



リファレンス・マニュアル

---

# Replication Agent™ 15.7.1

Linux、Microsoft Windows、および UNIX 版

ドキュメント ID : DC01887-01-1571-01

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
コマンド・リファレンス .....	5
Replication Agent コマンド・テーブル .....	5
log_system_name .....	11
lr_dump_marker .....	12
pdb_capabilities .....	13
pdb_date .....	13
pdb_execute_sql .....	14
pdb_gen_id .....	15
pdb_get_columns .....	16
pdb_get_databases .....	18
pdb_get_primary_keys .....	19
pdb_get_procedure_parms .....	20
pdb_get_procedures .....	22
pdb_get_sql_database .....	24
pdb_get_tables .....	24
pdb_ownerfilter .....	26
pdb_send_osuser_list .....	28
pdb_set_sql_database .....	30
pdb_setrepcol .....	31
pdb_setrepddl .....	36
pdb_setrepproc .....	45
pdb_setrepseq .....	54
pdb_setreptable .....	56
pdb_skip_op .....	69
pdb_thread_filter .....	72
pdb_truncate_xlog .....	73
pdb_version .....	75
pdb_xlog .....	75
quiesce .....	80
ra_admin .....	81

ra_config .....	86
ra_date .....	88
ra_deviceoffset .....	88
ra_devicepath .....	89
ra_downgrade .....	90
ra_downgrade_accept .....	91
ra_downgrade_prepare .....	93
ra_dump .....	94
ra_dumptran .....	95
ra_help .....	99
ra_helparchive .....	100
ra_helparticle .....	101
ra_helppdb .....	102
ra_helpdevice .....	103
ra_helpdeviceoffset .....	105
ra_helpfield .....	107
ra_helplocator .....	109
ra_helpop .....	110
ra_helptran .....	113
ra_helpuser .....	113
ra_license .....	115
ra_locator .....	116
ra_maintid .....	119
ra_marker .....	120
ra_migrate .....	121
ra_purge_first_open .....	122
ra_set_autocorrection .....	123
ra_set_login .....	125
ra_statistics .....	126
ra_statrack .....	137
ra_statrack_list .....	137
ra_status .....	138
ra_truncatearticles .....	139
ra_truncateddfilters .....	140
ra_truncateusers .....	141

ra_updatedevices .....	142
ra_updateusers .....	144
ra_version .....	145
ra_version_all .....	145
rasd_backup .....	146
rasd_helpbackup .....	147
rasd_removebackup .....	148
rasd_restore .....	149
rasd_trunc_schedule .....	150
resume .....	152
rs_create_repdef .....	155
rs_drop_repdef .....	157
rs_ticket .....	158
server_xlog .....	159
shutdown .....	161
suspend .....	162
test_connection .....	163
trace .....	165
<b>設定パラメータ .....</b>	<b>169</b>
Replication Agent 設定ファイル .....	169
設定ファイル・フォーマット .....	169
設定パラメータの変更 .....	170
Replication Agent 設定のコピー .....	170
設定パラメータ・リファレンス .....	171
admin_port .....	179
asm_password .....	180
asm_tns_connection .....	181
asm_tns_filename .....	181
asm_username .....	182
auto_adjust_block_count .....	183
column_compression .....	184
compress_ltl_syntax .....	185
connect_to_rs .....	185
ddl_password .....	186
ddl_username .....	186

dump_batch_timeout .....	187
filter_maint_userid .....	188
function_password .....	189
function_username .....	189
log_backup_files .....	190
log_directory .....	190
log_read_block_count .....	191
log_trace_verbos .....	192
log_wrap .....	192
lr_max_lobdata_cache .....	193
lr_max_op_queue_size .....	193
lr_max_scan_queue_size .....	193
lr_ntext_byte_order .....	194
lr_read_buffer_size .....	196
lr_send_trunc_partition_ddl .....	196
lti_batch_mode .....	197
lti_formatter_count .....	198
lti_max_buffer_size .....	198
lti_update_trunc_point .....	199
ltl_batch_size .....	200
ltl_big_endian_unitext .....	200
ltl_character_case .....	201
ltl_origin_time_required .....	202
ltl_send_only_primary_keys .....	202
ltm_admin_pw .....	204
ltm_admin_pw_min_length .....	204
ltm_admin_user .....	205
max_ops_per_scan .....	205
pdb_archive_path .....	206
pdb_archive_remove .....	207
pdb_auto_create_repdefs .....	208
pdb_automark_tables .....	210
pdb_auto_run_scripts .....	211
pdb_convert_datetime .....	212
pdb_dflt_column_repl .....	214

pdb_dflt_object_repl .....	214
pdb_ignore_unsupported_anydata .....	215
pdb_include_archives .....	217
pdb_skip_missing_user .....	217
pdb_support_large_identifier .....	218
pdb_timezone_file .....	219
pdb_xlog_device .....	219
pdb_xlog_prefix .....	220
pdb_xlog_prefix_chars .....	221
pds_connection_type .....	222
pds_dac_port_number .....	223
pds_database_name .....	223
pds_datasource_name .....	224
pds_host_name .....	224
pds_integrated_security .....	225
pds_password .....	226
pds_port_number .....	226
pds_retry_count .....	227
pds_retry_timeout .....	227
pds_server_name .....	227
pds_tns_connection .....	228
pds_tns_filename .....	228
pds_username .....	229
ra_admin_device .....	230
ra_admin_instance_prefix .....	230
ra_admin_prefix .....	231
ra_admin_prefix_chars .....	232
ra_admin_owner .....	233
ra_retry_count .....	234
ra_retry_timeout .....	234
rasd_backup_dir .....	235
rasd_database .....	235
rasd_mirror_tran_log .....	236
rasd_trace_log_dir .....	237
rasd_tran_log .....	238

rasd_tran_log_mirror .....	239
ra_standby .....	239
ra_statrack_interval .....	240
rman_enabled .....	241
rman_password .....	242
rman_username .....	242
rs_charset .....	243
rs_host_name .....	244
rs_packet_size .....	245
rs_password .....	246
rs_port_number .....	246
rs_replicate_owner_required .....	246
rs_retry_count .....	247
rs_retry_timeout .....	247
rs_source_db .....	248
rs_source_ds .....	248
rs_ticket_version .....	249
rs_username .....	249
rssd_charset .....	250
rssd_database_name .....	251
rssd_host_name .....	251
rssd_password .....	251
rssd_port_number .....	252
rssd_username .....	252
scan_sleep_increment .....	253
scan_sleep_max .....	254
skip_lr_errors .....	254
skip_ltl_errors .....	255
structured_tokens .....	255
truncation_interval .....	256
truncation_type .....	257
use_rssd .....	258
<b>追加の説明や情報の入手 .....</b>	<b>261</b>
サポート・センタ .....	261



Sybase EBF と Maintenance レポートのダウンロード .....	261
Sybase 製品およびコンポーネントの動作確認 .....	262
MySybase プロファイルの作成 .....	262
アクセシビリティ機能 .....	263
用語解説 .....	265
索引 .....	275

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

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

## 表記規則

## コマンド・リファレンス

Replication Agent コマンドについて、そのターゲット、構文、オプション、例、使用法などを説明します。

### Replication Agent コマンド・テーブル

このテーブルは、Replication Server がサポートしているコマンドをリストしています。ターゲット列は、パラメータが「すべて」のターゲットに使用されるか、Oracle、Microsoft SQL Server (MSSQL)、または IBM DB2 (UDB) の特定のターゲットに使用されるかを示します。

表 1 : Replication Agent コマンド

コマンド名	ターゲット	説明
log_system_name (11 ページ)	ALL	Replication Agent システムのログ・ファイルのパスを返します。
lr_dump_marker (12 ページ)	Oracle	system change number (SCN) を返すか設定し、最後にダンプされたプライマリ・データベースのポイントを表示します。
pdb_capabilities (13 ページ)	ALL	Replication Agent 機能のリストを返します。
pdb_date (13 ページ)	ALL	プライマリ・データ・サーバの現在の日付と時間を返します。
pdb_execute_sql (14 ページ)	ALL	現在のデータベースの指定された SQL 文を実行します。
pdb_gen_id (15 ページ)	ALL	データベース世代 ID の現在値を返すか、またはデータベース世代 ID の値を更新します。
pdb_get_columns (16 ページ)	ALL	指定されたテーブルのすべてのコラムのリストを返します。
pdb_get_databases (18 ページ)	ALL	プライマリ・データ・サーバにすべてのデータベースのリストを返します。
pdb_get_primary_keys (19 ページ)	ALL	指定されたテーブルのプライマリ・キーを構成するすべてのコラムのリストを返します。

コマンド名	ターゲット	説明
pdb_get_procedure_parms (20 ページ)	Oracle MSSQL	指定されたプロシージャのプロパティのリストを返します。
pdb_get_procedures (22 ページ)	Oracle MSSQL	指定されたデータベースのすべてのプロシージャのリストを返します。
pdb_get_sql_database (24 ページ)	ALL	SQL 文実施に指定されているデータベースの名前を返します。
pdb_get_tables (24 ページ)	ALL	指定されたデータベースのすべてのテーブルのリストを返します。
pdb_ownerfilter (26 ページ)	Oracle MSSQL	初期化のためにオブジェクトがフィルタされる所有者のリストを返します。このリストに所有者の名前を追加、あるいはリストから削除します。
pdb_send_osuser_list (28 ページ)	Oracle	プライマリ・データベース・ユーザの名前が、Replication Server® に送信された LTL にある対応するオペレーティング・システムのユーザ名と置き換えられるデータベース・ユーザのリストを返したり修正したりします。
pdb_set_sql_database (30 ページ)	ALL	SQL 文の実行に使用するデータベースを指定します。
pdb_setrepcol (31 ページ)	ALL	複製のマーク付けステータスを返します。また、すべてのマーク付けした列あるいは指定した列の複製を有効または無効にします。
pdb_setrepcddl (36 ページ)	Oracle MSSQL	DDL 複製ステータスを返し、DDL 文の複製を有効または無効にします。
pdb_setrepproc (45 ページ)	Oracle MSSQL	ストアド・プロシージャのマーク付けステータスを変更し、レポートします。
pdb_setrepseq (54 ページ)	Oracle	シーケンスの複製マーク付けステータスを変更し、レポートします。
pdb_setreptable (56 ページ)	ALL	テーブルの複製マーク付けステータスを変更し、レポートします。
pdb_skip_op (69 ページ)	ALL	処理中にスキップするレコードのリストを返すか、リストに対してレコードの識別子の追加または削除を行います。



コマンド名	ターゲット	説明
pdb_thread_filter (72 ページ)	Oracle	Oracle インスタンス redo ログ・スレッドまたは複製中のスレッドのすべてのアクティビティをフィルタし、フィルタされたスレッドのリストを表示します。
pdb_truncate_xlog (73 ページ)	ALL	Replication Agent プライマリ・データベースのトランザクション・ログをトランケートします。
pdb_version (75 ページ)	ALL	プライマリ・データ・サーバのタイプとバージョンを返します。
pdb_xlog (75 ページ)	ALL	トランザクション・ログ・オブジェクトの名前を返します。さらに、プライマリ・データベースに Replication Agent システム・オブジェクトを作成し、プライマリ・データベースから Replication Agent システム・オブジェクトを削除します。
quiesce (80 ページ)	ALL	トランザクション・ログと Replication Agent の内部キューに残っているすべてのデータが処理された後で、現在の Log Reader のアクティビティを停止し、Replication Agent を Admin 状態にします。
ra_admin (81 ページ)	ALL	Replication Agent システム・オブジェクトの名前を返します。さらに、プライマリ・データベースに Replication Agent システム・オブジェクトを作成するか、またはプライマリ・データベースから Replication Agent システム・オブジェクトを削除します。
ra_config (86 ページ)	ALL	設定パラメータのヘルプ情報を返します。また設定パラメータの値を設定します。
ra_date (88 ページ)	ALL	Replication Agent サーバの現在の日付と時刻を返します。
ra_deviceoffset (88 ページ)	MSSQL	Replication Agent システム・データベース (RASD) で記録されたログ・デバイスのロー・ディスク・デバイス読み取りオフセットを変更します。
ra_devicepath (89 ページ)	MSSQL	RASD で記録されたログ・デバイスのディスク・デバイス・パスを変更します。
ra_downgrade (90 ページ)	ALL	Replication Agent が以前のバージョンにダウングレードできるようにします。

コマンド名	ターゲット	説明
ra_downgrade_accept (91 ページ)	ALL	Replication Agent のダウングレード先インスタンスによって開始されたバージョン・ダウングレード・プロセスが完了します。
ra_downgrade_prepare (93 ページ)	ALL	Replication Agent がバージョン・ダウングレードできるようにします。
ra_dump (94 ページ)	ALL	プライマリ・データベースのトランザクション・ログにダンプ・マーカを記録します。
ra_dumptran (95 ページ)	Oracle	特定のデータベース・トランザクションのトラブルシューティングに使用される情報を返します。
ra_help (99 ページ)	ALL	Replication Agent コマンドのヘルプ情報を返します。
ra_helparchive (100 ページ)	Oracle	管理されているアーカイブ・ログのすべて、特定の redo ログ・スレッド、または特定の redo ログ・スレッドのアーカイブ・ログのメタデータ・リストを表示します。
ra_helparticle (101 ページ)	ALL	アーティクルに関する情報を RASD から返します。
ra_helpdb (102 ページ)	Oracle MSSQL	プライマリ・データベースに関する情報を RASD から返します。
ra_helpdevice (103 ページ)	Oracle MSSQL	プライマリ・データベース・ログ・デバイスに関する情報を RASD から返します。
ra_helpdeviceoffset (105 ページ)	MSSQL	プライマリ・データベース・ログ・デバイスに関するデバイス・オフセット情報を RASD ログ・デバイス・レポジトリから返します。
ra_helpfield (107 ページ)	ALL	フィールド (テーブル内の列またはストアド・プロシージャ内の入力パラメータ) に関する情報を RASD から返します。
ra_helplocator (109 ページ)	ALL	LTM ロケータ・フィールド値を返します。
ra_helpop (110 ページ)	ALL	特定のデータベース・トランザクション・ログ・オペレーションのトラブルシューティングに使用される情報を返します。
ra_helptran (113 ページ)	Oracle	すべてのオープン・トランザクションのリストを返します。

コマンド名	ターゲット	説明
ra_helpuser (113 ページ)	Oracle MSSQL	プライマリ・データベース・ユーザに関する情報を RASD から返します。
ra_license (115 ページ)	ALL	Replication Agent のライセンス情報とライセンスされている機能を返します。
ra_locator (116 ページ)	ALL	Replication Agent が格納している LTM ローケーターの現在の値を返し、変更します。
ra_maintid (119 ページ)	ALL	Replication Agent の接続のメンテナンス・ユーザを返します。
ra_marker (120 ページ)	ALL	プライマリ・データベースのトランザクション・ログにマーカーを記録します。
ra_migrate (121 ページ)	ALL	Replication Agent の各リリース間で必要なマイグレーション・タスクやダウングレード・タスクを行います。
ra_purge_first_open (122 ページ)	Oracle	オープン・トランザクションのリストから最初のオープン・トランザクションを削除します。
ra_set_autocorrection (123 ページ)	ALL	マーク付けされたテーブルのオートコレクションの有効化と無効化
ra_set_login (125 ページ)	ALL	Replication Agent の admin ユーザ・ログインとパスワードを設定します。
ra_statistics (126 ページ)	ALL	指定された Replication Agent コンポーネントまたはすべてのコンポーネントの統計を返し、すべてのコンポーネントの統計をリセットします。
ra_statrack (137 ページ)	ALL	統計追跡スレッドの開始と停止を行います。
ra_statrack_list (137 ページ)	ALL	統計グループを追跡リストに追加またはリストから削除、追跡リストの交換、現在追跡されている統計リストの表示を行います。
ra_status (138 ページ)	ALL	現在の Replication Agent の状態を返します。
ra_truncatearticles (139 ページ)	ALL	RASD のシステム・データ・レポジトリにある古いバージョンのプライマリ・データベース・アートをトランケートします。
ra_truncateddfilters (140 ページ)	Oracle	RASD でフィルタされる DDL コマンドの古いリストをトランケートします。

コマンド名	ターゲット	説明
ra_truncateusers (141 ページ)	Oracle MSSQL	RASD のシステム・データ・レポジトリにある古いバージョンのプライマリ・データベース・ユーザをトランケートします。
ra_updatedevices (142 ページ)	Oracle MSSQL	RASD のログ・デバイス・レポジトリを更新します。
ra_updateusers (144 ページ)	Oracle	プライマリ・データベースから RASD にユーザ情報を再ロードします。
ra_version (145 ページ)	ALL	Replication Agent バージョンを返します。
ra_version_all (145 ページ)	ALL	Replication Agent、プライマリ・データ・サーバ、Replication Server、通信ドライバのバージョンを返します。
rasd_backup (146 ページ)	Oracle MSSQL	Replication Agent システム・データベース (RASD) をバックアップします。
rasd_helpbackup (147 ページ)	Oracle MSSQL	RASD のバックアップのリストを表示します。
rasd_removebackup (148 ページ)	Oracle MSSQL	RASD のバックアップを削除します。
rasd_restore (149 ページ)	Oracle MSSQL	Replication Agent システム・データベース (RASD) をリストアします。
rasd_trunc_schedule (150 ページ)	Oracle MSSQL	レポジトリ・トランケーションの週ごとのスケジュールのリストを返します。また、特定のスケジュールを追加または削除します。
resume (152 ページ)	ALL	現在のアクティブなログの複写を開始し、Replication Agent を Replicating 状態にします。
rs_create_repdef (155 ページ)	Oracle MSSQL	マーク付けされた 1 つまたはすべてのテーブルおよびプロシージャについて、Replication Server で複写定義を作成します。
rs_drop_repdef (157 ページ)	Oracle MSSQL	設定された Replication Server で、テーブルおよびプロシージャの複写定義が削除されます。
rs_ticket (158 ページ)	ALL	プライマリ・データベースのトランザクション・ログに <b>rs_ticket</b> マーカを設定して、Replication Server の <b>rs_ticket</b> の処理をサポートします。

コマンド名	ターゲット	説明
server_xlog (159 ページ)	MSSQL	Microsoft SQL Server システム・リソース・データベースで Replication Agent のシステム・オブジェクトを作成または削除します。このコマンドは、Microsoft SQL Server の Replication Agent を初めて初期化する際に使用します。 <b>pdb_xlog init</b> より前に実行してください。
shutdown (161 ページ)	ALL	Replication Agent を停止します。
suspend (162 ページ)	ALL	すべての Log Reader アクティビティを直ちに停止し、コネクションを切断して、Replication Agent を Admin 状態にします。
test_connection (163 ページ)	ALL	Replication Agent の接続をテストします。
trace (165 ページ)	ALL	現在のトレース・フラグの設定を返します。また、指定したトレース・フラグを変更します。

## log\_system\_name

Replication Agent インスタンスのログ・ファイルのフル・パスを返します。

### 構文

```
log_system_name
```

### 使用法

- Replication Agent インスタンスを作成すると、ログ・ディレクトリがインスタンスのディレクトリ構造の一部として自動的に作成されます。**log\_directory** パラメータのデフォルト値は、そのディレクトリを指します。
- Microsoft Windows の Replication Agent ログ・ディレクトリのデフォルト・パスは

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

、Linux と UNIX の Replication Agent ログ・ディレクトリのデフォルト・パスは

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

です。

構文の説明は次のとおりです。

- **%SYBASE%** または **\$SYBASE** は Replication Agent インストール・ディレクトリです。

## コマンド・リファレンス

- *inst\_name* は Replication Agent インスタンスの名前です。
- **log\_directory** パラメータの値として有効なディレクトリ・パスを指定した場合、Replication Agent インスタンスは、そのシステム・ログ・ファイルを指定のディレクトリに配置します。  
**log\_directory** パラメータの値を **ra\_config** コマンドで変更した場合、新しい値は設定ファイルにすぐに記録されますが、Replication Agent インスタンスを停止および再開して、新しい値を有効にする必要があります。  
詳細については、**log\_directory** パラメータを参照してください。
- **log\_system\_name** コマンドが有効なのは、Replication Agent インスタンスが Admin 状態、Replicating 状態、または Replication Down 状態のときです。

### 参照：

- trace (165 ページ)
- ra\_config (86 ページ)

## lr\_dump\_marker

(Oracle のみ) system change number (SCN) を返すか設定し、最後にコミットされたトランザクションをプライマリ・データベース・ダンプに表示します。この値以上の SCN を持つログ・レコードが Replication Agent で検出されると、この値が Replication Server に送信されます。

---

**注意：** このコマンドは Oracle でのみ使用可能です。

---

### 構文

```
lr_dump_marker [scn]
```

### パラメータ

- **scn** – プライマリ・データベースが最後にダンプされたポイントを示す SCN。これは **dump database** マーカーで、ダンプの最も古いコミット済みトランザクションを示します。

### 使用法

- **lr\_dump\_marker** をオプションなしで呼び出すと、プライマリ・データベースが最後にダンプされたポイントをマークする SCN が返されます。SCN が指定されていない場合、**lr\_dump\_marker** は 0 を返します。
- SCN ダンプ・ポイントを設定するには、プライマリ・データベースが最後にダンプされたポイントをマークする有効な SCN を指定して **lr\_dump\_marker** を呼び出します。Oracle Recovery Manager (RMAN) ユーティリティでダンプを行っ

た場合は、RMAN の **list backup** コマンドを使用して、この SCN を取得できません。

- **lr\_dump\_marker** コマンドで設定された SCN は、Replication Agent が Replicating (再同期) 状態の場合に有効です。

