サーバのホスト・マシンのネットワーク名です。
Microsoft SQL Server	プライマリ・データ・サーバ名: <pre>ra_config pds_server_name, server</pre> ここで <i>server</i> は、プライマリ・データ・サーバの名前です。
IBM DB2 UDB	プライマリ・データ・サーバのホスト名: <pre>ra_config pds_hostname, pds_host</pre> ここで <i>pds_host</i> は、プライマリ・データ・サーバのホスト・マシンのネットワーク名です。 プライマリ・データベース・コネクションで使用されるデータ・ソース名 (DSN): <pre>ra_config pds_datasource_name, pds_dsn</pre> ここで <i>pds_dsn</i> は DSN です。

3. プライマリ・データ・サーバのポート番号を指定します。

```
ra_config pds_port_number, NNN
```


ここで *NNN* は、プライマリ・データ・サーバがコネクションを受信するネットワーク・ポートの数です。

4. プライマリ・データベース名を指定します。

```
ra_config pds_database_name, pdb
```

ここで *pdb* は、プライマリ・データベースのデータベース名です。

5. Replication Agent インスタンスへのプライマリ・データ・サーバのユーザ・ログイン名を指定します。

```
ra_config pds_username, ra_pds_user
```

ここで *ra_pds_user* は、Replication Agent がプライマリ・データ・サーバにログインするために使用するユーザ・ログイン名です。

6. Replication Agent ユーザ・ログインのパスワードを指定します。

```
ra_config pds_password, ra_pds_pwd
```

ここで *ra_pds_pwd* は、Replication Agent がプライマリ・データ・サーバにログインする際に使用するユーザ・ログイン名のパスワードです。

プライマリ・データベースのコネクション設定パラメータを設定したら、Replication Agent **test_connection PDS** コマンドを使用して、Replication Agent とプライマリ・データベース間の接続をテストします。

参照：

- ネットワーク接続テスト (60 ページ)

Replication Server のコネクション・パラメータの設定

Replication Server の接続を設定します。

1. Replication Agent 管理ポートにログインし、Replication Agent インスタンスが Admin 状態になっていることを確認します。

```
ra_status
```

インスタンスが Admin 状態になっていない場合は、Admin 状態に変更します。

```
suspend
```

2. Replication Server ホスト名を指定します。

```
ra_config rs_hostname, rs_host
```

ここで *rs_host* は、Replication Server のホスト・マシンのネットワーク名です。

3. Replication Server のポート番号を指定します。

```
ra_config rs_port_number, NNN
```

ここで *NNN* は、Replication Server がコネクションを受信するネットワーク・ポートの数です。

4. Replication Server 15.0 より前のバージョンを使用している場合は、Replication Server 文字セットを指定します。

```
ra_config rs_charset, charset
```

ここで *charset* と、Replication Server configuration (.cfg) ファイルの **RS_charset** 値を照合させます。Replication Server 設定ファイルの場所は \$SYBASE/REP-15_0/install/<instance>.cfg です。<instance> は Replication Server のインスタンスです。

注意： Replication Server 15.0.1 以降の場合、Replication Agent は Replication Server の **RS_charset** パラメータの値を使用します。Replication Agent の **rs_charset** パラメータは使用しません。

5. Replication Agent インスタンスの Replication Server のユーザ・ログイン名を指定します。

```
ra_config rs_username, ra_rs_user
```

ここで *ra_rs_user* は、Replication Agent がプライマリ Replication Server にログインする際に使用するユーザ・ログイン名です。

6. Replication Agent インスタンスのユーザ・ログインのパスワードを指定します。

```
ra_config rs_password, ra_rs_pwd
```

ここで *ra_rs_pwd* は、Replication Agent がプライマリ Replication Server にログインする際に使用するユーザ・ログイン名のパスワードです。

7. Replication Server プライマリ・データベース・コネクションのプライマリ・データ・サーバ名を指定します。

```
ra_config rs_source_ds, pds
```

ここで *pds* は、Replication Agent が LTL **connect source** コマンドで使用するプライマリ・データ・サーバ名です。

8. Replication Server プライマリ・データベース・コネクションのプライマリ・データベース名を指定します。

```
ra_config rs_source_db, pdb
```

ここで *pdb* は、Replication Agent が LTL **connect source** コマンドで使用するプライマリ・データベース名です。

ERSSD (または RSSD) のコネクション・パラメータの設定

ERSSD または RSSD への接続を設定します。

1. Replication Agent 管理ポートにログインし、Replication Agent インスタンスが Admin 状態になっていることを確認します。

```
ra_status
```

インスタンスが Admin 状態になっていない場合は、Admin 状態に変更します。

```
suspend
```

2. ERSSD ホスト名を指定します。

```
ra_config rssid_hostname, rssid_host
```

ここで *rssd_host* は ERSSD ホスト・マシンのネットワーク名です。

3. ERSSD のポート番号を指定します。

```
ra_config rssid_port_number, NNN
```

ここで *NNN* は、ERSSD サーバがコネクションを受信するネットワーク・ポートの数です。

4. ERSSD データベース名を指定します。

```
ra_config rssid_database_name, rssid_db
```

ここで *rssd_db* は ERSSD のデータベース名です。

5. Replication Agent インスタンスの ERSSD のユーザ・ログイン名を指定します。

```
ra_config rssid_username, ra_rssid_user
```

ここで *ra_rssid_user* は、Replication Agent が ERSSD にログインするために使用するユーザ・ログイン名です。

6. Replication Agent インスタンスのユーザ・ログインのパスワードを指定します。

```
ra_config rssid_password, ra_rssid_pwd
```

ここで *ra_rssid_pwd* は、Replication Agent が RSSD にログインする際に使用するユーザ・ログイン名のパスワードです。

プライマリ Replication Server と RSSD のコネクション設定パラメータを設定したら、Replication Agent の **test_connection RS** コマンドを使用して、Replication Agent、Replication Server および RSSD 間の接続をテストできます。

ネットワーク接続テスト

Replication Agent には、ネットワーク・コネクションをテストするための簡単な方法があります。

test_connection コマンドを実行すると、接続要求が送信され、以下のサーバとのネットワーク・コネクションが確認されます。

- プライマリ・データ・サーバ
- プライマリ Replication Server
- RSSD サーバ (設定してある場合)

注意： **use_rssd** 設定パラメータの値が **true** の場合は、**test_connection** コマンドを実行すると、Replication Server へのコネクティビティがテストされるときに、RSSD への Replication Agent のコネクティビティがテストされます。**use_rssd** 設定パラメータの値が **false** の場合は、**test_connection** コマンドを実行しても、RSSD への Replication Agent のコネクティビティはテストされません。

test_connection コマンドを実行すると、以下の場合はエラー・メッセージが返されます。

- Replication Agent 設定パラメータに記録されている接続仕様 (サーバ名、ポート番号、ユーザ・ログインなど) が正しくない。
- ネットワーク障害のため、Replication Agent がサーバへのコネクションを確立できない。
- サーバが停止しているため、Replication Agent がサーバへのコネクションを確立できない。

test_connection コマンドを実行しても、プライマリ・データベースの Replication Agent ユーザ・パーミッションは検証されません。**pds_username** パラメータと **pds_password** パラメータで指定されたユーザとパスワードでプライマリ・データ・サーバにログインできることが確認されるだけです。

test_connection コマンドを実行すると、Replication Agent ユーザ・ログイン (**rs_username** パラメータと **rs_password** パラメータで指定) にプライマリ Replication Server の **connect source** パーミッションが含まれていることが確認されます。

注意： ネットワーク・コネクティビティをテストするには、Admin 状態である必要があります。

参照：

- Replication Agent コネクティビティの設定 (51 ページ)

Replication Agent のコネクションの確認

Replication Agent の接続を確認します。

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. Replication Agent のネットワーク・コネクションをテストします。

```
test_connection
```

このコマンドを使用すると、ログインした Replication Agent インスタンスからのすべてのコネクションがテストされます。

注意： テストするコネクションを指定して、特定のコネクション (プライマリ・データ・サーバまたはプライマリ Replication Server のいずれか) をテストすることができます。

test_connection コマンドによってエラーが返された場合は、Replication Agent システム・ログを調べて障害の原因を判断してください。さらに、コネクションに関連付けられたサーバのシステム・ログをチェックして、障害の原因を確認することが必要な場合もあります。

test_connection コマンドの詳細については、『Replication Agent リファレンス・マニュアル』を参照してください。

Replication Agent の初期化

Replication Agent は、プライマリ・データベースが維持するネイティブのトランザクション・ログを使用してトランザクションを取得します。レプリケーションをサポートするため、Replication Agent はプライマリ・データベースにいくつかのオブジェクトを作成します。

注意： Replication Agent システム・データベース (RASD) を持つ Replication Agent を初期化する前に、DDL 操作によってデータベース・オブジェクトやスキーマが変更されることがないように、初期化中にデータベース・アクティビティが行われないための対策を講じてください。

参照：

- Replication Agent インスタンスの初期化 (83 ページ)

オブジェクト名プレフィクス

Replication Agent オブジェクトを作成する前に、オブジェクトの命名に使用されるオブジェクト名プレフィクス文字列を指定することができます。このプレフィク

ス文字列を設定することで、プライマリ・データベースの既存のデータベース・オブジェクトの名前との競合を避けることができます。

Replication Agent は、プレフィクス文字列を使用して、プライマリ・データベースのオブジェクトを見つけます。Replication Agent for Microsoft SQL Server と Replication Agent for UDB では、このプレフィクス文字列は **ra_admin_instance_prefix** パラメータの値によって定義されます。Replication Agent for Oracle では、プレフィクス文字列は **ra_admin_prefix** パラメータの値によって定義されます。

ra_config コマンドを使用して、**ra_admin_instance_prefix** または **ra_admin_prefix** の値を変更します。Replication Agent を初期化した後で **ra_admin_instance_prefix** または **ra_admin_prefix** の値を変更した場合、Replication Agent 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。

Replication Agent の初期化

Replication Agent インスタンスを初期化します。

前提条件

プライマリ・データベースに必要な設定作業が実行されているかどうかを確認するには、『Replication Agent プライマリ・データベース・ガイド』を参照してください。

手順

1. Replication Agent 管理ポートにログインします。
2. Replication Agent トランザクション・ログを一意に識別するプレフィクスを定義するには、次のコマンドを使用します。

```
ra_config ra_admin_instance_prefix, string
```

ここで *string* は 1～3 文字の文字列で、プライマリ・データベースで作成される Replication Agent オブジェクトのすべての名前のプレフィクスとして使用します。

注意： **ra_admin_instance_prefix** パラメータのデフォルト値は **ra_** です。この文字列がプライマリ・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しない限り、このデフォルト値を使用してください。

3. Replication Agent を初期化します。

```
ra_admin init
```

init オプションを付けて **ra_admin** コマンドを起動すると、Replication Agent は：

- プライマリ・データベースの互換性設定を確認します。

- プライマリ・データベースで実行される SQL スクリプトを作成します。このスクリプトは、Replication Agent オブジェクトをプライマリ・データベースに作成します。

RASD を使用する Replication Agent の場合、その RASD はプライマリ・データベースからの情報で初期化されます。

注意： Replication Agent が初期化されていないと、プライマリ・データベースのオブジェクトに複製のマークを付けることはできません。

4. Replication Agent が初期化され、そのオブジェクトがプライマリ・データベースに作成されたことを確認するには、次のように入力します。

```
ra_admin
```

オプションを付けずに **ra_admin** コマンドを起動すると、初期化が正常に完了している場合は、Replication Agent はプライマリ・データベースにあるオブジェクトのリストを返します。情報が返されない場合、Replication Agent は初期化されておらず、そのオブジェクトはプライマリ・データベースに存在しません。

Replication Agent が初期化され、プライマリ・データベース・コネクションと Replication Server コネクションの両方が正しく定義されている場合、Replication Agent インスタンスを複製状態にすることができます。

参照：

- Replication Agent の起動 (40 ページ)

プライマリ・データベース・オブジェクトのマーク付け

複製する個々のテーブルにはマークを付ける必要があります。テーブルへのマーク付けには、**pdb_setreptable** コマンドで明示的に行う方法と、**ra_admin init** 処理中に自動的に行う方法 (**pdb_automark_tables** 設定パラメータが true に設定されている場合) とがあります。

注意： Replication Agent for UDB では、**pdb_automark_tables** パラメータは使用できません。

テーブル、ストアド・プロシージャ、およびシーケンス (Oracle のみ) に複製のマーク付けがされ、そのオブジェクト (テーブル、ストアド・プロシージャ、またはシーケンス) で複製が有効化されている必要があります。LOB カラムで複製が有効になっていて、LOB カラムを含むテーブルに複製のマークが付けられ、そのテーブルの複製が有効になっている必要があります。

プライマリ・データベース内で複写のマークを付けられるオブジェクトには、以下の4つのタイプがあります。

- テーブル
- (Oracle および Microsoft SQL Server のみ) ストアド・プロシージャ
- ラージ・オブジェクト (LOB) カラム
- (Oracle および Microsoft SQL Server のみ) DDL
- (Oracle のみ) シーケンス

プライマリ・データベースのテーブルへのマーク付け

複写されるテーブルに対するトランザクションでは、プライマリ・データベースにあるプライマリ・テーブルに複写のマーク付けがされ、そのテーブルの複写が有効になっている必要があります。

前提条件

手順

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. テーブルがプライマリ・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_setreptable pdb_table
```

`pdb_table` は、複写のマークを付ける、プライマリ・データベースにあるテーブルの名前です。

- 指定されたテーブルにマーク付けされ、複写が有効化された情報を **pdb_setreptable** が返す場合、この手順を続ける必要はありません。
 - 指定されたテーブルがマーク付けされているが、複写が無効化されているという情報を `pdb_setreptable` が返した場合は、手順3を省略し、手順4に進んでそのテーブルの複写を有効化します。
 - 指定されたテーブルがマーク付けされていないという情報を `pdb_setreptable` が返した場合は、この手順を続けてテーブルに複写のマークを付けます。
3. テーブルに複写のマークを付け、複写で使用する名前を指定します。

- 同じテーブル名での複写定義を使用してテーブルに複写のマークを付けるには、次を使用します。

```
pdb_setreptable pdb_table, mark
```

- 別のテーブル名での複写定義を使用してテーブルに複写のマークを付けるには、次を使用します。

```
pdb_setreptable pdb_table, rep_table, mark
```


- `pdb_table` は、複製のマークを付ける、プライマリ・データベースにあるテーブルの名前です。
- `rep_table` は、このテーブルに対する複製定義の **with primary table named `rep_table`** 句で使用するテーブルの名前です。

構文の説明は次のとおりです。

- テーブルに複製のマーク付けをするときに、そのテーブルに対する Replication Server の複製定義が所有者指定の場合は、Replication Agent によって送信されるログ転送言語 (LTL) も所有者指定で、複製定義と一致するように指定する必要があります。このためには、次のように、**mark** キーワードの後に **owner** キーワードを使用します。

```
pdb_setreptable pdb_table, mark, owner
```

ここで `pdb_table` は、複製のマークを付ける、プライマリ・データベースにあるテーブルの名前です。

`pdb_dflt_object_repl` が true に設定されている場合 (デフォルト)、`pdb_setreptable` で複製のマーク付けをしたテーブルは、`pdb_setreptable` を正常に起動すると複製の準備が完了です。そして、このプロセスの手順 4 を省略できます。

`pdb_dflt_object_repl` が false の場合は、複製できるようにする前に、テーブルに対して複製を有効にする必要があります。

4. テーブルの複製を有効にします。

```
pdb_setreptable pdb_table, enable
```

すべてのユーザ・テーブルを一度にマーク付けするまたは有効にするには、次を使用します。

```
pdb_setreptable all, {mark|enable}
```

ここで、**mark** または **enable** は、データベース内のすべてのユーザ・テーブルに対するアクションを指定するキーワードです。

注意： `pdb_setreptable all, mark` を使用してプライマリ・データベースのすべてのテーブルに一度でマーク付けすることは、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付けしてください。

テーブルにマークが付けられ、そのテーブルの複製が有効になると、そのテーブルのデータに影響するトランザクションの複製を開始できます。

プライマリ・データベースのストアド・プロシージャへのマーク付け

プライマリ・データベースにあるストアド・プロシージャの呼び出しを複写するには、そのストアド・プロシージャに複写のマークが付けられ、そのストアド・プロシージャの複写が有効になっている必要があります。

前提条件

Oracle の場合、ストアド・プロシージャにマーク付け (またはマーク解除) する前に、DDL 複写を無効にする必要があります。

注意： UDB では、プロシージャ複写は使用できません。

手順

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. ストアド・プロシージャの複写が有効になっているかどうかを確認します。

```
pdb_setrepproc pdb_proc
```

`pdb_proc` は、複写のマークを付ける、プライマリ・データベース内のストアド・プロシージャの名前です。

- 指定されたストアド・プロシージャがマーク付けされ、複写が有効化されているという情報を `pdb_setrepproc` が返した場合、この手順を続ける必要はありません。
- 指定されたストアド・プロシージャがマーク付けされているが、複写が無効になっているという情報を `pdb_setrepproc` が返した場合は、手順 3 を省略し、手順 4 に進んでこのストアド・プロシージャの複写を有効にします。
- 指定されたストアド・プロシージャがマーク付けされていないという情報を `pdb_setrepproc` が返した場合は、この手順を続けてこのストアド・プロシージャに複写のマークを付けます。

3. 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。

- ファンクション複写定義がプライマリ・データベースのプロシージャと同じ名前の場合は、次のコマンドを使用します。

```
pdb_setrepproc pdb_proc, mark
```

- ファンクション複写定義の名前が、プライマリ・データベースのプロシージャと異なる場合は、次のコマンドを使用します。

```
pdb_setrepproc pdb_proc, rep_proc, mark
```

ここで、`rep_proc` は、このストアド・プロシージャに対する Replication Server 関数の複写定義の **with all procedures named rep_proc** 句で使用されるストアド・プロシージャの名前です。

`pdb_dflt_object_repl` が true に設定されている場合 (デフォルト)、`pdb_setrepproc` で複写のマーク付けをしたストアド・プロシージャは、`pdb_setrepproc` を正常

に呼び出すと複製の準備が完了です。そして、このプロセスの手順4を省略できます。

`pdb_dflt_object_repl` が `false` の場合は、複製できるようにする前に、ストアド・プロシージャに対して複製を有効にする必要があります。

4. マーク付けされたストアド・プロシージャの複製を有効にします。

```
pdb_setrepproc pdb_proc, enable
```

そのストアド・プロシージャにマーク付けがされ、そのストアド・プロシージャの複製が有効になっている場合、そのストアド・プロシージャの呼び出しの複製を開始できます。

プライマリ・データベースの LOB カラムの複製の有効化

複製される LOB カラムに影響するトランザクションでは、その LOB カラムを含むテーブルに複製のマークが付けられ、複製が有効になっている必要があります。

前提条件

`pdb_dflt_column_repl` が `true` の場合、(`pdb_setreptable` を呼び出して) テーブルにマーク付けをすると、そのテーブルのすべての LOB カラムは自動的に複製が有効になります。`pdb_dflt_column_repl` が `false` の場合(デフォルト)、複製できるようにする前に、個別に各 LOB カラムの複製を有効にする必要があります。

手順

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. LOB カラムの複製が有効になっているかどうかを確認します。

```
pdb_setrepcol pdb_table, pdb_col
```

構文の説明は次のとおりです。

- `pdb_table` は、LOB カラムを含んでいるマーク付けされたテーブルです。
- ここで `pdb_col` は、プライマリ・データベース内の LOB カラムです。

注意： Replication Agent for UDB では、“DATE” カラムを含むテーブルとともに `pdb_setrepcol` が呼び出される場合、プライマリ・テーブル内のプライマリ・キーに“DATE” カラムを含めることはできません。

LOB カラムの複製が有効にされているという情報を `pdb_setrepcol` が返した場合、この手順を続ける必要はありません。

LOB カラムの複製が有効化されていないという情報を `pdb_setrepcol` が返した場合は、この手順を続けてカラムの複製を有効にします。

3. LOB カラムの複製を有効にします。