## pdb\_capabilities

複写管理ツールで使用される Replication Agent 機能のリストを返します。

### 構文

```
pdb_capabilities
```

### 使用法

- **pdb\_capabilities** が呼び出されると、Replication Agent インスタンスの機能のリストを返します。
- **pdb\_capabilities** コマンドの目的は、複写管理ツールをサポートすることです。
- **pdb\_capabilities** コマンドが有効なのは、Replication Agent インスタンスが Admin 状態、Replicating 状態、または Replication Down 状態のときです。

## pdb\_date

プライマリ・データ・サーバの現在の日付と時間を返します。

### 構文

```
pdb_date
```

### 使用法

- **pdb\_date** が呼び出されると、プライマリ・データ・サーバから現在の日付と時刻を次の Sybase datetime データ型の形式で返します。

```
Current PDB Date
-----
Jan 11 2010 12:09:47.310
(1 row affected)
```

- **pdb\_date** コマンドが有効なのは、Replication Agent インスタンスが Admin 状態、Replicating 状態、または Replication Down 状態のときです。

### 参照：

- ra\_date (88 ページ)

## pdb\_execute\_sql

プライマリ・データ・サーバで現在のデータベースの SQL 文を実行します。

### 構文

```
pdb_execute_sql statement
```

### パラメータ

- **statement** – 二重引用符で囲まれた SQL 文の形式の文字列。

### 使用法

- Replication Agent インスタンスは、「現在の」データベースに対して指定した SQL 文を実行します。  
現在のデータベースは次のいずれかになります。
  - Replication Agent の **pds\_database\_name** 設定パラメータで指定されているプライマリ・データベースであるデフォルトの現在のデータベース、または
  - (Replication Agent インスタンスが現在接続されている) **pdb\_set\_sql\_database** コマンドで指定されたデータベース。
- 現在のデータベースを設定または変更するには、**pdb\_set\_sql\_database** コマンドを使用します。
- 現在のデータベースの名前を検出するには、**pdb\_get\_sql\_database** コマンドを使用します。

---

**注意：**現在のデータベースを設定または変更するために **pdb\_set\_sql\_database** コマンドが呼び出されない場合、**pdb\_get\_sql\_database** コマンドはデフォルトの現在のデータベースの名前を返します。

---

- **pdb\_execute\_sql** コマンドで指定された SQL 文は、二重引用符で囲まれた単一の SQL コマンドでなければなりません。例：

```
pdb_execute_sql "select * from Authors"
```

文字列は、実行のためにデータベースに直接渡されます。終了するためのコマンドは必要ありません。構文などの検証は実行されません。

- SQL 文の実行から返される結果はすべて、Replication Agent の管理ポートを介して、Replication Agent 管理クライアントに渡されます。
- **pdb\_execute\_sql** コマンドが有効なのは、Replication Agent インスタンスが Admin 状態、Replicating 状態、または Replication Down 状態のときです。

### 参照：

- [pdb\\_get\\_sql\\_database \(24 ページ\)](#)
- [pdb\\_set\\_sql\\_database \(30 ページ\)](#)



## pdb\_gen\_id

データベース世代 ID の現在値を返すか、またはデータベース世代 ID の値を更新します。

### 構文

```
pdb_gen_id [number]
```

### パラメータ

- **number** – データベース世代 ID を更新するときに使用する新しいデータベース世代 ID の値。0 ~ 32767 の番号でなければなりません。

### 例

- **例 1 –**

```
pdb_gen_id
```

このコマンドは、データベース世代 ID の現在値を返します。

- **例 2 –**

```
pdb_gen_id 10
```

このコマンドは、データベース世代 ID を値 **10** に更新します。

### 使用法

- **pdb\_gen\_id** をオプションなしで呼び出すと、Replication Agent のトランザクション・ログ・システム・テーブル (UDB) または RASD (Oracle または Microsoft SQL Server) に保管されているデータベース世代 ID の現在値が返されます。
- **pdb\_gen\_id** を **number** オプション指定で呼び出すと、Replication Agent のトランザクション・ログ・システム・テーブル (UDB) または RASD (Oracle または Microsoft SQL Server) に保管されているデータベース世代 ID の現在値が更新されます。データベース世代 ID の変更はすぐに有効になります。
- データベース世代 ID は、元のキュー ID の最初の 2 バイトです。データベース世代 ID は、トランザクションを再送信するときに Replication Agent に必要なリカバリ操作をサポートするために Replication Server で使用されます。リカバリ中、Replication Server で既に処理されたオペレーションを Replication Agent が再送信する必要がある場合、データベース世代 ID を変更して、そのオペレーションが既に処理済みであると Replication Server に認識されないようにすることができます。

## コマンド・リファレンス

- 元のキュー ID の詳細については、`ra_helplocator` を参照するか、または『Replication Agent プライマリ・データベース・ガイド』でご使用のプライマリ・データ・サーバに関するセクションを参照してください。
- Replication Agent のトランザクション・ログ (UDB) または RASD (Oracle または Microsoft SQL Server) が存在しない場合、`pdb_gen_id` コマンドはエラーを返します。
- パラメータ付きの `pdb_gen_id` コマンドが有効なのは、Replication Agent インスタンスが Admin 状態、Replicating 状態、または Replication Down 状態のときです。

### 参照：

- `ra_helplocator` (109 ページ)
- `ra_locator` (116 ページ)

## pdb\_get\_columns

プライマリ・データ・サーバにある現在のデータベースのテーブル内のカラムのリストを返します。

### 構文

```
pdb_get_columns [ownername, tablename[, colname]]
```

### パラメータ

- **ownername** – **tablename** オプションで指定されたテーブルの所有者のユーザ名。このオプションは、引用符で区切って大文字と小文字を指定できます。
- **tablename** – 情報が返される現在のデータベース内のテーブル名。このオプションは、引用符で区切って大文字と小文字を指定できます。
- **colname** – 情報が返されるカラムの名前。このオプションは、引用符で区切って大文字と小文字を指定できます。

### 例

- **例 1 –**

```
pdb_get_columns
```

このコマンドは、現在のデータベース内のすべてのユーザ・テーブルにあるすべてのカラムのリストを返します。

- **例 2 –**

```
pdb_get_columns bob, authors
```

このコマンドは、現在のデータベースのユーザ "bob" が所有する、テーブル *authors* のすべてのカラムのリストを返します。

• **例 3 –**

```
pdb_get_columns bob, authors, au_fname
```

このコマンドは、現在のデータベースのユーザ "bob" が所有する、テーブル *authors* のカラム *au\_fname* についての情報を返します。

## 使用法

---

**注意：** (Oracle と Microsoft SQL Server の場合) これらのコマンドの結果は、Replication Agent システム・データベース (RASD) から取得されます。(UDB の場合) これらのコマンドの結果は、プライマリ・データベースから直接取得されません。

---

- **pdb\_get\_columns** をオプションなしで呼び出すと、現在のデータベース内のすべてのユーザ・テーブルのすべてのカラムをリストする結果セットが返されます。
- **pdb\_get\_columns** を *ownername* オプションおよび *tablename* オプション指定で呼び出すと、現在のデータベース内の指定の所有者を持つ指定のテーブルのすべてのカラムをリストする結果セットが返されます。
- **pdb\_get\_columns** を *ownername* オプション、*tablename* オプション、および *colname* オプション指定で呼び出すと、現在のデータベース内の指定の所有者を持つ指定のテーブルの指定したカラムについての情報を持つ結果セットが返されます。
- **pdb\_get\_columns** コマンドは、*ownername* オプション、*tablename* オプション、および *colname* オプションでワイルドカード文字 (%) を受け入れます。
- 現在のデータベースは次のいずれかになります。
  - Replication Agent の **pds\_database\_name** 設定パラメータで指定されているプライマリ・データベースであるデフォルトの現在のデータベース、または
  - (Replication Agent インスタンスが現在接続されている) **pdb\_set\_sql\_database** コマンドで指定されたデータベース。
- 現在のデータベースを設定または変更するには、**pdb\_set\_sql\_database** コマンドを使用します。

---

**注意：** 現在のデータベースを設定または変更するために **pdb\_set\_sql\_database** コマンドが呼び出されない場合、**pdb\_get\_columns** コマンドは現在のデータベースの情報を返します。

---

- 現在のデータベースの名前を検出するには、**pdb\_get\_sql\_database** コマンドを使用します。

- (指定の所有者を持つ) 指定したテーブルが現在のデータベースに存在しない場合、または指定したカラムが指定のテーブルに存在しない場合、**pdb\_get\_columns** コマンドは 0 個のローを返します。
- **pdb\_get\_columns** コマンドが有効なのは、Replication Agent インスタンスが Admin 状態、Replicating 状態、または Replication Down 状態のときです。

### 参照：

- **pdb\_get\_databases** (18 ページ)
- **pdb\_get\_primary\_keys** (19 ページ)
- **pdb\_get\_procedure\_parms** (20 ページ)
- **pdb\_get\_procedures** (22 ページ)
- **pdb\_get\_tables** (24 ページ)

## **pdb\_get\_databases**

プライマリ・データ・サーバにあるすべてのユーザ・データベースのリストを返します。

---

**注意：** Oracle データ・サーバでは、複数のユーザ・データベースをサポートしていません。**pdb\_get\_databases** コマンドは、データベース・インスタンスの名前を返します。

---

### 構文

```
pdb_get_databases
```

### 使用法

- **pdb\_get\_databases** を呼び出すと、プライマリ・データ・サーバ内のすべてのユーザ・データベースをリストする結果セットが返されます。

---

**注意：** システム・データベースのタイプによって、プライマリ・データ・サーバ内のユーザ・データベースが結果セットに含まれる場合と含まれない場合があります。『Replication Agent プライマリ・データベース・ガイド』の特定のプライマリ・データ・サーバに関するセクションを参照してください。

---

- **pdb\_get\_databases** コマンドが有効なのは、Replication Agent インスタンスが Admin 状態、Replicating 状態、または Replication Down 状態のときです。

### 参照：

- **pdb\_get\_columns** (16 ページ)
- **pdb\_get\_primary\_keys** (19 ページ)
- **pdb\_get\_procedure\_parms** (20 ページ)
- **pdb\_get\_procedures** (22 ページ)

- `pdb_get_tables` (24 ページ)

## pdb\_get\_primary\_keys

プライマリ・データ・サーバにある現在のデータベースの指定したテーブル内のプライマリ・キー・カラムのリストを返します。

### 構文

```
pdb_get_primary_keys ownername, tablename
```

### パラメータ

- **ownername** – *tablename* で指定されたテーブルの所有者のユーザ名。このオプションは、引用符で区切って大文字と小文字を指定できます。
- **tablename** – プライマリ・キー・カラムの情報が返される現在のデータベース内のテーブル名。このオプションは、引用符で区切って大文字と小文字を指定できます。

### 使用法

---

**注意：** (Oracle と Microsoft SQL Server の場合) これらのコマンドの結果は、Replication Agent システム・データベース (RASD) から取得されます。(UDB の場合) これらのコマンドの結果は、プライマリ・データベースから直接取得されません。

---

- **pdb\_get\_primary\_keys** を呼び出すと、現在のデータベース内の指定の所有者を持つ指定したテーブルでプライマリ・キーとして定義されたすべてのカラムをリストする結果セットが返されます。
- **pdb\_get\_primary\_keys** コマンドは、*ownername* オプションでワイルドカード文字 (%) を受け入れますが、*tablename* オプションでは受け入れません。
- 現在のデータベースは次のいずれかになります。
  - Replication Agent の **pds\_database\_name** 設定パラメータで指定されているプライマリ・データベースであるデフォルトの現在のデータベース、または
  - Replication Agent インスタンスが現在接続されている、**pdb\_set\_sql\_database** コマンドで指定されたデータベース。(これは Oracle では有効ではありません。)
- 現在のデータベースを設定または変更するには、**pdb\_set\_sql\_database** コマンドを使用します。

---

**注意：** Oracle では、現在のデータベースを変更できません。

---

- 現在のデータベースの名前を検出するには、**pdb\_get\_sql\_database** コマンドを使用します。

- 指定の所有者を持つ指定したテーブルが現在のデータベースに存在しない場合、**pdb\_get\_primary\_keys** コマンドは 0 個のローを返します。
- **pdb\_get\_primary\_keys** コマンドが有効なのは、Replication Agent インスタンスが Admin 状態、Replicating 状態、または Replication Down 状態のときです。

### 参照：

- `pdb_get_columns` (16 ページ)
- `pdb_get_databases` (18 ページ)
- `pdb_get_procedure_parms` (20 ページ)
- `pdb_get_procedures` (22 ページ)
- `pdb_get_tables` (24 ページ)

## pdb\_get\_procedure\_parms

(Oracle と Microsoft SQL Server のみ) プライマリ・データ・サーバにある現在のデータベース内のプロシージャの入力パラメータのリストを返します。

---

**注意：** このコマンドは Oracle および Microsoft SQL Server でのみ使用可能です。

---

### 構文

```
pdb_get_procedure_parms [ownername, procname [, paramname]]
```

### パラメータ

- **ownername** – *procname* で指定されたプロシージャの所有者のユーザ名。このオプションは、引用符で区切って大文字と小文字を指定できます。
- **procname** – 情報が返される現在のデータベース内のプロシージャ名。このオプションは、引用符で区切って大文字と小文字を指定できます。
- **paramname** – 情報が返される入力パラメータの名前。このオプションは、引用符で区切って大文字と小文字を指定できます。

### 例

- **例 1 –**

```
pdb_get_procedure_parms
```

このコマンドは、現在のデータベース内のすべてのプロシージャに対するすべての入力パラメータのリストを返します。

- **例 2 –**

```
pdb_get_procedure_parms bob, sp_foo
```

このコマンドは、現在のデータベースのユーザ "bob" が所有する、*sp\_foo* という名前のプロシージャのすべての入力パラメータのリストを返します。

• **例 3 –**

```
pdb_get_procedure_parms bob, sp_foo, foo_count
```

このコマンドは、現在のデータベースのユーザ "bob" が所有する、プロシージャ *sp\_foo* の入力パラメータ *foo\_count* に関する情報を返します。

## 使用法

---

**注意：** (Oracle と Microsoft SQL Server の場合) これらのコマンドの結果は、Replication Agent システム・データベース (RASD) から取得されます。(UDB の場合) これらのコマンドの結果は、プライマリ・データベースから直接取得されません。

---

- **pdb\_get\_procedure\_parms** をオプションなしで呼び出すと、現在のデータベース内のすべてのプロシージャのすべての入力パラメータをリストする結果セットが返されます。
- **pdb\_get\_procedure\_parms** を *ownername* オプションおよび *procname* オプション指定で呼び出すと、現在のデータベース内の指定の所有者を持つ指定のプロシージャのすべての入力パラメータをリストする結果セットが返されます。
- **pdb\_get\_procedure\_parms** を *ownername* オプション、*procname* オプション、および *paramname* オプション指定で呼び出すと、現在のデータベース内の指定の所有者を持つ指定のプロシージャの指定した入力パラメータに関する情報を持つ結果セットが返されます。
- **pdb\_get\_procedure\_parms** コマンドは、*ownername* オプションと *procname* オプションでワイルドカード文字 (%) を受け入れます。
- 現在のデータベースは次のいずれかになります。
  - Replication Agent の **pds\_database\_name** 設定パラメータで指定されているプライマリ・データベースであるデフォルトの現在のデータベース、または
  - Replication Agent インスタンスが現在接続されている、**pdb\_set\_sql\_database** コマンドで指定されたデータベース。(これは Oracle では有効ではありません。)
- 現在のデータベースを設定または変更するには、**pdb\_set\_sql\_database** コマンドを使用します。

---

**注意：** Oracle では、現在のデータベースを変更できません。

---

- 現在のデータベースの名前を検出するには、**pdb\_get\_sql\_database** コマンドを使用します。
- 指定の所有者を持つ指定したプロシージャが現在のデータベースに存在しない場合、**pdb\_get\_procedure\_parms** コマンドは 0 個のローを返します。

- **pdb\_get\_procedure\_parms** コマンドが有効なのは、Replication Agent インスタンスが Admin 状態、Replicating 状態、または Replication Down 状態のときです。

**参照：**

- `pdb_get_columns` (16 ページ)
- `pdb_get_databases` (18 ページ)
- `pdb_get_primary_keys` (19 ページ)
- `pdb_get_procedures` (22 ページ)
- `pdb_get_tables` (24 ページ)

## **pdb\_get\_procedures**

(Oracle と Microsoft SQL Server のみ) プライマリ・データ・サーバにある現在のデータベース内のプロシージャのリストを返します。

---

**注意：** このコマンドは Oracle および Microsoft SQL Server でのみ使用可能です。

---

### **構文**

```
pdb_get_procedures [ownername, procname]
```

### **パラメータ**

- **ownername** – *procname* で指定されたプロシージャの所有者のユーザ名。このオプションは、引用符で区切って大文字と小文字を指定できます。
- **procname** – 情報が返される現在のデータベース内のプロシージャ名。このオプションは、引用符で区切って大文字と小文字を指定できます。

### **例**

- **例 1 –**

```
pdb_get_procedures
```

このコマンドは、現在のデータベース内のすべてのプロシージャのリストを返します。

- **例 2 –**

```
pdb_get_procedures bob, sp_foo
```

このコマンドは、現在のデータベースのユーザ "bob" が所有する、*sp\_foo* という名前前のプロシージャに関する情報を返します。



## 使用法

---

**注意：** (Oracle と Microsoft SQL Server の場合) これらのコマンドの結果は、Replication Agent システム・データベース (RASD) から取得されます。(UDB の場合) これらのコマンドの結果は、プライマリ・データベースから直接取得されません。

---

- **pdb\_get\_procedures** をオプションなしで呼び出すと、現在のデータベース内のすべてのプロシージャをリストする結果セットが返されます。
  - **pdb\_get\_procedures** を *ownername* オプションおよび *procname* オプション指定で呼び出すと、現在のデータベース内の指定の所有者を持つ指定したプロシージャに関する情報を持つ結果セットが返されます。
  - **pdb\_get\_procedures** コマンドは、*ownername* オプションと *procname* オプションでワイルドカード文字 (%) を受け入れます。
  - 現在のデータベースは次のいずれかになります。
    - Replication Agent の **pds\_database\_name** 設定パラメータで指定されているプライマリ・データベースであるデフォルトの現在のデータベース、または
    - Replication Agent インスタンスが現在接続されている、**pdb\_set\_sql\_database** コマンドで指定されたデータベース。(これは Oracle では有効ではありません。)
  - 現在のデータベースを設定または変更するには、**pdb\_set\_sql\_database** コマンドを使用します。
- 

**注意：** Oracle では、現在のデータベースを変更できません。

---

- 現在のデータベースの名前を検出するには、**pdb\_get\_sql\_database** コマンドを使用します。
- 指定の所有者を持つ指定したプロシージャが現在のデータベースに存在しない場合、**pdb\_get\_procedures** コマンドは 0 個のローを返します。
- **pdb\_get\_procedures** コマンドが有効なのは、Replication Agent インスタンスが Admin 状態、Replicating 状態、または Replication Down 状態のときです。

### 参照：

- **pdb\_get\_columns** (16 ページ)
- **pdb\_get\_databases** (18 ページ)
- **pdb\_get\_primary\_keys** (19 ページ)
- **pdb\_get\_procedure\_parms** (20 ページ)
- **pdb\_get\_tables** (24 ページ)

## pdb\_get\_sql\_database

現在のデータベースの名前を返します (存在する場合)。

### 構文

```
pdb_get_sql_database
```

### 使用法

- **pdb\_get\_sql\_database** を呼び出すと、現在のデータベースの名前が返されます。
- 現在のデータベースを設定するために **pdb\_set\_sql\_database** コマンドが呼び出されない場合、デフォルトの現在のデータベースが返されます。
- 現在のデータベースは次のいずれかになります。
  - Replication Agent の **pds\_database\_name** 設定パラメータで指定されているプライマリ・データベースであるデフォルトの現在のデータベース、または
  - Replication Agent インスタンスが現在接続されている、コマンドで指定されたデータベース。(これは Oracle では有効ではありません。)
- 現在のデータベースを設定または変更するには、**pdb\_set\_sql\_database** コマンドを使用します。

---

**注意：** Oracle では、現在のデータベースを変更できません。

- **pdb\_get\_sql\_database** コマンドが有効なのは、Replication Agent インスタンスが Admin 状態、Replicating 状態、または Replication Down 状態のときです。

### 参照：

- `pdb_execute_sql` (14 ページ)
- `pdb_set_sql_database` (30 ページ)

## pdb\_get\_tables

プライマリ・データ・サーバにある現在のデータベース内のユーザ・テーブルのリストを返します。

### 構文

```
pdb_get_tables [ownername, tablename]
```

### パラメータ

- **ownername** – *tablename* で指定されたテーブルの所有者のユーザ名。このオプションは、引用符で区切って大文字と小文字を指定できます。

- **tablename** – 情報が返される現在のデータベース内のテーブル名。このオプションは、引用符で区切って大文字と小文字を指定できます。

## 例

### 例 1 –

```
pdb_get_tables
```

このコマンドは、現在のデータベース内のすべてのユーザ・テーブルのリストを返します。

### 例 2 –

```
pdb_get_tables bob, authors
```

このコマンドは、現在のデータベースのユーザ "bob" が所有する、テーブル *authors* に関する情報を返します。

## 使用法

---

**注意：** (Oracle と Microsoft SQL Server の場合) これらのコマンドの結果は、Replication Agent システム・データベース (RASD) から取得されます。(UDB の場合) これらのコマンドの結果は、プライマリ・データベースから直接取得されません。