```
pdb_setrepcol pdb_table, pdb_col, enable
```

その LOB カラムの複写が有効になると、そのカラムのデータに影響するトランザクションの複写を開始できます。

DDL 複写

DDL を複写するには、**pdb_setrepcddl** を使用してフィルタリング・ルールを適切に設定します。

ddl_username と **ddl_password** も設定する必要があります。**pdb_setrepcddl** コマンドの使用方法の詳細については、『Replication Agent リファレンス・マニュアル』を参照してください。

注意： UDB では、DDL 複写は使用できません。

DDL を複写するには：

- Replication Agent で、スタンバイ・データベースですべての DDL コマンドを実行する権限のある、ユニークなユーザ名の入力が必要です。
- Replication Server には、その定義内に **replicate DDL** セットを持つ、データベース・レベルの複写定義が必要です。

設定プロパティ *ddl_username* とデータベース・レベルの複写定義の詳細については、『Replication Agent リファレンス・マニュアル』を参照してください。

Replication Agent の設定要件

Replication Agent を設定する場合は、以下の要件を確認します。

プライマリ・データベース
プライマリ・データベースを設定します：

- プライマリ・データベースに Replication Agent のユーザ・ログイン名を追加し、そのユーザに、複写をサポートするために必要なタスクを実行できるように適切なパーミッションを付与します。
- プライマリ・データベースに、メンテナンス・ユーザのログイン名を追加します (Replication Server の **create connection** コマンドで指定)。
- (Oracle のみ)：サブリメンタル・ロギングを有効にします。

Replication Agent

Replication Agent インスタンスを設定します：

- プライマリ・データベース、Replication Server、および RSSD との各ネットワーク通信に対するコネクション設定が正しく設定されていることを確認します。
- (Microsoft SQL Server) 各プライマリ・データベース・サーバを初期化します。また、リモート TCP/IP コネクションとリモート DAC コネクションを許可するように、データベースを設定します。
- **ra_admin init** コマンドを使用して、Replication Agent を初期化します。このコマンドは、プライマリ・データベースが複写の準備ができているか検証し、プライマリ・データベース内に Replication Agent のシステム・オブジェクトを設定し、そして、RASD を初期化します (Oracle および Microsoft SQL Server のみ)。

Replication Server

Replication Server を設定します：

- **connect source** パーミッションと **create object** パーミッションが付与された Replication Agent ユーザ・ログイン名を使用します。
- RSSD 用の Replication Agent ユーザ・ログイン名を識別または作成します。
- プライマリ・データベース用とスタンバイ・データベース用の、データベース複写定義とサブスクリプションを定義します。
- RSSD では、異機種データ型のサポート・スクリプトを適用します。
- (Oracle) Replication Server のバージョンが 15.0 以前の場合、Replication Agent とともに配布されたスクリプトを適用して、Oracle エラー・クラスを正しく定義します。

RSSD

Replication Server のバージョンが 15.2 以降で、Replication Server の **using profile** 句を用いた **create connection** コマンドを使用して、すでにコネクションを作成済みの場合は、接続プロファイルによって RSSD オブジェクトはロードされているため、これらの手順は省略できます。Replication Server のバージョンが 15.0 以前の場合、次のようにして、Replication Server と RSSD との両方で Oracle エラー・クラスを正しく定義します。

- Replication Server で、`$$SYBASE/RAX-15_5/scripts/oracle/oracle_create_error_class_1_rs.sql` スクリプトを実行します。
- RSSD データベースで、`$$SYBASE/RAX-15_5/scripts/oracle/oracle_create_error_class_2_rssd.sql` スクリプトを実行します。
- Replication Server で、`$$SYBASE/RAX-15_5/scripts/oracle/oracle_create_error_class_3_rs.sql` スクリプトを実行します。

複数の Replication Agent

Replication Server の Multi-Path Replication™ 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、Multi-Path Replication 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持つ必要があります。
- 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持つ必要があります。このパラメータは Replication Agent for Oracle インスタンスをグループ内で識別するために使用されます。

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、**all**、**marked**、**unmarked** の各キーワードを使用します。

これらのパラメータとコマンド・キーワードの詳細については、『Replication Agent リファレンス・マニュアル』および『Replication Agent プライマリ・データベース・ガイド』を参照してください。

管理

Replication Agent の管理タスクおよび手順を確認します。

Replication Agent ソフトウェアのインストールの詳細については、『Replication Agent インストール・ガイド』を参照してください。

注意：この項の手順の例では、Replication Agent 管理ポートへのログインに使用する Open Client アプリケーションとして `isql` が示されていますが、任意の Open Client (または Open Client 互換の) アプリケーションを使用できます。

参照：

- 設定 (13 ページ)

Replication Agent インスタンスのステータスの判断

Replication Agent のステータスは、Replication Agent インスタンスの現在の状態とアクティビティによって構成されます。

Replication Agent の現在の状態を判断するには、以下のようになります。

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. Replication Agent インスタンスの現在の状態を取得します。

```
ra_status
```

このコマンドは、次のように Replication Agent インスタンスの現在の状態と、現在のアクティビティがあればそれを返します。

```
State Action
```

```
-----  
ADMIN Waiting for operator command  
(1 row affected)
```

Replication Agent の状態

Replication Agent は次のいずれかの状態になります。

状態は次のとおりです。

- Admin – インスタンスでは他の複製システム・コンポーネントへの接続が確立されていませんが、設定パラメータの変更やトランザクション・ログまたは RASD の管理など、管理コマンドを実行するために使用できます。

Replication Agent インスタンスが Admin 状態になっている場合は、複製プロセスは発生しません。

- Replicating – インスタンスはトランザクション・ログのスキャン、ログ・レコードおよび変更セット・データの処理、およびプライマリ Replication Server への LTL コマンドの送信など、通常の複製プロセスを実行しています。一部の管理コマンドは実行できません。
- Replicating (Resynchronization) – インスタンスが再起動され、プライマリ・データベースおよびレプリケート・データベースと再同期しています。
- Replication Down – エラーのため、複製を停止しています。

Replication Agent インスタンスの状態は、次のいずれかによって変更できます。

- Replication Agent が複製トランザクションを処理している間に発生する外部イベント (Replication Server コネクションでのネットワーク・エラーなど)
- オペレータの介入 (Replication Agent の状態を変更するコマンドの呼び出しなど)

状態変更イベントが発生してから Replication Agent インスタンスが実際に新しい状態になるまで、インスタンスは「遷移中」であるといえます。状態遷移中は、管理コマンドが無視される場合があります。

Admin 状態

Replication Agent インスタンスは、以下の場合に Admin 状態になります。

- インスタンスがデフォルト状態で起動された。
- インスタンスが **ra** ユーティリティの **-admin** オプションを使用して起動された。
- Replication Agent が複製状態になっているときに、Replication Agent の **quiesce** コマンドまたは **suspend** コマンドが呼び出された。

Admin 状態では、Replication Agent インスタンスは実行中ですが、プライマリ Replication Server (または、設定されている場合は RSSD) またはプライマリ・データベースへのコネクションが確立されていません。

Replication Agent インスタンスが Admin 状態になっているときは、Replication Agent 設定パラメータの値の変更を含むほとんどの管理タスクを実行できます。

注意： Admin 状態では、必要に応じてプライマリ・データベースへのコネクションを開いて、プライマリ・データベースからの結果を要求するコマンドを処理できます。

複製状態

Replication Agent インスタンスは、以下の場合に複製状態になります。

- インスタンスが **ra** ユーティリティの **-replicate** オプションを使用して起動された。
- Replication Agent が Admin 状態になっているときに、Replication Agent の **resume** コマンドが呼び出された。

注意： Replication Agent インスタンスは、プライマリ Replication Server でプライマリ・データベースへのコネクションが作成されている場合にのみ、複写状態に移行します。『Replication Agent プライマリ・データベース・ガイド』を参照してください。

複写状態では、新しい Replication Agent インスタンスはプライマリ・データベースとのコネクションとプライマリ Replication Server とのコネクション (および設定されている場合は RSSD とのコネクション) が維持され、その Log Reader コンポーネントは、複写するトランザクション用のトランザクション・ログをスキャンします。

Replication Agent インスタンスがトランザクション・ログ内のすべてのレコードの処理を終了した場合、状態は **Replicating** として続きます。Replication Agent インスタンスがログの最後に到達した場合は、次のようになります。

- Log Reader コンポーネントのログ・スキャン・プロセスが、**scan_sleep_increment** および **scan_sleep_max** 設定パラメータの値に従って「スリープ」します。
- Log Transfer Interface (LTI) コンポーネントが、Log Reader から受信したすべての変更セットを処理し、Replication Server にすべての LTL を送信し終わったら、新しい複写トランザクションがログに発生して Log Reader によってスキャンされるまで、複写スループットは発生しません。
- Replication Agent のインスタンスは、他のイベントによって Admin 状態または Replication Down 状態にならない限り、複写状態のままです。

Replicating (Resynchronization) 状態

Replication Agent は、**resume resync** コマンドが呼び出され、正常終了すると、**Replicating (Resynchronization)** 状態になります。この状態では、Replication Agent はプライマリ・データベースおよびレプリケート・データベースと再同期します。Replication Agent は、プライマリ・データベースおよびレプリケート・データベースとの再同期が完了すると、複写状態に戻ります。再同期の詳細については、『Replication Server 異機種間複写ガイド』と『Replication Server 管理ガイド』を参照してください。**resume** コマンドの詳細については、『Replication Agent リファレンス・マニュアル』を参照してください。

Replication Down 状態

Replication Agent のインスタンスは、以下の場合に Replication Down 状態になります。

- インスタンスが複製状態のときにリカバリ不可能なエラーが発生した。
- ネットワーク障害または通信エラーが原因で、プライマリ・データベースまたはプライマリ Replication Server とのコネクションが切断された。

Replication Agent は、Replication Down 状態になる前に接続が切断された場合、この接続の設定パラメータに記録された値を使用して接続を再度確立しようとしません。再度確立できなかった場合、Replication Agent のインスタンスは Replication Down 状態になります。

エラーが解決すると、Replication Agent は複製状態に復帰できます。

注意： Replication Down 状態での Replication Agent の動作は Admin 状態での操作と同じですが、Replication Down 状態が Replication Agent エラー中に発生する点のみが異なります。

Replication Agent のステータスの変更

Replication Agent インスタンスのステータスはその現在の動作状態を示し、このステータスにより、実行できる管理タスクを判断できます。

通常、次の 2 つの場合にのみ、Replication Agent インスタンスのステータスを変更します。

- 特定の管理作業またはメンテナンス作業を行う場合 (複製状態から Admin 状態に変更)
- 管理作業またはメンテナンス作業の終了後、またはエラーからの復旧後に、通常の複製作業にリストアする場合 (Admin 状態から複製状態に変更)

複製状態から Admin 状態へ

Replication Agent インスタンスのステータスを複製状態から Admin 状態に変更するには、**quiesce** コマンドまたは **suspend** コマンドを使用します。

quiesce コマンドと **suspend** コマンドの詳細については、『Replication Agent リファレンス・マニュアル』を参照してください。

Admin 状態から複製状態へ

Replication Agent インスタンスのステータスを Admin 状態から複製状態に変更するには、**resume** コマンドを使用します。

resume コマンドの詳細については、『Replication Agent リファレンス・マニュアル』を参照してください。

参照：

- Replication Agent での複製の停止 (77 ページ)
- 複製の開始 (76 ページ)

Replication Agent の統計

Replication Agent は、複製状態では、その内部コンポーネントのパフォーマンスに関する情報を記録しています。この情報を使用して Replication Agent のパフォーマンスを調整したり、問題をトラブルシューティングしたりできます。

Replication Agent のパフォーマンスに関する情報は、**ra_statistics** コマンドを使用して取得できます。**ra_statistics** コマンドを使用すると、統計カウンタをリセットすることもできます。

注意： Replication Agent インスタンスが複製状態に移行するごとに、統計カウンタは自動的にリセットされます。

ra_statistics コマンドと Replication Agent のパフォーマンス統計に関する詳細については、『Replication Agent リファレンス・マニュアル』を参照してください。

Replication Agent インスタンスの停止

各 Replication Agent インスタンスは、複製システムの他のすべてのコンポーネントや他の Replication Agent インスタンスとは関係なく起動および停止できます。

Replication Agent インスタンスを停止すると、そのプロセスもホスト・マシンで終了します。

注意： Replication Agent のすべての複製プロセスは、インスタンスを停止せずに停止できます。

Replication Agent インスタンスを停止するには、管理ポートにログインし、**shutdown** コマンドを起動します。**shutdown** コマンドには、次の2つのオプションがあります。

- 通常の停止 – まず Replication Agent インスタンスをクワイース状態にしてから、インスタンスを停止し、そのプロセスを終了します。
- 即時停止 – 最初にクワイース状態にせずに、Replication Agent インスタンスを停止し、そのプロセスをすぐに停止します。この方法を使用するには、**immediate** キーワードを、**shutdown** コマンドを起動するときに使用します。

注意： Replication Agent インスタンスが状態遷移中の場合、オプションを指定しないと (通常の停止)、**shutdown** コマンドは無視されます。**shutdown immediate** は、ある状態からの遷移を含め、どの状態でも無視されません。

Replication Agent インスタンスを通常どおりに停止するには：

- トランザクション・ログの読み込みを停止します。

- プライマリ・データベースへのコネクションを削除します。
- すでに内部キューにあるトランザクションの処理を終了します。
- 内部キューのトランザクションの LTL を正常に送信したら、Replication Server へのコネクションを削除します。
- プロセスを停止します。

参照：

- Replication Agent の起動 (40 ページ)
- Replication Agent での複製の停止 (77 ページ)

Replication Agent インスタンスの停止

Replication Agent を停止します。

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. 次のようにして、**shutdown** コマンドを起動します。

- Replication Agent インスタンスを通常に停止する場合：

```
shutdown
```

- Replication Agent インスタンスのステータスにかかわらず、即時停止する場合：

```
shutdown immediate
```

このコマンドは、最初にクワイーズせずに、Replication Agent インスタンスを即時に停止し、終了させます。

shutdown コマンドの詳細については、『Replication Agent リファレンス・マニュアル』を参照してください。

複製の開始

Replication Agent インスタンスの複製を開始します。

前提条件

プライマリ・データベースからトランザクションの複製を行う前に、「Replication Agent コネクティビティの設定」のすべての手順を完了している必要があります。また、Replication Agent インスタンスが実行中であることを確認します。

手順

Replication Agent インスタンスが複製状態のときは、プライマリ・データベースとのコネクションとプライマリ Replication Server とのコネクション (および設定され

ている場合は RSSD との接続)が維持され、その Log Reader コンポーネントは、複製するトランザクション用のトランザクション・ログをスキャンします。

1. Replication Agent 管理ポートにログインし、Replication Agent インスタンスが Admin 状態になっていることを確認します。

```
ra_status
```

2. Replication Agent インスタンスの複製を開始します。

```
resume
```

resume コマンドが呼び出された後、Replication Agent インスタンスは Admin 状態から複製状態になります。

3. **ra_status** コマンドを使用して、Replication Agent インスタンスが複製状態になっていることを確認します。

注意： Replication Agent インスタンスは、プライマリ Replication Server でプライマリ・データベースへの接続が作成されている場合にのみ、複製状態に移行します。Replication Server でのプライマリ・データベース・接続の作成の詳細については、『Replication Agent プライマリ・データベース・ガイド』を参照してください。

Replication Agent インスタンスが複製状態のときは、複製されるトランザクション用のトランザクション・ログがスキャンされ、プライマリ Replication Server に LTL が送信されます。

resume コマンドと Replication Agent が複製プロセスを開始する方法の詳細については、『Replication Agent リファレンス・マニュアル』を参照してください。

参照：

- Replication Agent コネクティビティの設定 (51 ページ)
- トラブルシューティング (127 ページ)
- Replication Agent の起動 (40 ページ)

Replication Agent での複製の停止

ここでは、Replication Agent インスタンスの複製の停止方法について説明します。

Replication Agent で複製を停止すると、以下の処理が実行されます。

- 内部 Log Reader コンポーネントと Log Transfer Interface コンポーネントによって通常の複製プロセスが停止します。

- プライマリ・データベースへのオープンなコネクションが解放され、Replication Server へのコネクションが削除されます。
- Replication Agent インスタンスは複写状態から Admin 状態になります。

Replication Agent インスタンスが Admin 状態の場合、このインスタンスは実行中で、管理コマンドを実行できますが、プライマリ・データベースへのコネクションとプライマリ Replication Server へのコネクション (および設定されている場合は RSSD へのコネクション) は維持されず、複写トランザクションは処理されません。

一部の管理作業では、Replication Agent インスタンスが Admin 状態である必要があります。正常に稼働している複写システムでは、これらの作業を実行するには、Replication Agent の複写を停止する必要があります。

Replication Agent の複写を停止するには、次の 2 つの方法があります。

- Replication Agent インスタンスをクワイース状態にして、複写をおだやかに停止します。
- Replication Agent インスタンスをサスペンドして、複写を即時に停止します。

Replication Agent インスタンスのクワイース

Replication Agent インスタンスをクワイース状態にすると、複写プロセスは緩やかに停止し、ログからのすべてのトランザクションは確実に読み込まれ、Replication Server へ確実に送信されます。

Replication Agent をクワイース状態にするには、次の手順に従います。

- (Oracle および Microsoft SQL Server) Log Reader コンポーネントは、オペレーションがなくなるまで、つまり Log Reader がログの最後に達するまで、トランザクション・ログからオペレーションの読み込みを続けます。Log Reader は、ログからスキャンした最後のオペレーションの処理を完了するまで、Log Transfer Interface コンポーネントに変更セット・データを送信し続けます。
- (UDB) Log Reader コンポーネントは、現在のスキャンが完了すると、トランザクション・ログからのオペレーションの読み込みを停止します。ログからスキャンした最後のオペレーションの処理を完了するまで、Log Transfer Interface コンポーネントに変更セット・データを送信し続けます。
- Log Transfer Interface コンポーネントは、Log Reader から受信した最後の変更セットの処理が完了すると直ちに、Replication Server への LTL コマンドの送信を停止します。
- Log Transfer Interface コンポーネントが入力キューの処理とその結果の LTL の送信を完了すると、Replication Agent インスタンスは、プライマリ・データベー

スへのすべてのコネクション (開いているものがあれば) を解放し、Replication Server (および、接続されている場合は RSSD) へのコネクションを削除します。

- Replication Agent インスタンスは複製状態から Admin 状態になります。
- 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. Replication Agent インスタンスをクワイイス状態にします。

```
quiesce
```

quiesce を実行後、Replication Agent インスタンスは複製状態から Admin 状態になります。

- 3. Replication Agent インスタンスが Admin 状態になっていることを確認します。

```
ra_status
```

注意： 内部キューが満杯で、プライマリ・データベースが依然としてログ・ファイルに新しいアクティビティを記録している場合、**quiesce** コマンドを実行すると、クワイイス処理が完了するまでにしばらく時間がかかり、Replication Agent インスタンスが Admin 状態への移行を完了するまで、多少の遅れが出る場合があります。

quiesce コマンドとその処理の詳細については、『Replication Agent リファレンス・マニュアル』を参照してください。

Replication Agent インスタンスのサスペンド

Replication Agent インスタンスをサスペンドすると、複製プロセスが即時に停止します。

Replication Agent をサスペンドするには、次の手順に従います。

- Log Reader コンポーネントは、トランザクション・ログのスキャンを直ちに停止し、Log Transfer Interface コンポーネントは Replication Server への LTL コマンドの送信を直ちに停止します。
- Replication Agent の内部キュー (Log Reader コンポーネントと Log Transfer Interface コンポーネントの入力キューと出力キュー) にあるすべてのデータは、それ以上処理されずにフラッシュされます。
- Replication Agent インスタンスは、プライマリ・データベースへのすべてのコネクション (開いているものがあれば) を解放し、Replication Server (および、接続されている場合は RSSD) へのコネクションを削除します。
- Replication Agent インスタンスは複製状態から Admin 状態になります。
- 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. Replication Agent インスタンスの複製をサスペンドします。

```
suspend
```

suspend を実行後、Replication Agent インスタンスは複製状態から Admin 状態になります。

3. Replication Agent インスタンスが Admin 状態になっていることを確認します。

```
ra_status
```

suspend コマンドの詳細については、『Replication Agent リファレンス・マニュアル』を参照してください。

複製システム・コンポーネントの移動

Replication Agent インスタンスとプライマリ・データベース・ログを、あるホスト・マシンから別のホスト・マシンに移動できます。

次のようなコンポーネントの移動が必要な場合があります。

- Replication Agent インスタンスを同じプラットフォームに移動します。
- Replication Agent インスタンスを別のプラットフォームに移動します。
- Oracle ログを移動します。

同じプラットフォームへの Replication Agent インスタンスの移動

Replication Agent インスタンスを、同じオペレーティング・システムおよびハードウェアを使用する他のホスト・マシンに移動します。

次に例を示します。

- Replication Agent インスタンス `inst1` を `host1` から `host2` に移動します。
- Replication Agent インスタンスは次のロケーションにインストールされています。
 - (Linux または UNIX) `/opt/sybase/RAX-15_n`
 - (Windows) `C:\¥sybase¥RAX-15_n``n` は 0、1、2、または 5 です。

1. Replication Agent を `host2` にインストールします。`host1` と同じインストール・ディレクトリ構造を使用することをおすすめします。

注意： `host2` にインストールする Replication Agent のバージョン文字列は、`host1` の Replication Agent のバージョン文字列と同じにしてください。

2. `host2` で、プライマリ・データベースの必要に応じて次のいずれかをインストールします。
 - JDBC ドライバ (Oracle または Microsoft SQL Server)
 - IBM DB2 クライアント

CLASSPATH 環境変数を更新します。

3. Replication Agent for UDB (すべてのバージョン)、Replication Agent for Oracle 15.7.1、または Replication Agent for Microsoft SQL Server 15.0 のインスタンスを移動する場合は、この手順を省略します。
 - (Linux または UNIX) プライマリ Oracle データベースのオンライン・ログおよびアーカイブ・ログが格納されているドライブをマウントします。
 - (Windows) Replication Agent for Oracle バージョン 15.6 以前、または Replication Agent for Microsoft SQL Server バージョン 15.1 以降の場合は、ネットワーク・ドライブを、プライマリ・データベースのトランザクション・ログおよびアーカイブ・ログが格納されているロケーションにマップします。
4. inst1 (host1 上) を停止します。
5. inst1 インスタンス・ディレクトリを host2 に (host1 から) コピーします。
 - (Linux または UNIX) \$SYBASE/RAX-15_n
 - (Windows) %SYBASE%\RAX-15_n
6. inst1 インスタンスを参照する、interfaces ファイル (Linux または UNIX) または sql.ini ファイル (Windows) があれば更新します。エントリのホスト名を host2 に (host1 から) 変更します。

Sybase Control Center (SCC) を使用して Replication Agent インスタンスをモニタする場合は、SCC で inst1 インスタンスのホスト名を更新します。

7. host2 の Replication Agent を host1 上と同じインストール・ディレクトリにインストールするように選択した場合は、この手順を省略します。
host2 上のインストール・ディレクトリが host1 で使用されているディレクトリと異なる場合は、\$SYBASE/RAX-15_n/inst1/inst1.cfg 設定ファイル (Linux または UNIX) または %SYBASE%\RAX-15_n\inst1\inst1.cfg 設定ファイル (Windows) で次のプロパティを編集します。
 - **asa_install_dir**
 - **lob_cache_directory**
 - **log_directory**
 - **rasd_backup_dir**
 - **rasd_database**
 - **rasd_trace_log_dir**
 - **rasd_tran_log**
 - **rasd_tran_log_mirror**

注意：すべての Replication Agent のバージョンまたはタイプで、上記のすべてのプロパティが使用されるとは限りません。

たとえば、Linux または UNIX で **log_directory** を編集するには、以下を

```
log_directory=<host1_install_dir>/RAX-15_n/inst1/log
```

次のように変更します。

```
log_directory=<host2_install_dir>/RAX-15_n/inst1/log
```

<host1_install_dir> と <host2_install_dir> は、host1 および host2 上の Replication Agent のインストール・パスです。

Windows では、パス名に円記号を2つ使用します。たとえば、次のパス名は

```
log_directory=C¥:¥¥sybase¥¥RAX-15_5¥¥inst1¥¥log
```

次のように変更します。

```
log_directory=C¥:¥¥software¥¥syb¥¥RAX-15_5¥¥inst1¥¥log
```

8. host2 で、inst1 を Admin 状態で起動します。
9. Replication Agent for UDB (すべてのバージョン)、Replication Agent for Oracle 15.7.1、または Replication Agent for Microsoft SQL Server 15.0 のインスタンスを移動する場合は、この手順を省略します。
Replication Agent for Oracle バージョン 15.6 以前、または Replication Agent for Microsoft SQL Server バージョン 15.1 以降の場合は、デバイス・パスを調整します。
10. inst1 インスタンス (host2 上) を再開します。

インスタンスは、移動前に停止した場所から複製を続行します。

Replication Agent の管理

Replication Agent for Oracle、Replication Agent for Microsoft SQL Server、および Replication Agent for UDB の管理作業とメンテナンス作業をレビューします。

Replication Agent の管理作業とメンテナンス作業には以下のものがあります。

- Replication Agent の初期化
- Replication Agent の初期化解除
- トランザクション・ログのトランケーション
- プライマリ・データベース内の Replication Agent オブジェクトのバックアップ

Replication for Oracle

Replication Agent for Oracle は、Oracle LogMiner を使用して複写されたトランザクションを取り込みます。プライマリ・データベース内に作成されたオブジェクトは、ストアド・プロシージャの複写を容易にします。これらのデータベース・オブジェクトには、定期的なメンテナンスは必要ありません。

設定により異なりますが、Replication Agent が、アーカイブされたトランザクション・ログにアクセスする場合 (デフォルト)、またはオンライン・トランザクシ

ン・ログのみを処理する場合があります。redo ログとアーカイブ・ログ・ファイルの詳細については、『Replication Agent プライマリ・データベース・ガイド』を参照してください。

Replication Agent for Microsoft SQL Server

Replication Agent for Microsoft SQL Server は、ネイティブの Microsoft SQL Server ログを使用して複製されたトランザクションを取り込みます。プライマリ・データベース内に作成されたオブジェクトは、複製を容易にします。これらのデータベース・オブジェクトには、定期的なメンテナンスは必要ありません。

プライマリ・データベース・トランザクション・ログの自動トランケート方法については、『Replication Agent プライマリ・データベース・ガイド』を参照してください。

Replication Agent for UDB

Replication Agent for UDB は、ネイティブの DB2 ログを使用して複製されたトランザクションを取り込みます。Replication Agent for UDB はプライマリ・データベース内にオブジェクトを作成しシステム・データを格納します。しかし、これらのデータベース・オブジェクトには、定期的なメンテナンスは必要ありません。

設定により異なりますが、Replication Agent が、オンライン・トランザクション・ログのみを処理する場合(デフォルト)、または、アーカイブされたトランザクション・ログにアクセスする場合があります。オンライン・トランザクション・ログ・ファイル、アーカイブされたトランザクション・ログ・ファイル、およびログのトランケーションについては、『Replication Agent プライマリ・データベース・ガイド』を参照してください。

Replication Agent インスタンスの初期化

Replication Agent インスタンスを初期化し、必要とされるオブジェクトをプライマリ・データベースに作成します。

前提条件

Replication Agent のインスタンスが実行中であり、プライマリ・データベースへの接続が確立されている必要があります。

手順

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. この Replication Agent インスタンスに関連付けられているオブジェクトがすでにプライマリ・データベースに存在するかどうかを判断します。

```
ra_admin
```

Replication Agent オブジェクトが存在しない場合、**ra_admin** コマンドを実行しても何も情報は返されません。この手順を続行して Replication Agent を初期化

します。この手順では、複写をサポートするプライマリ・データベース内にオブジェクトも作成されます。

注意：

ra_admin コマンドは、Replication Agent for Microsoft SQL Server と Replication Agent for UDB のオブジェクトを **ra_admin_instance_prefix** 設定パラメータの値に基づいて検索し、Replication Agent for Oracle のオブジェクトを **ra_admin_prefix** パラメータの値に基づいて検索します。トランザクション・ログの作成後に **ra_admin_instance_prefix** パラメータまたは **ra_admin_prefix** パラメータの値が変更されると、**ra_admin** コマンドで以前に作成されたオブジェクトを見つけることができません。

Replication Agent オブジェクトがプライマリ・データベース内に存在する場合、**ra_admin** コマンドによってオブジェクトの一覧が返されます。

Replication Agent インスタンスのオブジェクトが存在する場合は、この手順を完了する必要はありません。

3. トランザクション・ログ・コンポーネントのデータベース・オブジェクト名のプレフィクスに特定の文字列を使用するには、**ra_config** コマンドを使用して、**ra_admin_instance_prefix** の値 (Microsoft SQL Server および UDB) または **ra_admin_prefix** (Oracle) パラメータを設定します。

```
ra_config ra_admin_instance_prefix, XXX
```

ここで **XXX** は、**ra_admin_instance_prefix** または **ra_admin_prefix** パラメータの新しい値となる 1～3 文字の文字列で、オブジェクトの作成時にデータベース・オブジェクト名に使用されるプレフィクス文字列となります。デフォルト値は **ra_** です。

注意： **ra_admin_prefix_chars** パラメータの値では、プレフィクス文字列で使用できるアルファベット以外の文字が指定されます (**ra_admin_instance_prefix** または **ra_admin_prefix** パラメータの値)。プライマリ・データ・サーバによって、データベース・オブジェクト名に使用できる文字が制限されることがあります。データベースごとに使用可能な文字の詳細については、『Replication Agent プライマリ・データベース・ガイド』を参照してください。

また、**ra_config** を使用して、**ra_admin_instance_prefix** (Microsoft SQL Server および UDB) または **ra_admin_prefix** (Oracle) パラメータの現在の値を判断することもできます。

```
ra_config ra_admin_instance_prefix
```

ra_config を呼び出して値を指定せずに設定パラメータを指定すると、パラメータの現在の値が返されます。

4. お使いの Replication Agent で RASD が使用されている場合は、プライマリ・データベースをクワイスさせなければならないことがあります。
- (Replication Agent for Oracle) 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は、プライマリ・データベースのクワイスを回避できる場合があります。データベース再同期の設定の詳細については、『Replication Server 管理ガイド』を参照してください。
 - (Replication Agent for Microsoft SQL Server) プライマリ・データベースをクワイスするか、他の方法でデータベース・オブジェクトまたはスキーマが変更される可能性があるすべての DDL 操作を阻止する必要があります。適切なパーミッションまたは権限を持つユーザ・ログインを使用してプライマリ・データ・サーバにログインし、プライマリ・データベースをクワイスします(または、データベース・オブジェクトまたはスキーマが変更される可能性があるすべての DDL 操作を阻止するのに必要なコマンドを実行します)。
 - (Replication Agent for UDB) RASD がいないため、特に対処は必要はありません。
5. Replication Agent のインスタンスを初期化し、プライマリ・データベースにオブジェクトを作成します。

```
ra_admin init
```

注意： `init` キーワードを使用して `ra_admin` を呼び出すと、Replication Agent オブジェクト (現在、`ra_admin_instance_prefix` または `ra_admin_prefix` パラメータで指定されているプレフィクス文字列を使用) がすでにプライマリ・データベースに存在する場合、エラー・メッセージが返されます。

`init` オプションを使用して `ra_admin` コマンドを呼び出すと、Replication Agent では次のことが行われます。

- プライマリ・データベースの互換性設定を確認します。
- プライマリ・データベースで実行される SQL スクリプトを作成します。このスクリプトは、Replication Agent オブジェクトを作成します。

RASD を使用する Replication Agent の場合、その RASD はプライマリ・データベースからの情報で初期化されます。

注意： スクリプトを生成するだけで、実行しないように Replication Agent を設定するには、`pdb_auto_run_scripts` パラメータの値を `false` に設定してから `ra_admin` コマンドを呼び出します。トランザクション・ログの作成を完了するには、`pdb_auto_run_scripts` を `true` に設定し、`ra_admin init` コマンドを再度実行する必要があります。このスクリプトは参照時に使われるだけです。プライマリ・データベース内でスクリプトを手動で実行しても、Replication Agent インスタンスは初期化されません。

初期化が成功すると、スクリプトは `scripts/xlog/installed` ディレクトリ内の `partinit.sql` ファイルに保存されます。

初期化が成功しなかった場合、プライマリ・データベースは変更されず、スクリプトは `scripts/xlog` ディレクトリ内の `partinit.sql` ファイルに保存されません。Replication Agent のエラー・ログ、さらに必要に応じてプライマリ・データベースのエラー・ログを確認し、初期化が成功しなかった理由を突き止めます。

参照：

- Replication Agent コネクティビティの設定 (51 ページ)
- Replication Agent の起動 (40 ページ)
- トラブルシューティング (127 ページ)

Replication Agent インスタンスの初期化解除

Replication Agent インスタンスの初期化を解除し、そのオブジェクトをプライマリ・データベースから削除します。

前提条件

プライマリ・データベースからそのオブジェクトを削除し Replication Agent の初期化を解除するには、Replication Agent インスタンスが Admin 状態で実行中であることが必要です。

手順

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. Replication Agent オブジェクトがプライマリ・データベース内に存在することを確認します。

```
ra_admin
```

Replication Agent オブジェクトがプライマリ・データベース内に存在しない場合、`ra_admin` コマンドでオブジェクトに関する情報は返されません。オブジェクトが存在しない場合、この手順を完了する必要はありません。

注意： `ra_admin` コマンドは、`ra_admin_instance_prefix` 設定パラメータの現在の値に基づいて Replication Agent オブジェクトを探します。Replication Agent インスタンスが初期化された後で `ra_admin_instance_prefix` パラメータの値が変更された場合、`ra_admin` コマンドは以前に作成された Replication Agent オブジェクトを見つけることができません。

この Replication Agent インスタンス用のオブジェクトが存在する場合、**ra_admin** コマンドでそのオブジェクトの名前のリストが返されます。この手順を続行して、プライマリ・データベースからオブジェクトを削除します。

3. プライマリ・データベースのマーク付けされたすべてのテーブルの複写を無効にします。

```
pdb_setreptable all, disable
```

pdb_setreptable コマンドを **all** キーワードと **disable** キーワードを使用して起動すると、Replication Agent によって、プライマリ・データベース内のマーク付けされたすべてのテーブルの複写が無効になります。

4. プライマリ・データベースのマーク付けされたすべてのプロシージャの複写を無効にします。

```
pdb_setrepproc all, disable
```

5. プライマリ・データベースのマーク付けされたすべてのテーブルのマーク付けを解除します。

```
pdb_setreptable all, unmark
```

pdb_setreptable コマンドを **all** キーワードと **unmark** キーワードを使用して起動すると、Replication Agent によって、プライマリ・データベース内のマーク付けされたすべてのテーブルから複写のマーク付けが削除されます。

注意： **pdb_setreptable all, unmark** を使用してプライマリ・データベースのすべてのテーブルから一度でマーク付けを解除することは、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付けを解除してください。

6. プライマリ・データベースのマーク付けされたすべてのプロシージャのマーク付けを解除します。

```
pdb_setrepproc all, unmark
```

pdb_setrepproc コマンドを **all** キーワードと **unmark** キーワードを使用して起動すると、Replication Agent によって、プライマリ・データベース内のマーク付けされたすべてのプロシージャから複写のマーク付けが削除されます。通常、プライマリ・データベース内のどのオブジェクトにも複写のマーク付けがされている場合は、Replication Agent のトランザクション・ログを削除することはできません。

7. Replication Agent オブジェクトを削除します。

```
ra_admin deinit
```

deinit キーワードを付けて **ra_admin** コマンドを起動すると、プライマリ・データベース内に Replication Agent オブジェクトが存在しない場合は、エラー・メッセージが返されます。

deinit キーワードを付けて **ra_admin** コマンドを起動した後、Replication Agent は、プライマリ・データベースからオブジェクトを削除し、Replication Agent の初期化を解除するスクリプトを作成します。

注意： **ra_admin** コマンドを起動する前に、**pdb_auto_run_scripts** パラメータの値を **false** に設定しておくこと、Replication Agent がスクリプトを作成するだけで、実行しないように設定できます。Replication Agent オブジェクトの削除を完了するには、**pdb_auto_run_scripts** を **true** に設定し、**ra_admin init** コマンドを再実行する必要があります。このスクリプトは参照時に使われるだけです。プライマリ・データベース内でスクリプトを手動で実行しても、Replication Agent インスタンスは初期化解除されません。

初期化解除が成功した場合、スクリプトは `RAX-15_5¥inst_name¥scripts ¥xlog¥installed` ディレクトリの `partdeinit.sql` という名前のファイルに保存されます。

初期化解除が成功しなかった場合、スクリプトは `RAX-15_5¥inst_name ¥scripts¥xlog` ディレクトリの `partdeinit.sql` という名前のファイルに保存されます。

参照：

- Replication Agent の起動 (40 ページ)

強制的な初期化解除

スクリプトの実行の失敗後に残っている Replication Agent オブジェクトを強制的に削除します。

前提条件

エラーによってスクリプトの実行に失敗した場合は、プライマリ・データベースのエラー・ログと Replication Agent システム・ログを参照してエラーを評価し、是正アクションが必要かどうかを判断します。

手順

deinit キーワードを指定して **ra_admin** コマンドを呼び出すと、Replication Agent によって `partdeinit.sql` スクリプトが作成されます。このスクリプトが正常に実行されると、すべての Replication Agent オブジェクトがプライマリ・データベースから削除されます。何らかの理由で `partdeinit.sql` スクリプトが失敗すると、Replication Agent オブジェクトがプライマリ・データベースから削除される場合と、プライマリ・データベースに残る場合があります。スクリプトの実行の失敗後に Replication Agent オブジェクトの削除を完了するには、次の手順に従います。

ra_admin を呼び出します。


```
ra_admin deinit, force
```

force キーワードを使用すると、Replication Agent は、エラーが発生しても `partdeinit.sql` スクリプトが終了するまでスクリプトの実行を継続します。

トランザクション・ログのトランケーション

Replication Agents for Oracle、Replication Agents for Microsoft SQL Server、および Replication Agents for UDB は、トランザクション・ログの自動および手動のトランケーションをサポートしています。

自動のログ・トランケーションはいつでも有効、または無効にでき、自動のログ・トランケーションを有効または無効にすることにより、Replication Agent のトランザクション・ログはいつでも手動でトランケートできます。

注意： データベースのタイプと Replication Agent の設定により異なりますが、Replication Agent は、データベースのオンライン・ログまたはアーカイブされたログのいずれかをトランケートします。データベースのアーカイブ・ログをトランケートするように Replication Agent を設定することをおすすめします。詳細については、『Replication Agent プライマリ・データベース・ガイド』を参照してください。

Replication Agent がそのトランザクション・ログを、自動またはオンデマンド (手動) のいずれかでトランケートする場合、そのトランケーション・ポイントは、プライマリ Replication Server から受信した最後の LTM ロケータにより決定されます。

自動トランケーション

トランザクション・ログの自動トランケーションには、次の 2 つのオプションがあります。

- 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、自動でトランケートする
- 指定した時間間隔ごとに定期的にトランケートする

Replication Agent がそのトランザクション・ログを自動または手動のいずれかでトランケートする場合、そのトランケーション・ポイントは、プライマリ Replication Server から受信した最後の LTM ロケータにより決定されます。

自動ログ・トランケーションの有効化

トランザクション・ログを自動的にトランケートします。

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. 自動ログ・トランケーションを有効にし、自動トランケーションのタイプを指定します。

- 指定した時間間隔での自動ログ・トランケーションを有効にするには：

```
ra_config truncation_type, interval
ra_config truncation_interval, N
```

ここで、*N*は自動トランケーションを行う間隔(単位は分)です。

注意： `truncation_interval` の最大値は **720** です。

- 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の、自動ログ・トランケーションを有効にするには：

```
ra_config truncation_type, locator_update
```

`truncation_interval` 設定パラメータと `truncation_type` 設定パラメータの詳細については、『Replication Agent リファレンス・マニュアル』を参照してください。

自動ログ・トランケーションの無効化

自動トランケーションを無効にします。

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. 自動ログ・トランケーションを無効にするには：

```
ra_config truncation_type, command
```

注意： `truncation_type` パラメータの値が `interval` で、`truncation_interval` パラメータの値が **0** (ゼロ) の場合、自動ログ・トランケーションは実質的に無効です。

Replication Agent トランザクション・ログの手動によるトランケーション

自動ログ・トランケーションが無効になっている場合、Replication Agent のトランザクション・ログは手動で定期的にトランケートする必要があります。

トランザクション・ログを手動でトランケートするには：

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. Replication Agent のトランザクション・ログをトランケートします。

```
pdb_truncate_xlog
```

`pdb_truncate_xlog` コマンドは非同期であり、緊急のエラーが発生しない限り、成功または失敗を返しません。

`pdb_truncate_xlog` コマンドの詳細については、『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 の自動ログ・トランケーション機能の代替手段としては、Scheduler ユーティリティを使用して、スクリプトで `pdb_truncate_xlog` コマンドを実行します。

プライマリ・データベースのバックアップ

Replication Agent では、プライマリ・データベースにある Replication Agent オブジェクトの自動バックアップと自動リストアをサポートしていません。

プライマリ・データベース・ベンダが提供するデータベース・バックアップ・ユーティリティを使用して、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・オブジェクトを定期的にバックアップすることをおすすめします。

Replication Agent システム・データベース (RASD) の管理

Replication Agent for Oracle と Replication Agent for Microsoft SQL Server では埋め込みデータベースを使用します。これは、RASD 用 SQL Anywhere によって管理されます。

RASD

Replication Agent for Oracle または Replication Agent for Microsoft SQL Server の各インスタンスは、RASD の情報によって、トランザクション・ログのデータベース構造やスキーマ・オブジェクトを認識します。

注意： RASD は、Replication Agent for UDB では使用できません。

Replication Agent インスタンスを作成する場合、RASD は自動的に作成されますが、**ra_admin init** コマンドを使用して、Replication Agent インスタンスを初期化するまで情報は含まれません。Replication Agent インスタンスを作成する場合、そのインスタンスは：

- プライマリ・データベースに問い合わせ、データベース構造またはスキーマに関する情報を取得します。
- データベース・スキーマに関する情報をその RASD に格納します。

注意： Replication Agent の初期化は、複製システムを設定するのに必要な作業の 1 つで、前提条件がいくつかあります。

RASD に最初にデータが入力された後、その内容は、通常の複製中に (介入なしに)、プライマリ・データベースと自動的に同期されます。

複製が行われない場合、RASD の内容は古くなります (プライマリ・データベースと同期されないため)。このため、使用する前に再構築する必要があります。

DDL コマンド

大半の一般的なデータ定義言語 (DDL) コマンド、およびプライマリ・データベース内で実行されたシステム・プロシージャはトランザクション・ログに記録され、

レプリケート・データベースに複写されます。Replication Agent がこれらの DDL トランザクションの複写を処理するときに、RASD は自動的に更新されます。

注意：DDL 複写は、Replication Agent for UDB では使用できません。

DDL コマンドまたはシステム・プロシージャがプライマリ・データベース・スキーマ内で変更され、Replication Agent がそのコマンドまたはプロシージャを認識できず、自動的にその RASD を更新する場合、その後のトランザクションが RASD に記録されていないオブジェクトのデータを変更すると、複写の失敗が発生します。そのような場合、プライマリ・データベースをクワイシし、Replication Agent を再初期化して、RASD の更新を強制する必要があります。

既存のデータベース・オブジェクトに影響する DDL トランザクションを処理するごとに、Replication Agent は対象オブジェクトのメタデータの新しいバージョンを RASD に作成します。各オブジェクトのバージョンは、DDL トランザクションが変更した LTM ロケータ値によって識別されます。

オブジェクトの以前のバージョンは、システム・リカバリに十分な期間、RASD に保持する必要があります。たとえば、DDL によって変更される前の、あるオブジェクトに関連したトランザクションを再生しようとする、そのオブジェクトの現在のバージョンではエラー (またはデータの矛盾) になります。

注意：Replication Agent は、リストアされたトランザクション・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。

オブジェクト・バージョンと LTM ロケータ値

Replication Agent は、現在のオブジェクト・バージョン文字列と現在の LTM ロケータ値を比較することによって、各オブジェクトのどのバージョンを使用するかを決定します。現在の LTM ロケータ値がそのオブジェクト・バージョンの値以上の場合、現在のオブジェクト・メタデータが使用されます。現在の LTM ロケータ値がそのオブジェクト・バージョンの値よりも小さい場合、以前のバージョンのオブジェクト・メタデータが使用されます。

定期的にトランケートしないと、RASD のサイズは無制限に増殖し、オブジェクト・メタデータのバージョンがどんどん追加されていきます。

参照：

- Replication Agent インスタンスの作成 (13 ページ)
- RASD のトランケーション (98 ページ)
- RASD の更新 (93 ページ)

無効なデバイスのパス

ra_devicepath コマンドで無効なデバイスのパスを指定すると、その後の再初期化に失敗します。また、RASD 内のログ・デバイス情報はすべてクリアされます。

注意： **ra_devicepath** コマンドは Replication Agent for Microsoft SQL Server でのみ使用できます。

たとえば、次の **ra_devicepath** コマンドでのパス指定は無効です。