---

- **pdb\_get\_tables** をオプションなしで呼び出すと、現在のデータベース内のすべてのユーザ・テーブルをリストする結果セットが返されます。

---

**注意：** **pdb\_get\_tables** コマンドを呼び出すと、システム・テーブルは、いくつかのプライマリ・データ・サーバによって返される場合と返されない場合があります。

---

- **pdb\_get\_tables** を *ownername* オプションおよび *tablename* オプション指定で呼び出すと、現在のデータベース内の指定の所有者を持つ指定したテーブルに関する情報を持つ結果セットが返されます。
- **pdb\_get\_tables** コマンドは、*ownername* オプションと *tablename* オプションでワイルドカード文字 (%) を受け入れます。
- 現在のデータベースは次のいずれかになります。
  - Replication Agent の **pds\_database\_name** 設定パラメータで指定されているプライマリ・データベースであるデフォルトの現在のデータベース、または
  - Replication Agent インスタンスが現在接続されている、**pdb\_set\_sql\_database** コマンドで指定されたデータベース。(これは Oracle では有効ではありません。)
- 現在のデータベースを設定または変更するには、**pdb\_set\_sql\_database** コマンドを使用します。

---

**注意：** Oracle では、現在のデータベースを変更できません。

---

- 現在のデータベースの名前を検出するには、**pdb\_get\_sql\_database** コマンドを使用します。
- (指定の所有者を持つ) 指定したテーブルが現在のデータベースに存在しない場合、**pdb\_get\_tables** コマンドは 0 個のローを返します。
- **pdb\_get\_tables** コマンドが有効なのは、Replication Agent インスタンスが Admin 状態、Replicating 状態、または Replication Down 状態のときです。

**参照：**

- **pdb\_get\_columns** (16 ページ)
- **pdb\_get\_databases** (18 ページ)
- **pdb\_get\_primary\_keys** (19 ページ)
- **pdb\_get\_procedure\_parms** (20 ページ)
- **pdb\_get\_procedures** (22 ページ)

## **pdb\_ownerfilter**

(Oracle と Microsoft SQL Server のみ) オブジェクトが初期化でフィルタされる所有者のリストを返します。また、所有者をリストに追加またはリストから削除します。

---

**注意：** このコマンドは Oracle および Microsoft SQL Server でのみ使用可能です。

---

### **構文**

```
pdb_ownerfilter [ {add | remove}, owner ]
```

### **パラメータ**

- **add – add** キーワードは、指定した所有者が所有するすべてのオブジェクトを除外します。この所有者が所有するすべてのオブジェクトは、初期化のためにマーク付けできません。
- **remove – remove** キーワードは、指定した所有者のフィルタを削除します。この所有者が所有するすべてのオブジェクトは、初期化のためにマーク付けできます。Oracle では、"SYS" の所有者を削除できません。
- **owner** – フィルタリングに使用する所有者の名前。

*owner* オプションは、引用符で区切って大文字と小文字を指定できます。

大文字と小文字を混在させる必要がある場合は、名前を区切ります。このパラメータは、引用符で区切って大文字と小文字を指定できます。例：

```
"Owner", "oWnEr"
```

## 例

### • 例 1 –

```
pdb_ownerfilter
```

このコマンドは、初期化のためにオブジェクトが除外されるすべての所有者のリストを返します。

### • 例 2 –

```
pdb_ownerfilter add, SYSTEM
```

このコマンドは、複製のためにオブジェクトが除外される所有者のリストに "system" ユーザを追加します。

### • 例 3 –

```
pdb_ownerfilter remove, SYSTEM
```

このコマンドは、初期化のためにオブジェクトが除外される所有者のリストから "system" ユーザを削除します。

## 使用法

- **pdb\_ownerfilter** を使用すると、初期化中に Replication Agent システム・データベースに読み込まれるオブジェクトの数を制限できます (**pdb\_xlog init** を参照)。**pdb\_xlog init** が処理されると、**pdb\_ownerfilter** リスト内のオブジェクトと所有者は読み込まれません。RASD のサイズを縮小したり時間を短縮したりして、オブジェクトが複製されていない所有者または大部分のオブジェクトが複製されない所有者をリストに追加することにより初期化を実行できます。

---

**注意：**所有者がリストになくても、複製 (コマンド **pdb\_setreptable**、**pdb\_setrepproc** を使用) のためにマーク付けされたすべてのオブジェクトは RASD に読み込まれます。このリストは、初期化処理に影響しますが、複製には影響しません (複製の実行は、所有者のフィルタリングではなくマーク付けステータスに基づいています)。

---

- **pdb\_ownerfilter** が呼び出されると、指定したキーワードとオプションに従って機能が決まります。
- 複数のキーワードとオプションを指定する場合は、それぞれカンマで区切る必要があります。カンマの前後にはオプションで空白を挿入できます。例：  

```
pdb_ownerfilter add, system
```
- **pdb\_ownerfilter** をキーワードなしで呼び出すと、オブジェクトが除外されるユーザのリストが返されます。
- **pdb\_ownerfilter** コマンドが有効なのは、Replication Agent インスタンスが Admin 状態または Replication Down 状態のときだけです。

## コマンド・リファレンス

- Oracle では、"SYS" の所有者を削除できません。
- Oracle の場合、初期化後には **pdb\_setreptable** と **pdb\_setrepproc** を使用してどのオブジェクトでも複写できます。ただし、次のオブジェクトは例外で、いつでも複写できるわけではありません。
  - "SYS" の所有者に所有されるオブジェクト。
  - 名前が VS\$ で始まるすべてのシステム・テーブル。
  - 名前が DBMS で始まるすべてのシステム・プロシージャまたはパッケージ。

### 参照：

- **pdb\_setrepproc** (45 ページ)
- **pdb\_setreptable** (56 ページ)
- **pdb\_xlog** (75 ページ)
- **ra\_admin** (81 ページ)
- **ra\_config** (86 ページ)

## **pdb\_send\_osuser\_list**

(Oracle のみ) データベース・ユーザのリストを返したり修正したりします。このリストにあるプライマリ・データベースのユーザ名が、Replication Server に送信された LTL 内の対応するオペレーティング・システムのユーザ名と置き換えられます。

---

**注意：** このコマンドは Oracle でのみ使用可能です。

---

### 構文

```
pdb_send_osuser_list [ { add | remove }, { user | all } ]
```

### パラメータ

- **add – add** キーワードを使用して、プライマリ・データベースのユーザ名をユーザのリストに追加します。このリストにあるプライマリ・データベースのユーザ名が、Replication Server に送信された LTL 内のオペレーティング・システムのユーザ名と置き換えられます。ユーザ名を 1 つ追加するには、次のように **add** キーワードの後に **user** パラメータを続けます。

```
pdb_send_osuser_list add, user
```

有効なすべてのプライマリ・データベースのユーザ名をリストに追加するには、次のように **add** キーワードの後に **all** キーワードを続けます。

```
pdb_send_osuser_list add, all
```

- **remove – remove** キーワードを使用して、プライマリ・データベースのユーザ名をユーザのリストから削除します。このリストにあるプライマリ・データ

ベースのユーザ名が、Replication Server に送信された LTL 内のオペレーティング・システムのユーザ名と置き換えられます。ユーザ名を 1 つ削除するには、次のように **remove** キーワードの後に *user* パラメータを続けます。

```
pdb_send_osuser_list remove, user
```

すべてのユーザ名をリストから削除するには、次のように **remove** キーワードの後に **all** キーワードを続けます。

```
pdb_send_osuser_list remove, all
```

ユーザのリストにあるすべてのユーザ名を表示するには、次のように **pdb\_send\_osuser\_list** コマンドのみを使用します。このリストにあるプライマリ・データベースのユーザ名が、Replication Server に送信された LTL 内のオペレーティング・システムのユーザ名に置き換えられます。

**pdb\_send\_osuser\_list**

## 例

### • 例 1 –

```
pdb_send_osuser_list add, dbuser1
```

このコマンドは、プライマリ・データベースのユーザ名 *dbuser1* をリストに追加します。*dbuser1* に対応するオペレーティング・システムのユーザ名が *osuser1* である場合、Replication Agent が Replication Server に送信する LTL には、ユーザ名 *dbuser1* ではなく *osuser1* が含まれます。

### • 例 2 –

```
pdb_send_osuser_list add, all
```

このコマンドは、有効なすべてのプライマリ・データベースのユーザ名をリストに追加します。Replication Agent が Replication Server に送信する LTL には、リスト内の有効なすべてのプライマリ・データベースのユーザ名に対応するオペレーティング・システムのユーザ名が含まれます。

### • 例 3 –

```
pdb_send_osuser_list remove, dbuser1
```

このコマンドは、プライマリ・データベースのユーザ名 *dbuser1* をリストから削除します。*dbuser1* に対応するオペレーティング・システムのユーザ名が *osuser1* である場合、Replication Agent が Replication Server に送信する LTL には、プライマリ・データベースのユーザ名 *osuser1* ではなく *dbuser1* が含まれます。

### • 例 4 –

```
pdb_send_osuser_list remove, all
```

このコマンドは、すべてのプライマリ・データベースのユーザ名をリストから削除します。Replication Agent が Replication Server に送信する LTL には、オペレーティング・システムのユーザ名ではなく、プライマリ・データベースのユーザ名が含まれます。

### • 例 5 –

```
pdb_send_osuser_list
```

このコマンドは、すべてのデータベース・ユーザをリストします。このリスト内にあるプライマリ・データベースのユーザ名が、Replication Server に送信された LTL 内の対応するオペレーティング・システムのユーザ名と置き換えられます。

### 使用法

- パラメータ付きの **pdb\_send\_osuser\_list** コマンドが有効なのは、Replication Agent インスタンスが Admin 状態、Replicating 状態、または Replication Down 状態のときのみです。
- パラメータ付きの **pdb\_send\_osuser\_list** コマンドは、**pdb\_send\_osuser\_list** を呼び出したときにプライマリ・データベースにログインしているデータベース・ユーザにのみ影響します。

## pdb\_set\_sql\_database

現在のデータベースを SQL 文の実行に使用されるように設定します。

### 構文

```
pdb_set_sql_database database
```

### パラメータ

- **database** – Replication Agent が SQL 文 (クエリ) を実行できるプライマリ・データ・サーバ内のデータベースの名前。大文字と小文字を指定するには、このパラメータを引用符で区切ります。

### 使用法

- **pdb\_set\_sql\_database** を呼び出すと、Replication Agent が SQL クエリを実行できる「現在の」データベースが設定されます。

---

**注意：** **pdb\_set\_sql\_database** コマンドは、Oracle では有効ではありませんが、複数のデータベースを持つデータベース・サーバをサポートするその他の Replication Agent との接続を提供するために含まれています。

---

- Replication Agent は、**pdb\_set\_sql\_database** で指定したデータベース名を確定化しません。



無効なデータベース名を指定すると、次の Replication Agent コマンドのいずれかが呼び出されるまでエラーは返されません。

- **pdb\_execute\_sql**
- **pdb\_get\_columns**
- **pdb\_get\_primary\_keys**
- **pdb\_get\_procedure\_parms**
- **pdb\_get\_procedures**
- **pdb\_get\_tables**
- 現在のデータベースの名前を検出するには、**pdb\_get\_sql\_database** を使用します。

---

**注意：**現在のデータベースを設定するために **pdb\_set\_sql\_database** コマンドが呼び出されない場合、**pdb\_get\_sql\_database** コマンドは、Replication Agent の **pds\_database\_name** 設定パラメータで指定したプライマリ・データベースであるデフォルトの現在のデータベースを返します。

---

- **pdb\_set\_sql\_database** コマンドが有効なのは、Replication Agent インスタンスが Admin 状態、Replicating 状態、または Replication Down 状態のときです。

#### 参照：

- **pdb\_execute\_sql** (14 ページ)
- **pdb\_get\_sql\_database** (24 ページ)

## pdb\_setrepcol

LOB カラムの複写ステータスを返します。また、マーク付けしたテーブル内の LOB カラムの複写を有効または無効にします。

### 構文

```

pdb_setrepcol [
  {
    enable
  |
  disable
  |
  tablename[, colname[, { enable | disable[, force] } ] ]
  |
  all, { enable | disable[, force] }
  }
]

```

### パラメータ

- **tablename – colname** オプションで指定したカラムを含むプライマリ・データベース内のユーザ・テーブルの名前。

*tablename* オプションは、それぞれの要素がピリオドで区切られた所有者指定 (所有者名を含む) の場合もあります。例：

```
owner.table
```

*tablename* オプションは、引用符で区切って大文字と小文字を指定できます。

大文字と小文字を混在させる必要がある場合は、名前を区切ります。例：

```
"Owner".table
```

```
"Owner"."Table"
```

大文字と小文字を区別しない *tablename* オプションの各要素は、前の例のように区切る必要があります。

---

**注意：** *ltl\_character\_case* パラメータの値と一致しない大文字と小文字をオブジェクト名で使用する必要がある場合、オブジェクト名を区切る必要があります。

---

オブジェクト名にスペースやピリオドなどの英数字以外の文字が含まれる場合、引用符で区切る必要があります。例：

```
"table name"
```

```
owner."table name"
```

オブジェクト名にピリオドが含まれる場合、所有者指定と引用符による区切りの両方を行う必要があります。例：

```
owner."table.name"
```

```
"table.owner"."table.name"
```

- **colname** – *tablename* オプションで指定されたユーザ・テーブル内の LOB カラム名。

*colname* オプションは、引用符で区切って大文字と小文字を指定できます。

大文字と小文字を混在させる必要がある場合は、名前を区切ります。例：

```
"Colname"
```

```
"COlName"
```

---

**注意：** *ltl\_character\_case* パラメータの値と一致しない大文字と小文字をカラム名で使用する必要がある場合、カラム名を区切る必要があります。詳細については、*ltl\_character\_case* を参照してください。

---

- **all** – 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すべての LOB カラムを表すキーワード。 **all** キーワードを使用することにより、有効化または無効化のオペレーションを、マーク付けされたテーブル内のすべての LOB カラムに適用できます。
- **enable** – LOB カラムの複製の有効化を表すキーワード。
- **disable** – LOB カラムの複製の無効化を表すキーワード。

- **force** – LOB カラムで複写を強制的に無効化することを表すキーワード。

**disable** キーワードの後に **force** キーワードを続けると、**pdb\_setrepcol** コマンドは指定した LOB カラムの複写をすぐに無効にします。**disable** キーワードと **all** キーワードの後に **force** キーワードを続けると、**pdb\_setrepcol** コマンドは、プライマリ・データベースにあるマーク付けされたすべての LOB カラムの複写をすぐに無効にします。

## 例

- 例 1 –

```
pdb_setrepcol
```

このコマンドは、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有効なすべての LOB カラムについて、複写情報を返します。

- 例 2 –

```
pdb_setrepcol authors
```

このコマンドは、プライマリ・データベース内の *authors* という名前のテーブルに対して定義されたすべての LOB カラムについて、複写情報を返します。

- 例 3 –

```
pdb_setrepcol authors, picture
```

このコマンドは、プライマリ・データベース内のテーブル *authors* の "picture" というカラムの複写情報を返します。

- 例 4 –

```
pdb_setrepcol authors, picture, enable
```

このコマンドは、プライマリ・データベース内のテーブル *authors* のカラム *picture* について、複写を有効にします。

- 例 5 –

```
pdb_setrepcol all, disable
```

このコマンドは、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すべての LOB カラムについて、複写を無効にします。

## 使用法

- カラムの名前を変更したり、カラムを削除して新しいカラムを元の名前で作成したりする場合は、この新しいカラムからの複写を明示的に有効または無効にする必要があります。これは、新しいカラムには、元のカラムからの複写ス

データ関連情報がないためです。マーク付け情報は、カラム名ではなく、カラム番号に基づいて内部で保持されます。

- **pdb\_setrepcol** が呼び出されると、指定したキーワードとオプションに従って機能が決まります。
- 複数のキーワードまたはオプションを指定する場合は、それぞれカンマで区切る必要があります。カンマの前後にはオプションで空白を挿入できます。例：  
`pdb_setrepcol all, disable`
- **pdb\_setrepcol** コマンドでカラム名を指定する場合、有効な LOB カラムの名前を使用する必要があります。
- **pdb\_setrepcol** コマンドでは、次のような項目をテーブル名として指定できません。
  - プライマリ・データベースのシステム・テーブル
  - エイリアスまたは同義語
  - ビュー
  - Replication Agent トランザクション・ログ・オブジェクト
- プライマリ・データベース内のカラム名がキーワードと同じである場合、文字列 **col=** をカラム名の先頭に追加することによって識別できます。例：  
`pdb_setrepcol tablename, col=enable, disable`
- **pdb\_setrepcol** コマンドで LOB カラムの複写を有効にする場合、プライマリ・データベース内の **date** データ型または **time** データ型を変換するように Replication Agent を構成しないでください。
- オプションを指定しないで、または 1 つ指定して **pdb\_setrepcol** を呼び出すと、プライマリ・データベース内の LOB カラムの有効なステータスに関する情報が返されます。
  - **pdb\_setrepcol** をオプションなしで呼び出すと、プライマリ・データベースで複写が有効になっているすべての LOB カラムのリストが返されます。

---

**注意：** **pdb\_setrepcol** コマンドをオプションなしで呼び出すと、**enable** キーワードを指定して **pdb\_setrepcol** コマンドを呼び出した場合と同じ結果が得られます。

---

- **pdb\_setrepcol** をテーブル名を指定して呼び出すと、指定したプライマリ・テーブル内のすべての LOB カラムの有効なステータスに関する情報が返されます。
- **pdb\_setrepcol** を **enable** キーワード指定で呼び出すと、プライマリ・データベースで複写が有効になっているすべての LOB カラムのリストが返されます。
- **pdb\_setrepcol** を **disable** キーワード指定で呼び出すと、プライマリ・データベースで複写が無効になっているすべての LOB カラムのリストが返されます。

無効としてリストされている LOB カラムでは、トランザクションは複写用に  
取り出されません。

- **pdb\_setrepcol** を、有効なプライマリ・テーブル名と有効な LOB カラム名を指定し、キーワードを指定しないで呼び出すと、プライマリ・データベース内の指定したテーブルの指定した LOB カラムの有効なステータスに関する情報が返されます。
- **pdb\_setrepcol** を **all** キーワード指定で呼び出すと、次のキーワード (**enable** または **disable**) で指定された操作がプライマリ・データベース内のマーク付けされたテーブルのすべての LOB カラムに適用されます。
  - **pdb\_setrepcol** を **all** キーワードおよび **enable** キーワード指定で呼び出すと、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すべての LOB カラムについて、複写が有効になります。
  - **pdb\_setrepcol** を **all** キーワードおよび **disable** キーワード指定で呼び出すと、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すべての LOB カラムについて、複写が無効になります。
- **pdb\_setrepcol** を、有効なプライマリ・テーブル名と有効な LOB カラム名の後に 1 つまたは複数のキーワードを指定して呼び出すと、キーワード (**enable** または **disable**) で指定された操作は、指定したプライマリ・テーブル内の指定した LOB カラムに適用されます。
  - **pdb\_setrepcol** を、テーブル名と LOB カラム名に加えて **enable** キーワードを指定して呼び出すと、プライマリ・データベース内の指定した LOB カラムについて、複写が有効になります。
  - **pdb\_setrepcol** を、テーブル名と LOB カラム名に加えて **disable** キーワードを指定して呼び出すと、プライマリ・データベース内の指定した LOB カラムについて、複写が無効になります。

指定したテーブル名と LOB カラム名の組み合わせがプライマリ・データベースに存在しない場合、**pdb\_setrepcol** コマンドはエラーを返します。

- Replication Agent トランザクション・ログがプライマリ・データベース (UDB) に存在しない場合、または RASD が初期化されない場合 (Oracle または Microsoft SQL Server)、**pdb\_setrepcol** コマンドはエラーを返します。
- "DATE" カラムを含んでいるテーブルを指定して **pdb\_setrepcol** を呼び出す場合、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。これは Replication Agent for UDB に該当しますが、Replication Agent for Oracle や Microsoft SQL Server には該当しません。

#### 参照：

- **pdb\_setrepproc** (45 ページ)
- **pdb\_setreptable** (56 ページ)
- **ra\_config** (86 ページ)

- `ltl_character_case` (201 ページ)

## **pdb\_setrepddl**

(Oracle と Microsoft SQL Server のみ) DDL 複写ステータスを返し、DDL 文の複写を有効または無効にします。

---

**注意：** このコマンドは Oracle および Microsoft SQL Server でのみ使用可能です。

---

### **構文**

Replication Agent for Oracle の場合

```
pdb_setrepddl [  
  {  
    tablename  
    | procname  
    | sequence_name  
    | objects, all  
    | user, { all | user } ]  
  | stmt, { all | ddl_statement | ddl_statement_keyword }  
  | owner, { all | ownername }  
  } ]  
  [  
    { enable[, override] | disable[, override] | default }  
    | { enable, { all | marked | unmarked } }  
  ]  
]
```

Replication Agent for Microsoft SQL Server の場合

```
pdb_setrepddl [enable|disable]
```

### **パラメータ**

- **override (Oracle のみ)** – DDL 文の複写を有効にし、既存のフィルタリング・ルールをオーバーライドするには、**pdb\_setrepddl** コマンドの後ろに **enable** キーワードと **override** キーワードを指定します。

```
pdb_setrepddl enable, override
```

DDL 文の複写を無効にし、既存のフィルタリング・ルールをオーバーライドするには、**pdb\_setrepddl** コマンドの後ろに **disable** キーワードと **override** キーワードを指定します。

```
pdb_setrepddl enable, override
```

- **tablename (Oracle のみ)** – プライマリ・データベース内のユーザ・テーブルの名前です。テーブルに関連した DDL の複写を有効または無効にするには、**tablename** パラメータを使用します。

```
pdb_setrepddl tablename [ , { enable[ , override] | disable |  
default } ]
```

テーブルの現在のフィルタ設定をリストするには、*tablename* パラメータのみを入力します。

```
pdb_setrepddl tablename
```

- **procname (Oracle のみ)** – プライマリ・データベース内のプロシージャの名前です。プロシージャ名に関連した DDL の複写を有効または無効にするには、*procname* パラメータを使用します。

```
pdb_setrepddl procname [ , { enable[, override] | disable | default } ]
```