```
1> ra_devicepath 1, C:¥invalid_path¥invalid1.log
2> go
```

ra_helpdevice コマンドの結果は、このパスに対応するデバイス状態が“INVALID”であることを示します。無効なパスのために、その後の再初期化は失敗します。

```
1> ra_admin refresh
2> go
Msg 32000, Level 20, State 0:
Server 'myserver', Procedure 'ra_admin refresh', Line 1:
Command <ra_admin refresh> failed - Replication initialization
failed because: C:¥invalid_path¥invalid1.log (The system cannot find
the file specified)
```

この初期化の失敗の後、**ra_helpdevice** コマンドは情報を返しません。ログ・デバイス・レポジトリがクリアされているためです。

ログ・デバイス・レポジトリがクリアされるのを回避するには、**ra_devicepath** コマンドでログ・デバイス・レポジトリを更新する前に、新しいデバイスのパスをすべて確認します。

RASD の更新

RASD は、通常の複製アクティビティ中に自動的に更新されます。

注意： RASD が破損したと疑われる場合にのみ、Sybase サポート・センタの推奨に従って、RASD を強制的に更新してください。

RASD を強制的に更新するには、次の手順に従います。

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. Replication Agent インスタンスの現在の状態を取得します。

```
ra_status
```

3. Replication Agent が Admin 状態の場合、この手順を省略して、手順 4 に進みます。

Replication Agent が複製状態の場合：

- a) Replication Agent インスタンスの複製をサスペンドします。

```
suspend
```

- b) Replication Agent インスタンスが Admin 状態になっていることを確認します。

```
ra_status
```

4. Replication Agent を再初期化し、RASD を更新するよう強制します。

```
ra_admin refresh
```

注意： Replication Agent for Microsoft SQL Server と Replication Agent for UDB では、**ra_admin refresh** コマンドによって、マーク付け情報または設定は上書きされません。また、RASD 内のすべてのログ情報がプライマリ・データ・サーバから返されるものと一致する場合は、RASD 内のログ・デバイスへの既存のパス情報も上書きされません。

RASD 内の特定の各トランザクション・ログまたは各デバイスに対して、プライマリ・データ・サーバから返される情報と一致しない情報が1つでもあれば、**ra_admin refresh** は、そのトランザクション・ログまたはデバイスに対する RASD 記録を、プライマリ・データ・サーバから返される情報で上書きします。

5. Replication Agent インスタンスの複写をレジュームします。

```
resume
```

6. Replication Agent が複写状態になっていることを確認します：

```
ra_status
```

参照：

- [トラブルシューティング \(127 ページ\)](#)

ログ・デバイス・レポジトリの更新

Replication Agent は、Replication Agent インスタンスが初期化される際に、プライマリ・ログ・デバイスに関する情報を RASD に格納します。RASD 内のログ・デバイス情報はログ・デバイス・レポジトリとして参照されます。

RASD 内の他の情報とは異なり、ログ・デバイス・レポジトリは、**ra_updatedevices** コマンドを使用していつでも更新できます。

注意： プライマリ・データベースでログ・デバイスを追加する、削除する、拡張する、または移動する場合、Replication Agent のログ・デバイス・レポジトリを更新する必要があります。Oracle ASM を使用して redo ログを管理し、ASM ディスク・グループからディスクを追加または削除する場合、デバイス・レポジトリを更新する必要があります。プライマリ・データベースでのログ・デバイスのすべ

ての変更は、Replication Agent のログ・デバイス・レポジトリを更新することによって調整することをおすすめします。

ログ・デバイス・レポジトリを更新すると、Replication Agent では以下の処理が実行されます。

- プライマリ・データベースに、すべてのログ・デバイスに関する情報を問い合わせる。
- プライマリ・データベースから返された情報を、ログ・デバイス・レポジトリに記録された情報と比較する。
- プライマリ・データベースから返される情報が以下の場合は、その新しい情報でログ・デバイス・レポジトリを更新する。
 - プライマリ・データベースから返された情報と一致しない、ログ・デバイス・レポジトリ内の既存のログ・デバイスに関する情報がある場合、または
 - プライマリ・データベースから返された情報に、新しいログ・デバイスに関する情報がある場合。

プライマリ・サイトでのログ・デバイスへのパスが、スタンバイ・サイトでの対応するログ・デバイスへのパスと異なる場合、**ra_devicepath** を使用して RASD に記録されているそのログ・デバイスへのパスを指定する必要があります。

注意： **ra_devicepath** コマンドは Replication Agent for Microsoft SQL Server でのみ使用できます。Replication Agent のログ・デバイス・レポジトリを更新するときに、プライマリ・データベースをクワイースする必要はありません。

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. Replication Agent インスタンスの現在の状態を取得します。

```
ra_status
```

3. Replication Agent が Admin 状態の場合、この手順を省略して、手順 4 に進みます。

Replication Agent が複製状態の場合：

- a) Replication Agent インスタンスの複製をサスペンドします。

```
suspend
```

- b) Replication Agent インスタンスが Admin 状態になっていることを確認します。

```
ra_status
```

4. プライマリ・データベースでのログ・デバイスの変更を、Replication Agent のログ・デバイス・レポジトリを更新することによって調整する場合、Replication Agent が Admin 状態になった後で、プライマリ・データベースでログ・デバイスを変更します。

5. RASD のログ・デバイス・レポジトリを更新します：

```
ra_updatedevices
```

6. ログ・デバイスへのパスを指定するには、**ra_devicepath** を使用します：

```
ra_devicepath device, dev_path
```

構文の説明は次のとおりです。

- *device* はデバイス ID です。
- *dev_path* は、Replication Agent がそのログ・デバイスにアクセスするために使用する (オプションの) 代替用のパスです。

注意： パスを指定する必要があるログ・デバイスごとに、**ra_devicepath** を毎回起動する必要があります。

7. Replication Agent インスタンスの複写を開始します。

```
resume
```

プライマリ・データベースでのログ・デバイスの変更を反映する必要があるごとに、何回でも、ログ・デバイス・レポジトリを更新できます。

RASD のバックアップ

どのデータベースでも同様ですが、デバイスに障害があった場合のデータ消失を避けるために、RASD を定期的にバックアップする必要があります。各バックアップは別々のディレクトリに保存され、その名前はバックアップをする日付と時刻で構成されます。最後に実行されたバックアップは、その日付のディレクトリ以外に、`backup` ディレクトリにも保存されます。

注意： RASD は常に、バックアップしてから、トランケートすることをおすすめします。また、プライマリ・データベースのリストアの際に、RASD が相対的に同じポイントでリストアされるように、RASD のバックアップとプライマリ・データベースのバックアップを同期する必要があります。

Replication Agent は、**rasd_backup_dir** 設定パラメータで指定されたディレクトリに RASD バックアップ・ファイルを配置します。Replication Agent インスタンスがどの状態にあるときでも、いつでも、RASD をバックアップできます。

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. RASD をバックアップします：

```
rasd_backup
```

バックアップが正常に終了すると、Replication Agent から確認のメッセージが返されます。

Replication Agent が、**rasd_backup_dir** パラメータで指定されたディレクトリを見つけられない場合、またはそのディレクトリに RASD バックアップ・ファイルを書

きだめない場合 (パーミッション問題などで)、エラーが返されます。これらのエラーの原因を修正しないと、RASD を正常にバックアップできません。

RASD のリストア

RASD が破損した場合 (デバイスの障害などで)、最新のバックアップ・ファイルからデータベースをバックアップできます。

Replication Agent は、**rasd_backup_dir** 設定パラメータで指定されたディレクトリから RASD バックアップ・ファイルを取得します。**rasd_backup_dir** パラメータの詳細については、『Replication Agent リファレンス・マニュアル』を参照してください。

注意： RASD をリストアするには、Replication Agent インスタンスが Admin 状態である必要があります。

バックアップ・ファイルから RASD をリストアするには、以下の手順に従います。

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. Replication Agent インスタンスの現在の状態を取得します。

```
ra_status
```

3. Replication Agent が Admin 状態の場合、この手順を省略して、手順 4 に進みます。

Replication Agent が複写状態の場合：

- a) Replication Agent インスタンスの複写をサスペンドします。

```
suspend
```

- b) Replication Agent インスタンスが Admin 状態になっていることを確認します。

```
ra_status
```

4. RASD をリストアします：

```
rasd_restore backup_directory
```

ここで、**backup_directory** には、リストアするバックアップが含まれます (たとえば、2012-07-04_14.21.34)。パラメータを付けずに **rasd_restore** を実行すると、最新のバックアップでのみリストアされます。

リストアが正常に終了すると、Replication Agent から、RASD のリストアが成功した旨の確認のメッセージが返されます。

Replication Agent が、**rasd_backup_dir** パラメータで指定されたディレクトリを見つけれない場合、またはそのディレクトリの RASD バックアップ・ファイ

ルを読み込めない場合 (パーミッション問題などで)、エラーが返されます。これらのエラーの原因を修正しないと、RASD をリストアできません。

5. Replication Agent インスタンスの複写をレジュームします。

```
resume
```

参照：

- [トラブルシューティング \(127 ページ\)](#)

RASD のトランケーション

RASD が無制限に増殖するのを防ぐために、プライマリ・データベース・オブジェクトのメタデータの古いバージョンを定期的にトランケートできます。

注意： トランケートする前に、**rasd_backup** を使用して RASD をバックアップします。

RASD には、次の 2 つのタイプのデータベース・オブジェクトに対する定義が格納されています。

- アーティクル - 複写のマーク付けがされたテーブルとストアド・プロシージャ。
- ユーザ - プライマリ・データベース内のトランザクションを適用するデータベース・ユーザ。

ra_truncatearticles、**ra_truncateusers**、**rasd_trunc_schedule** の各コマンドを使用して、RASD のサイズを管理します。これらのコマンドの詳細については、『[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 インスタンスがどの状態にあるときでも、いつでも、RASD をトランケートできます。

参照：

- [RASD のバックアップ \(96 ページ\)](#)

RASD での旧バージョンのアーティクルのトランケート

RASD からアーティクルをトランケートします。

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. RASD 内のアーティクルをトランケートします：

```
ra_truncatearticles NNN
```

ここで、*NNN* は、最も古い、現在のバージョン以外の維持されるアーティクルを識別する LTM ロケータ値です。

指定した LTM ロケータの値よりも小さい、現在のバージョン以外のすべてのバージョンのすべてのアーティクルは、RASD からトランケートされます。現在 (最新) のバージョンのアーティクルが LTM ロケータ値で識別されるバージョンよりも古い場合、そのアーティクルはトランケートされません。

RASD での旧バージョンのユーザのトランケート

RASD からユーザをトランケートします。

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. RASD でユーザをトランケートします。

```
ra_truncateusers NNN
```

NNN は、最も古い、現在のバージョン以外の維持されるユーザを識別する LTM ロケータ値です。

指定した LTM ロケータの値よりも小さい、現在以外のすべてのバージョンのすべてのユーザは、RASD からトランケートされます。現在 (最新) バージョンのユーザが LTM ロケータ値で識別されるバージョンよりも古い場合、そのユーザはトランケートされません。

デフォルトのホストとポート番号の設定

埋め込み SQL Anywhere データベースである RASD は、Replication Agent が起動されると起動します。デフォルトでは、SQL Anywhere ホストの値は **localhost** で、SQL Anywhere のポート番号は Replication Agent のポート番号に **1** を加えた値です。

これらの値がホスト環境と競合しているために Replication Agent インスタンスを起動できない場合は、インスタンスの構成ファイルにある Replication Agent 設定パラメータの **asa_host** と **asa_port** を編集してホスト値とポート番号を変更します。

例：

```
$$SYBASE/RAX-15_5/<instance>/<instance>.cfg
```

<instance> は、Replication Agent インスタンスの名前です。

注意： これらの設定を変更した後、Replication Agent を再起動する必要があります。

複写済みトランザクションとプロシージャの特定

Sybase トランザクション複写システムでは、Replication Agent のコンポーネントと Replication Server のコンポーネントの両方に、複写するトランザクションを特定 (選択) できる機能があります。プライマリ・データベースで、すべてのトランザ

クシオンを複製したりすべてのデータを変更する操作を行ったりする必要はありません。

複製対象のトランザクションを選択する機能は、データベース内のすべてのテーブルではなく、一部のテーブルを使用するアプリケーションをサポートする複製システムを実装する必要がある場合に特に役立ちます。

テーブルにマークを付けることによって、トランザクションが複製されているプライマリ・データベースの特定のテーブルを識別します。マーク付けされているテーブルのデータに影響を及ぼすトランザクションは、複製トランザクションと呼ばれています。

注意： マーク付きテーブルとマーク解除テーブルの両方のテーブルのデータに影響があるトランザクションの場合、マーク付きテーブルのデータに影響を与えるオペレーションのみが複製されます。

ストアド・プロシージャにマークを付けることによって、適用ファンクションとして複製される予定のプライマリ・データベース内の特定のプロシージャを特定(選択)します。マーク付けされたプロシージャがプライマリ・データベースで起動されると、そのプロシージャの入力パラメータの値とともにその起動もレプリケート・データベースに複製されます。

複製対象のプロシージャを選択する機能が特に役立つのは、レプリケーション・システムを実装してストアド・プロシージャを使用するアプリケーションをサポートする必要がある場合や、プロシージャを1回呼び出すことによって生成されるさまざまな個別データの変更処理よりも1回のプロシージャ呼び出しを複製することのほうが効果的である場合です。

注意： テーブルまたはストアド・プロシージャに複製のマークを付けるには、Replication Agent トランザクション・ログ・オブジェクトを作成する必要があります。

参照：

- Replication Agent システム・データベース (RASD) の管理 (91 ページ)

テーブルのマーク付けとマーク解除

複製する個々のテーブルにはマークを付ける必要があります。テーブルにマークを付けるには、`pdb_setreptable` コマンドで明示的に行う方法と、`ra_admin init` 処理中に自動的に行う方法 (`pdb_automark_tables` 設定パラメータが `true` に設定されている場合) とがあります。

注意： `pdb_automark_tables` 設定パラメータは、UDB ではサポートされていません。

プライマリ・データベース内のテーブルのデータに影響を及ぼすトランザクションを複製するには、該当するテーブルに複製のマーク付けがされており、そのテーブル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そのテーブルの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、テーブルがマーク付けされると複製が自動的に有効になります。

参照：

- マーク付けされたテーブルの複製の有効化と無効化 (106 ページ)

Replication Agent for Oracle でのテーブルのマーク付け

Replication Agent for Oracle でテーブルに複製のマークが付けられていると、Replication Agent は RASD に接続して該当のテーブルの RASD アーティクルにそのテーブルのマーク状態を記録します。

テーブルにマークが付けられると、そのテーブルのデータに影響を及ぼすその後の処理が複製されます。

Replication Agent for Microsoft SQL Server でのテーブルのマーク付けとマーク解除

Replication Agent for Microsoft SQL Server でテーブルに複製のマークが付けられていると、Replication Agent はプライマリ・データベースにログインしてコマンドを実行し、Microsoft SQL Server トランザクション・ログでの変更内容のロギングをオンにします。

Replication Agent for Microsoft SQL Server でテーブルの複製のマークが外されていると、Replication Agent はプライマリ・データベースにログインしてコマンドを実行し、Microsoft SQL Server トランザクション・ログでの変更内容のロギングをオフにします。

Replication Agent for UDB でのテーブルのマーク付け

Replication Agent for UDB は、テーブルが複製のマーク付けをされるとプライマリ・データベースでタスクを実行します。

Replication Agent：

- プライマリ・データベースにログインし、テーブルの **DATA CAPTURE** オプションの値を **DATA CAPTURE CHANGES** に設定します。
- プライマリ・データベース内にある Replication Agent のマーク付けされているオブジェクトのテーブルにローを追加します。マーク付けされているオブジェクトのテーブル内の各ローには、プライマリ・データベースで複製のマーク付けがされているテーブルの属性が一覧表示されます。

Replication Agent for UDB でのテーブルのマーク解除

Replication Agent for UDB は、テーブルのマークが解除されるとプライマリ・データベースでタスクを実行します。

Replication Agent :

- プライマリ・データベースにログインし、テーブルの **DATA CAPTURE** オプションの値を、テーブルがマーク付けされる前の値にリストアします。
- Replication Agent のマーク付けオブジェクトのテーブルにあるテーブル・ローを削除します。

テーブルのマークを解除すると、そのテーブルのデータに影響を及ぼすその後のオペレーションは無視されます (複製されません)。

プライマリ・データベース・テーブルの複製のマーク付け

Replication Agent インスタンスを使用してテーブルに複製のマーク付けをします。

前提条件

テーブルに複製のマークを付けるには、Replication Agent トランザクション・ログ・オブジェクトを作成する必要があります。

手順

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. テーブルがプライマリ・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_setreptable pdb_table
```

pdb_table は、複製のマークを付ける、プライマリ・データベースにあるテーブルの名前です。

指定されたテーブルに複製のマークが付けられているという情報を pdb_setreptable が返した場合は、この手順を続ける必要はありません。

指定されたテーブルがマーク付けされていないという情報を pdb_setreptable が返した場合は、この手順を続けてテーブルに複製のマークを付けます。

3. データベース複製定義のみでテーブル複製定義が存在しない場合、複製の前にテーブル複製定義は追加されず、次のいずれかになります。
 - a) レプリケート・データベースのテーブルがプライマリ・データベースのテーブルと同じ名前の場合は、以下を使用します。

```
pdb_setreptable pdb_table, mark
```

- b) レプリケート・データベースのテーブルがプライマリ・データベースのテーブルと異なる名前の場合は、以下を使用します。

```
pdb_setreptable pdb_table, rep_table, mark
```

ここで、*rep_table* はレプリケート・データベース内のテーブルの名前です。

4. テーブル複写定義が存在しているかテーブル複写定義が複写前に追加される場合は、データベース複写定義の有無にかかわらず、次のいずれかを実行します。

- a) テーブル複写定義のプライマリ・テーブルがプライマリ・データベース内のテーブルと同じ名前の場合は、以下を使用します。

```
pdb_setreptable pdb_table, mark
```

注意：レプリケート・データベース内のテーブル名がテーブル複写定義のプライマリ・テーブルと同じ名前の場合、プライマリ Replication Server の複写定義で **with all tables named** 句を使用できます。次に例を示します。

```
create replication definition my_table_repdef
with primary at data_server.database
with all tables named pdb_table ...
```

レプリケート・データベース内のテーブルの名前がテーブル複写定義のプライマリ・テーブルの名前と異なる場合、テーブル複写定義をレプリケート・データベース内のテーブルにマップする必要があります。次に例を示します。

```
create replication definition my_table_repdef
with primary at data_server.database
with primary table named pdb_table
with replicate table named rep_table ...
```

- b) テーブル複写定義のプライマリ・テーブルとプライマリ・データベースのテーブルの名前が異なる場合は、次を使用します。

```
pdb_setreptable pdb_table, rdpri_table, mark
```

ここで、*rdpri_table* はテーブル複写定義のプライマリ・テーブルの名前です。テーブル複写定義は、レプリケート・データベースのテーブルにマップする必要があります。

注意：レプリケート・データベース内のテーブル名がテーブル複写定義のプライマリ・テーブルと同じ名前の場合、プライマリ Replication Server の複写定義で **with all tables named** 句を使用できます。次に例を示します。

```
create replication definition my_table_repdef
with primary at data_server.database
with all tables named rdpri_table ...
```

レプリケート・データベース内のテーブルの名前がテーブル複写定義のプライマリ・テーブルの名前と異なる場合、テーブル複写定義をレプリケート・データベース内のテーブルにマップする必要があります。次に例を示します。

```
create replication definition my_table_repdef
with primary at data_server.database
with primary table named rdpri_table
with replicate table named rep_table ...
```

5. テーブルに複製のマークを付ける際に、所有者指定の複製定義と一致したら、オプションで、テーブルの所有者を含めるように指定します。
 - 所有者モードが設定されている場合、Replication Agent の複製定義と一致すると、その所有者の名前が使用されます。
 - 所有者モードが設定されていない場合は、一致する複製定義の名前でも Replication Agent はその所有者の名前を使用しません。

所有者指定の複製定義と一致する場合にテーブル所有者を含めるように指定するには、**owner** キーワードを **mark** キーワードの後に続けて使用します。

```
pdb_setreptable pdb_table, mark, owner
```

pdb_table は、複製のマークを付けるテーブルの名前です。

注意： プライマリ・データベースから返されるテーブル所有者の名前は、そのテーブルの複製定義に指定されている所有者の名前と同一である必要があります。

6. 次のように **pdb_dflt_object_repl** の状態に応じて手順を実行します。
 - **true** (デフォルト) の場合、複製のマーク付けをされているテーブルは **pdb_setreptable** が正常に返された後、直ちに複製の準備が整います。次の手順 (**pdb_setreptable** を使用してマーク付けしたテーブルの複製を有効にする) を省略できます。
 - **false** の場合、次の手順で説明されているとおりにテーブルの複製を有効にする必要があります。
7. テーブルの複製を有効にします。

```
pdb_setreptable pdb_table, enable
```

テーブルの複製を有効にすると、Replication Agent はそのテーブル内のデータに影響を及ぼすトランザクションを複製し始めます。

プライマリ・データベース・テーブルのマーク解除

プライマリ・データベース・テーブルのマークを解除します。

前提条件

UDB に限って、マークを解除する際の Replication Agent は管理状態でなければなりません。

手順

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. テーブルがプライマリ・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_setreptable pdb_table
```

`pdb_table` は、マークを解除する、プライマリ・データベースにあるテーブルの名前です。

指定されたテーブルがマーク付けされているという情報を `pdb_setreptable` が返した場合は、この手順を続けてテーブルのマークを解除します。

指定されたテーブルがマーク付けされていないという情報を `pdb_setreptable` が返した場合は、この手順を続ける必要はありません。

3. テーブルの複写を無効にします。