プロシージャの現在のフィルタ設定をリストするには、*procname* パラメータのみを入力します。

```
pdb_setrepddl procname
```

- **sequence\_name (Oracle のみ)** – プライマリ・データベース内のユーザ・シーケンスの名前です。シーケンスに関連した DDL の複写を有効または無効にするには、*sequence\_name* パラメータを使用します。

```
pdb_setrepddl sequence_name [ , { enable[, override] | disable | default } ]
```

シーケンスの現在のフィルタ設定をリストするには、*sequence\_name* パラメータのみを入力します。

```
pdb_setrepddl sequence_name
```

- **objects, all (Oracle のみ)** – **objects** キーワードは、**all** キーワードとともに使用する必要があります。次のようなすべてのオブジェクトの DDL 文の複写を有効または無効にできます。

```
pdb_setrepddl objects, all [ , { enable[, override] | disable | default } ]
```

フィルタされた DDL 文のすべてのオブジェクトをリストするには、

**pdb\_setrepddl** コマンドの後に **objects, all** キーワードを続けます。

```
pdb_setrepddl objects, all
```

- **user (Oracle のみ)** – **user** キーワードを使用すると、プライマリ・データベース・ユーザが実行する DDL 文の複写を有効または無効にできます。指定されたユーザからの DDL の複写を有効または無効にするには、*user* パラメータを使用します。

```
pdb_setrepddl user, user [ , { enable[, override] | disable | default } ]
```

ユーザの現在のフィルタ設定をリストするには、次のように **user** キーワードの後に *user* パラメータを続けます。

```
pdb_setrepddl user, user
```

複写からフィルタされる DDL 文のデータベース・ユーザをリストするには、次のように **user** キーワードのみを入力します。

```
pdb_setrepddl user
```

すべてのユーザに対して DDL 文の複写を有効または無効にするには、次のように **user** キーワードの後に **all** キーワードを続けます。

```
pdb_setrepddl user, all [ , { enable[, override] | disable | default } ]
```

- stmt (Oracle のみ) – stmt** キーワードを使用すると、DDL 文の複写を有効または無効にできます。特定の文のために DDL の複写を有効または無効にするには、一重引用符または二重引用符で囲まれた DDL 文のフォームで文字列を含む *ddl\_statement* パラメータを使用します。

```
pdb_setrepddl stmt, ddl_statement [ , { enable[, override] | disable | default } ]
```

特定の DDL 文の現在のフィルタ設定をリストするには、次のように **stmt** キーワードの後に *ddl\_statement* パラメータを続けます。

```
pdb_setrepddl stmt, ddl_statement
```

文のセット全体に対して DDL の複写を有効または無効にするには、DDL 文キーワード形式の文字列を含む *ddl\_statement\_keyword* パラメータを使用します。

```
pdb_setrepddl stmt, ddl_statement_keyword [ , { enable[, override] | disable | default } ]
```

表 2 : ddl 文キーワード・パラメータ

キーワードの値	フィルタされた DDL 文
cluster	alter cluster、create cluster、drop cluster、truncate cluster
context	alter context、drop context
dimension	alter dimension、create dimension、drop dimension
directory	alter directory、drop directory
function	alter function、create function、drop function
index	alter index、create index、drop index
indextype	alter indextype、create indextype、drop indextype
java	alter java、create java、drop java
library	alter library、drop library



キーワードの値	フィルタされた DDL 文
materialized_view	alter materialized view、 create materialized view、 drop materialized view
operator	alter operator、 create operator、 drop operator
outline	alter outline、 create outline、 drop outline
package	alter package、 create package、 drop package
package body	alter package body、 create package body、 drop package body
point	alter restore point、 drop restore point
procedure	alter procedure、 create procedure、 drop procedure
profile	alter profile、 create profile、 drop profile
role	alter role、 create role、 drop role
sequence	alter sequence、 create sequence、 drop sequence
synonym	alter (public) synonym、 drop (public) synonym
table	alter table、 create table、 drop table
trigger	alter trigger、 create trigger、 drop trigger
type	alter type、 create type、 drop type
type body	alter type body、 create type body、 drop type body
user	alter user、 create user、 drop user
view	alter view、 create view、 drop view

すべての DDL 文の複写を有効または無効にするには、次のように **stmt** キーワードの後に **all** キーワードを続けます。

```
pdb_setrepddl stmt, all [ , { enable[, override] | disable | default } ]
```

複写からフィルタされる DDL 文をリストするには、次のように **stmt** キーワードのみを入力します。

```
pdb_setrepddl stmt
```

- owner (Oracle のみ)** – **owner** キーワードを使用すると、特定のユーザが所有するオブジェクトに影響を与える DDL 文の複写を有効または無効にできます。特定のユーザが所有するオブジェクトに影響する DDL 文の複写を有効または無効にするには、ユーザを指定するために *ownername* パラメータを使用します。

```
pdb_setrepddl owner, ownername [ , { enable[, override] | disable
| default } ]
```

オブジェクト所有者に対する現在のフィルタ設定をリストするには、次のように **owner** キーワードの後に *ownername* パラメータを続けます。

```
pdb_setrepddl owner, ownername
```

複写からフィルタされる DDL 文のオブジェクト所有者をリストするには、次のように **owner** キーワードのみを入力します。

```
pdb_setrepddl owner
```

すべての所有者に対してオブジェクトに影響を与える DDL 文の複写を有効または無効にするには、次のように **owner** キーワードの後に **all** キーワードを続けます。

```
pdb_setrepddl owner, all [ , { enable[, override] | disable |
default } ]
```

- **enable** – Replication Agent for Oracle の場合、**enable** キーワードを使用すると、**pdb\_setrepddl** コマンドに他のキーワードとパラメータで指定されたとおりに DDL 文の複写を有効にすることができます。DDL 文の以前のフィルタリングを上書きするには、次のように **enable** キーワードの後に **override** キーワードを続けます。

Replication Agent for Microsoft SQL Server の場合、**enable** キーワードを使用すると、プライマリ・データベースに対する DDL 文の複写を有効にできます。

複数の Replication Agent と Replication Server Multi-Path™ Replication (Oracle のみ) の場合:

- **all** は、Replication Agent グループ内の呼び出し元 Replication Agent インスタンスに対してすべての DDL の複写を有効にします。例：
 

```
pdb_setrepddl enable, all
```
- **marked** は、Replication Agent グループ内の呼び出し元 Replication Agent インスタンスによってマーク付けされたオブジェクトにのみ DDL の複写を有効にします。
 

```
pdb_setrepddl enable, marked
```
- **unmarked** は、Replication Agent グループ内のどの Replication Agent インスタンスによってもマーク付けされていないオブジェクトに対する DDL の複写を有効にします。
 

```
pdb_setrepddl enable, unmarked
```
- **disable** – Replication Agent for Oracle の場合、**disable** キーワードを使用すると、**pdb\_setrepddl** コマンドに他のキーワードとパラメータで指定されたとおりに DDL 文の複写を無効にすることができます。

Replication Agent for Microsoft SQL Server の場合、**disable** キーワードを使用すると、プライマリ・データベースへの DDL 文の複写を無効にできます。デフォルトで、DDL 文の複写は Replication Agent for Microsoft SQL Server で無効になっています。

Replication Agent for Microsoft SQL Server の DDL 複写のステータスを返すには、**pdb\_setrepddl** キーワード (**disable** キーワードまたは **enable** キーワードなし) を使います。

- **default (Oracle のみ)** – 明示的に、特定の所有者、オブジェクト、DDL 文への複写が有効または無効になっていない場合、**default** キーワードは次の階層を基に DDL 文のフィルタリングを有効または無効にします。
  1. オブジェクト:オブジェクトへの任意の DDL フィルタリング・ルールは監視されますが、オブジェクトの所有者、オブジェクトに影響を与える文、またはユーザを含むルールは監視されません。
  2. 所有者:オブジェクトの所有者への任意の DDL フィルタリング・ルールは監視されますが、オブジェクトに影響を与える文、またはユーザを含むルールは監視されません。
  3. 文:DDL 文への任意の DDL フィルタリング・ルールは監視されますが、ユーザを含むルールは監視されません。
  4. ユーザ:ユーザを含む任意の DDL フィルタリング・ルールは監視されます。
- **enable (複数の Replication Agent の場合)** – Replication Agent for Oracle の場合、**enable** キーワードを使用すると、**pdb\_setrepddl** コマンドに他のキーワードとパラメータで指定されたとおりに DDL 文の複写を有効にすることができます。DDL 文の以前のフィルタリングを上書きするには、次のように **enable** キーワードの後に **override** キーワードを続けます。

Replication Agent for Microsoft SQL Server の場合、**enable** キーワードを使用すると、プライマリ・データベースに対する DDL 文の複写を有効にできます。

## 例

- 例 1 –

```
pdb_setrepddl stmt
```

(Oracle のみ) このコマンドは、複写からフィルタされた DDL 文をリストします。

- 例 2 –

```
pdb_setrepddl stmt, 'create index'
```

(Oracle のみ) このコマンドは、**create index** 文の最新フィルタ設定をリストします。

- 例 3 –

```
pdb_setrepddl stmt, 'create index', disable
```

(Oracle のみ) このコマンドにより、Replication Agent は複製から **create index** 文をフィルタします。

- 例 4 –

```
pdb_setrepddl stmt, index, disable
```

(Oracle のみ) このコマンドにより、Replication Agent は複製から **alter index** 文、**create index** 文、および **drop index** 文をフィルタします。

- 例 5 –

```
pdb_setrepddl owner
```

(Oracle のみ) このコマンドは、複製からフィルタされる DDL 文のオブジェクト所有者をリストします。

- 例 6 –

```
pdb_setrepddl owner, myuser
```

(Oracle のみ) このコマンドは、オブジェクト所有者の最新フィルタ設定をリストします。

- 例 7 –

```
pdb_setrepddl owner, myuser, disable
```

(Oracle のみ) このコマンドにより、Replication Agent は myuser というユーザによって所有されているオブジェクトに影響を与える DDL 文をフィルタします。

- 例 8 –

```
pdb_setrepddl myuser.mytable
```

(Oracle のみ) このコマンドは、myuser.mytable というテーブルの最新フィルタ設定をリストします。

- 例 9 –

```
pdb_setrepddl myuser.mytable, disable
```

(Oracle のみ) このコマンドにより、Replication Agent は myuser.mytable というテーブルに影響を与えるすべての DDL 文をフィルタします。

- 例 10 –

```
pdb_setrepddl owner, myuser, disable
```

```
pdb_setrepddl myuser.mytable, enable
```

(Oracle のみ) これらのコマンドにより、Replication Agent は myuser.mytable テーブルではなく myuser というユーザによって所有されているオブジェクトに影響を与えるすべての DDL 文をフィルタします。

- **例 11 –**

```
pdb_setrepddl owner, all, disable
```

```
pdb_setrepddl owner, myuser, enable
```

(Oracle のみ) これらのコマンドにより、Replication Agent は myuser というユーザによって所有されているオブジェクトに影響を与える DDL 文のみを複製します。

- **例 12 –**

```
pdb_setrepddl user, myuser, disable
```

(Oracle のみ) このコマンドにより、Replication Agent は myuser というユーザから DLL をフィルタします。

- **例 13 –**

```
pdb_setrepddl enable, override
```

(Oracle のみ) このコマンドは DLL 複製を有効にし、任意の既存のフィルタ設定を上書きします。

- **例 14 –**

```
pdb_setrepddl owner, myuser, enable, override
```

(Oracle のみ) このコマンドは myuser というユーザから DDL の複製を有効にし、任意の既存のフィルタ設定を上書きします。

- **例 15 –**

```
pdb_setrepddl
```

このコマンドは、プライマリ・データベース内の現在の DDL 複製のステータスを返します。

- **例 16 –**

```
pdb_setrepddl enable
```

このコマンドは、この時点後、プライマリ・データベースに発行された DDL コマンドの複製を有効にします。

- **例 17 –**

```
pdb_setrepddl disable
```

このコマンドは、この時点後、プライマリ・データベースに発行された DDL コマンドの複写を無効にします。

### 使用法

- **pdb\_setrepddl** コマンドを使用して DDL 複写を有効にするだけでなく、Replication Agent に **ddl\_username** パラメータと **ddl\_password** パラメータを設定する必要があります。
- DDL を有効にするデータベース複写定義は、DDL 複写にとって必要です。**use\_rssd** 設定パラメータが **true** に設定されている場合、Replication Agent **resume** コマンドが呼び出される前に、データベース複写定義は、プライマリ Replication Server の RSSD に存在する必要があります。
- プライマリとレプリケートのデータベースに同一の構文を持つと DDL 文のみが複写されます。Replication Agent は、プライマリ・データベース内の文の構文を使用して DDL 文を送信します。
- Replication Agent が初期化されていない場合、**pdb\_setrepddl** コマンドはエラーを返します。
- Replication Agent インスタンスが Admin 状態または Replication Down 状態の場合、**pdb\_setrepddl** コマンドは **enable** オプションと **disable** オプションのみが使用できます。
- DDL 複写と複数の Replication Agent に対して **all** キーワード、**marked** キーワードおよび **unmarked** キーワードを使用する場合、以下のようになります。

- **all** キーワードを省略しても、すべての DDL の複写を有効にできます。

```
pdb_setrepddl enable
```

- **marked** キーワードと **unmarked** キーワードは、お互いを上書きしません。たとえば、Replication Agent グループ内の Replication Agent インスタンスによってマーク付けられていないオブジェクトの DDL および呼び出し元 Replication Agent インスタンスによってマーク付けられているオブジェクトの DDL の両方の複写を有効にするには、次のようにします。

```
pdb_setrepddl enable, unmarked  
pdb_setrepddl enable, marked
```

ただし、**marked** と **unmarked** は前回の **all** キーワードの実行を上書きし、**all** キーワードは前回の **marked** と **unmarked** のキーワードの実行を上書きしません。

### 参照：

- **ddl\_password** (186 ページ)
- **asm\_username** (182 ページ)

## pdb\_setrepproc

(Oracle および Microsoft SQL Server のみ) ストアド・プロシージャの複製マーク付けステータスを返します。指定されたプロシージャに複製のマークを付け、マーク付けされたすべてのプロシージャまたは指定されたプロシージャのマークを外し、すべてのマーク付けされたプロシージャまたは指定されたプロシージャの複製を有効または無効にします。

---

**注意：** このコマンドは Oracle および Microsoft SQL Server でのみ使用可能です。

---

### 構文

```

pdb_setrepproc
  [ { procname[, repname,] { mark
                                | unmark[, force]
                                | enable
                                | disable }
    | all, { unmark[, force]
            | enable
            | disable }
  } ]

```

ストアド・プロシージャの複製マーク付けステータスを返すには、次のようにします。

```

pdb_setrepproc [ {procname|mark|unmark|enable|disable } ]

```

マーク付けされたすべてのストアド・プロシージャのマークを外したり、有効または無効にするには、次のようにします。

```

pdb_setrepproc all, {unmark[, force]|enable|disable}

```

複製済みの名前で、指定されたストアド・プロシージャに複製のマークを付けるには、次のようにします。

```

pdb_setrepproc procname, [repname,] mark

```

指定するストアド・プロシージャのマークを外したり、有効または無効にするには、次のようにします。

```

pdb_setrepproc procname, {unmark[, force]|enable|disable}

```

---

**注意：** Microsoft SQL Server の場合、**pdb\_dflt\_object\_repl** 設定パラメータは **pdb\_setrepproc** コマンドに何も影響しません。

---

### パラメータ

- **procname** – プライマリ・データベース内のユーザ・ストアド・プロシージャ名です。

*procname* オプションは、引用符で区切って大文字と小文字を指定できます。

大文字と小文字を混在させる必要がある場合は、名前を区切ります。例：

"Proc"

**注意：** `lfl_character_case` パラメータの設定と一致しない大文字と小文字をオブジェクト名で使用する必要がある場合、オブジェクト名を区切る必要があります。

オブジェクト名にスペースやピリオドなどの英数字以外の文字が含まれる場合、引用符で区切る必要があります。例：

"proc name"

"proc.name"

オブジェクト名にピリオドが含まれる場合、所有者指定と引用符による区切りの両方を行う必要があります。例：

owner."proc.name"

"proc.owner"."proc.name"

- **rename** – プライマリ・ストアド・プロシージャのファンクション複写定義で指定されるストアド・プロシージャの名前です。

*rename* オプションは、引用符で区切って大文字と小文字を指定できます。詳しくは、前述の *procname* オプションの説明を参照してください。

複写済みの名前を指定することで、プライマリ・データベース内のものとは名前が異なるレプリケート・データベース内のストアド・プロシージャ呼び出しに、ストアド・プロシージャ呼び出しを複写できます。

**注意：** `pdb_setrepproc` コマンドで指定する複写済みの名前は、プライマリ・データベース・コネクションの Replication Server ファンクション複写定義の **with primary function named** 句で指定された名前と一致する必要があります。Replication Agent はファンクション複写定義を検証できませんが、存在しない場合、プライマリ・データベースからのファンクション複写は失敗します。

- **all** – プライマリ・データベース内のすべてのユーザ・ストアド・プロシージャを参照するキーワードです。 **all** キーワードを使用することにより、すべてのユーザ・ストアド・プロシージャにマークを付けるか、すべてのマーク付けされたストアド・プロシージャにマーク解除オペレーション、有効化オペレーション、または無効化オペレーションを適用することができます。
- **mark** –

**注意：** **mark** キーワードを使用する場合、所有者を指定する必要があります。

ユーザ・ストアド・プロシージャの複写マーク付けを参照するキーワードです。

- **unmark** – マーク付けされたストアド・プロシージャのマーク解除を参照するキーワードです。
- **force** – マーク解除オペレーションを参照するキーワードです。 **unmark** キーワードの後に **force** キーワードを続けると、 `pdb_setrepproc` コマンドはストア



ド・プロシージャの有効化ステータスを確認せずに、プライマリ・データベース内の指定されたストアド・プロシージャのマークをすぐに外します。 **unmark** キーワードおよび **all** キーワードの後に **force** キーワードを続けると、**pdb\_setreproc** コマンドは有効化ステータスに関係なく、プライマリ・データベース内でマーク付けされたすべてのストアド・プロシージャから複写マークをすぐに外します。

また、**force** キーワードは、マーク解除のプロセス中にエラーが発生した場合でも、マークが付いていないスクリプトの実行を強制します。通常、スクリプト実行中にエラーが発生すると、スクリプトは完了まで実行せずにすぐに終了します。

---

**注意：**スクリプトの Oracle プロシージャでのマーク付けおよびマーク解除での使用方法について理解を深めるには、「[使用法](#)」の項を確認します。

---

前のスクリプト実行に失敗し、マーク解除のオペレーションが完了しないままの状態になった場合、**force** キーワードは便利です。強制スクリプトの実行中にエラーが発生すると、**pdb\_setreproc** コマンドは次のようなメッセージを返します。

```
Errors were encountered and ignored during FORCED script
execution. See error log for details.
```

- **enable** – マーク付けされたストアド・プロシージャへの複写の有効化を参照するキーワードです。
- **disable** – マーク付けされたストアド・プロシージャへの複写の無効化を参照するキーワードです。

## 例

- 例 1 –

```
pdb_setreproc
```

このコマンドは、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ストアド・プロシージャの複写マーク付け情報を返します。

- 例 2 –

```
pdb_setreproc authors
```

このコマンドは、プライマリ・データベース内の“authors”というユーザ・ストアド・プロシージャの複写マーク付け情報を返します。

- 例 3 –

```
pdb_setreproc authors, mark
```

このコマンドは、プライマリ・データベース内の“authors”というユーザ・ストアド・プロシージャをマーク付けします。

• 例 4-

```
pdb_setrepproc authors, enable
```

このコマンドは、プライマリ・データベース内の“authors”というマーク付けされたストア・プロシージャへの複写を有効にします。

• 例 5-

```
pdb_setrepproc all, unmark
```

このコマンドは、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ストア・プロシージャのマーク付けを解除します。

## 使用法

- マーク付けされたプロシージャが名前変更または削除されて、元の名前付きの新しいプロシージャが作成された場合、新しいプロシージャには元のプロシージャのマーク付けに関連する情報がないため、その新しいプロシージャをマーク付けする必要があります。マーク付け情報は、オブジェクト名ではなくオブジェクト ID 別に内部で保持されます。
- **pdb\_setrepproc** コマンドの使用法は、Replication Server で作成した複写定義の種類によって異なります。ファンクション複写定義なしでデータベース複写定義を作成した場合、**pdb\_setrepproc** コマンドの複写プロシージャは、複写データベースのプロシージャを参照します。ただし、ファンクション複写定義を作成した場合は、**pdb\_setrepproc** コマンドのレプリケート・プロシージャは、ファンクション複写定義の名前を参照し、複写データベース内のプロシージャにマップされる必要があるのは、ファンクション複写定義です。ファンクション複写定義が存在せず、複写前に追加もされないが、データベース複写定義のみが存在する場合、これらのコマンドを使用してプロシージャを複写のためにマーク付けします。
- 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・データベース内のプロシージャと同じ名前の場合は、次のようにします。

```
pdb_setrepproc pdb_proc, mark
```

構文の説明は次のとおりです。

**pdb\_proc** は、複写のマークを付ける、プライマリ・データベース内のプロシージャの名前です。

- 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・データベース内のプロシージャと異なる名前の場合は、次のようにします。

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

構文の説明は次のとおりです。

**pdb\_proc** は、複写のマークを付ける、プライマリ・データベース内のプロシージャの名前です。

**rep\_proc** 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。

ファンクション複写定義が存在するか、複写前に追加される場合、データベース複写定義が存在するかどうかにはかかわらず、これらのコマンドを使用してプロシージャを複写のためにマーク付けします。

- ファンクション複写定義がプライマリ・データベース内のプロシージャと同じ名前の場合は、次のようにします。

```
pdb_setrepproc pdb_proc, mark
```

構文の説明は次のとおりです。

**pdb\_proc** は、複写のマークを付ける、プライマリ・データベース内のプロシージャの名前です。

- レプリケート・データベース内のプロシージャもファンクション複写定義と同じ名前の場合、プライマリ Replication Server の複写定義の “deliver as” 句を使用する必要はありません。例：

```
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。例：

```
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・データベース内のプロシージャと異なる場合は、次のようにします。

```
pdb_setrepproc pdb_proc , rdpri_proc, mark
```

構文の説明は次のとおりです。

**pdb\_proc** は、複写のマークを付ける、プライマリ・データベース内のプロシージャの名前です。

**rdpri\_proc** はファンクション複写定義の名前です。

- レプリケート・データベース内のプロシージャもファンクション複写定義と同じ名前の場合、プライマリ Replication Server の複写定義の “deliver as” 句を使用する必要はありません。例：

```
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。例：

```
create function replication definition rdpri_proc with
primary at data_server.database deliver as 'rep_proc' ...
```

- 複数のキーワードとオプションを指定する場合は、それぞれカンマで区切る必要があります。カンマの前後にはオプションで空白を挿入できます。例：

```
pdb_setrepproc all, unmark, force
```

- **pdb\_setrepproc** コマンドでストアド・プロシージャの名前を指定する場合、有効なユーザ・ストアド・プロシージャの名前を使用する必要があります。
- **pdb\_setrepproc** コマンドでストアド・プロシージャの名前として、以下の項目を指定することはできません。
  - システム・プロシージャ
  - Replication Agent トランザクション・ログ・プロシージャ
- 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は、キーワードの名前と同じ場合、`proc=` 文字列をストアド・プロシージャの名前の先頭に追加することによって識別されます。例：

```
pdb_setrepproc proc=unmark, mark
```
- **unmark** キーワードを使用してストアド・プロシージャから複製マークを外す場合、Replication Agent はそのストアド・プロシージャの複製が無効になり、トランザクション・ログにストアド・プロシージャに保留中の(未処理の)オペレーションがないことを確認します。複製がプロシージャに対して無効ではない場合、またはトランザクション・ログにそのプロシージャの保留中オペレーションがある場合は、**pdb\_setrepproc** はエラーを返します。
- **pdb\_setrepproc** は、オプションなしで、または1つのオプションで呼び出されると、プライマリ・データベース内のストアド・プロシージャに関するマーク付け情報を返します。
  - **pdb\_setrepproc** はオプションなしで呼び出されると、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のリストを返します。

---

**注意：** **pdb\_setrepproc** コマンドをオプションなしで呼び出すと、**mark** キーワードのみを指定して **pdb\_setrepproc** コマンドを呼び出した場合と同じ結果を生成します。

---

- **pdb\_setrepproc** はプロシージャ名で呼び出されると、指定されたプロシージャに関する完全なマーク付け情報を返します。
- **pdb\_setrepproc** を **mark** キーワード指定で呼び出すと、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のリストを返します。
- **pdb\_setrepproc** を **unmark** キーワード指定で呼び出すと、プライマリ・データベース内のマークが付いていない全プロシージャのリストを返します。
- **pdb\_setrepproc** を **enable** キーワード指定で呼び出すと、複製が現在有効になっているプライマリ・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のリストを返します。
- **pdb\_setrepproc** を **enable** キーワード指定で呼び出すと、複製が現在有効になっているプライマリ・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のリストを返します。

複製のためにマーク付けされているストアド・プロシージャは RASD で記録されます。その他のユーザ・プロシージャはマーク解除プロシージャと見なされます。

---

**注意：** Replication Agent システムのプロシージャは、マーク解除プロシージャのリストには含まれません。また、これらのプロシージャの同義語、エイリアスも含まれていません。

---

マーク解除または無効としてリストされたプロシージャの場合、呼び出しは複製用に取り出されません。

- **pdb\_setrepproc** を **all** キーワードおよびアクション・キーワード (**unmark**、**enable**、**disable**) 指定で呼び出すと、指定されたアクションはプライマリ・データベース内のすべてのユーザ・ストアド・プロシージャ、またはプライマリ・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に適用されます。
  - **pdb\_setrepproc** を **all** キーワードおよび **unmark** キーワード指定で呼び出すと、プライマリ・データベース内にあるマーク付けされたすべてのプロシージャから複製マークが外されます。  
**unmark** キーワードの後に **force** キーワードを指定すると、複製がまだ有効になっているプロシージャを含む、マーク付けされたすべてのプロシージャのマークを強制的にすぐ外することができます。
  - **pdb\_setrepproc** を **all** キーワードおよび **enable** キーワード指定で呼び出すと、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の複製が有効になります。
  - **pdb\_setrepproc** を **all** キーワードおよび **disable** キーワード指定で呼び出すと、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の複製が無効になります。
- **pdb\_setrepproc** を有効なユーザ・ストアド・プロシージャ名の後にアクション・キーワード (**mark**、**unmark**、**enable**、または **disable**) 指定で呼び出すと、指定されたアクションは指定されたプロシージャに適用されます。
  - **pdb\_setrepproc** をプロシージャ名および **mark** キーワード指定で呼び出すと、プライマリ・データベース内にある指定されたプロシージャが複製のためにマーク付けされます。
  - **pdb\_setrepproc** をプロシージャ名および **unmark** キーワード指定で呼び出すと、プライマリ・データベース内にある指定されたプロシージャから複製マークが外されます。
  - **pdb\_setrepproc** をプロシージャ名および **enable** キーワード指定で呼び出すと、プライマリ・データベース内にある指定のマーク付けされたプロシージャについて、複製が有効になります。
  - **pdb\_setrepproc** をプロシージャ名および **disable** キーワード指定で呼び出すと、プライマリ・データベース内にある指定のマーク付けされたプロシージャについて、複製が無効になります。

---

**注意：** 結果は同じなので、**[enable | disable]** ではなく **[mark | unmark]** を使用します。

---

- プライマリ・データベースに存在しないストアド・プロシージャ名を指定する場合、**pdb\_setrepproc** コマンドはエラーを返します。
- **pdb\_setrepproc** をプロシージャの名前と複写済みの名前に続けて **mark** キーワードを使用して呼び出すと、プライマリ・プロシージャは、指定した複写済みの名前を使用して複写のためにマーク付けされます。  
指定したプライマリ・プロシージャ名がプライマリ・データベースに存在しない場合、**pdb\_setrepproc** コマンドはエラーを返します。  
複写済みの名前を指定すると、プロシージャ呼び出しは、プライマリ・プロシージャとは異なる名前のレプリケート・データベースのプロシージャに複写されます。

---

**注意：** **pdb\_setrepproc** コマンドで指定する複写済みの名前は、プライマリ・データベース・コネクションの Replication Server ファンクション複写定義の名前と一致する必要があります。Replication Agent はファンクション複写定義を検証できませんが、存在しない場合、プライマリ・データベースからのファンクション複写は失敗します。

---

- RASD が初期化されていない場合 (Oracle および Microsoft SQL Server)、**pdb\_setrepproc** コマンドはエラーを返します。

Oracle の場合:

Oracle でストアド・プロシージャの複写をサポートするには、複写のためにマーク付けされたストアド・プロシージャを変更する必要があります。変更は、Oracle トランザクション・ログのストアド・プロシージャの実行を記録するために必要です。変更の結果として、Oracle でストアド・プロシージャをマーク付けおよびマーク解除する場合、次のような動作を検討します。

- 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 複写を無効にし、変更をスタンバイに複写しないようにマーク付けまたはマーク解除した後、再び有効にします。
- 複写のためにストアド・プロシージャをマーク付けおよびマーク解除するには、Replication Agent を削除した後にプロシージャを再作成する必要があります。ただし、Replication Agent は元のプロシージャで定義されているものと同じ権限を再作成のプロシージャにもすべて設定します。

---

**注意：** マーク付けされたストアド・プロシージャの Replication Agent コメントを削除または変更しないでください。

---

- **pdb\_setrepproc** が複写のためにプロシージャをマーク付けするために呼び出される場合、Replication Agent は、次のことを実行します。
  - ユーザ・プロシージャを変更して、入力パラメータの値を取り込み、Replication Agent トランザクション・ログ・レコードを生成するコードを追加します。

- プライマリ・データベース内の Replication Agent のトランザクション・ログに必要なプロシージャを作成する SQL スクリプトを生成します。
- 生成されたスクリプトを `partmark.sql` ファイルに保存します。このファイルは `RAX-15_5¥inst_name¥scripts¥procname` ディレクトリ (`inst_name` は Replication Agent インスタンスの名前で、`procname` はマーク付けされたストアド・プロシージャの名前です) にあります。このスクリプトは手動で実行することはできません。参照目的でのみ示されています。

---

**注意：** `pdb_auto_run_scripts` 設定パラメータの値が `false` の場合、`partmark.sql` スクリプトは保存され、自動的に実行されません。スクリプトを手動で実行することはできません。プロシージャのマーク付けを完了するには、まず `pdb_auto_run_scripts` を `true` に設定し、`pdb_setrepproc` コマンドを再実行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ためにスクリプトを実行し、プライマリ・データベースにトランザクション・ログ・オブジェクトを作成します (`pdb_auto_run_scripts` 設定パラメータの値が `true` の場合)。
- スクリプトが正常に完了した後、`partmark.sql` ファイルを `RAX-15_5¥inst_name¥scripts¥procname¥installed` ディレクトリに移動します。
- `mark` スクリプトが失敗した場合、それはファイル (`partmark.sql`) に保存されます。このファイルは、`RAX-15_5¥inst_name¥scripts ¥procname` ディレクトリにあります。テーブルはマーク付けされず、トランザクション・ログ・オブジェクトは作成されません。スクリプトは `mark.sql` ファイルを表示すると確認できます。
- `pdb_setrepproc` がマーク付けされたストアド・プロシージャのマークを外すために呼び出される場合、Replication Agent は、次のことを実行します。
  - ユーザ・プロシージャを変更して、入力パラメータの値を取り込み、トランザクション・ログ・レコードを生成する Replication Agent コードを削除します。
  - プライマリ・データベース内のトランザクション・ログに必要なテーブルとプロシージャを削除する SQL スクリプトを生成します。
  - 生成されたスクリプトを `partunmark.sql` というファイル (`RAX-15_5¥inst_name¥scripts¥procname` ディレクトリ内) に保存します。ここで、`inst_name` は Replication Agent インスタンスの名前で、`procname` はマークが外されるストアド・プロシージャの名前です。Oracle の場合、この `partunmark.sql` スクリプトは手動で実行できません。参照目的でのみ示されています。

---

**注意：** `pdb_auto_run_scripts` 設定パラメータの値が `false` の場合、`partunmark.sql` スクリプトは保存され、自動的に実行されません。ス

クリプトを手動で実行することはできません。プロシージャのマーク解除を完了するには、まず **pdb\_auto\_run\_scripts** を true に設定し、**pdb\_setrepproc** コマンドを再実行する必要があります。

- ストアド・プロシージャのマークを外すためにスクリプトを実行し、プライマリ・データベース内のトランザクション・ログ・オブジェクトを削除します (**pdb\_auto\_run\_scripts** 設定パラメータの値が true の場合)。
- スクリプトが正常に完了した後、partunmark.sql ファイルを RAX-15\_5¥inst\_name¥scripts¥procname¥installed ディレクトリに移動します。
- unmark スクリプトが失敗した場合、そのスクリプトはファイル partunmark.sql (RAX-15\_5¥inst\_name¥procname¥scripts ディレクトリ内) に保存されます。ストアド・プロシージャはマーク解除されず、トランザクション・ログ・オブジェクトは削除されません。スクリプトは partunmark.sql ファイルを表示することで、確認できます。

unmark スクリプトの実行で任意のデータベース・オブジェクトに致命的なエラーが発生した場合、**pdb\_setrepproc** コマンドは次のメッセージを返します。

```
Could not unmark the following objects: ...
See error log for details.
```

- **pdb\_setrepproc** コマンドは、Boolean 型の引数を持つ Oracle ストアド・プロシージャの複写に使用されます。詳細については、『Replication Agent プライマリ・データベース・ガイド』の「Replication Agent for Oracle」の「BOOLEAN 引数を持つストアド・プロシージャの複写」を参照してください。

**参照：**

- pdb\_setrepcol (31 ページ)
- pdb\_setreptable (56 ページ)
- ra\_config (86 ページ)

**pdb\_setrepseq**

(Oracle のみ) シーケンスの複写マーク付けステータスを返します。指定されたシーケンスに複写のマークを付け、すべてのマーク付けされたシーケンスまたは指定されたシーケンスのマークを外し、すべてのマーク付けされたシーケンスまたは指定されたシーケンスへの複写を有効または無効にします。

**注意：** このコマンドは Oracle でのみ使用可能です。

**構文**

```
pdb_setrepseq
  [ { sequence_name, [repname,] { mark
```



```

| unmark[, force]
| enable
| disable }
| all, { mark
| unmark[, force]
| enable
| disable }
} ]

```

シーケンスの複写マーク付けステータスを返すには、次のようにします。

```
pdb_setrepseq [ { sequence_name|mark|unmark|enable|disable } ]
```

マーク付けされたすべてのシーケンスをマーク解除、有効または無効にするには、次のようにします。

```
pdb_setrepseq all, {unmark[, force] |enable|disable}
```

指定されたシーケンスをマーク付け、マーク解除、有効または無効にするには、次のようにします。

```
pdb_setrepseq sequence_name, {mark|unmark[, force] |enable|disable}
```

複写済みの名前で指定されたシーケンスに複写のマークを付けるには、次のようにします。

```
pdb_setrepseq sequence_name, repname, mark
```

## パラメータ

- **sequence\_name** – プライマリ・データベース内のユーザ・シーケンスの名前です。*sequence\_name* オプションは、引用符で区切って、大文字と小文字を指定できます。大文字と小文字を混在させる必要がある場合は、名前を区切りませ。例：

```
"Sequence"
```

*sequence\_name* パラメータは、それぞれの要素がピリオドで区切られた所有者指定 (プライマリ・シーケンスの所有者名を含む) の場合もあります。例：

```
owner.sequence
```

**注意：** *lfl\_character\_case* パラメータの設定と一致しない大文字と小文字をオブジェクト名で使用する必要がある場合、オブジェクト名を区切る必要があります。オブジェクト名にスペースやピリオドなどの英数字以外の文字が含まれる場合、引用符で区切る必要があります。たとえば、"sequence name" または `owner."sequence name"` となります。

- **repname** – プライマリ・サイトでのシーケンス名とは異なるものにする場合、レプリケート・サイトで更新されるシーケンスの複写済みの名前です。*repname* オプションは、引用符で区切って大文字と小文字を指定できます。詳しくは、前述の *sequence\_name* パラメータの説明を参照してください。複写済みの名前を指定することで、プライマリ・データベース内のものとは名前が異なるレプリケート・データベース内のシーケンスに、シーケンス更新を複写で

きます。*rename* オプションは、それぞれの要素がピリオドで区切られた所有者指定 (複写シーケンスの所有者名を含む) の場合もあります。例：

```
reowner.rename
```

- **all** – プライマリ・データベース内のすべてのユーザ・シーケンスを参照するキーワードです。*all* キーワードを使用することにより、すべてのユーザ・シーケンスのマークを外したり、マーク付けされたすべてのシーケンスに有効化または無効化のオペレーションを適用できます。
- **mark** – ユーザ・シーケンスの複写のマーク付けを参照するキーワードです。
- **unmark** – ユーザ・シーケンスの複写のマーク解除を参照するキーワードです。
- **force** – マーク解除オペレーションを参照するキーワードです。**unmark** キーワードの後に **force** キーワードを続けると、**pdb\_setrepseq** コマンドはシーケンスの有効化ステータスを確認せずに、プライマリ・データベース内の指定されたシーケンスのマークをすぐに外します。**unmark** キーワードと **all** キーワードの後に **force** キーワードを続けると、**pdb\_setrepproc** コマンドは、有効化ステータスに関係なく、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シーケンスから複写マークをすぐに外します。
- **enable** – マーク付けされたシーケンスの複写の有効化を参照するキーワードです。
- **disable** – マーク付けされたシーケンスの複写の無効化を参照するキーワードです。

### 使用法

- **pdb\_setrepseq** が呼び出されると、指定したキーワードとオプションに従って機能が決まります。
- 複数のキーワードとオプションを指定する場合は、それぞれカンマで区切る必要があります。カンマの前後にはオプションで空白を挿入できます。例：  

```
pdb_setrepseq all, unmark, force
```
- **pdb\_setrepseq** コマンドのシーケンスを指定する場合、有効なユーザ・シーケンスの名前を使用する必要があります。

### pdb\_setreptable

複写のマーク付けステータスを返します。すべてのユーザ・テーブルまたは指定されたテーブルに複写のマークを付ける、マーク付けられたすべてのテーブルまたは指定されたテーブルのマークを外す、マーク付けされたすべてのテーブルまたは指定されたテーブルへの複写を有効または無効にします。

### 構文

Oracle および Microsoft SQL Server の場合:

```
pdb_setreptable  
[ { tablename[, [rename,] { mark [ { , immediate
```

```

| , owner[, force] } ]
| unmark[, force]
| enable
| disable}]
| all, { mark
| unmark[, force]
| enable
| disable }
} ]

```

**IBM DB2 の場合:**

```

pdb_setreptable
[ { tablename[, [repname,] { mark [ , owner[, force] ]
| unmark[, force]
| enable
| disable }}
| all, { mark
| unmark[, force]
| enable
| disable }
} ]

```

複写のマーク付けステータスを返すには、次のようにします。

```

pdb_setreptable tablename

```

すべてのユーザ・テーブルにマークを付ける (Oracle および Microsoft SQL Server でのみ使用可能) には、次のようにします。

```

pdb_setreptable all, mark

```

すべてのマーク付けされたテーブルに対して、マークを外したり有効化または無効化するには、次のようにします。

```

pdb_setreptable all, {unmark[, force]|enable|disable}

```

指定されたテーブルに対して、マークを付けたり外したり、有効化または無効化するには、次のようにします。

```

pdb_setreptable tablename, {mark[, owner][, force] |
unmark[, force] |enable|disable}

```

**注意：** `pdb_setreptable all, mark` または `pdb_setreptable all, unmark` を使用してプライマリ・データベース内のすべてのテーブルから一度にマークを付けたり外したりすることは、Replication Agent for UDB ではサポートされていません。テーブルごとに個別にマークを付けたり外したりする必要があります。

複写済みの名前指定されたテーブルに複写のマークを付けるには、次のようにします。

```

pdb_setreptable tablename, repname, mark[, owner][, force]

```

マークが付けられているかいないかに関わらず、すべてのオカレンスについて指定されたテーブルに複写のマークをすぐに付けるには、次のようにします。

```

pdb_setreptable tablename, mark[, immediate]

```

マーク付けされたすべてのテーブルのリストを返すには、次のようにします。

```
pdb_setreptable
```

### パラメータ

- **tablename** – プライマリ・データベース内の有効なユーザ・テーブルの名前です。Replication Agent は、指定されたプライマリ・テーブルに関する完全なマーク付け情報を返します。システム・テーブル、ビュー、または Replication Agent トランザクション・ログ・テーブルをプライマリ・テーブルとして指定することはできません。

*tablename* パラメータは、それぞれの要素がピリオドで区切られた所有者指定 (プライマリ・テーブルの所有者名を含む) の場合もあります。例：

```
owner.table
```

このパラメータは、引用符で区切って大文字と小文字を指定できます。

大文字と小文字を混在させる必要がある場合は、名前を区切ります。例：

```
"Owner".table
```

```
"Owner"."Table"
```

大文字と小文字を区別しない *tablename* オプションの各要素は、前の例のように区切る必要があります。

オブジェクト名にスペースやピリオドなどの英数字以外の文字が含まれる場合、引用符で区切る必要があります。例：

```
"table name"
```

```
owner."table name"
```

オブジェクト名にピリオドが含まれる場合、所有者指定と引用符による区切りの両方を行う必要があります。例：

```
owner."table.name"
```

```
"table.owner"."table.name"
```

- **repname** – プライマリ・テーブルの複製定義で指定されているテーブルの名前です。

---

**注意：** `pdb_setreptable` コマンドで指定する複製済みの名前は、プライマリ・データベース・コネクションの Replication Server 複製定義の **with primary table named** 句で指定したテーブル名と一致する必要があります。Replication Agent は複製定義を確定化できませんが、それが存在しない場合、または **with primary table named** 句が `pdb_setreptable` で指定した複製済みの名前と一致しない場合は、プライマリ・テーブルからの複製は失敗します。

---

*repname* オプションは、それぞれの要素がピリオドで区切られた所有者指定 (複製テーブルの所有者名を含む) の場合もあります。例：