```
pdb_setreptable pdb_table, disable
```

4. テーブルのマークを解除します。

```
pdb_setreptable pdb_table, 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_setreptable pdb_table, unmark, force
```

5. テーブルに複写のマークが付けられていないことを確認します。

```
pdb_setreptable pdb table
```

注意：プライマリ・データベースでマークを付けられているすべてのオブジェクトのマークは、プライマリ・データベースが UDB でない限り、`pdb_setreptable` コマンドを `all` キーワードを使用して呼び出すと外せます。

DDL 複写の有効化と無効化

DDL 複写を有効にする前に、`ddl_username` および `ddl_password` 設定パラメータを、DDL コマンド実行時に Replication Server がレプリケート・データベースで使用するユーザ名に設定する必要があります。このユーザ名は、Replication Server レプリケート・コネクションで設定されたメンテナンス・ユーザとは異なる名前であればなりません。

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：DDL 複写は Oracle および Microsoft SQL Server でのみ使用可能です。複写されない DDL コマンドの詳細については、『Replication Agent プライマリ・データベース・ガイド』を参照してください。

一時的に DDL の複写をサスペンドするには、`pdb_setrepddl` コマンドを使用して DDL の複写を無効にします。DDL の複写を再開するには、`pdb_setrepddl` コマンド

を使用して複写を有効にします。`pdb_setrepddl` コマンド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。

注意： DDL を複写するには、Replication Server にデータベースレベルの複写定義が必要であり、その定義で `replicate DDL` が設定されていなければなりません。データベースレベルの複写定義の作成の詳細については、『Replication Agent リファレンス・マニュアル』を参照してください。

マーク付けされたテーブルの複写の有効化と無効化

プライマリ・データベースでメンテナンス操作を行う場合など、マーク付けされたテーブルの複写を一時的に停止するために、プライマリ・データベース内のその他のテーブルの複写に影響を及ぼすことなく、マーク付けされたテーブルの複写を無効にすることができます。この場合、該当のテーブルから複写を再開する準備が整えば、データベースのその他のテーブルに影響を及ぼすことなくそのテーブルの複写を有効にできます。

テーブルのデータに影響を及ぼすトランザクションを複写するには、該当するテーブルに複写のマーク付けがされており、そのテーブルの複写が有効になっている必要があります。

Replication Agent for UDB には、プライマリ・データベースでマーク付けされている各テーブルのエントリを含んでいるマーク付けされたオブジェクトのテーブルがあります。マーク付けされている各テーブル・ローは、マーク付けされているテーブルで複写が有効または無効になっていることを示すフラグを含んでいます。Replication Agent for Oracle と Replication Agent for Microsoft SQL Server は、RASD にアーティクルがあります。アーティクルとはテーブルと 1 対 1 の関係にあるオブジェクトで、マーク付けされたインジケータを有します。

マークが付けられているオブジェクトで複写が無効になっている場合、マーキングのインフラストラクチャは有効なままですが、該当のオブジェクトのトランザクションは Replication Server に送信されません。

参照：

- テーブルのマーク付けとマーク解除 (100 ページ)

マーク付けされたテーブルの複写の有効化

マーク付けされたテーブル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. テーブルがプライマリ・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_setreptable pdb_table
```

`pdb_table` は、複写を有効にする、プライマリ・データベースにあるテーブルの名前です。

指定されたテーブルがマーク付けされ、複写が無効化されているという情報を `pdb_setreptable` が返した場合、この手順を続けてテーブルの複写を有効にします。

注意： テーブルの複写を有効または無効にする前に、テーブルを複写するようにマーク付けしてください。

3. テーブルの複写を有効にします。

```
pdb_setreptable pdb_table, enable
```

テーブルの複写を有効にすると、そのテーブル内のデータに影響を及ぼすトランザクションが複写用に取り出されます。

4. ここで、テーブルで複写が有効になっていることを確認します。

```
pdb_setreptable pdb_table
```

マーク付けされたテーブルの複写の無効化

マーク付けされたテーブル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. テーブルがプライマリ・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_setreptable pdb_table
```

`pdb_table` は、複写を無効にする、プライマリ・データベースにあるテーブルの名前です。

指定されたテーブルがマーク付けされ、その複写が有効化されているという情報を `pdb_setreptable` が返した場合、この手順を続けてテーブルの複写を無効にします。

注意： テーブルの複写を有効または無効にする前に、テーブルを複写するようにマーク付けしてください。

3. テーブルの複写を無効にします。

```
pdb_setreptable pdb_table, disable
```

テーブルの複写を無効にすると、複写を再度有効にするまで、そのテーブル内のデータに影響を及ぼすトランザクションが複写用に取り出されることはありません。

4. ここで、テーブルで複写が無効になっていることを確認します。

```
pdb_setreptable pdb_table
```

LOB カラムの複写の有効化と無効化

このマニュアルでは、プライマリ・データベース・ベンダが使用する実際のデータ型名に関係なく、ラージオブジェクト (LOB) データ型を含んでいるすべてのカラムを LOB カラムと呼びます。LOB カラムに影響を及ぼすトランザクションを複写するには、該当のカラムの複写を有効にする必要があります。

LOB カラムを含んでいるテーブルに複写のマーク付けをして複写を有効にするだけでなく、LOB カラムそれぞれの複写を有効にする必要があります。

- **pdb_dflt_column_repl** パラメータの値が true である場合、テーブルがマーク付けされるとそのテーブル内のすべての LOB カラムの複写が自動的に有効になります。
- **pdb_dflt_column_repl** パラメータの値が false であると、テーブルがマーク付けされたときにテーブル内の LOB カラムの複写が自動的に有効になることはありません。

テーブルが複写のマーク付けをされておりテーブルでの複写が有効になっていても、そのテーブルの LOB カラムではマーク付けも有効化も行われていないと、LOB カラムに影響を及ぼすトランザクションの一部は複写されません。テーブルが複写のマーク付けをされておりテーブルの複写が有効になっている場合、LOB カラム以外のすべてのカラムに影響を及ぼすトランザクション部分が複写されません。

参照：

- テーブルのマーク付けとマーク解除 (100 ページ)

マーク付けされたテーブルの LOB カラムの複写の有効化

LOB カラム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. LOB カラムの複製が有効になっているかどうかを確認します。

```
pdb_setrepcol pdb_table, pdb_col
```

- **pdb_table** は、LOB カラムを含んでいるマーク付けされたテーブルです。
- **pdb_col** は、複写を有効にする LOB カラムです。

LOB カラムの複写が無効化されているという情報を **pdb_setrepcol** が返した場合は、この手順を続けてカラムの複写を有効にします。

注意： LOB カラムの複写を有効または無効にする前に、LOB カラムを含んでいるテーブルに複写のマークを付けてください。

3. LOB カラムの複写を有効にします。

```
pdb_setrepcol pdb_table, pdb_col, enable
```

LOB カラムで複写が有効になっていると (さらに、LOB カラムを含んでいるテーブルで複写が有効になっている場合)、LOB カラムのデータに影響を及ぼすトランザクションが複写されます。

4. ここで、LOB カラムで複写が有効になっていることを確認します。

```
pdb_setrepcol pdb_table, pdb_col
```

マーク付けされたテーブルの LOB カラムの複写の無効化

LOB カラム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. LOB カラムの複製が有効になっているかどうかを確認します。

```
pdb_setrepcol pdb_table, pdb_col
```

構文の説明は次のとおりです。

- pdb_table は、LOB カラムを含んでいるマーク付けされたテーブルです。
- pdb_col は、複写を無効にする LOB カラムです。

LOB カラムの複写が有効化されているという情報を pdb_setrepcol が返した場合は、この手順を続けてカラムの複写を無効にします。

注意： LOB カラムの複写を有効または無効にする前に、LOB カラムを含んでいるテーブルに複写のマークを付けてください。

3. LOB カラムの複写を無効にします。

```
pdb_setrepcol pdb_table, pdb_col, disable
```

LOB カラムの複写を無効にすると、LOB カラムの複写を再度有効にしない限り、LOB カラムのデータに影響を及ぼすトランザクションは複写されません。

4. ここで、LOB カラムで複写が無効になっていることを確認します。

```
pdb_setrepcol pdb_table, pdb_col
```

ストアド・プロシージャのマーク付けとマーク解除

Replication Agent は、プライマリ・データベースでのストアド・プロシージャの起動を複写することによって、Replication Server のファンクション複写をサポートしています。

Replication Agent は、適用ファンクションと要求ファンクションの両方を複写できます。

- 適用ファンクションはプライマリ・データベースで実行されるストアド・プロシージャであり、プライマリ・データベースのデータに影響を及ぼすトランザクションを生成します。
- 要求ファンクションは特定のデータベース (レプリケート・データベースなど) で起動されるストアド・プロシージャで、別のデータベース (プライマリ・データベースなど) で実行されます。

注意： UDB ではストアド・プロシージャの複写はサポートされていません。

要求ファンクションで使用される特定のユーザとパスワードを提供する以外に、Replication Agent がこれらの2つのファンクションの種類を区別することはありません。要求ファンクションを使用する場合は、**function_username** および **function_password** 設定パラメータを指定する必要があります。

適用および要求ファンクションの詳細については、『Replication Server 管理ガイド』の「複写ファンクションの管理」を参照してください。

プライマリ・データベースで呼び出されるストアド・プロシージャを複写するには、そのストアド・プロシージャに複写のマークが付けられ、そのストアド・プロシージャの複写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写を有効にすることは類似しています)。

注意： ストアド・プロシージャに複写のマークを付けることと、そのストアド・プロシージャで複写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、ストアド・プロシージャにマークが付けられると複写が自動的に有効になります。

マーク付けされたストアド・プロシージャがマーク付けされたテーブルに影響を及ぼす処理を実行する場合、そのマーク付けされたテーブルに影響を及ぼす処理は複写のために取り出されません。マーク付けされたストアド・プロシージャの起動のみが複写されます。

ストアド・プロシージャに複写のマークを付けると、Replication Agent はそのストアド・プロシージャのシャドーロー・プロシージャを作成します。また Replication Agent for Oracle および Replication Agent for UDB は、新規のプロシージャを挿入することによってマークを付けられているストアド・プロシージャを変更します。

- 関連するシャドーロー・プロシージャを実行するための最初の手順
- パラメータが異なるシャドーロー・プロシージャを再実行するための最後の手順

マークが付けられているストアド・プロシージャの複写を一時的にサスペンドするには (プライマリ・データベースでデータベースのメンテナンス処理が実行される場合など)、そのストアド・プロシージャの複写を無効にすることができます。

複製のマークが付けられているオブジェクトのマークを外す、そのオブジェクトの複製を容易にするために作成されたトランザクション・ログ・オブジェクトがプライマリ・データベースから削除されます。

Replication Server のファンクション複製機能の詳細については、『Replication Server 管理ガイド』を参照してください。

参照：

- ストアド・プロシージャの複製の有効化と無効化 (114 ページ)

ストアド・プロシージャの複製のマーク付け

ストアド・プロシージャに複製のマークを付けます。

前提条件

ストアド・プロシージャに複製のマーク付けをするには、Replication Agent トランザクション・ログ・オブジェクトを作成する必要があります。

Oracle の場合、ストアド・プロシージャのマーク付けを実行する際に、DDL 複製を無効にする必要があります。ストアド・プロシージャにマークを付けるとそのストアド・プロシージャが変更されるため、マーク付けによる変更がレプリケート・サイトに複製されないように最初に DDL 複製を無効にする必要があります。

手順

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. ストアド・プロシージャの複製が有効になっているかどうかを確認します。

```
pdb_setrepproc pdb_proc
```

pdb_proc は、複製のマークを付ける、ストアド・プロシージャの名前です。

- 指定されたストアド・プロシージャがマーク付けされたという情報を pdb_setrepproc が返した場合、この手順を続ける必要はありません。
 - 指定されたストアド・プロシージャがマーク付けされていないという情報を pdb_setrepproc が返した場合は、この手順を続けてこのストアド・プロシージャに複製のマークを付けます。
3. データベース複製定義のみでファンクション複製定義が存在しない場合、複製の前にファンクション複製定義は追加されず、次のいずれかになります。
 - レプリケート・データベースのプロシージャがプライマリ・データベースのプロシージャと同じ名前の場合は、次のコマンドを使用します。

```
pdb_setrepproc pdb_proc, mark
```
 - レプリケート・データベースのプロシージャがプライマリ・データベースのプロシージャと異なる名前の場合は、次のコマンドを使用します。

```
pdb_setrepproc pdb_proc, rep_proc, mark
```

rep_proc は、レプリケート・データベース内のプロシージャの名前です。

4. ファンクション複写定義が存在しているか複写定義が複写前に追加される場合は、データベース複写定義の有無にかかわらず、次のいずれかを実行します。

- ファンクション複写定義がプライマリ・データベースのプロシージャと同じ名前の場合は、次のコマンドを使用します。

```
pdb_setrepproc pdb_proc, mark
```

- レプリケート・データベースのストアド・プロシージャがファンクション複写定義と同じ名前である場合、**deliver as** 句を使用する必要はありません。次に例を示します。

```
create function replication definition pdb_proc
with primary at data_server.database ...
```

- レプリケート・データベースのプロシージャの名前がファンクション複写定義の名前と異なる場合、ファンクション複写定義をレプリケート・データベースのプロシージャにマップする必要があります。次に例を示します。

```
create function replication definition pdb_proc
with primary at data_server.database
deliver as 'rep_proc' ...
```

- ファンクション複写定義の名前が、プライマリ・データベースのプロシージャと異なる場合は、次のコマンドを使用します。

```
pdb_setrepproc pdb_proc, rdpri_proc, mark
```

rdpri_proc は、ファンクション複写定義の名前です。ファンクション複写定義は、レプリケート・データベースのプロシージャにマップする必要があります。

- レプリケート・データベースのストアド・プロシージャがファンクション複写定義と同じ名前である場合、**deliver as** 句を使用する必要はありません。次に例を示します。

```
create function replication definition rdpri_proc
with primary at data_server.database ...
```

- レプリケート・データベースのプロシージャの名前がファンクション複写定義の名前と異なる場合、ファンクション複写定義をレプリケート・データベースのプロシージャにマップする必要があります。次に例を示します。

```
create function replication definition rdpri_proc
with primary at data_server.database
deliver as 'rep_proc' ...
```

5. マーク付けされたストアド・プロシージャの複写を有効にします。

```
pdb_setrepproc pdb_proc, enable
```

ストアド・プロシージャで複写が有効になると、プライマリ・データベースにストアド・プロシージャの呼び出しの複写を開始できます。

注意： ストアド・プロシージャが Oracle にあり、ストアド・プロシージャにマークを付けるときに DDL 複写を無効にした場合は、DDL 複写を再び有効にしてください。ストアド・プロシージャにマークを付けるとそのストアド・プロシージャが変更されるため、マーク付けによる変更がスタンバイ・サイトに複写されないように最初に DDL 複写を無効にする必要があります。

参照：

- DDL 複写の有効化と無効化 (105 ページ)

ストアド・プロシージャのマーク解除

ストアド・プロシージャのマークを解除すると、Replication Agent は、そのストアド・プロシージャにマークが付けられたときに作成されたトランザクション・ログ・オブジェクトを削除します。Oracle の場合、ストアド・プロシージャのマーク解除を実行する際に、DDL 複写を無効にする必要があります。

ストアド・プロシージャのマークを解除するには、次の手順を実行します。

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. ストアド・プロシージャの複写が有効になっているかどうかを確認します。

```
pdb_setrepproc pdb_proc
```

pdb_proc は、複写のマークを解除する、ストアド・プロシージャの名前です。

- 指定されたストアド・プロシージャがマーク付けされているという情報を pdb_setrepproc が返した場合は、この手順を続けてストアド・プロシージャの複写のマークを解除します。
- 指定されたストアド・プロシージャがマーク付けされていないという情報を pdb_setrepproc が返した場合、この手順を続ける必要はありません。

3. ストアド・プロシージャの複写を無効にします。

```
pdb_setrepproc pdb_proc, disable
```

4. ストアド・プロシージャのマークを解除します。

```
pdb_setrepproc pdb_proc, 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_setrepproc pdb_proc, unmark, force
```

5. ストアド・プロシージャに複写のマークが付けられていないことを確認します。

```
pdb_setrepproc pdb_proc
```

pdb_setrepproc を all キーワードを使って呼び出すことによって、プライマリ・データベースでマーク付けされているすべてのストアド・プロシージャのマークを解除できます。

注意： ストアド・プロシージャが Oracle にあり、ストアド・プロシージャにマークを付けるときは DDL 複写を無効にした場合は、DDL 複写を再び有効にしてください。ストアド・プロシージャにマークを付けるとそのストアド・プロシージャが変更されるため、マーク付けによる変更がスタンバイ・サイトに複写されないように最初に DDL 複写を無効にする必要があります。

参照：

- DDL 複写の有効化と無効化 (105 ページ)

ストアド・プロシージャの複写の有効化と無効化

ストアド・プロシージャの複写を一時的にサスペンドするには、`pdb_setrepproc` コマンドを使用してマーク付けされているストアド・プロシージャの複写を無効にします。マーク付けされているストアド・プロシージャの複写を再開する準備が整ったら、`pdb_setrepproc` コマンドを使用して複写を有効にします。

注意： UDB では、プロシージャの複写はサポートされていません。

プライマリ・データベースでのストアド・プロシージャの呼び出しを複写するには、ストアド・プロシージャに複写のマーク付けをし、そのストアド・プロシージャの複写を有効にする必要があります。デフォルトでは、どのプロシージャにも複写のマーク付けはされていません。

ストアド・プロシージャに複写のマークを付けることと、そのストアド・プロシージャで複写を有効にすることは別のことです。

参照：

- ストアド・プロシージャのマーク付けとマーク解除 (109 ページ)

マーク付けされたストアド・プロシージャの複写の有効化

ストアド・プロシージャ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. ストアド・プロシージャの複写が有効になっているかどうかを確認します。

```
pdb_setrepproc pdb_proc
```

`pdb_proc` は、複写を有効にするストアド・プロシージャの名前です。

指定されたストアド・プロシージャがマーク付けされ、その複写が無効になっているという情報を `pdb_setrepproc` が返した場合、この手順を続けてそのストアド・プロシージャの複写を有効にします。

注意： ストアド・プロシージャの複写を有効または無効にする前に、ストアド・プロシージャに複写のマークを付けてください。

3. マーク付けされたストアド・プロシージャの複写を有効にします。

```
pdb_setrepproc pdb_proc, enable
```

ストアド・プロシージャの複写を有効にすると、そのストアド・プロシージャの呼び出しが複写されます。

4. ここで、ストアド・プロシージャで複写が有効になっていることを確認します。

```
pdb_setrepproc pdb_proc
```

マーク付けされたストアド・プロシージャの複写の無効化

ストアド・プロシージャ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. ストアド・プロシージャの複写が有効になっているかどうかを確認します。

```
pdb_setrepproc pdb_proc
```

pdb_proc は、複写を無効にするストアド・プロシージャの名前です。

指定されたストアド・プロシージャがマーク付けされ、その複写が有効になっているという情報を pdb_setrepproc が返した場合、この手順を続けてそのストアド・プロシージャの複写を無効にします。

注意： ストアド・プロシージャの複写を有効または無効にする前に、ストアド・プロシージャに複写のマークを付けてください。

3. ストアド・プロシージャの複写を無効にします。

```
pdb_setrepproc pdb_proc, disable
```

ストアド・プロシージャの複写を無効にすると、複写を再度有効にするまで、そのストアド・プロシージャの呼び出しが複写用に取り出されることはありません。

4. ここで、ストアド・プロシージャで複写が無効になっていることを確認します。

```
pdb_setrepproc pdb_proc
```

Oracle シーケンスのマーク付けとマーク解除

Replication Agent は、プライマリ・データベースでのシーケンスの複写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、そのすべて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 (テーブルにマークを付けることとテーブルの複写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す。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、スタンバイ・サイトにはプライマリ・サイトで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。

注意：シーケンスの複製は Oracle でのみサポートされています。

マーク付けされたシーケンスの複製を一時的にサスペンドするには、そのシーケンスの複製を無効にします。

参照：

- シーケンスの複製の有効化と無効化 (118 ページ)
- シーケンスのマーク解除 (117 ページ)

シーケンスの複製のマーク付け

Oracle シーケンスに複製のマークを付けます。

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. シーケンスの複製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` は、複製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がマーク付けされているという情報を `pdb_setrepseq` が返した場合は、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がマーク付けされていないという情報を `pdb_setrepseq` が返した場合は、この手順を続けてシーケンスに複製のマークを付けます。

3. シーケンスに複製のマークを付けます。。

`pdb_setrepseq` を使用すると、複製されるプライマリ・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・サイトで追加するシーケンスの名前が異なる場合にシーケンスに複製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, mark
```

注意： 指定した名前と異なる名前のシーケンスを複製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製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 は、レプリケート・データベース内のシーケンスの名前です。

注意： レプリケート・データベース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製する場合、そのレプリケート・データベース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。

次に、**pdb_dflt_object_repl** の値に応じた手順を示します。

- true** (デフォルト) – **pdb_setrepseq** で複製のマークを付けられているシーケンスの複製準備を整えるには、**pdb_setrepseq** を正常に呼び出します。この手順の手順 4 を省略できます。
- false** – 複製できるようにする前に、シーケンスに対して複製を有効にする必要があります。

4. シーケンスの複製を有効にします。

```
pdb_setrepseq pdb_seq, enable
```

シーケンスで複製が有効になると、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製を開始できます。

シーケンスのマーク解除

Oracle シーケンスのマークを解除します。

- 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。
- シーケンスの複製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 は、マーク解除するシーケンスの名前です。