```
repowner.reptable
```

また、*rename* オプションは、引用符で区切って大文字と小文字を指定することもできます。詳しくは、前述の *tablename* オプションの説明を参照してください。

---

**注意：** 複写テーブル名にピリオド (たとえば、*table.name*) が含まれ、所有者指定がない場合、Replication Agent の **use\_rssd** パラメータ値を **true** に設定する必要があります。

---

- **all** – プライマリ・データベース内のすべてのテーブルを参照するキーワードです。**all** キーワードを使用することにより、すべてのユーザ・テーブルにマークを付けたり、すべてのマーク付けされたテーブルにマーク解除オペレーション、有効化オペレーション、または無効化オペレーションを適用したりできます。
- **mark** – テーブルのマーク付けを参照するキーワードです。Replication Agent は、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リストを返します。
- **owner** – マーク付けオペレーションを参照するキーワードです。

**owner** キーワードは、**SEND OWNER** モードをオンにします。複写定義にテーブル所有者を指定すると、常に **owner** キーワードを使用する必要があります (**SEND OWNER** モードを有効にする場合)。

**owner** モードは、LTL が任意のテーブル・レベルの複写定義がこのテーブルと一致する所有者指定である必要があることを Replication Server に告げるように、フラグを設定します。

**owner** モードを設定する場合、複写定義は所有者指定である必要があります。**owner** モードを設定しない場合、複写定義は所有者指定である必要はありません。

- **unmark** – マーク付けされたテーブルのマーク解除を参照するキーワードです。Replication Agent は、プライマリ・データベース内のすべてのマーク解除テーブルのリストを返します。
- **force** – マーク解除オペレーションまたはマーク付けオペレーションで使用されるキーワードです。

---

**注意：** **mark** キーワードおよび **force** キーワードと、**pdb\_setreptable** コマンドの併用は、UDB および Oracle の場合のみ有効です。

---

- **unmark** キーワードの後に **force** キーワードを続けると、**pdb\_setreptable** コマンドはストアド・プロシージャの有効化ステータスを確認せずに、プライマリ・データベース内の指定されたテーブルの複写マークを外します。**unmark** キーワードと **all** キーワードの後に **force** キーワードを続けると、**pdb\_setreptable** コマンドは、有効化ステータスに関係なく、プライマリ・

データベース内のすべてのマーク付けされたテーブルから複写マークをすぐに削除します。

また、**force** キーワードは、マーク解除のプロセス中にエラーが発生した場合でも、マークが付いていないスクリプトの実行を強制します。通常、スクリプト実行中にエラーが発生すると、スクリプトは完了まで実行せずすぐに終了します。前のスクリプト実行に失敗し、マーク解除のオペレーションが完了しないままの状態になった場合、**force** キーワードは便利です。

強制スクリプトの実行中にエラーが発生すると、**pdb\_setreptable** コマンドは次のようなメッセージを返します。

```
Errors were encountered and ignored during FORCED script execution. See error log for details.
```

- UDB および Oracle の場合:

**mark** キーワードの後に **force** キーワードを続けると、**pdb\_setreptable** コマンドによって、複写のマークが付けられたサポート対象外のデータ型を持つ 1 つまたは複数のカラムをテーブルに含めることができます。サポート対象外のカラムのデータは、Replication Server に送信されません。結果として、複写データベースが受信するデータにはサポート対象外カラムに挿入されるものがないため、どの複写テーブルにもサポート対象外カラムに対応するデフォルトのカラム値が必要です。

**force** キーワードを **all** キーワードと組み合わせることはできません。サポート対象外のデータ型を持つテーブルには、**pdb\_setreptable** コマンドと **force** キーワードを使用して、個別にマークを付ける必要があります (サポート対象外のデータ型を持つカラムがある場合は、自動的に、またはデフォルトでマークが付けられることはありません)。

さらに、**pdb\_automark\_tables** 設定パラメータが **true** の場合、サポート対象外のデータ型を持つテーブルは自動的にマークが付けられません。

**pdb\_setreptable** コマンドと、**mark** キーワードおよび **force** キーワードを使用して、サポート対象外のデータ型を持つテーブルに個別にマークを付ける必要があります。サポート対象とサポート対象外のデータ型のリストについては、『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。

---

**注意：** 複写定義が **rs\_create\_repdef** コマンドを使用して作成される場合、**force** キーワードを使用してマークが付けられたテーブルでは、サポート対象のデータ型を持つカラムのみが複写定義にリストされます。サポート対象外のデータ型を持つすべてのカラムは、複写定義から除外されます。

---

- **enable** – マーク付けされたテーブルの複写の有効化を参照するキーワードです。Replication Agent は、複写が有効化されたプライマリ・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リストを返します。

- **disable** – マーク付けされたテーブルの複製の無効化を参照するキーワードです。Replication Agent は、複製が無効化されたプライマリ・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リストを返します。
- **immediate** – すべてのオカレンスについてテーブルを即座にマーク付けできるようにするキーワード。このキーワードは Microsoft SQL Server および Oracle 専用です。

## 例

- **例 1 –**

```
pdb_setreptable authors
```

このコマンドは、プライマリ・データベース内の "authors" という名前のテーブルについて、複製マーク付け情報を返します。

- **例 2 –**

```
pdb_setreptable mark
```

このコマンドは、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について、複製マーク付け情報を返します。

- **例 3 –**

```
pdb_setreptable disable
```

このコマンドは、プライマリ・データベース内で複製が無効になっているすべてのマーク付きテーブルについて、複製マーク付け情報を返します。

- **例 4 –**

```
pdb_setreptable all, unmark, force
```

このコマンドは、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について、強制的にマークを外します。

- **例 5 –**

```
pdb_setreptable all, enable
```

このコマンドは、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について、複製を有効にします。

- **例 6 –**

```
pdb_setreptable authors, mark
```

このコマンドは、プライマリ・データベース内の "authors" という名前のテーブルを、複製のためにマーク付けします。複製定義内のプライマリ・テーブル名は、**authors** である必要があります。

- **例 7 –**

```
pdb_setreptable authors, mark, owner
```

このコマンドは、LTL で **OWNER\_MODE** が有効になるように、プライマリ・データベース内の "authors" という名前のテーブルを、複写のためにマーク付けします。したがって、このテーブル用に作成されたテーブル複写定義もすべて所有者指定である必要があります。

- **例 8 –**

```
pdb_setreptable ptable, rtable, mark, owner
```

複写定義内のプライマリ・テーブル名は、以下である必要があります。

```
powner.rtable
```

- **例 9 –**

```
pdb_setreptable ptable, rowner.rtable, mark, owner
```

複写定義内のプライマリ・テーブル名は、以下である必要があります。

```
rowner.rtable
```

- **例 10 –**

```
pdb_setreptable ptable, rowner.rtable, mark, owner
```

複写定義内のプライマリ・テーブル名は、以下である必要があります。

```
rowner.rtable
```

- **例 11 –**

```
pdb_setreptable authors, enable
```

このコマンドは、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 "authors" について、複写を有効にします。

- **例 12 –**

```
pdb_setreptable table=mark, enable
```

このコマンドは、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 "mark" について、複写を有効にします。

- **例 13 –**

```
pdb_setreptable authors, unmark, force
```

このコマンドは、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 "authors" について、強制的にマークを解除します。

- **例 14 – (UDB および Oracle のみ)**

```
pdb_setreptable authors, mark, force
```



このコマンドは、テーブル **"authors"** にサポート対象外のデータ型のカラムが含まれていても、このテーブルを強制的にマーク付けします。サポート対象外のデータ型を含むカラムは複写されません。

## 使用法

### オプションを使用しない場合

- **pdb\_setreptable** をオプションなしで呼び出すと、Replication Agent はプライマリ・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リストを返します。
- 複写のためにマーク付けされたテーブルは、マーク付けされたオブジェクト・テーブルにリストされます。その他のユーザ・テーブルはすべてマーク解除テーブルと見なされます。

---

**注意：** Replication Agent のトランザクション・ログ・テーブルおよび隠しテーブルは、マーク解除テーブルのリストには含まれません。また、これらのデータベース・オブジェクトの同義語、ビュー、エイリアスも含まれません。

---

マーク解除または無効としてリストされたテーブルの場合、トランザクションは複写用に取り出されません。

### テーブルのマーク付け

- テーブルが複写用にマーク付けされ、**owner** モードが **on** に設定されている場合、複写定義の **"with primary table named"** 句または **"with all tables named"** 句に所有者名が含まれている必要があります。**owner** モードの設定と複写定義内の所有者名が一致しない場合、その複写定義は使用されません。

例：

- **pdb\_setreptable** の発行 (**owner** モードは **on** に設定):

```
pdb_setreptable "mytable", mark, owner
```

これにより **rs\_create\_repdef** コマンドはプライマリおよびレプリケート・データベース用にこの複写定義を生成します。Replication Server はこれを受信することを想定しています。

```
create replication definition ra$0xda_"mytable"
```

```
with primary at ora102.dco
with primary table named "qa4user"."mytable"
with replicate table named "qa4user"."mytable"
.
.
```

- **pdb\_setreptable** の発行 (**owner** モードは **off** に設定):

```
pdb_setreptable "mytable", mark
```

これにより **rs\_create\_repdef** コマンドはプライマリおよびレプリケート・データベース用にこの複製定義を生成します。Replication Server はこれを受信することを想定しています。

```
create replication definition ra$0xda_ "mytable"

with primary at oral02.dco
with primary table named "mytable"

with replicate table named "qa4user"."mytable"
.
.
.
```

- マーク付けされたテーブルの名前が変更されるか、そのテーブルが削除されて、元の名前で新しいテーブルが作成された場合、新しいテーブルには元のテーブルのマーク付けに関する情報はないため、そのテーブルを明示的にマーク付けする必要があります。マーク付け情報は、テーブル名ではなくオブジェクト ID 別に内部で保持されます。
- 以前にマーク付けされていたテーブル名を使用して新しいテーブルを作成した場合、**mark** オプションを指定して **pdb\_setreptable** コマンドを実行することにより、新しいテーブルにマーク付けする必要があります。これは、以前のテーブルのマークを解除していない場合でも行う必要があります。
- (UDB のみ) **mark** スクリプトが失敗した場合、そのスクリプトはファイル (mark.sql) に保存されます。このファイルは RAX-15\_¥inst\_name ¥scripts¥tablename ディレクトリにあります。テーブルはマーク付けされず、トランザクション・ログ・オブジェクトは更新されません。スクリプトは mark.sql ファイルを表示すると確認できます。

### テーブルのマーク解除

- **unmark** キーワードを使用してプライマリ・テーブルから複製のマーク付けを削除する場合、Replication Agent はそのテーブルの複製が無効になり、トランザクション・ログにそのテーブルに対する保留中の (未処理の) オペレーションがないことを確認します。複製が無効な場合、またはトランザクション・ログにそのテーブルに対する保留中のオペレーションがある場合、**pdb\_setreptable** はエラーを返します。
- **unmark** キーワードを使用してプライマリ・テーブルから複製のマーク付けを削除する場合、**force** キーワードも指定すると、複製が無効かどうかにかかわらず、プライマリ・テーブルから複製のマーク付けを即座に削除することができます。
- (UDB のみ) **unmark** スクリプトが失敗した場合、そのスクリプトはファイル (unmark.sql) に保存されます。このファイルは RAX-15\_¥inst\_name ¥tablename¥scripts ディレクトリにあります。テーブルはマーク解除されず、トランザクション・ログ・オブジェクトは削除されません。スクリプトは unmark.sql ファイルを表示すると確認できます。unmark スクリプトの実行

でいずれかのデータベース・オブジェクトに致命的なエラーが発生した場合、**pdb\_setreptable** コマンドは次のメッセージを返します。

```
Could not unmark the following objects: ...
See error log for details.
```

## キーワード

- 複数のキーワードとオプションを指定する場合は、それぞれカンマで区切る必要があります。カンマの前後にはオプションで空白を挿入できます。例：

```
pdb_setreptable all, unmark, force
```

- プライマリ・データベース内のテーブル名がキーワードと同じである場合、**table=string** を名前の先頭に追加することで識別できます。例：

```
pdb_setreptable table=unmark, mark
```

これはプライマリ・データベース名にも複写済みの名前にも当てはまりません。

## アクション・キーワード

- **pdb\_setreptable** を、有効なユーザ・テーブル名の後にアクション・キーワード (**mark**、**unmark**、**enable**、または **disable**) を指定して呼び出すと、指定のアクションが指定のテーブルに適用されます。
- **pdb\_setreptable** をテーブル名および **mark** キーワード指定で呼び出すと、プライマリ・データベース内にある指定のテーブルが複写のためにマーク付けされます。

**注意：** 個々のテーブルがマーク付けされている場合、所有者フィルタ・リストはチェックされません。これにより、ユーザは所有者フィルタ・リスト内に所有者があるテーブルをマーク付けすることができます。

- **pdb\_setreptable** をテーブル名および **unmark** キーワード指定で呼び出すと、プライマリ・データベース内にある指定のテーブルから複写のためのマーク付けが削除されます。

指定したテーブルのマークを即座に強制的に外す場合、複写がまだ有効になっているテーブルのマークを外す場合、またはエラーを無視して以前に失敗したマーク解除オペレーションを続行するためにスクリプトを強制的に実行する場合は、**force** キーワードを **unmark** キーワードの後に指定します。

- **pdb\_setreptable** をテーブル名および **enable** キーワード指定で呼び出すと、プライマリ・データベース内にある指定のマーク付けされたテーブルについて、複写が有効になります。
- **pdb\_setreptable** をテーブル名および **disable** キーワード指定で呼び出すと、プライマリ・データベース内にある指定のマーク付けされたテーブルについて、複写が無効になります。

**disable** スクリプトの実行でいずれかのデータベース・オブジェクトに致命的なエラーが発生した場合、**pdb\_setreptable** コマンドは次のメッセージを返します。

```
Could not disable the following objects: ...
See error log for details.
```

## all キーワード

- **pdb\_setreptable** を **all** キーワードおよびアクション・キーワード (**mark**、**unmark**、**enable**、または **disable**) 指定で呼び出すと、指定したアクションはプライマリ・データベース内のすべてのテーブル、またはプライマリ・データベース内のマーク付けされたすべてのテーブルに適用されます。
- **pdb\_setreptable** を **all** キーワードおよび **mark** キーワード指定で呼び出すと、プライマリ・データベース内にあるすべてのユーザ・テーブルが複写のためにマーク付けされます。

---

**注意：**所有者フィルタ・リスト内のユーザが所有するテーブルは、マーク付けされません。ただし、個々のテーブルはどれでもマーク付けできます。

---

- **pdb\_setreptable** を **all** キーワードおよび **unmark** キーワード指定で呼び出すと、プライマリ・データベース内にあるマーク付けされたすべてのテーブルから複写のためのマーク付けが削除されます。  
マーク付けされたすべてのテーブルのマークを即座に強制的に外す場合、複写がまだ有効になっているテーブルのマークを外す場合、またはエラーを無視して以前に失敗したマーク解除オペレーションを続行するためにスクリプトを強制的に実行する場合は、**force** キーワードを **unmark** キーワードの後に指定します。
- **pdb\_setreptable** を **all** キーワードおよび **enable** キーワード指定で呼び出すと、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について、複写が有効になります。
- **pdb\_setreptable** を **all** キーワードおよび **disable** キーワード指定で呼び出すと、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について、複写が無効になります。

## 複写定義のタイプ

- **pdb\_setreptable** コマンドの使用法は、Replication Server で作成した複写定義のタイプによって異なります。テーブル複写定義なしでデータベース複写定義を作成した場合、**pdb\_setreptable** コマンドの複写プロシージャは、レプリケート・データベース内のテーブルを参照します。ただし、テーブル複写定義を作成した場合、**pdb\_setreptable** コマンドのレプリケート・テーブルは、テーブル複写定義の名前を参照し、レプリケート・データベース内のテーブルにマップされる必要があるのは、テーブル複写定義です。
- テーブル複写定義が存在せず、複写前に追加もされないが、データベース複写定義のみが存在する場合、これらのコマンドを使用してテーブルを複写のためにマーク付けします。
  - レプリケート・データベース内の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
```

ここで、**pdb\_table** は複製のためにマーク付けするプライマリ・データベース内のテーブルの名前です。

- レプリケート・データベース内のテーブルがプライマリ・データベース内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。

- テーブル複製定義が存在するか、複製前に追加される場合、データベース複製定義が存在するかどうかにかかわらず、これらのコマンドを使用してテーブルを複製のためにマーク付けします。

- テーブル定義内のプライマリ・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
```

レプリケート・データベース内のテーブルもテーブル複製定義と同じ名前の場合、プライマリ Replication Server の複製定義で **with all tables named** 句を使用できます。例：

```
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がテーブル複製定義内のプライマリ・テーブルとは異なる名前の場合、テーブル複製定義はレプリケート・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。例：

```
create replication definition my_table_repdef with primary at
data_server.database
with primary table named pdb_table
with replicate table name rep_table ...
```

- テーブル複製定義の名前がプライマリ・データベース内のテーブルと異なる場合は、以下を使用します。

```
pdb_setreptable pdb_table, rdpri_table, mark
```

ここで、**rdpri\_table** は複製定義内のプライマリ・テーブルの名前です。

レプリケート・データベース内のテーブルもテーブル複製定義内のプライマリ・テーブルと同じ名前の場合、プライマリ Replication Server の複製定義で **with all tables named** 句を使用できます。例：

```
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がテーブル複製定義内のプライマリ・テーブルとは異なる名前の場合、テーブル複製定義はレプリケート・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。例：

```
create replication definition my_table_repdef
with primary at data_server.database
```

```
with primary table named rdpri_table  
with replicate table name rep_table ...
```

### 複写済みの名前と owner キーワード

- **pdb\_setreptable** を、プライマリ・テーブル名と複写済みの名前に続けて **mark** キーワードを指定して呼び出すと、プライマリ・テーブルは指定した複写済みの名前前で複写のためにマーク付けされます。

複写済みの名前を指定することで、プライマリ・テーブルとは異なる名前のレプリケート・データベース内のテーブルにトランザクションを複写できます。

---

**注意：** **pdb\_setreptable** コマンドで指定する複写済みの名前は、プライマリ・データベース・コネクションの Replication Server 複写定義の **with all tables named** 句で指定したテーブル名と一致する必要があります。Replication Agent は複写定義を確定化できませんが、それが存在しない場合、または **with all tables named** 句が **pdb\_setreptable** で指定した複写済みの名前と一致しない場合、プライマリ・テーブルからの複写は失敗します。

---

- また、**owner** キーワードを **mark** キーワードの後で指定すると、プライマリ・テーブルに対するオペレーションが複写された場合に、プライマリ・テーブルの所有者名が `owner.tablename` の形式でレプリケート・テーブル名に付加されます。

---

**注意：** レプリケート所有者名の付いた所有者指定のレプリケート・テーブル名を使用する場合、**owner** キーワードを **pdb\_setreptable** コマンドで使用します。修飾されていないレプリケート・テーブル名を指定すると、プライマリ・テーブル所有者名がレプリケート・テーブル名とともに LTL に送信されます。

---

### サポート対象外のデータ型

- (UDB および Oracle) 複写がサポートされていないデータ型のカラムがテーブルに含まれている場合、**pdb\_setreptable** コマンドに **mark** キーワードを指定して実行すると、次のようなエラーが発生して失敗する可能性があります。

```
Command <pdb_setreptable> failed - Table <MYTABLE> could not be  
marked because:The table contains an  
unsupported data type.
```

サポート対象外のデータ型のカラムを複写から除外して、テーブルを強制的にマーク付けするには、**force** キーワードを **pdb\_setreptable** コマンドに追加します。

### エラー

- Replication Agent トランザクション・ログがプライマリ・データベース (UDB) に存在しない場合、または RASD が初期化されない場合 (Oracle および Microsoft SQL Server)、**pdb\_setreptable** コマンドはエラーを返します。

- 指定したテーブル名がプライマリ・データベースに存在しない場合、**pdb\_setreptable** コマンドはエラーを返します。
- 指定したプライマリ・テーブル名がプライマリ・データベースに存在しない場合、**pdb\_setreptable** コマンドはエラーを返します。
- enable スクリプトの実行でいずれかのデータベース・オブジェクトに致命的なエラーが発生した場合、**pdb\_setreptable** コマンドは次のメッセージを返します。

```
Could not enable the following objects: ...
See error log for details.
```

### カラム名のスペース

- スペースのあるカラム名を含むテーブルを複製するには、**structured\_tokens** を **true** に設定する必要があります。

### エイリアス

- エイリアスまたは同義語を **pdb\_setreptable** コマンドでプライマリ・テーブルとして指定すると、エイリアスまたは同義語が参照する実際のテーブルが処理対象になります。実際のテーブル名は、プライマリ Replication Server に送られるテーブル名です。

### 参照：

- [pdb\\_setrepcol](#) (31 ページ)
- [pdb\\_setrepproc](#) (45 ページ)
- [ra\\_config](#) (86 ページ)

## **pdb\_skip\_op**

処理中にスキップするオペレーションのリストを返すか、オペレーション・リストに対してオペレーションの追加または削除を行います。レコード・ロケータの形式はデータベースによって異なります。

### **構文**

```
pdb_skip_op [
{
  add , { locator | { scn, thread, rba | lsn } }
|
  remove, { all | locator | { scn, thread, rba | lsn } }
}
]
```

## パラメータ

- **add** – 指定された ID を、スキップするレコードの識別子のリストに追加します。
- **remove** – 指定された ID を、スキップする識別子のリストから削除します。
- **locator** – スキップするオペレーションを識別する LTM ロケータのリストからのロケータ・キーワード。
- **scn** – (Oracle の場合) system change number (SCN) キーワードで、スキップする指定のログ・レコードが識別されます。
- **thread** – (Oracle の場合) スキップするオペレーションの redo ログ・スレッドの thread キーワード。
- **rba** – (Oracle の場合) スキップするログ・レコードの record byte address (RBA) キーワード。
- **lsn** – (UDB の場合) スキップするログ・レコードの system change number (LSN) キーワード。
- **all** – スキップする識別子のリストですべての ID を追加または削除できます。

## 例

- **例 1 –**

```
pdb_skip_op
```

このコマンドをパラメータなしで実行すると、スキップするレコードの識別子のリストが返されます。

- **例 2 –**

```
pdb_skip_op add, id
```

このコマンドは、スキップする識別子のリストに ID を追加します。

- **例 3 –**

```
pdb_skip_op remove, id
```

このコマンドは、スキップする識別子のリストから ID を削除します。

- **例 4 –**

```
pdb_skip_op remove, all
```

このコマンドは、スキップする識別子のリストからすべての ID を削除します。

- **例 5 –**

```
pdb_skip_op add, locator
```



このコマンドは、ロケーションで参照されるオペレーションを、スキップする識別子のリストに追加します。

## 使用法

- **pdb\_skip\_op** コマンドでは、問題のあるレコードをスキップできるので、Replication Agent の再初期化を回避することができます。
- スキップしたレコードは、警告メッセージとしてシステム・ログに書き込まれます。
- **pdb\_skip\_op** コマンドが有効なのは、Replication Agent が Admin 状態または Replication Down 状態のときだけです。
- 識別子の形式はデータベースによって異なります。

- Microsoft SQL Server の場合、識別子は RBA で、形式は次のとおりです。

```
virtualfileseqno:pageid:operationsseqno
```

構文の説明は次のとおりです。

- **virtualfileseqno** は仮想ログ・シーケンス番号 (LSN) です。
- **pageid** はページ番号です。
- **operationsseqno** はオペレーションのシーケンス番号です。

LSN は、Microsoft SQL Server DBCC コマンドを使用して取得することも、ロケータから解析することもできます。例：

- 10 進形式の LSN：

```
489:112:27
```

- 16 進形式の LSN：

```
0x00000489:00000112:0027
```

- Oracle の場合、識別子には system change number (SCN)、redo ログ・スレッド、およびレコード・バイト・アドレス (RBA) が含まれます。形式は次のとおりです。

```
wrap.base.sub, thread, lsn.blknum.blkoffset
```

構文の説明は次のとおりです。

- **wrap** は SCN ラップ番号です。
- **base** は SCN ベース番号です。
- **sub** は SCN サブインデックスです。
- **thread** はオペレーションが発生した redo ログ・スレッド番号です。
- **lsn** は RBA ログ・シーケンス番号です。
- **blknum** は RBA ブロック番号です。
- **blkoffset** はレコードが存在するブロックにオフセットした RBA です。

値は、上に示したとおり引用符で囲み、それぞれの項目をピリオドで区切って **pdb\_skip\_op** コマンドで一緒に指定する必要があります。

例：

## コマンド・リファレンス

```
'0000.012345678.00', '1', '0012.0000444.0000123'
```

すべての値は、次のように識別子の前に "0x" を付けることにより 16 進値として記述することができます。

```
'0x000c.00001bc.000007b'
```

または

```
'0x000.00BC614E'
```

### 参照：

- ra\_helplocator (109 ページ)
- ra\_locator (116 ページ)

## pdb\_thread\_filter

(Oracle RAC のみ) Oracle インスタンス redo ログ・スレッドまたは複製中のスレッドのすべてのアクティビティをフィルタし、フィルタされたスレッドのリストを表示します。

### 構文

```
pdb_thread_filter [  
  add, thread_id  
  |  
  remove, { thread_id | all }  
]
```

### パラメータ

- **add** – 指定されたスレッド ID をフィルタされているスレッドのリストに追加します。
- **remove** – 指定されたスレッド ID またはすべてのスレッド ID をフィルタされているスレッドのリストから削除します。
- **thread\_id** – フィルタされているスレッドのリストに追加されるまたはリストから削除されるスレッド ID。
- **all** – フィルタされているスレッドのリストからすべてのスレッドを削除します。

### 例

- **例 1** –

```
pdb_thread_filter add, 1
```

スレッド 1 のフィルタ・アクティビティ。

- 例 2 –

```
pdb_thread_filter remove, 1
```

フィルタされているスレッドのリストからスレッド 1 を削除します。

- 例 3 –

```
pdb_thread_filter remove, all
```

フィルタされているスレッドのリストからすべてのスレッドを削除します。

- 例 4 –

```
pdb_thread_filter
```

フィルタされているスレッドのリストを表示します。

## 使用法

- **pdb\_thread\_filter** を正しく使用しないとデータが消失することがあります。
- **pdb\_thread\_filter** は、フィルタされているスレッドが実行されていないときに、Sybase サポート・センタの指示のもとでのみ使用してください。

## pdb\_truncate\_xlog

Replication Agent プライマリ・データベースのトランザクション・ログをオンデマンドでトランケートします。

- **pdb\_truncate\_xlog** の動作は、設定パラメータ **pdb\_include\_archives** の値によって変わります。
  - **pdb\_include\_archives** が **false** の場合、Replication Agent によって処理済みのオンライン redo ログをアーカイブするアーカイブ・プロセスがトリガされます。
  - **pdb\_include\_archives** が **true** の場合、**pdb\_archive\_path** で指定されたパスから古いアーカイブ redo ログ・ファイルが削除されます。

---

**注意：** **pdb\_archive\_path** ディレクトリからの古いアーカイブ・ログ・ファイルのトランケーションは、**pdb\_archive\_remove** パラメータが **true** に設定されている場合にのみ実行されます。

---

- UDB では、**pdb\_archive\_remove** が **true** に設定されている場合、アーカイブされたトランザクション・ログ・ファイルは、**pdb\_archive\_path** 設定パラメータで指定されたロケーションから削除されます。プライマリ・トランザクション・ログ・ファイルは、Replication Agent トランケーション・ポイント・ロケータで参照された **lsn** を含むログ・ファイルの直前のログ・ファイルまで削除されます。
- Microsoft SQL Server の場合、プライマリ・データベース・ログはトランケートされます。Replication Server の LTM ロケータに基づき、Replication Agent はプ

ライマリ・データベースに問い合わせ、トランケートできる最新のトランザクション ID を取得し、最新のトランザクションの前にトランザクション・ログ領域を再使用可能としてマーク付けします。

**pdb\_truncate\_xlog** の実行時に Replication Agent が各種のデータベースにどのように影響するかの詳細については、『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。

### 構文

```
pdb_truncate_xlog
```

### 使用法

- **pdb\_truncate\_xlog** が呼び出されると、Replication Agent は、プライマリ Replication Server から受け取った最新のトランケーション・ポイントに基づいて、プライマリ・データベースのトランザクション・ログを即座にトランケートします。トランケーション・ポイントは、LTM ロケータに含まれる情報の一部です。
- プライマリ Replication Server から LTM ロケータを更新するには、**ra\_locator** コマンドを使用します。
- **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** コマンドの成功または失敗を判断するには、Replication Agent のシステム・ログを調べる必要があります。
- Replication Agent のプライマリ・データベース・ログが存在しない場合、またはコネクション障害が発生した場合、**pdb\_truncate\_xlog** コマンドはエラー・メッセージを返します。
- **ra\_config** コマンドを使用して必要な自動トランケーションのタイプを指定できます。トランザクション・ログのサイズを管理するのに自動トランケーションでは十分でない場合は、**pdb\_truncate\_xlog** コマンドを使用してトランザクション・ログをトランケートできます。
- **pdb\_truncate\_xlog** コマンドが有効なのは、Replication Agent のインスタンスが Admin 状態、Replicating 状態、または Replication Down 状態のときです。

### 参照：

- ra\_config (86 ページ)
- ra\_locator (116 ページ)

## pdb\_version

プライマリ・データ・サーバのタイプとバージョンを返します。

### 構文

```
pdb_version
```

### 使用法

返される実際の結果は、プライマリ・データ・サーバのタイプによって異なります。

### 参照：

- ra\_version (145 ページ)
- ra\_version\_all (145 ページ)

## pdb\_xlog

Replication Agent システム・オブジェクトの名前を返します。さらに、プライマリ・データベースに Replication Agent システム・オブジェクトを作成するか、またはプライマリ・データベースから Replication Agent システム・オブジェクトを削除します。

**注意：** `ra_admin` および `ra_locator` を、廃止された `pdb_xlog` の代わりに使用します。

Oracle および Microsoft SQL Server の場合、`pdb_xlog` は Replication Agent がプライマリ・データベースからシステム・データを取得するためのパーミッションが有効であることを確認します。さらに、プライマリ・データベースの状態を確認して、アーカイブ処理がオンかオフかを判断し、プライマリ・データベースからのシステム・データを RASD にロードします。

### 構文

```
pdb_xlog [{ init | create | remove } [, force ] | move_truncpt ]
```

### パラメータ

- **init** – プライマリ・データベースに Replication Agent システム・オブジェクトを作成するためのキーワード。
- **create** – トランザクション・ログを作成するためのキーワード。廃止されました。代わりに **init** キーワードを使用します。
- **remove** – トランザクション・ログを削除するためのキーワード。

- **force – remove** オペレーションまたは **init** オペレーションを参照するキーワード。
- **move\_truncpt** – トランケーション・ポイントを移動するキーワード。

### 例

#### 例 1 –

```
pdb_xlog init
```

このコマンドによって Replication Agent が初期化され、必要なトランザクション・ログ・ベース・コンポーネントが作成されます。

```
pdb_xlog init, force
```

このコマンドによって Replication Agent が再初期化され、必要なトランザクション・ログ・ベース・コンポーネントが作成または再ロードされます。

```
pdb_xlog remove
```

このコマンドによって Replication Agent トランザクション・ログ・ベース・コンポーネントが削除されます。

```
pdb_xlog remove, force
```

このコマンドによって Replication Agent トランザクション・ログ・ベース・コンポーネントがすべて削除され、削除中に発生する個々のエラーはすべて無視されます。

```
pdb_xlog move_truncpt
```

このコマンドは、トランザクション・ログのトランケーション・ポイントを現在のトランザクション・ログの末尾に移動します。

### 使用法

- **pdb\_xlog** コマンドをオプションなしで呼び出すと、プライマリ・データベースにあるすべての Replication Agent システム・オブジェクトの実際の名前(同義語やエイリアスではなく)が返されます。Oracle および Microsoft SQL Server の場合、Replication Agent を初期化すると、コンポーネント名とプライマリ・データベース・インスタンス名が返されます。

Replication Agent オブジェクト名の詳細については、『Replication Agent プライマリ・データベース・ガイド』で特定のプライマリ・データ・サーバに関するセクションを参照してください。

- **pdb\_xlog** をオプションなしで呼び出し、Replication Agent システム・オブジェクトがプライマリ・データベースに存在しない場合、または RASD が初期化されていない場合 (Oracle および Microsoft SQL Server)、このコマンドは情報を返しません。

- **pdb\_xlog** を **init** キーワード指定で呼び出すと、トランケーション・ポイントがプライマリ・データベース・トランザクション・ログの末尾に設定されます。

---

**注意：** Microsoft SQL Server の場合、**pdb\_xlog init** プロセス中、Replication Agent は **pds\_dac\_port\_number** を使用して Microsoft SQL Server に接続できます。『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。

---

- **pdb\_xlog** を **init** キーワードおよび **force** キーワード指定で呼び出すと、Replication Agent がまだ初期化されていない場合、トランケーション・ポイントはログの末尾に移動されます。ただし、Replication Agent が初期化済みである場合、トランケーション・ポイントは移動されません。

---

**注意：** **pdb\_xlog init** を **force** キーワードとともに使用するのには、Sybase テクニカル・サポートからアドバイスを受けた場合のみにしてください。

---

- **pdb\_xlog** を **move\_truncpt** キーワード指定で呼び出すと、トランケーション・ポイントは Replication Agent コンポーネントへの変更なしでログの末尾 (Oracle の場合、現在のオンライン redo ログの末尾) に移動されます。Replication Agent が初期化されていない場合、**move\_truncpt** オプションの効果はありません。

---

**注意：** **move\_truncpt** オプションによって設定されたログ内のロケーションよりも前に発生するログ開始ポイントを Replication Server が要求することを回避するには、Replication Server のプライマリ・コネクションの LTM ロケータ値をゼロにする必要があります。プライマリ・データベース・コネクションに対して Replication Server System Database (RSSD) コマンド **rs\_zeroltm** を実行することにより、LTM ロケータの値をゼロにします。

---

**pdb\_xlog move\_truncpt** を使用してセカンダリ・トランケーション・ポイントをプライマリ・データベース・トランザクション・ログの末尾に移動すると、ログに記録された DDL コマンドをスキップする危険性があります。DDL コマンドは、Replication Agent システム・データベース (RASD) 内に格納された情報の更新のために Replication Agent によって使用された可能性があります。一部のログ・レコードの処理をスキップしたために RASD の内容が正しくない場合は、コマンド **pdb\_xlog init, force** を使用して RASD 内のすべてのスキーマ情報を強制的に更新できます。RASD に格納された単一のオブジェクトのスキーマのみに関心がある場合は、そのオブジェクトのみのマークを外して、再度マーク付けします。これにより、そのオブジェクトのスキーマが RASD に再度読み込まれます。

- **pdb\_xlog** を **init** キーワード指定で呼び出すと、Replication Agent は次のように動作します。
  - プライマリ・データベースの Replication Agent テーブルとプロシージャを作成する SQL スクリプトを生成します。

- 生成されたスクリプトを `partinit.sql` というファイル (`RAX-15_5¥inst_name¥scripts¥xlog` ディレクトリ内) に保存します。  
`inst_name` は Replication Agent インスタンスの名前です。

---

**注意：** `pdb_auto_run_scripts` 設定パラメータの値が `false` の場合、`partinit.sql` スクリプトは保存されますが実行されません。ただし、スクリプトを手動で実行することはできません。Replication Agent の初期化を完了するには、まず `pdb_auto_run_scripts` を `true` に設定し、次に `pdb_xlog init` コマンドを再実行します。

---

- スクリプトを実行してプライマリ・データベースに Replication Agent システム・オブジェクトを作成します (`pdb_auto_run_scripts` 設定パラメータの値が `true` の場合)。
- スクリプトが正常に完了した後、`partinit.sql` ファイルを `RAX-15_5¥inst_name¥scripts¥xlog¥installed` ディレクトリに移動します。
- `create` スクリプトが失敗した場合、そのスクリプトはファイル `partinit.sql` (`RAX-15_5¥inst_name¥scripts¥xlog` ディレクトリ内) に保存され、トランザクション・ログは作成されません。スクリプトは `partinit.sql` ファイルを表示すると確認できます。
- `pdb_xlog` を `init` キーワード指定で呼び出し、Replication Agent オブジェクトがプライマリ・データベースにすでに存在する場合、または RASD が初期化済みの場合 (Oracle および Microsoft SQL Server)、`pdb_xlog` はエラー・メッセージを返します。
- `pdb_xlog` を `remove` キーワード指定で呼び出すと、Replication Agent は次のように動作します。
  - UDB では、`pdb_xlog remove` がプライマリ・データベースから JAR ファイルを削除します (JAR は `pdb_xlog init` コマンドによってインストールされます)。Replication Agent for UDB の初期化を解除するには、`pdb_xlog remove` を使用し、トランザクション・ストアード・プロシージャと JAR をデータベースから削除する必要があります。
  - プライマリ・データベース内の Replication Agent システム・オブジェクトに必要なテーブルとプロシージャを削除する SQL スクリプトを生成します。
  - 生成されたスクリプトを `partdeinit.sql` というファイル (`RAX-15_5¥inst_name¥scripts¥xlog` ディレクトリ) に保存します。ここで、`inst_name` は Replication Agent インスタンスの名前です。

---

**注意：** `pdb_auto_run_scripts` 設定パラメータの値が `false` の場合、`partdeinit.sql` スクリプトは保存されますが自動的に実行されません。スクリプトを手動で実行することはできません。Replication Agent の初



期化解除を完了するには、まず **pdb\_auto\_run\_scripts** を true に設定し、次に **pdb\_xlog remove** コマンドを再実行します。

- スクリプトを実行してプライマリ・データベースから Replication Agent オブジェクトを削除します (**pdb\_auto\_run\_scripts** 設定パラメータの値が true の場合)。
- スクリプトが正常に完了した後、partdeinit.sql ファイルを RAX-15\_5¥inst\_name¥scripts¥xlog¥installed ディレクトリに移動します。
- スクリプトが失敗した場合、そのスクリプトはファイル partdeinit.sql (RAX-15\_5¥inst\_name¥scripts¥xlog ディレクトリ) に保存され、Replication Agent オブジェクトはプライマリ・データベースから削除されません。スクリプトは partdeinit.sql ファイルを表示すると確認できます。
- **pdb\_xlog** を **remove** キーワードの後に **force** キーワード指定で呼び出すと、partdeinit.sql スクリプトはエラーが発生しても実行を続けます。 **force** キーワードは、前の **remove** オペレーションが失敗し、partdeinit.sql スクリプトがエラーによって終了した場合に役立ちます。
- **pdb\_xlog** を **remove** キーワード指定で呼び出し、Replication Agent オブジェクトがプライマリ・データベースに存在しない場合、または RASD が初期化されていない場合 (Oracle または Microsoft SQL Server)、**pdb\_xlog** はエラー・メッセージを返します。
- **pdb\_xlog** を **remove** キーワード指定で呼び出し、プライマリ・データベース内の任意のオブジェクトがまだ複写のためにマーク付けされている場合、**pdb\_xlog** はエラー・メッセージを返します。  
**pdb\_setreproc** コマンドおよび **pdb\_setreptable** コマンドを使用して、プライマリ・データベース内のどのストアド・プロシージャとテーブルがまだマーク付けされているかを調べることができます。また、**pdb\_setrepddl** コマンドを使用して DDL が有効かどうかを調べることもできます。  
プライマリ・データベース内でオブジェクトがマーク付けされている場合も、**pdb\_xlog** を **remove** キーワードの後に **force** キーワードを指定して使用し、マーク付けされたすべてのオブジェクトのマークを外してからトランザクション・ログ・オブジェクトを削除できます。
- **pdb\_xlog** をオプションなしで呼び出すと、Replication Agent インスタンスが Admin、Replicating、または Replication Down の状態にあるときにコマンドが有効になります。
- **pdb\_xlog** を **init** キーワードまたは **remove** キーワード指定で呼び出すと、コマンドは Replication Agent インスタンスが Admin または Replication Down の状態にあるときにのみ有効になります。
- **pdb\_xlog init** コマンドは、これらの権限が **pds\_username** に付与されていることを確認します。

- EXECUTE\_CATALOG\_ROLE
- SELECT ON V\_\$LOGMNR\_CONTENTS
- SELECT ON V\_\$LOGMNR\_LOGS

これらの権限は、`ra_dumptran` コマンドおよび `ra_helpop` コマンドが正しく機能するために必要です。これらの権限は、複製には必要なく、デバッグおよびトラブルシューティングに使用する `ra_dumptran` コマンドおよび `ra_helpop` コマンドを使用する場合にのみ必要です。`pdb_xlog init` を呼び出すときにこれらの権限が付与されていない場合は、警告メッセージが返され、Replication Agent ログ・ファイルに記録されます。

- Replication Agent トランザクション・ログの詳細については、『Replication Agent プライマリ・データベース・ガイド』でご使用のプライマリ・データ・サーバに関するセクションを参照してください。

### 参照：

- `pdb_setrepcol` (31 ページ)
- `pdb_setrepproc` (45 ページ)
- `pdb_setreptable` (56 ページ)
- `ra_admin` (81 ページ)
- `ra_locator` (116 ページ)

## quiesce

Replicating 状態で処理しているすべての Replication Agent を停止し、Replication Agent インスタンスを Admin 状態にします。

### 構文

```
quiesce
```

### 使用法

- **quiesce** コマンドを呼び出すと、Replication Agent インスタンス内で現在行われているすべての複製処理が終了されます。
- Log Reader コンポーネントは、スキャンがログの最後に到達すると、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。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

データベースへのすべての接続を解放し、プライマリ Replication Server (および、接続されている場合は RSSD) への接続を削除します。

- Replication Agent インスタンスは Replicating 状態から Admin 状態になります。
- **quiesce** コマンドの呼び出し時に Replication Agent の内部キューが満杯である場合、クワイース処理の完了に時間がかかる可能性があります。また、Replication Agent インスタンスが Admin 状態への移行を完了するのが遅れる場合があります。
- **quiesce** コマンドはプライマリ・ログ内のすべてのデータが読み込まれ、Replication Server に送られるまで待ってから、Replication Agent を Admin 状態に移行します。
- Replication Agent インスタンスが Admin 状態の場合に **quiesce** コマンドを実行すると、エラーが返されます。
- **quiesce** コマンドは Replication Agent インスタンスが Replicating 状態の場合にのみ有効です。

---

**注意：** **suspend** コマンドのアクションは **quiesce** コマンドと同様ですが、**suspend** コマンドでは Replication Agent の処理が即座に停止され、内部キュー内のすべてのデータがフラッシュされることが異なります。

---

#### 参照：

- ra\_status (138 ページ)
- resume (152 ページ)
- shutdown (161 ページ)
- suspend (162 ページ)

## ra\_admin

Replication Agent システム・オブジェクトの名前を返します。さらに、プライマリ・データベースに Replication Agent システム・オブジェクトを作成するか、またはプライマリ・データベースから Replication Agent システム・オブジェクトを削除します。

Oracle および Microsoft SQL Server の場合、**ra\_admin** は Replication Agent がプライマリ・データベースからシステム・データを取得するためのパーミッションが有効であることを確認します。Oracle の場合、**ra\_admin** はさらに、プライマリ・データベースの状態を確認し、アーカイブ処理がオンかオフかを判断し、プライマリ・データベースからのシステム・データを RASD にロードします。

---

**注意：** **ra\_admin** および **ra\_locator** を、廃止された **pdb\_xlog** の代わりに使用します。

---

## 構文

```
ra_admin [ { init | refresh | deinit[, force] } ]
```

## パラメータ

- **init** – プライマリ・データベースに Replication Agent システム・オブジェクトを作成するためのキーワード。
- **refresh** – Replication Agent を再初期化し、必要なトランザクション・ログ・ベース・コンポーネントが作成または再ロードされるようにするキーワード。
- **deinit** – トランザクション・ログを削除するためのキーワード。
- **force** – **deinit** オペレーションを参照するキーワードです。

## 例

### 例 1 –

```
ra_admin init
```

このコマンドによって Replication Agent が初期化され、必要なトランザクション・ログ・ベース・コンポーネントが作成されます。

### 例 2 –