- 指定されたシーケンスがマーク付けされているという情報を **pdb_setrepseq** が返した場合、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がマーク付けされていないという情報を **pdb_setrepseq** が返した場合、この手順を続ける必要はありません。

3. シーケンスの複製を無効にします。

```
pdb_setrepseq pdb_seq, disable
```

4. シーケンスのマークを解除します。

```
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. シーケンスに複写の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

マーク付けされているシーケンスの複写を一時的にサスペンドするには、**pdb_setrepseq** コマンドを使用します。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：

- シーケンスの複写のマーク付け (116 ページ)

マーク付けされたシーケンスの複写の有効化

Oracle 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` は、複写を有効にするシーケンスの名前です。

指定されたシーケンスがマーク付けされ、その複写が無効化されているという情報を `pdb_setrepseq` が返した場合、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. ここで、シーケンスで複写が有効になっていることを確認します。

```
pdb_setrepseq pdb_seq
```

マーク付けされたシーケンスの複写の無効化

Oracle 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がマーク付けされ、その複写が有効にされているという情報を pdb_setrepseq が返した場合、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
```

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

4. ここで、シーケンスで複写が無効になっていることを確認します。

```
pdb_setrepseq pdb_seq
```

Replication Agent のチューニング

Replication Agent 設定パラメータの一部を調整することによって、Replication Agent のパフォーマンスをチューニングまたは最適化します。

ra_config コマンドを使用すると、Replication Agent 設定パラメータの設定または変更を実行できます。

Replication Agent は、**ra_config** または **ra_set_login** が呼び出されると必ず設定ファイル全体を上書きするため、設定ファイルの編集は行わないでください。また、各 Replication Agent インスタンスは起動時にのみ設定ファイルを読み込みます。インスタンスが停止されて再起動される前に新しい設定パラメータの値を反映させるには、**ra_config** コマンドを使用する必要があります。

Replication Agent のインスタンスが管理状態であれば、すべての Replication Agent 設定パラメータを変更できますが、Replication Agent のインスタンスがレプリケーション状態のときには変更できない設定パラメータもあります。

ra_config コマンドおよび Replication Agent の設定パラメータの詳細については、『Replication Agent リファレンス・マニュアル』を参照してください。

カスタマイズにチューニング

通常、Replication Agent のデフォルトの設定値は最適なパフォーマンスを提供します。

ただし、特定の環境に適合させたり最適化したりするために設定内容を変更しなければならない状況が発生する場合があります。

Replication Agent システム・ログのサイズと容量の調整

デフォルトでは、Replication Agent が生成するシステム・ログのサイズは固定されています。ディスクが消費され続けないように、システム・ログがロールオーバーすることがあります。

ログ・サイズを大きくしてバックアップ・ファイルの数を調整し、ログ・データを長期間保存するようにすることができます。これらのパラメータ値を小さくすると、お使いの環境の未使用スペースを増やせます。

1. Replication Agent 管理ポートにログインし、Replication Agent インスタンスが Admin 状態になっていることを確認します。

```
ra_status
```

2. プライマリ・データベースの Replication Agent 設定パラメータの値を設定します。バックアップ・ファイルのサイズと数を増やすには、これらの値を大きくします。お使いの環境で利用できるスペースを増やすには、これらの値を小さくします。

```
ra_config log_backup_files, n
```

```
ra_config log_wrap, m
```

スピンの防止

Replication Agent では、設定パラメータの **scan_sleep_increment** と **scan_sleep_max** を使用して、ログの末尾に到達したときにスキャンを一時的に停止します。これによって、Replication Agent がログの末尾で継続して「スピン」しないようにします。

トランザクションはスリープ状態にあるために、デフォルトでは Replication Agent が最長で 60 秒間一時的に停止してから新しいトランザクションが表示されます。トランザクションの最長可能待機時間をデフォルトの 60 秒間よりも短くする必要がある場合は、スキャン・パラメータの値を小さくします。その結果、ログが末尾に達すると CPU 使用率が増加します。

反対に、待機時間よりも CPU 使用率の最大化がより重大な懸案事項である場合は、これらのパラメータ値を増やし、次回増加するトランザクションに待機時間を設ける代わりに、非アクティブなログでの Replication Agent の CPU 使用率を少なくすることができます。

注意： これらのパラメータは、ログが末尾に到達して複写する追加のアクティビティが存在しない場合にのみ効果があります。デフォルトでは、ログが末尾に到達していなくても、Replication Agent は (一時停止せずに) 直ちに再スキャンします。

パラメータと変数

次に、パフォーマンスとチューニングに影響を与えるパラメータを一覧表示します。

インスタンス作成時のパラメータとデフォルト

大部分の環境下で、Replication Agent は、これらのデフォルトの設定プロパティ値を使用してインスタンスを作成するときに最適なパフォーマンスを発揮します。

パラメータ	デフォルト
<code>column_compression</code>	true
<code>compress_lti_syntax</code>	true
<code>lti_batch_mode</code>	true
<code>lti_formatter_count</code>	3
<code>lti_max_buffer_size</code>	5000
<code>lti_update_trunc_point</code>	10000
<code>ltl_batch_size</code>	40000
<code>ltl_origin_time_required</code>	false
<code>ltl_send_only_primary_keys</code>	true
<code>rs_packet_size</code>	2048
<code>structured_tokens</code>	true
<code>use_rssd</code>	true

パフォーマンスのスループットに影響を及ぼすパラメータ

次の設定プロパティはパフォーマンスのスループットに影響を及ぼす可能性があり、ハードウェア・リソース、プライマリ・データベース・スキーマ、トランザクション・プロファイルなどの環境要因に基づいたチューニングが必要になることがあります。

パラメータ	デフォルト
log_read_block_count	64
lr_max_op_queue_size	1000
lr_max_scan_queue_size	1000
lri_max_buffer_size	5000
rasd_object_cache_size	1000

注意： log_read_block_count パラメータは、Microsoft SQL Server データベースでのみ使用されます。

複写環境、システム・リソースに加えて、データベース・スキーマやトランザクション・プロファイルに応じて、最大メモリ設定もパフォーマンスのスループットに影響を及ぼす可能性があります。Replication Agent の最大メモリ設定は、RA_JAVA_MAX_MEM 環境変数が設定されている場合、この変数によって決まります。

パフォーマンスの遅延時間に影響を及ぼすパラメータ

次の設定パラメータはパフォーマンスの遅延時間に影響を及ぼします。しかし、これらのパラメータのデフォルト値はパフォーマンスのスループットに対して優先順位が付けられており、パフォーマンスの遅延時間を短縮するためにこれらのパラメータを調整すると、パフォーマンスのスループットに悪影響を及ぼすことがあります。

パラメータ	デフォルト
dump_batch_timeout	5
scan_sleep_increment	5
scan_sleep_max	60

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

参照：

- デフォルトのヒープ・サイズ (135 ページ)

最適なパフォーマンスのチューニング

Replication Agent を最適なパフォーマンスにチューニングするには、複写環境と Replication Agent の統計に関する知識が必要です。それぞれの環境で最適なパフォーマンスを実現するには、試行錯誤しなければならない可能性があります。

メモリ使用

最初に考えなければならないことは、Replication Agent で使用できる最大メモリ量です。設定、スキーマ、およびトランザクション・プロファイルに割り当てられるメモリが少ないと、Java Virtual Machine (JVM) ガーベジ・コレクションのパフォーマンスが大幅に低下する可能性があります。メモリの使用に関する情報は、5つの Replication Agent 統計値から得られます。特に、時間をかけてメモリの最大使用量のパーセンテージをモニタすることによって、メモリ容量が設定に十分であるかどうかを確認できます。

キュー (またはバッファ) およびキャッシュ・サイズのパラメータは、メモリ使用量を決定する主要な設定パラメータです。**ra_statistics** コマンドを使用してキューとキャッシュ・サイズを使用されているメモリのパーセンテージとともにモニタすると、設定を変更して使用可能なメモリ量を調整するべきかどうか、反対に、使用可能なメモリ量を変更して設定を調整するべきかどうかを判断するのに役立つ可能性があります。最大メモリ使用のパーセンテージは、キューおよびキャッシュ・サイズを小さくすることで抑える必要があります。結果として、ガーベジ・コレクションが低下してパフォーマンスが向上します。一方、使用可能な最大メモリを増加させてもパフォーマンスが向上します。一般的に、メモリ使用量が最大値の 50% 以下に低下しない、ガーベジ・コレクションが Replication Agent のパフォーマンスを妨げる可能性があります。

スレッド

Replication Agent はマルチスレッド・アプリケーションであり、複数のスレッドは処理時にアクティブになっています。通常は、ログ・レコードが読み込まれ、順番に処理されます。つまり、たいていの場合、各スレッドは別のスレッドに依存しており、1つのスレッドのボトルネックによってシステム全体の複写パフォーマンスが低下する可能性があるということです。

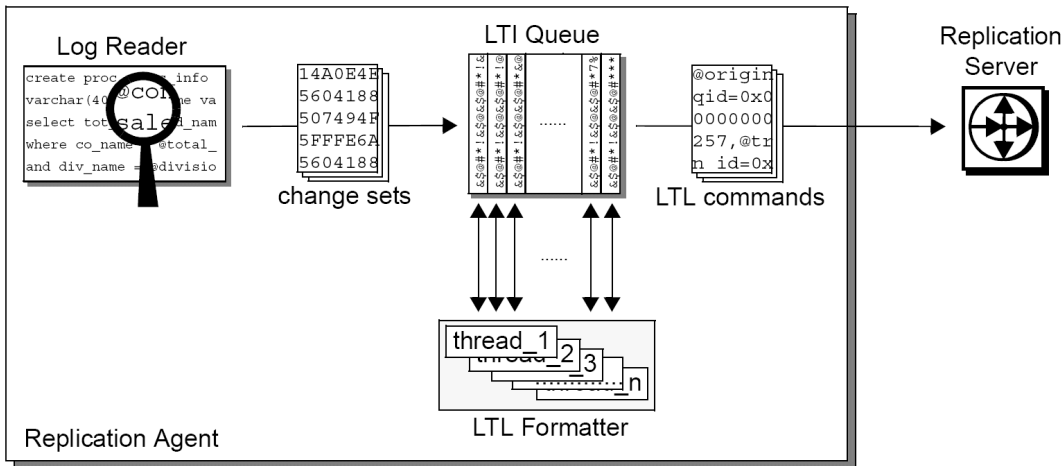
並列 LTL フォーマット

Replication Agent では LTL コマンドを並列でフォーマットでき、マルチプロセッサ・マシンのパフォーマンスを向上させます。LTL フォーマッタは、LTL コマンドに変更セットを変換する Replication Agent の内部コンポーネントです。

Replication Agent の 15.6 より前のバージョンでは、LTL フォーマッタは、Log Transfer Interface (LTI) キューの項目を逐次的にしか処理できなかったため、パフォーマンス上のボトルネックになっていました。今バージョンの Replication Agent では、LTI キューの項目を並列で処理するマルチスレッド LTL フォーマッタ

が使用可能になりました。この結果、マルチプロセッサを実行するマシンのパフォーマンスが向上します。

図 4 : Replication Agent の LTL フォーマット



Replication Agent では、LTI キューのサイズと LTL フォーマッタ・スレッド数を設定するために、`lti_max_buffer_size` と `lti_formatter_count` の 2 つの設定パラメータを使用します。また、Replication Agent には、パフォーマンスとチューニングに役に立つ LTI 統計値が用意されています。

統計とキューのボトルネック

長期間にわたって分析される Replication Agent 統計値は、Replication Agent のボトルネックを特定します。キュー (またはバッファ) サイズの統計値によって、Replication Agent のボトルネックが発生する場所が示されることがあります。たとえば、Log Transfer Interface (LTI) のインバウンド・キューが限度容量に達しているかそれに近い状態であるのに LTI アウトバウンド・キューの容量がゼロかそれに近い状態である場合、LTL がフォーマットされる LTI インバウンド・キューが複製パフォーマンスを低下させるボトルネックである可能性があります。このように、キューのサイズで特定領域の問題を特定することができます。

- 限度容量に達しているかそれに近い LTI アウトバウンド・キュー・サイズは、Replication Agent が Replication Server で待機していることを示します。この場合、Replication Agent のすべてのキューも限度容量に達しているかそれに近い状態です。
- LTI インバウンド・キュー・サイズが限度容量に達しているかそれに近い状態で、LTI アウトバウンド・キュー・サイズがゼロかそれに近い状態である場合は、Replication Agent の LTL のフォーマットにボトルネックがあることを示します。これは、プライマリ・データベースの文字セットと Replication Server の

文字セットが異なり、Replication Agent が文字セット間で文字データを変換しなければならないことが原因になっている可能性があります。あるいは、RSSD が全く使用されていない (`use_rssd` パラメータが `false` に設定されている) か、`column_compression` および `lti_send_only_primary` キー・パラメータによって決まる最小のカラム機能が使用されていないために、Replication Server RSSD が効率的に使用されていないことがこのボトルネックの原因になっている場合があります。

- Log Reader のオペレーション・キュー・サイズが限度容量に達しているかそれに近い状態で、LTI キュー・サイズがゼロかそれに近い状態である場合は、Replication Agent のオペレーション処理にボトルネックがあることを示します。これは、RASD アクセスまたはキャッシュ設定に関する問題が原因である可能性があります。
- キュー・サイズが限度容量に達しているかそれに近い状態である場合、Log Reader スキャン・キューは Replication Agent の処理で使用される最初のキューであるため、ログ・スキャンのボトルネックを示している可能性があります。このボトルネックは、読み取りブロック・カウントが低すぎるか、Replication Agent がローカルではなくネットワークを介してログを読み取る必要があるために、入出力が効率的に行われていないことが原因である可能性があります。

文字セットの変換

複製環境における文字セットの違いも複製パフォーマンスに影響を及ぼす可能性があります。たとえば、プライマリ・データベースの文字セットと Replication Server の文字セットが異なる場合、Replication Agent は複製されるすべての文字データに対して文字セット変換を実行します。Replication Agent が文字セット変換を実行する必要がある場合は、複製開始時に Replication Agent のシステム・ログに警告メッセージが記録されます。

参照：

- 使用可能なメモリ (134 ページ)

エンドツーエンドのチューニング

複製システム全体で最良の Replication Server パフォーマンスとエンドツーエンド・パフォーマンスを実現するには、最適である Replication Agent のパフォーマンスを低下させることが有益である場合もあります。

たとえば、High Volume Adaptive Replication (HVAR) を使用する Oracle と ASE 間の複製システムでは、オートコレクション用の同一の Replication Agent 設定を使用することで、最良のエンドツーエンド複製パフォーマンスが実現される可能性があります。テーブルでオートコレクションの複製が設定されると、そのテーブルでは `column_compression` パラメータと `lti_send_only_primary_keys` パラメータが無視され、プライマリ Oracle データベースがすべてのカラムを記録するように設定さ

れます。これによって Replication Agent のパフォーマンスは低下しますが、複製システム全体のパフォーマンスは向上します。

HVAR と Oracle

UDB または Microsoft SQL Server の場合、HVAR を利用するためにプライマリ・データベースを再設定する必要はありません。Oracle はデフォルトで更新オペレーションに対して最小カラムのみを記録するため、複製のマークが付けられているテーブル内のすべてのカラムでサブリメンタル・ロギングを有効にする必要があります。

Replication Agent for Oracle からのサブリメンタル・ロギングを有効にするには、**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』で説明)。

注意： 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。オートコレクション機能の制限の詳細については、『Replication Agent プライマリ・データベース・ガイド』を参照してください。

トラブルシューティング

Replication Agent および複製システムの基本的なトラブルシューティング手順を確認します。

複製システムでは、コマンドと複製の2種類のエラーが発生します。コマンド・エラーは、複製システムを設定するときに発生します。問題のトラブルシューティングに役立つ特定のエラー・メッセージが表示されます。複製エラーは、複製システムの設定後、複製トランザクションがレプリケート・データベースに表示されない場合に発生します。

複製を妨げる問題が発生しても、複製システム・コンポーネントからエラー・メッセージが返されない場合があります。たとえば、コンポーネントでは、複製の開始を妨げる問題が設定にあることを認識できない場合があります。

機能している Replication Agent システム、すなわち、事前に複製されたトランザクションが正常に処理されているシステムでは、発生するほとんどのシステムの問題で、1つ以上のシステム・コンポーネントからエラー・メッセージが返されます。ただし、複製を妨げる問題の中には、システム・コンポーネントによってエラーとは解釈されないものもあります。この場合、複製は失敗しますが、エラー・メッセージは返されません。

コマンド・エラー

コマンドによって返されたエラー・メッセージまたはログ・ファイルに表示されたエラー・メッセージをトラブルシューティングします。

接続拒否

エラー・メッセージ: <jdbc:sybase:Tds:localhost:5001/emb> に接続できませんでした: JZ006: IOException が検出されました:

java.net.ConnectException: 接続が拒否されました: connectJZ006:

説明: Replication Agent は、*localhost* という名前のホストとポート 5001 を使用して Sybase サーバに接続しようとしたのですが、サーバが見つからないことを示すエラーが発生しました。

対処法: 通常、これは設定エラーです。Replication Agent が接続しようとしているサーバが実行されていないか、Replication Agent で設定されたホストおよびポートの情報が正しくありません。

- サーバが実行中であることを確認します。

- Replication Agent で設定されたホストおよびポートの情報が正しいことを確認します。
- 上記の項目を確認した後、コネクションをテストします。

参照：

- コネクション設定パラメータの設定 (55 ページ)
- ネットワーク接続テスト (60 ページ)

Replication Agent のエラー

Replication Agent のエラーをトラブルシューティングします。

プライマリ・データベース・オブジェクトに、複写のマークがついているかの確認

Sybase トランザクション複写システムでは、Replication Agent および Replication Server の両方で、複写するオブジェクトを選択できます。プライマリ・データベースですべてのオブジェクトまたはすべてのデータ変更オペレーションを複写する必要はありません。

プライマリ・データベース・オブジェクト (テーブルまたはストアド・プロシージャ) が複写されない場合、複写するオブジェクトがマーク付けされていることを確認します。

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. オブジェクトが既にマーク付けされているかを調べます。

- テーブルの場合

```
pdb_setreptable pdb_table
```

ここで `pdb_table` は、複写のマークが付けられているかどうかを確認するプライマリ・データベースにあるテーブルの名前です。

- LOB カラムの場合

```
pdb_setrepcol pdb_table, pdb_col
```

構文の説明は次のとおりです。

- `pdb_table` は、LOB カラムを含んでいるマーク付けされたテーブルです。
- `pdb_col` は、LOB カラムです。
- ストアド・プロシージャの場合

```
pdb_setrepproc pdb_proc
```


`pdb_proc` は、複製のマークが付けられているかどうかを確認するプライマリ・データベース内のストアド・プロシージャの名前です。

- DDL の場合

```
pdb_setrepddl object
```

オブジェクトは、テーブル、プロシージャ、または `sequence` 名です。

`pdb_setrepddl` コマンド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。

- シーケンスの場合

```
pdb_setrepseq pdb_seq
```

`pdb_seq` は、複製のマークが付けられているかどうかを確認するシーケンスの名前です。

参照：

- [Replication Agent の現在のステータスの確認 \(129 ページ\)](#)
- [プライマリ・データベース・ログへのアクセス \(131 ページ\)](#)
- [Replication Agent ログ \(132 ページ\)](#)
- [Replication Agent オペレーションとパフォーマンスを確認するための `ra_statistics` の使用 \(133 ページ\)](#)
- [使用可能なメモリ \(134 ページ\)](#)
- [LTL のデバッグ \(137 ページ\)](#)
- [省略された DDL コマンド \(138 ページ\)](#)
- [JVM スレッド・ダンプ \(139 ページ\)](#)
- [LOB カラムの複製の有効化と無効化 \(108 ページ\)](#)
- [ストアド・プロシージャのマーク付けとマーク解除 \(109 ページ\)](#)
- [DDL 複製の有効化と無効化 \(105 ページ\)](#)
- [Oracle シーケンスのマーク付けとマーク解除 \(115 ページ\)](#)
- [プライマリ・データベース・テーブルの複製のマーク付け \(102 ページ\)](#)

Replication Agent の現在のステータスの確認

Replication Agent インスタンスの状態は、そのインスタンスが複製状態か Admin 状態かを示します。

Replication Agent インスタンスが Admin 状態の場合、複製されません。

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. Replication Agent インスタンスの現在の状態を取得します。

```
ra_status
```

トラブルシューティング

このコマンドは、次のように Replication Agent インスタンスの現在の状態と、現在のアクティビティがあればそれを返します。

```
State Action
-----
ADMIN Waiting for operator command
(1 row affected)
```

『Replication Agent リファレンス・マニュアル』を参照して、**ra_status** コマンドの詳細を取得します。

Replication Agent インスタンスがいずれかの状態の場合、推奨されている対処法を実行します。

State	Action
Admin	Replication Agent の resume コマンドを実行することにより、複製を開始して、Replication Agent のインスタンスを複製状態にします。 resume コマンドを呼び出した後、Replication Agent インスタンスが Admin 状態に戻った場合、インスタンスが複製状態になることを妨げる未解決の問題が少なくとも 1 つはあるということになります。
Replicating	特に対処は必要ありません。インスタンスは正常に稼働しています。
Replicating (Resynchronization)	特に対処は必要ありません。インスタンスは、プライマリ・データベースとレプリケート・データベースの再同期中です。再同期が完了すると、Replication Agent インスタンスは自動的に複製状態に移行します。
Replication Down	複製はエラーが原因で停止しました。Replication Agent エラー・ログを確認し、診断してエラーを修正します。その後、Replication Agent のインスタンスを再開します。

参照：

- プライマリ・データベース・オブジェクトに、複製のマークがついているかの確認 (128 ページ)
- プライマリ・データベース・ログへのアクセス (131 ページ)
- Replication Agent ログ (132 ページ)
- Replication Agent オペレーションとパフォーマンスを確認するための ra_statistics の使用 (133 ページ)
- 使用可能なメモリ (134 ページ)
- LTL のデバッグ (137 ページ)
- 省略された DDL コマンド (138 ページ)
- JVM スレッド・ダンプ (139 ページ)
- Replication Agent の状態 (71 ページ)

- 複写の開始 (76 ページ)

プライマリ・データベース・ログへのアクセス

Replication Agent がプライマリ・データベース・ログを読み込むことができない場合は、Replication Agent に障害が発生する場合があります。

ユーザー・アカウント制御 (UAC) が有効になっていて、そのインスタンスの起動時に「管理者として実行」オプションを指定しないと、Replication Agent for Oracle または Replication Agent for Microsoft SQL Server のインスタンスが Windows で正しく機能しなくなる場合があります。この問題は、Microsoft Windows Vista、Microsoft Windows Server 2008、Microsoft Windows Server 2008 R2、Microsoft Windows 7 で Replication Agent にプライマリ・データベース・ログの読み取りパーミッションがない場合に発生します。Replication Agent システム・ログ・ファイルには、プライマリ・データベース・ログへのアクセスの問題を示すメッセージが含まれている場合があります。

- Replication Agent for Oracle の場合

```
java.io.FileNotFoundException:  
C:¥SOME¥PATH¥REDO01.LOG (Access is denied)
```

- Replication Agent for Microsoft SQL Server の場合

```
Failed to synchronize log device with message  
C:¥SOME¥PATH¥MSSQL¥DATA¥pdb_log.ldf (Access is denied)
```

参照：

- プライマリ・データベース・オブジェクトに、複写のマークがついているかの確認 (128 ページ)
- Replication Agent の現在のステータスの確認 (129 ページ)
- Replication Agent ログ (132 ページ)
- Replication Agent オペレーションとパフォーマンスを確認するための ra_statistics の使用 (133 ページ)
- 使用可能なメモリ (134 ページ)
- LTL のデバッグ (137 ページ)
- 省略された DDL コマンド (138 ページ)
- JVM スレッド・ダンプ (139 ページ)

- 4つ目のカラムは、エラーが発生した Java クラスを識別します。

注意： 5つ目と6つ目のカラムは、設定プロパティの `log_trace_verbose` が `true` に設定されている場合にのみ表示されます。

- 5つ目のカラムにはメソッドが含まれます。
- 6つ目のカラムには行番号が含まれます。
- 最後のカラムは、テキストによるメッセージの説明です。

注意： 場合によっては、特定のカラムの情報がこれらの説明と一致しないことがあります。このような場合、サポート・センタがメッセージの生成元を判断するために使用するその他の情報が生成されます。

参照：

- プライマリ・データベース・オブジェクトに、複製のマークがついているかの確認 (128 ページ)
- Replication Agent の現在のステータスの確認 (129 ページ)
- プライマリ・データベース・ログへのアクセス (131 ページ)
- Replication Agent オペレーションとパフォーマンスを確認するための `ra_statistics` の使用 (133 ページ)
- 使用可能なメモリ (134 ページ)
- LTL のデバッグ (137 ページ)
- 省略された DDL コマンド (138 ページ)
- JVM スレッド・ダンプ (139 ページ)

Replication Agent オペレーションとパフォーマンスを確認するための `ra_statistics` の使用

`ra_statistics` コマンドは、Replication Agent オペレーションおよびパフォーマンスの評価に使用できるアクティビティ関連の統計を返します。

最初にコマンドを実行したときに返される統計と、機能することが分かっているコマンドを正常に複製した後に返される統計を比較することにより、統計上の差異を分析して、問題のある箇所をトラブルシュートできます。統計を使用すると、次のようなインスタンスの動作を確認することができます。

- トランザクション・ログのスキャン
- 複製されたトランザクションの処理
- Replication Server への LTL の送信

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. 複製状態にあることを確認します。そうでない場合、状態を Admin に変更します。
3. すべての Replication Agent コンポーネントと Java VM の統計を取得します。

```
ra_statistics
```

4. 返された統計を保存して、比較のベースラインとして使用します。
5. 複製されていないオブジェクトに対して、アクティビティを実行します。たとえば、複製されていないテーブルを更新します。
6. 手順 2 を繰り返します。

注意： Replication Agent がトランザクションを処理するための十分な時間を取ってください。

7. 新しく返された統計アクティビティをベースラインと比較します。
Total Maintenance User operations filtered に対して返された値が増加している場合、この Replication Server コネクションに対し、Replication Server メンテナンス・ユーザとしてトランザクションを実行していることとなります。デフォルトでは、これらのトランザクションは Replication Server に送信されません。異なるユーザとしてプライマリ・データベースに接続するか、**filter_maint_userid** の設定値を **false** に設定する必要があります。

ra_statistics コマンドの詳細については、『Replication Agent リファレンス・マニュアル』を参照してください。

参照：

- プライマリ・データベース・オブジェクトに、複製のマークがついているかの確認 (128 ページ)
- Replication Agent の現在のステータスの確認 (129 ページ)
- プライマリ・データベース・ログへのアクセス (131 ページ)
- Replication Agent ログ (132 ページ)
- 使用可能なメモリ (134 ページ)
- LTL のデバッグ (137 ページ)
- 省略された DDL コマンド (138 ページ)
- JVM スレッド・ダンプ (139 ページ)
- コネクション設定パラメータの設定 (55 ページ)

使用可能なメモリ

メモリ不足を診断してトラブルシューティングします。

メモリが不足すると、次のエラー・メッセージが表示されます。

```
java.lang.OutOfMemoryError
```

メモリが不足すると、Replication Agent の複写状態が終了するか Replication Agent サーバ全体で実行が停止します。

JRE で使用できるメモリ量の調整をサポートするために、Replication Agent bin ディレクトリ内にあるすべての実行可能スクリプト(またはバッチ・ファイル)は、RA_JAVA_MAX_MEM という名前の環境変数を参照します。すべての Replication Agent インスタンスの RUN スクリプトも、RA_JAVA_MAX_MEM 環境変数を参照します。

参照：

- プライマリ・データベース・オブジェクトに、複写のマークがついているかの確認 (128 ページ)
- Replication Agent の現在のステータスの確認 (129 ページ)
- プライマリ・データベース・ログへのアクセス (131 ページ)
- Replication Agent ログ (132 ページ)
- Replication Agent オペレーションとパフォーマンスを確認するための ra_statistics の使用 (133 ページ)
- LTL のデバッグ (137 ページ)
- 省略された DDL コマンド (138 ページ)
- JVM スレッド・ダンプ (139 ページ)

デフォルトのヒープ・サイズ

Replication Agent は実行可能プログラムや RUN スクリプトで RA_JAVA_MAX_MEM 環境変数の設定を行わず、JVM を有効にしてデフォルトの最大ヒープ・サイズを使用します。

Java 6 では、デフォルトの最大ヒープ・サイズは次の値よりも大きくなります。

- 物理メモリの 4 分の 1
- 1GB

デフォルトの初期および最大ヒープ・サイズの詳細およびデフォルト値の設定方法についての推奨事項については、Java 6 のドキュメントを参照してください。

デフォルトの最大メモリ設定

デフォルトの最大メモリは、次の 3 つの方法で設定できます。

RA_JAVA_MAX_MEM を次のように設定できます。

- RUN スクリプト・ファイルで設定
- ra スクリプト・ファイルで設定

- 環境変数として設定

RUN スクリプト・ファイルで RA_JAVA_MAX_MEM を設定すると、環境変数設定および ra スクリプト・ファイルの設定が上書きされます。RUN スクリプト・ファイルで RA_JAVA_MAX_MEM を設定しない場合、環境変数設定は、ra スクリプト・ファイルで設定されたすべての値を上書きします。ra スクリプト・ファイルでの RA_JAVA_MAX_MEM 設定は、RA_JAVA_MAX_MEM が RUN スクリプト・ファイルおよび環境変数で設定されていない場合にのみ使用されます。

Replication Agent のすべてのインスタンス、または 1 つの特定のインスタンスに対し、JRE に使用可能な最大メモリ量の JVM デフォルト値を上書きできます。

すべてのインスタンスのデフォルト値の上書き

Replication Agent のすべてのインスタンスのデフォルト最大メモリ設定を上書きします。

1. ra スクリプト・ファイルを開いて編集します。

- Windows では、%SYBASE%\RAX-15_5\bin\ra.bat ファイルを編集します。
- UNIX では、\$SYBASE/RAX-15_5/bin/ra.sh ファイルを編集します。

2. ra スクリプト・ファイルの行のコメントを解除します。

- Windows では、この行のコメントを解除します。

```
set RA_JAVA_MAX_MEM=512m
```

- UNIX では、これらの行のコメントを解除します。

```
RA_JAVA_MAX_MEM=512m  
export RA_JAVA_MAX_MEM
```

注意：UNIX では、等号の前後にスペースを入れることはできません。

3. 必要に応じて、「512m」を複写環境に適した値で置き換えます。

注意：ここで、「m」は、メガバイトの略です。

4. ra スクリプト・ファイルを保存します。

単一インスタンスのデフォルトの上書き

Replication Agent の単一インスタンスのデフォルト最大メモリ設定を上書きします。

1. RUN スクリプトを開いて、編集します。

- Windows では、%SYBASE%\RAX-15_5\<インスタンス>\RUN_<インスタンス>.bat ファイルを編集します。

- UNIX では、\$SYBASE/RAX-15_5/<インスタンス>/RUN_<インスタンス>.sh ファイルを編集します。

ここで、<インスタンス>は、Replication Agent インスタンスの名前です。

2. RUN スクリプト・ファイルの行のコメントを解除します。

- Windows では、この行のコメントを解除します。

```
set RA_JAVA_MAX_MEM=512m
```

- UNIX では、これらの行のコメントを解除します。

```
RA_JAVA_MAX_MEM=512m
export RA_JAVA_MAX_MEM
```

注意：UNIX では、等号の前後にスペースを入れることはできません。

3. 必要に応じて、「512m」を複写環境に適した値で置き換えます。

注意：ここで、「m」は、メガバイトの略です。

4. RUN スクリプト・ファイルを保存します。

LTL のデバッグ

ログ転送言語 (LTL) は、Replication Server と通信または Replication Server に複写データを分散するために使用される構文です。

LTL は、Replication Agent からの主要な出力です。

問題のある LTL をデバッグするには、次を実行します。

1. 管理者ログインを使用して、実行中の Replication Agent のインスタンスにログインします。
2. Replication Agent インスタンスが Admin 状態になっていることを確認します。

```
ra_status
```

Replication Agent インスタンスが Admin 状態になっていない場合は、Replication Agent インスタンスをクワイース状態にします。

```
quiesce
```

3. プライマリ・データベースの Replication Agent 設定パラメータの値を設定します。

```
trace LTITRACELTL, true
```

4. Replication Agent インスタンスの複写を開始します。

```
resume
```

5. 新しい複写アクティビティが生成されたら、ログ・ディレクトリの LTITRACELTL.log ファイルをチェックして、問題をデバッグします。

参照：

- プライマリ・データベース・オブジェクトに、複製のマークがついているかの確認 (128 ページ)
- Replication Agent の現在のステータスの確認 (129 ページ)
- プライマリ・データベース・ログへのアクセス (131 ページ)
- Replication Agent ログ (132 ページ)
- Replication Agent オペレーションとパフォーマンスを確認するための `ra_statistics` の使用 (133 ページ)
- 使用可能なメモリ (134 ページ)
- 省略された DDL コマンド (138 ページ)
- JVM スレッド・ダンプ (139 ページ)

冗長 LTL の生成

デフォルトでは、Replication Server に送信されるデータ量を減らすため、Replication Agent によって生成された LTL は圧縮されます。問題をデバッグするために、冗長出力が必要な場合、これらの設定パラメータを変更します。

冗長 LTL を出力するには、次を実行します。

1. 管理者ログインを使用して、実行中の Replication Agent のインスタンスにログインします。
2. Replication Agent インスタンスが Admin 状態になっていることを確認します。

```
ra_status
```

Replication Agent インスタンスが Admin 状態になっていない場合は、Replication Agent インスタンスをクワイース状態にします。

```
quiesce
```

3. プライマリ・データベースの Replication Agent 設定パラメータの値を設定します。

```
ra_config column_compression, false
ra_config compress_ltl_syntax, false
ra_config structured_tokens, false
```

4. 新しい複製アクティビティが生成されたら、ログ・ディレクトリの `LTITRACELTL.log` ファイルをチェックして、問題をデバッグします。

省略された DDL コマンド

`ra_locator move_truncpt` コマンドを使用して、プライマリ・データベースのトランザクション・ログの終わりまでトランケーション・ポイントを移動した場合、RASD とプライマリ・データベースとの同期を維持するために、Replication Agent によって読み取られた DDL コマンドが省略されます。

プライマリ・データベースの内容と RASD の内容の不整合を解決するには、次のいずれかを実行します。

- **ra_admin refresh** コマンドを使用して RASD を更新します。これで RASD はすべてのオブジェクトに対するプライマリ・データベースの現在のスキーマ情報で更新されます。
- DDL の影響を受けたオブジェクトが 1 つのみであることが明らかな場合、オブジェクトのマークを外して、再度マークを付けます。

参照：

- プライマリ・データベース・オブジェクトに、複写のマークがついているかの確認 (128 ページ)
- Replication Agent の現在のステータスの確認 (129 ページ)
- プライマリ・データベース・ログへのアクセス (131 ページ)
- Replication Agent ログ (132 ページ)
- Replication Agent オペレーションとパフォーマンスを確認するための ra_statistics の使用 (133 ページ)
- 使用可能なメモリ (134 ページ)
- LTL のデバッグ (137 ページ)
- JVM スレッド・ダンプ (139 ページ)

JVM スレッド・ダンプ

スレッド・ダンプには Replication Agent JVM のアクティブなスレッドの情報が含まれているため、Replication Agent のデッドロックまたはパフォーマンスの問題のトラブルシューティングに役に立ちます。

Windows マシンでのスレッド・ダンプの取得

Replication Agent を起動したコマンド・ウィンドウで、[Ctrl] キーを押しながら、[Break] キーを押します。

UNIX または *Linux* マシンでのスレッド・ダンプの取得

- **ps** および **grep** を使用して、Replication Agent が実行中の JVM のプロセス ID を検索します。

```
%>ps -ef | grep RAX-15_5
```

e オプションと **f** オプションは、実行中のプロセスごとの全出力を示します。**grep** コマンドを実行すると、文字列 “RAX-15_5” の結果がフィルタされます。次のような結果が返されます。

```
sybase 12345 67890 0 14:21 pts/1 00:00:22  
/software/sybase/RAX-15_5/JRE6/bin/java -server
```

トラブルシューティング

12345 は Replication Agent Java プロセスのプロセス ID (PID) です。

- **kill** コマンドと **QUIT** オプションを使用してプロセスを終了します。

```
%>kill -QUIT pid
```

pid は最後の手順で取得した PID です。

UNIX または Linux では、通常はバックグラウンドで Replication Agent プロセスが起動されます。このため、Replication Agent プロセスが起動されたコンソールは、スレッド・ダンプを取得するときに使用できなくなる可能性があります。JVM では Replication Agent プロセスが起動されたコンソールにスレッド・ダンプを常に出力するため、Replication Agent を起動する場合、常にプロセスの出力をファイルにリダイレクトする必要があります。例：

```
%>RUN_inst_name >! output.txt
```

ここで、*inst_name* は、Replication Agent インスタンスのサーバ名です。

注意： AIX では、プロセスの出力に、スレッド・ダンプが含まれている `javacore` ファイルのパスが含まれています。

参照：

- プライマリ・データベース・オブジェクトに、複製のマークがついているかの確認 (128 ページ)
- Replication Agent の現在のステータスの確認 (129 ページ)
- プライマリ・データベース・ログへのアクセス (131 ページ)
- Replication Agent ログ (132 ページ)
- Replication Agent オペレーションとパフォーマンスを確認するための `ra_statistics` の使用 (133 ページ)
- 使用可能なメモリ (134 ページ)
- LTL のデバッグ (137 ページ)
- 省略された DDL コマンド (138 ページ)

Replication Server のトラブルシューティングの問題

最も一般的な複製の問題を確認するには、Replication Server コマンドを使用します。

Replication Server の問題の診断および解決方法の詳細については、『Replication Server トラブルシューティング・ガイド』を参照してください。

複写定義とサブスクリプションの確認

適切な情報で複写定義を作成したか、すべての複写定義のサブスクリプションを定義してアクティブ化したかを確認します。

Replication Server のステータスとオペレーションのチェック

Replication Server には、ステータスとオペレーションのチェックのために使用できるいくつかの **admin** コマンドがあります。

1. 「sa」パーミッションを持つユーザー・ログインで Replication Server にログインします。
2. Replication Server の現在のステータスを確認します。

```
admin health
```

この例で示されるとおり、このコマンドは Replication Server の現在のステータスを返します。

Mode	Quiesce	Status
NORMAL	FALSE	HEALTHY

Replication Server のステータスが、SUSPECT の場合、**admin who_is_down** コマンドを使用して、停止しているか、他のサーバへの接続を試行している Replication Server スレッドを確認します。

3. Replication Server プライマリ・データベース・コネクション (Replication Agent からプライマリ Replication Server へのコネクション) の現在のステータスを確認します。

```
admin show_connections
```

admin who, dsi コマンドを使用して、プライマリ Replication Server の Replication Agent コネクションについての詳細を取得することもできます。

注意： **admin show_connections** コマンドまたは **admin who, dsi** コマンド出力を使用して、プライマリ・データ・サーバおよびプライマリ・データベース名が、プライマリ Replication Server の Replication Agent コネクションに対して適切かどうかを確認します。

『Replication Server リファレンス・マニュアル』を参照して、**admin show_connections** コマンドおよび **admin who** コマンドについての詳細を取得してください。

rs_username ログインに適切なパーミッションが付与されたかの確認

Replication Server での Replication Agent ログインを確認します。

Replication Server の **connect source lti** コマンド。

- Replication Agent によって使用される Replication Server データベース・コネクションがプライマリ Replication Server に存在するかどうかを確認します。
- Replication Agent の **rs_username** パラメータで指定されたログイン名に、データ・ソースとしてプライマリ Replication Server に接続するパーミッションがあるかどうかを確認します。
- プライマリ Replication Server がサポートする LTL の最高位番号のバージョンを示すバージョン文字列を返します。

このタスクに必要なパラメータ値については、『Replication Agent インストール・ガイド』の「インストールおよび設定ワークシート」を参照してください。

1. **rs_username** 設定パラメータで指定された Replication Agent ユーザ・ログイン名でプライマリ Replication Server にログインします。
2. 次のコマンドを実行します。

```
connect source lti pds.pdb version
```

構文の説明は次のとおりです。

- *pds* は、Replication Agent の **rs_source_ds** 設定パラメータに指定された値です。
- *pdb* は、Replication Agent の **rs_source_db** 設定パラメータに指定された値です。
- *version* は、推奨される LTL バージョン番号です。

LTL バージョン番号の値に **999** と入力します。Replication Server は、サポートする LTL の最高位番号のバージョンを返します。

3. プライマリ Replication Server を **rs_username** として切断し、管理者ログインで Replication Agent のインスタンスにログインして、**resume** コマンドを呼び出します。

connect source lti コマンドの詳細については、『Replication Server デザイン・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。

ステابل・キューの確認

Replication Server のステابل・キューを確認して、どのトランザクションが処理または無視され、トランザクションが開いているか (コミットされていないか) を判断します。

SQM および SQT スレッドの情報を表示するには、次を実行します。

1. プライマリ Replication Server にログインして、**admin who, sqm** コマンドを実行します。
2. 結果を表示して、検出および無視された重複メッセージの数、および Replication Server ステابل・キューに書き込まれるブロック数を判断します。
3. プライマリ Replication Server で、**admin who, sql** コマンドを実行します。
4. 結果を表示して、オープン・トランザクションを検出します。

『Replication Server リファレンス・マニュアル』を参照して、**admin who** コマンドについての詳細を取得してください。

パフォーマンスのモニタリング

プライマリ・データベースおよびレプリケート・データベースにそれぞれ存在する **rs_ticket** および **rs_ticket_report** Replication Server ストアド・プロシージャを使用して、Replication Server のパフォーマンスを監視します。Replication Agent では、Replication Agent の **rs_ticket** コマンドを通じて、これらのストアド・プロシージャへのサポートを提供します。

rs_ticket および **rs_ticket_report** Replication Server ストアド・プロシージャの詳細については、『Replication Server リファレンス・マニュアル』を参照してください。
rs_ticket Replication Agent コマンドの詳細については、『Replication Agent リファレンス・マニュアル』を参照してください。

プライマリ・データへのサブスクリプションのマテリアライズ

Sybase 以外のデータベースのプライマリ・テーブルにサブスクリプションをマテリアライズするバルク・マテリアライゼーションと、バルク・マテリアライゼーションを使用して、プライマリ・データベースのプライマリ・テーブルからの複写を設定する方法について説明します。

マテリアライゼーション

マテリアライゼーションは、サブスクリプションの作成とアクティブ化、およびプライマリ・データベースからレプリケート・データベースへのデータのコピーによるレプリケート・データベースの初期化です。

プライマリ・データベースからデータを複写するには、プライマリ・データベースの状態と一致した状態になるように、各レプリケート・データベースを設定し、データを読み込んでおく必要があります。

Sybase Replication Server では、次の 2 種類のサブスクリプション・マテリアライゼーションがサポートされています。

- バルク・マテリアライゼーション – サブスクリプションを手動で作成してアクティブ化し、複写システムのコントロール外のデータ・アンロード・ユーティリティとデータ・ロード・ユーティリティを使用してレプリケート・データベースにデータを読み込むプロセスです。
- アトミック・マテリアライゼーション – Replication Server コマンドを使用して、サブスクリプションを作成し、レプリケート・データベースにデータを読み込むプロセスです。

注意： Replication Agent では、アトミック・マテリアライゼーションはサポートされていません。

サブスクリプション・マテリアライゼーション・メソッドの詳細については、『Replication Server 管理ガイド』を参照してください。

バルク・マテリアライゼーション

Sybase では、Sybase 以外のデータベースでプライマリ・データのサブスクリプションをマテリアライズするために、バルク・マテリアライゼーションの使用を推奨しています。

バルク・マテリアライゼーションを使用するには、次を実行します。

プライマリ・データへのサブスクリプションのマテリアライズ

- サブスクリプションの定義、アクティブ化、および確定化を行う (またはマテリアライゼーションを伴わないサブスクリプションを作成する)。
- プライマリ・サイトでサブスクリプション・データをダウンロードする。
- ダウンロードしたデータをレプリケート・データベースに移動する。
- レプリケート・テーブルにデータをロードする。
- レプリケート・データベースが複写トランザクションを受け取ることができるように、レプリケート Replication Server からレプリケート・データベースへのデータベース・コネクションをレジュームする。
- Replication Agent インスタンスで複写をレジュームする。

Sybase 以外のデータベースでのプライマリ・データのサブスクリプションには、次の2つのバルク・マテリアライゼーションのオプションがあります。

- アトミック・バルク・マテリアライゼーション
- ノンアトミック・バルク・マテリアライゼーション

データのアンロードとロード

サブスクリプション・マテリアライゼーション処理の一部は、プライマリ・テーブルからサブスクリプション・データをアンロードして、レプリケート・テーブルにロードできるようにすることです。

「サブスクリプション・データ」とは、サブスクリプションによって要求されるプライマリ・テーブル内のデータのことで、

アンロード・ユーティリティ

データ・アンロード・ユーティリティは、通常、プライマリ・データ・サーバ・ソフトウェアに付属しています。OEM で提供される任意のアンロード・ユーティリティを使用することも、データベース・アンロード・ユーティリティを使用することもできます。

ロード・ユーティリティ

レプリケート・データベースのデータ・サーバとして Adaptive Server Enterprise を使用している場合は、**bcp** を使用して、レプリケート・データベースにサブスクリプション・データをロードします。レプリケート・データベースのデータ・サーバとして Sybase 以外のデータ・サーバを使用している場合は、任意のロード・ユーティリティを使用してレプリケート・データベースにサブスクリプション・データをロードできます。

『Sybase Adaptive Server Enterprise ユーティリティ・プログラム』を参照し、**bcp** を使用して、Adaptive Server Enterprise でレプリケート・データベースにサブスクリプション・データをロードする方法を取得してください。

アトミック・バルク・マテリアライゼーション

アトミック・バルク・マテリアライゼーションは、テーブルのコピーが作成される間、プライマリ・テーブルを更新するすべてのアプリケーションをサスペンドできることを前提とします。このコピーは、レプリケート・サイトにロードされます。

プライマリ・データへの更新をサスペンドできる場合、この方法を使用してプライマリ・データベースからデータを取得できます。

アトミック・バルク・マテリアライゼーションの準備

アトミック・バルク・マテリアライゼーションの準備が完了しているかを確認します。

アトミック・バルク・マテリアライゼーションの手順を開始する前に、以下のことを確認してください。

- プライマリ・テーブルが存在し、格納データがある。
- プライマリ・テーブル (またはプライマリ・テーブルにある複写対象のカラム) に対して、所有権または **select** 権限を持つユーザ ID がある。
- レプリケート・テーブルが存在し、適切なカラムを格納している。
- 複写システムにおいて、すべての Replication Server を適切に設定している。
- プライマリ Replication Server で複写定義を正しく作成している。
- プライマリ・データベースで、Replication Agent トランザクション・ログを適切に作成している。
- プライマリ・データベースのプライマリ・テーブルに、マーク付けされ有効化された複写がある。
- Replication Agent インスタンスを開始し、そのインスタンスを複写状態にしている。

アトミック・バルク・マテリアライゼーションの実行

アトミック・バルク・マテリアライゼーションを実行します。

1. システム管理者として、レプリケート Replication Server にログインします (**sa**)。

```
isql -Usa -Psa_password -SRRS_servername
```

構文の説明は次のとおりです。

- *sa* は、システム管理者のユーザ ID。
- *sa_password* は、システム管理者のユーザ ID のパスワード。
- *RRS_servername* は、レプリケート Replication Server の名前です。

2. レプリケート Replication Server でサブスクリプションを定義します。

```
1> define subscription subscription_name
2> for replication_definition
3> with replicate at dataserver.database
4> [where search_conditions]
5> go
```

dataserver.database 名は、レプリケート・データベースに対して使用する名前と一致する必要があります。

3. プライマリ Replication Server とレプリケート Replication Server で、サブスクリプションをチェックします。このコマンドを使用して、サブスクリプションのステータスが **DEFINED** であることを確認します。

```
1> check subscription subscription_name
2> for replication_definition
3> with replicate at dataserver.database
4> go
```

4. プライマリ・トランザクションのアクティビティが行われないように、プライマリ・テーブルをロックします。これにより、マテリアライゼーション実行の間、プライマリ・テーブルに対する更新を避けることができます。
5. サイトの推奨されるデータベース・アンロード方法を使用して、プライマリ・サイトでサブスクリプション・データをアンロードし、プライマリ・テーブルからデータを選択またはダンプします。

注意： プライマリ・テーブルからサブスクリプション・データをアンロードする場合は、複写定義で指定されているカラムと、サブスクリプションで指定されているローのみを選択するようにしてください。

6. レプリケート Replication Server で、**with suspension** オプションを使用して、サブスクリプションをアクティブ化します。

```
1> activate subscription subscription_name
2> for replication_definition
3> with replicate at dataserver.database
4> with suspension
5> go
```

7. プライマリ Replication Server とレプリケート Replication Server で、サブスクリプションがアクティブになるまで待機します。プライマリ Replication Server とレプリケート Replication Server の両方で、**check subscription** を実行して、サブスクリプションのステータスが **ACTIVE** であることを確認します。

レプリケート Replication Server で、サブスクリプションのステータスが **ACTIVE** である場合は、レプリケート・データベースに対するデータベース・コネクションがサスペンドされています。

8. プライマリ・テーブルを読み込みおよび書き込みアクセスに戻します (ロックの解除)。

- サブスクリプション・データを、**bcp** またはサイトで推奨されるデータベース・ロード・ユーティリティを使用して、レプリケート・データベースにロードします。

注意：レプリケート・テーブルにサブスクリプション・データをロードする前に、Replication Agent または Replication Server 関数文字列によって実行されるデータ操作が (*datetime* 変換など)、アンロード・ファイルに適用されていることを確認します。

- レプリケート Replication Server から、レプリケート・データベースに対するデータベース・コネクションをレジュームします。

```
1> resume connection
2> to dataserver.database
3> go
```

- レプリケート Replication Server でサブスクリプションを確定化します。

```
1> validate subscription subscription_name
2> for replication_definition
3> with replicate at dataserver.database
4> go
```

- プライマリ Replication Server とレプリケート Replication Server の両方でサブスクリプションが有効になるまで待機した後、**check subscription** を実行して、ステータスが **VALID** であることを確認します。

この手順を完了すると、サブスクリプションが作成され、レプリケート・データはプライマリ・データと一致しています。複写は継続中です。

参照：

- トラブルシューティング (127 ページ)

ノンアトミック・バルク・マテリアライゼーション

ノンアトミック・バルク・マテリアライゼーションは、テーブルのコピーが作成される間、プライマリ・テーブルを更新するアプリケーションをサスペンドできないことを前提とします。そのため、ノンアトミック・マテリアライゼーションでは、レプリケート・データベースがプライマリ・データベースと同期されるように、Replication Server の「オートコレクション」機能を使用する必要があります。

注意：プライマリ・テーブルの複写定義に **replicate minimal columns** 機能が設定されている場合、ノンアトミック・マテリアライゼーションを使用することはできません。

ノンアトミック・バルク・マテリアライゼーションの準備

ノンアトミック・バルク・マテリアライゼーションの準備のためにタスクを実行します。

ノンアトミック・バルク・マテリアライゼーションの手順を開始する前に、以下のことを確認してください。

- プライマリ・テーブルが存在し、格納データがある。
- プライマリ・テーブル (またはプライマリ・テーブルにある複製対象のカラム) に対して、所有権または **select** 権限を持つユーザ ID がある。
- レプリケート・テーブルが存在し、適切なカラムを格納している。
- 複製システムにおいて、すべての Replication Server を適切に設定している。
- プライマリ Replication Server で複製定義を正しく作成している。
- プライマリ・データベースで、Replication Agent トランザクション・ログを適切に作成している。
- プライマリ・データベースのプライマリ・テーブルに、マーク付けされ有効化された複製がある。
- Replication Agent インスタンスを開始し、そのインスタンスを複製状態にしている。

ノンアトミック・バルク・マテリアライゼーションの実行

ノンアトミック・バルク・マテリアライゼーションを実行します。

1. システム管理者として、レプリケート Replication Server にログインします (**sa**)。

```
isql -Usa -Psa_password -SRRS_servername
```

構文の説明は次のとおりです。

- *sa* は、システム管理者のユーザ ID。
- *sa_password* は、システム管理者のユーザ ID のパスワード。
- *RRS_servername* は、レプリケート Replication Server の名前です。

2. レプリケート Replication Server でオートコレクション機能をオンにします。

```
1> set autocorrection on
2> for replication_definition
3> with replicate at dataserver.database
4> go
```

3. レプリケート Replication Server でサブスクリプションを定義します。

```
1> define subscription subscription_name
2> for replication_definition
3> with replicate at dataserver.database
4> with suspension
5> go
```

dataserver.database 名は、レプリケート・データベースに対して使用する名前と一致する必要があります。

4. プライマリ・データベースで、**rs_marker** を呼び出して、サブスクリプションをアクティブ化します。
5. プライマリ Replication Server とレプリケート Replication Server で、サブスクリプションをチェックします。このコマンドを使用して、サブスクリプションのステータスが **ACTIVE** であることを確認します。

```
1> check subscription subscription_name
2> for replication_definition
3> with replicate at dataserver.database
4> go
```

レプリケート Replication Server で、サブスクリプションのステータスが **ACTIVE** である場合は、レプリケート・データベースに対するデータベース・コネクションがサスペンドされています。

6. サイトの推奨されるデータベース・アンロード方法を使用して、プライマリ・サイトでサブスクリプション・データをアンロードし、プライマリ・テーブルからデータを選択またはダンプします。

注意： プライマリ・テーブルからサブスクリプション・データをアンロードする場合は、複写定義で指定されているカラムと、サブスクリプションで指定されているローのみを選択するようにしてください。

7. プライマリ・データベースで、**rs_marker** を呼び出して、サブスクリプションを確定化します。
8. プライマリ Replication Server とレプリケート Replication Server の両方でサブスクリプションが有効になるまで待機した後、**check subscription** を実行して、ステータスが **VALID** であることを確認します。
9. サブスクリプション・データを、**bcp** またはサイトで推奨されるデータベース・ロード・ユーティリティを使用して、レプリケート・データベースにロードします。

注意： サブスクリプション・データをレプリケート・テーブルにロードする前に、Replication Agent (*datetime* 変更など) または Replication Server 関数文字列によって実行されるデータ処理がアンロード・ファイルに適用されているかを確認します。

10. レプリケート Replication Server から、レプリケート・データベースに対するデータベース・コネクションをレジュームします。

```
1> resume connection
2> to dataserver.database
3> go
```

11. プライマリ Replication Server とレプリケート Replication Server の両方でサブスクリプションが有効になるまで待機した後、**check subscription** を実行して、ステータスが **VALID** であることを確認します。

レプリケート Replication Server でサブスクリプションのステータスが **VALID** の場合、レプリケート・データベースはプライマリ・データベースと同期されているため、オートコレクションをオフにできます。

12. レプリケート Replication Server でオートコレクション機能をオフにします。

```
1> set autocorrection off
2> for replication_definition
3> with replicate at dataserver.database
4> go
```

この手順を完了すると、サブスクリプションが作成され、レプリケート・データはプライマリ・データと一致しています。複製は継続中です。

複製コマンド言語 (RCL) コマンドについては、『Replication Server Commands Reference』を参照してください。Replication Server の設定とマテリアライゼーション・メソッドの詳細については、『Replication Server 管理ガイド』を参照してください。

参照：

- [トラブルシューティング \(127 ページ\)](#)

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に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。複製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。データ複製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製とは異なります。「データの複製」と対比。
- **データの複製** – 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。データの複製はデータの分配とは異なります。複製データはリモート・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製」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

Adaptive Server Enterprise
 ユーザ・ログイン 53
 Administrator GUI
 起動 37
 Administrator GUI の起動 37
 agt_service ユーティリティ 16, 17
 Replication Agent の起動 47
 構文 17

B

bcp (バルク・コピー) ユーティリティ 149, 151

C

charset 16
 connect source パーミッション 53, 69

D

DDL
 複写の有効化 105
 DDL 複写の有効化 105

I

IBM DB2 Universal Database 6, 7

J

Java Runtime Environment (JRE) 10
 JDBC ドライバ 9

L

Log Administrator コンポーネント 10
 Log Reader コンポーネント 10
 Log Transfer Interface (LTI) コンポーネント 10
 ログ転送言語 (LTL) 6
 Log Transfer Manager (LTM) コンポーネント 10

LTI

次を参照：Log Transfer Interface (LTI) コンポーネント

LTM

次を参照：Log Transfer Manager (LTM) コンポーネント

M

maxmem 16
 Microsoft SQL Server 6, 7

O

Oracle 6, 7

P

pdb_auto_run_scripts 設定パラメータ 85, 88
 pdb_dflt_object_repl 設定パラメータ 101, 104, 110, 117
 pdb_setrepcol コマンド 108, 109
 pdb_setrepproc コマンド 87, 115, 116
 pdb_setrepseq コマンド 119
 pdb_setreptable コマンド 64, 65, 87, 102, 105-107
 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 65, 87

R

ra_admin コマンド 83, 86, 88
 ra_admin ユーティリティ 19
 ra_admin_instance_prefix 設定パラメータ 86
 ra_config コマンド 84, 119
 ra_set_login コマンド 51
 ra_statistics コマンド 75, 133, 134
 ra_status コマンド 71
 RASD 91, 99
 更新 91, 94, 96

索引

- トランケート 98, 99
- バックアップ 96
- リストア 96
- 更新の強制 94
- ra ユーティリティ 18
 - Replication Agent の起動 43
 - 構文 18
- Replication Agent 16-19
 - Log Administrator コンポーネント 10
 - Log Reader コンポーネント 10
 - Log Transfer Interface (LTI) コンポーネント 10
 - Log Transfer Manager (LTM) コンポーネント 10
 - RASD 91, 99
 - RUN スクリプト 46
 - Windows サービス・ツールでの起動 48
 - インスタンスのコピー 39
 - インスタンスの作成 38
 - インスタンスの削除 34
 - インスタンスの初期化 91, 92
 - 管理者ログイン 50
 - 管理ポート 10
 - システム・ログ・ファイル 132
 - システム・データ・レポジトリ 91, 96
 - スクリプト・ディレクトリ 86
 - チューニング 121
 - 統計、パフォーマンス 75
 - トラブルシューティング 127
 - トランザクション・ログのプレフィクス 84, 86
 - パフォーマンスのチューニング 119
 - パフォーマンスの統計 75
 - ベース・ディレクトリ 14
 - ホスト・マシン 13, 22
 - ログベース設計 6, 7
 - 設定要件 68, 69
 - 通信方法 9
 - 複写の開始 77
 - 複写の停止 77
- Replication Agent for Microsoft SQL Server 6, 7
- Replication Agent for Oracle 6, 7
- Replication Agent for UDB 6, 7
- Replication Agent インスタンス 16-19
- Replication Agent インスタンスのコピー 39

- Replication Agent インスタンスの作成 38
- Replication Agent のインスタンスの削除 34
- Replication Agent の初期化 91, 92
- Replication Agent の起動 16, 18
- Replication Monitoring Services (RMS) 8
- Replication Server
 - connect source パーミッション 53, 69
 - サブスクリプション 6, 7, 141
 - ステابل・キュー 143
 - トラブルシューティング 143
 - 複写定義 6, 7, 141
 - プライマリ Replication Server 6, 7
 - レプリケート Replication Server 6, 7
- Replication Server システム・データベース
 - 次を参照：RSSD
- RSSD 6, 7
- RUN スクリプト
 - Replication Agent 46

S

- shutdown コマンド
 - immediate オプション 75
- SYBASE 環境変数 15, 16

W

- Windows サービス・ツール
 - Replication Agent の起動 48

あ

- アトミック・バルク・マテリアライゼーション 145

い

- インスタンス、Replication Agent 16-19
 - 名前 13
 - 削除 34
 - 初期化 91, 92
 - 設定要件 69
- インスタンス名 16-19
- インストール 16

え

エラー

Replication Agent の起動 19
複製の失敗 128

か

環境、SYBASE 変数 15, 16
管理者ユーティリティ 37
管理者ログイン 50
管理ポート 10, 19

き

キュー

Replication Server 143

く

クライアント・ポート 19
Replication Server 59

こ

更新

RASD 91, 94
システム・データ・レポジトリ 91

構文

agt_service ユーティリティ 17
ra ユーティリティ 18

コマンド

pdb_setrepcol 108, 109
pdb_setrepproc 87, 115, 116
pdb_setrepseq 119
pdb_setreptable 64, 65, 87, 102, 105–107
ra_admin 83, 86, 88
ra_config 84, 119
ra_set_login 51
ra_statistics 75, 133, 134
ra_status 71

コンポーネント

Replication Agent 10
複製システム 5

さ

サブスクリプションのマテリアライズ
バルク・マテリアライゼーション 145

し

シーケンス

マーク解除 117
マーク付け 115, 117

システム・データ・レポジトリ 91, 96, 99
更新 91
更新の強制 94

す

スクリプト

ディレクトリ 86
トランザクション・ログの作成 86

ステابل・キュー

次も参照：キュー

ストアド・プロシージャ

マーク解除 87, 113
マーク付け 66
複製の無効化 110

ストアド・プロシージャのマーク解除 113

せ

設定

Replication Agent 69

設定パラメータ

pdb_auto_run_scripts 85, 88
pdb_dflt_object_repl 101, 104, 110, 117
ra_admin_instance_prefix 84, 86
ra_admin_prefix 84
チューニングの推奨事項 120, 121
パフォーマンスのチューニング 121
既存のインスタンスからのコピー 31

そ

即時停止 75

ち

チューニング

Replication Agent 121
Replication Agent のパフォーマンス 119

索引

つ

通信

- JDBC ドライバ 9
- Replication Agent プロトコル 9

て

- データベース
 - データのアンロード 146
- データベース・オブジェクト
 - DDL 68, 105
 - LOB カラム 67
- データベースからのデータのアンロード 146
- データ定義言語
 - 次を参照： DDL
- テーブル、プライマリ
 - 次を参照： プライマリ・テーブル

と

- 統計、パフォーマンス 75
- トラブルシューティング 127, 143
 - Replication Server 143
 - サブスクリプション 141
 - 複写定義 141
 - 起動エラー 19
- トランケート
 - RASD 98, 99
- トランザクション・ログ 82
 - プレフィクス 84, 86
 - 作成スクリプト 86

な

- 名前 16-19
 - Replication Agent インスタンス 13
 - RSSD ユーザ・ログイン 53

の

- ノンアトミック・バルク・マテリアライゼーション 145

は

- バージョン 18

- パーミッション
 - Replication Server の connect source 53, 69
- パーミッションの付与
 - Replication Server の connect source 53
- バックアップ
 - RASD 96
- パフォーマンス
 - チューニング 119
 - 統計 75
- バルク・マテリアライゼーション 145
 - アトミック 145
 - ノンアトミック 145

ひ

- 表記規則
 - 構文 1
 - スタイル 1

ふ

- ファイル
 - Replication Agent スクリプト・ディレクトリ 86
 - Replication Agent ベース・ディレクトリ 14
 - システム・ログ・ファイル 132
- 複写システム・コンポーネントの移動 80
 - Replication Agent インスタンスの移動 80
- 複写定義 6, 7
 - トラブルシューティング 141
- 複数の Replication Agent 11
- プライマリ・データベース
 - トランザクション・ログ 82, 88
- プライマリ・テーブル 6
 - サブスクリプション 6, 7, 141
 - データのアンロード 146
 - マーク解除 87
 - 複写の無効化 87, 106
- プライマリ・テーブルに対するサブスクリプション 6, 7, 141
- プレフィクス、トランザクション・ログ 84, 86

へ

ベース・ディレクトリ、Replication Agent

ヘルプ 17

変数、SYBASE 環境 15, 16

ほ

ホスト・マシン

Replication Agent 13, 22

ま

マーク解除

シーケンス 117

ストアド・プロシージャ 87

プライマリ・テーブル 87

マーク付け

ストアド・プロシージャ 66

プライマリ・テーブル 102

マルチパス複写 11

む

無効化

ストアド・プロシージャの複写 110

テーブルの複写 106

も

文字セット 42

ゆ

ユーザ ID

Adaptive Server Enterprise ユーザ・ログイン

53

ユーティリティ

Administrator GUI 37

agt_service ユーティリティ 16

bcp (バルク・コピー) ユーティリティ 149,
151

ra_admin ユーティリティ 19

ra ユーティリティ 18

よ

要件

Replication Agent の設定 68

り

リストア

RASD 96

れ

レポジトリ

システム・データ 91, 96, 99

ろ

ログ・ファイル

Replication Agent システム・ログ 132

ログベースの Replication Agent 6, 7