```
ra_admin refresh
```

このコマンドによって Replication Agent が再初期化され、必要なトランザクション・ログ・ベース・コンポーネントが作成または再ロードされます。

### 例 3 –

```
ra_admin deinit
```

このコマンドによって Replication Agent トランザクション・ログ・ベース・コンポーネントが削除されます。

### 例 4 –

```
ra_admin deinit, force
```

このコマンドによって Replication Agent トランザクション・ログ・ベース・コンポーネントがすべて削除され、削除中に発生する個々のエラーはすべて無視されます。

## 使用法

- **ra\_admin** コマンドをオプションなしで呼び出すと、プライマリ・データベースにあるすべての Replication Agent システム・オブジェクトの実際の名前 (同義語やエイリアスではなく) を返します。Oracle および Microsoft SQL Server の場合、Replication Agent を初期化すると、コンポーネント名とプライマリ・データベース・インスタンス名が返されます。

Replication Agent オブジェクト名の詳細については、『Replication Agent プライマリ・データベース・ガイド』で特定のプライマリ・データ・サーバに関するセクションを参照してください。

- **ra\_admin** をオプションなしで呼び出し Replication Agent システム・オブジェクトがプライマリ・データベースに存在しない場合、または RASD が初期化されていない場合 (Oracle および Microsoft SQL Server)、このコマンドは情報を返しません。
- **ra\_admin** を **init** キーワード指定で呼び出すと、トランケーション・ポイントがプライマリ・データベース・トランザクション・ログの末尾に移動します。

---

**注意：** Microsoft SQL Server の場合、**ra\_admin init** プロセス中、Replication Agent は **pds\_dac\_port\_number** を使用して Microsoft SQL Server に接続できます。『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。

---

- **ra\_admin** を **init** キーワード指定で呼び出すと、Replication Agent は次のように動作します。
  - プライマリ・データベースの Replication Agent テーブルとプロシージャを作成する SQL スクリプトを生成します。
  - 生成されたスクリプトを `partinit.sql` というファイル (RAX-15\_5¥inst\_name¥scripts¥xlog ディレクトリ内) に保存します。inst\_name は Replication Agent インスタンスの名前です。

---

**注意：** **pdb\_auto\_run\_scripts** 設定パラメータの値が `false` の場合、`partinit.sql` スクリプトは保存されますが実行されません。ただし、スクリプトを手動で実行することはできません。Replication Agent の初期化を完了するには、まず **pdb\_auto\_run\_scripts** を `true` に設定し、次に **ra\_admin init** コマンドを再実行します。

---

- スクリプトを実行してプライマリ・データベースに Replication Agent システム・オブジェクトを作成します (**pdb\_auto\_run\_scripts** 設定パラメータの値が `true` の場合)。
- スクリプトが正常に完了した後、`partinit.sql` ファイルを RAX-15\_5¥inst\_name¥scripts¥xlog¥installed ディレクトリに移動します。
- `create` スクリプトが失敗した場合、そのスクリプトはファイル `partinit.sql` (RAX-15\_5¥inst\_name¥scripts¥xlog ディレクトリ内) に保存され、トランザクション・ログは作成されません。スクリプトは `partinit.sql` ファイルを表示すると確認できます。
- **ra\_admin** を **init** キーワード指定で呼び出し、Replication Agent オブジェクトがプライマリ・データベースにすでに存在する場合、または RASD が初期化済みの

場合 (Oracle および Microsoft SQL Server)、**ra\_admin** はエラー・メッセージを返します。

- **ra\_admin** を **refresh** キーワード指定で呼び出すと、Replication Agent がまだ初期化されていない場合、トランケーション・ポイントはログの末尾に移動されます。ただし、Replication Agent が初期化済みである場合、トランケーション・ポイントは移動されません。

---

**注意：** Sybase サポート・センタからアドバイスを受けた場合にのみ、**ra\_admin refresh** を使用してください。

---

- **ra\_admin** を **deinit** キーワード指定で呼び出すと、Replication Agent は次のように動作します。
  - UDB では、**ra\_admin** がプライマリ・データベースから JAR ファイルを削除します (JAR は **ra\_admin init** コマンドによってインストールされます)。UDB に対して Replication Agent の初期化を解除するには、**ra\_admin deinit** を使用し、トランケーション・ストア・プロシージャと JAR をデータベースから削除する必要があります。
  - プライマリ・データベース内のシステム・オブジェクトに必要なテーブルとプロシージャを削除する SQL スクリプトを生成します。
  - 生成されたスクリプトを `partdeinit.sql` というファイル (RAX-15\_5¥inst\_name¥scripts¥xlog ディレクトリ) に保存します。ここで、`inst_name` は Replication Agent インスタンスの名前です。

---

**注意：** **pdb\_auto\_run\_scripts** 設定パラメータの値が `false` の場合、`partdeinit.sql` スクリプトは保存されますが自動的に実行されません。スクリプトを手動で実行することはできません。Replication Agent の初期化解除を完了するには、まず **pdb\_auto\_run\_scripts** を `true` に設定し、次に **ra\_admin deinit** コマンドを再実行します。

---

- スクリプトを実行してプライマリ・データベースから Replication Agent オブジェクトを削除します (**pdb\_auto\_run\_scripts** 設定パラメータの値が `true` の場合)。
- スクリプトが正常に完了した後、`partdeinit.sql` ファイルを `RAX-15_5¥inst_name¥scripts¥xlog¥installed` ディレクトリに移動します。
- スクリプトが失敗した場合、そのスクリプトはファイル `partdeinit.sql` (RAX-15\_5¥inst\_name¥scripts¥xlog ディレクトリ) に保存され、Replication Agent オブジェクトはプライマリ・データベースから削除されません。スクリプトは `partdeinit.sql` ファイルを表示すると確認できます。
- **ra\_admin** を **deinit** キーワードの後に **force** キーワード指定で呼び出すと、`partdeinit.sql` スクリプトはエラーが発生しても実行を続けます。 **force**

キーワードは、前の remove オペレーションが失敗し、partdeinit.sql スクリプトがエラーによって終了した場合に役立ちます。

- **ra\_admin** を **deinit** キーワード指定で呼び出し、Replication Agent オブジェクトがプライマリ・データベースにすでに存在する場合、または RASD が初期化済みの場合 (Oracle および Microsoft SQL Server)、**ra\_admin** はエラー・メッセージを返します。
- **ra\_admin** を **deinit** キーワード指定で呼び出し、プライマリ・データベース内の任意のオブジェクトがまだ複製のためにマーク付けされている場合、**ra\_admin** はエラー・メッセージを返します。

**pdb\_setrepproc** コマンドおよび **pdb\_setreptable** コマンドを使用して、プライマリ・データベース内のどのストアド・プロシージャとテーブルがまだマーク付けされているかを調べることができます。また、**pdb\_setreppddl** コマンドを使用して DDL が有効かどうかを調べることができます。

プライマリ・データベース内でオブジェクトがマーク付けされている場合も、**ra\_admin** を **deinit** キーワードの後に **force** キーワードを指定して使用し、マーク付けされたすべてのオブジェクトのマーク付けを解除してからトランザクション・ログ・オブジェクトを削除できます。

- **ra\_admin** をオプションなしで呼び出すと、Replication Agent インスタンスが、Admin、Replicating、または Replication Down の状態にあるときにコマンドが有効になります。
- **ra\_admin** を **init** キーワードまたは **deinit** キーワード指定で呼び出すと、コマンドは Replication Agent インスタンスが Admin または Replication Down の状態にある場合にのみ有効になります。
- **ra\_admin init** コマンドは、これらの権限が **pds\_username** に付与されたことを確認します。
  - EXECUTE\_CATALOG\_ROLE
  - SELECT ON V\_\$LOGMNR\_CONTENTS
  - SELECT ON V\_\$LOGMNR\_LOGS

これらの権限は、**ra\_dumptran** コマンドおよび **ra\_helpop** コマンドが正しく機能するために必要です。これらの権限は、複製には必要なく、デバッグおよびトラブルシューティングに使用する **ra\_dumptran** コマンドおよび **ra\_helpop** コマンドを使用する場合にのみ必要です。**ra\_admin init** を呼び出すときにこれらの権限が付与されていない場合は、警告メッセージが返され、Replication Agent ログ・ファイルに記録されます。

- Replication Agent トランザクション・ログの詳細については、『Replication Agent プライマリ・データベース・ガイド』でご使用のプライマリ・データ・サーバに関するセクションを参照してください。

#### 参照：

- **pdb\_setrepcol** (31 ページ)

## コマンド・リファレンス

- `pdb_setreproc` (45 ページ)
- `pdb_setreptable` (56 ページ)
- `ra_admin` (81 ページ)
- `ra_locator` (116 ページ)
- `pdb_xlog` (75 ページ)

### ra\_config

Replication Agent 設定パラメータのヘルプ情報を返すか、指定された設定パラメータの値を設定します。

#### 構文

```
ra_config [param [, value]]
```

#### パラメータ

- **param** – Replication Agent 設定パラメータの名前。
- **value** – *param* オプションで指定した設定パラメータに割り当てられる値です。キーワード **default** を使用すると、指定したパラメータをデフォルト値に設定できます。

#### 例

- 例 1 –

```
ra_config
```

**ra\_config** をパラメータなしで発行すると、すべての Replication Agent 設定パラメータのリストが返されます。

- 例 2 –

```
ra_config use_rssd
```

このコマンドは、**use\_rssd** 設定パラメータの現在値を返します。

- 例 3 –

```
ra_config scan_sleep_max, 60
```

このコマンドは、**scan\_sleep\_max** パラメータの値を 60 に変更します。

#### 使用法

- **ra\_config** をオプションなしで呼び出すと、すべての Replication Agent 設定パラメータのリストが返されます。



- **ra\_config** を *param* オプション指定で呼び出すと、指定した設定パラメータのみの情報が返されます。
- **ra\_config** を *param* オプションおよび *value* オプション指定で呼び出すと、指定した設定パラメータの設定が *value* オプションで指定した値に変更されます。
- キーワード **default** を *value* オプションの代わりに使用すると、設定パラメータをデフォルト値にリセットできます。例：  

```
ra_config use_rssd, default
```
- 次の情報は、各設定パラメータに対して返されます。
  - パラメータ名 – パラメータの名前。
  - パラメータ型 – パラメータ値のデータ型 (たとえば、文字列型、数値型、ブール型など)。
  - 現在の値 – **ra\_config** の呼び出し時に有効なパラメータの値。
  - 保留中の値 – 現在の値と異なる場合、**ra\_config** コマンドを前回呼び出したときにパラメータが設定された値で、まだ有効になっていない値。
  - デフォルト値 – Replication Agent インスタンス設定ファイルの作成時のパラメータの値。
  - 有効な値 – 数値の範囲や特定の文字列のリストなど、パラメータに指定できる値。
  - カテゴリ – パラメータの値の影響を受ける Replication Agent コンポーネントを指します。
  - 再起動 – 値の変更を有効にするために Replication Agent インスタンスをシャットダウンして再起動する必要があるパラメータを指します。
- **ra\_config** をオプションなしで、または **param** オプションのみを指定して呼び出した場合、コマンドは Replication Agent インスタンスが Admin、Replicating、または Replication Down 状態にあるときに有効です。
- Replication Agent インスタンスが Replicating 状態にある場合に、Admin 状態でのみ変更可能なパラメータに対して **ra\_config** を **param** オプションおよび **value** オプション指定で呼び出すと、エラーが返されます。
- **ra\_config** を **param** オプションおよび **value** オプション指定で呼び出すと、コマンドは Replication Agent インスタンスが Admin 状態または Replication Down 状態のときに常に有効です。

**参照：**

- ra\_set\_login (125 ページ)
- ra\_help (99 ページ)

## ra\_date

Replication Agent インスタンスから現在の日付と時刻を返します。

### 構文

```
ra_date
```

### 使用法

- **ra\_date** が呼び出されると、Replication Agent インスタンスから現在の日付と時刻を次の Sybase *datetime* データ型の形式で返します。

```
Current RA Date
-----
          Jan 11 2010 12:09:47.310
(1 row affected)
```

- **ra\_date** コマンドが有効なのは、Replication Agent インスタンスが Admin 状態、Replicating 状態、または Replication Down 状態のときです。

### 参照：

- [pdb\\_date](#) (13 ページ)
- [ra\\_config](#) (86 ページ)

## ra\_deviceoffset

(Microsoft SQL Server のみ)RASD で記録されたログ・デバイスのロー・ディスク・デバイス読み取りオフセットを変更します。

### 構文

```
ra_deviceoffset device_id, device_offset
```

### パラメータ

- **device\_id** – このデバイス ID は、**ra\_helpdevice** コマンドを使用して取得できます。
- **device\_offset** – Replication Agent がログ・スキャンを開始するロー・デバイスのオフセット。

### 例

- **例 1** –

```
ra_deviceoffset 1,20
```

このコマンドは、次のようにログ・デバイス ID "1" へのロー・デバイス読み取りオフセットを指定します。

20

## 使用法

- **ra\_devicepath** を呼び出すと、Replication Agent は、RASD 内の指定のログ・デバイスについて指定されたロー・デバイス読み取りオフセットを記録します。
- RASD に格納されたログ・デバイスに関する情報を取得するには、**ra\_helpdevice** コマンドを使用します。
- **ra\_updatedevices** を呼び出すと、デバイス・オフセットはデフォルト値にリセットされません。デフォルトのデバイス読み取りオフセットは 0 です。
- Replication Agent インスタンスが Replicating 状態のときに **ra\_deviceoffset** を呼び出すと、エラーが返されます。
- **ra\_deviceoffset** コマンドが有効なのは、Replication Agent インスタンスが Admin 状態、Replicating 状態、または Replication Down 状態のときのみです。

### 参照：

- ra\_helpdevice (103 ページ)
- pdb\_xlog (75 ページ)
- ra\_admin (81 ページ)
- ra\_updatedevices (142 ページ)

## ra\_devicepath

(Microsoft SQL Server のみ) RASD で記録されたログ・デバイスのディスク・デバイス・パスを変更します。

### 構文

```
ra_devicepath device, dev_path
```

### パラメータ

- **device** – デバイス ID。
- **dev\_path** – *device* オプションで指定したデバイスのディスク・ログ・デバイスを指すパス。

### 例

- **例 1** –

```
ra_devicepath 3,
```

```
d:¥software¥oracle¥devices¥redo001.log
```

このコマンドは、次のようにログ・デバイス ID "3" へのディスク・デバイス・パスを指定します。

```
d:¥software¥oracle¥devices¥redo001.log
```

### 使用法

- **ra\_devicepath** を呼び出すと、Replication Agent は、RASD 内の指定のログ・デバイスについて指定されたディスク・デバイス・パスを記録します。
- RASD に格納されたログ・デバイスに関する情報を取得するには、**ra\_helpdevice** コマンドを使用します。
- **ra\_updatedevices** を、**ra\_devicepath** でデバイス・パスを設定した後に呼び出す場合、ログ・デバイスのデフォルト・パスを変更する必要があるときは、**ra\_devicepath** を再度使用してパスをリセットする必要があります。デフォルト・デバイス・パスは、プライマリ・データ・サーバによって返されたデバイス・パスです。
- Replication Agent インスタンスが Replicating 状態のときに **ra\_devicepath** を呼び出すと、エラーが返されます。
- **ra\_devicepath** コマンドが有効なのは、Replication Agent インスタンスが Admin 状態または Replication Down 状態のときだけです。

### 参照：

- **ra\_helpdevice** (103 ページ)
- **pdb\_xlog** (75 ページ)
- **ra\_admin** (81 ページ)
- **ra\_updatedevices** (142 ページ)

### ra\_downgrade

**ra\_downgrade** コマンドは Replication Agent が以前のバージョンにダウングレードできるようにします。

**ra\_downgrade** コマンドは RASD の内容をエクスポート・ファイルにコピーします。このファイルが Replication Agent のダウングレード先インスタンス (前のバージョン) によって適用され、ダウングレードが完了します。

---

**注意：** **ra\_downgrade\_prepare** コマンドおよび **ra\_downgrade\_accept** コマンドは廃止されました。できる限り **ra\_downgrade** コマンドと **ra\_migrate** コマンドを使用してください。『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。

---

## 構文

```
ra_downgrade [list]
```

## パラメータ

- **list** – RASD の内容がエクスポートされるファイルの名前と場所を表示します。

## 使用法

- 前のバージョンの Replication Agent にダウングレードするための準備として、**ra\_downgrade** は、プライマリ・データベースの Replication Agent システム・オブジェクトを変更し、Replication Agent のダウングレード先インスタンスと一致させます。
- Oracle および UDB では、**ra\_downgrade** は、Replication Agent システム・データベース (RASD) の内容を *timestamp.export* というファイルに抽出することも行います。ここで、*timestamp* は **ra\_downgrade** の呼び出し時に取得されたタイムスタンプです。このファイルのデフォルトの保存場所は、ダウングレード先である Replication Agent インスタンス (前のバージョン) の **rasd\_backup\_dir** 設定パラメータによって指定されたディレクトリの配下にある **import** サブディレクトリです。**ra\_downgrade** が正常に実行されると、このファイルのパスが返されます。
- Microsoft SQL Server の場合は、**ra\_downgrade** は Replication Agent システム・データベース (RASD) の内容を *rasd.export* という名前のファイルに抽出します。
- **ra\_downgrade** コマンドが有効なのは、Replication Agent のインスタンスが Admin 状態または Replication Down 状態のときです。

## 参照：

- [ra\\_downgrade\\_accept](#) (91 ページ)
- [ra\\_downgrade\\_prepare](#) (93 ページ)
- [ra\\_migrate](#) (121 ページ)

## ra\_downgrade\_accept

(Oracle および Microsoft SQL Server のみ) **ra\_downgrade\_accept** コマンドは、Replication Agent のダウングレード先 (前のバージョン) のインスタンスによって実行されます。

このコマンドは、Replication Agent のダウングレード元のインスタンス (後のバージョン) によって実行される **ra\_downgrade\_prepare** コマンドによって開始されたダウングレード・プロセスを完了します。

---

**注意：** **ra\_downgrade\_prepare** コマンドおよび **ra\_downgrade\_accept** コマンドは廃止されました。できる限り **ra\_downgrade** コマンドと **ra\_migrate** コマンドを使用して

ください。『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。

---

### 構文

```
ra_downgrade_accept export_file
```

### パラメータ

- **export\_file – ra\_downgrade\_prepare** コマンドの実行時に RASD の抽出先となるファイル名。これは、ファイル名のみの場合も、絶対パスを含むファイル名の場合もあります。ファイル名は *timestamp.export* という形式になります。ここで *timestamp* は **ra\_downgrade\_prepare** の呼び出しと同時に取得したタイムスタンプです。

---

**注意：** **export\_file** パラメータは、Microsoft SQL Server と Oracle の場合のみ使用します。**ra\_downgrade\_accept** コマンドを UDB で使用する場合、パラメータは指定しません。

---

### 使用法

- (Microsoft SQL Server および Oracle のみ) **ra\_downgrade\_accept** コマンドは、指定されたファイルから RASD を復元します。このファイルのデフォルトの保存場所は、ダウングレード先である Replication Agent インスタンス (前のバージョン) の **rasd\_backup\_dir** 設定パラメータによって指定されたディレクトリの配下にある **import** サブディレクトリです。**ra\_downgrade\_prepare** が正常に実行された場合は、このファイルのパスが返されます。
- **ra\_downgrade\_accept** コマンドを使用するには、Replication Agent がプライマリ・データベースと接続できなければなりません。
- (Microsoft SQL Server および Oracle のみ) **ra\_downgrade\_accept** コマンドが正常に実行された場合は、Replication Agent がシャットダウンします。
- **ra\_downgrade\_accept** コマンドが有効なのは、Replication Agent のインスタンスが Admin 状態または Replication Down 状態のときです。

### 参照：

- ra\_downgrade (90 ページ)
- ra\_downgrade\_prepare (93 ページ)
- ra\_migrate (121 ページ)

## ra\_downgrade\_prepare

(Oracle および Microsoft SQL Server のみ) **ra\_downgrade\_prepare** コマンドは、Replication Agent を前のバージョンにダウングレードする準備をします。

Replication Agent for Microsoft SQL Server または Replication Agent for Oracle で **ra\_downgrade\_prepare** コマンドを実行すると、RASD の内容がエクスポート・ファイルにコピーされます。このファイルが Replication Agent のダウングレード先インスタンス (前のバージョン) によって適用され、ダウングレードが完了します。

---

**注意:** **ra\_downgrade\_prepare** コマンドおよび **ra\_downgrade\_accept** コマンドは廃止されました。できる限り **ra\_downgrade** コマンドと **ra\_migrate** コマンドを使用してください。『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。

---

### 構文

```
ra_downgrade_prepare { list | target_inst_path }
```

### パラメータ

- **list** – エクスポート可能な RASD テーブルのリストを示すキーワード。
- **target\_inst\_path** – ダウングレード先に指定した Replication Agent インスタンス (前のバージョン) の絶対パス。

### 使用法

- 前のバージョンの Replication Agent にダウングレードするための準備として、**ra\_downgrade\_prepare** コマンドはまず、プライマリ・データベースの Replication Agent システム・オブジェクトを変更し、Replication Agent のダウングレード先インスタンスと一致させます。
- (Microsoft SQL Server および Oracle のみ) **ra\_downgrade\_prepare** コマンドは、Replication Agent システム・データベース (RASD) の内容を *timestamp.export* というファイルに抽出することも行います。ここで、*timestamp* は **ra\_downgrade\_prepare** の呼び出し時に取得されたタイムスタンプです。このファイルのデフォルトの保存場所は、ダウングレード先である Replication Agent インスタンス (前のバージョン) の **rasd\_backup\_dir** 設定パラメータによって指定されたディレクトリの配下にある *import* サブディレクトリです。**ra\_downgrade\_prepare** が正常に実行された場合は、このファイルのパスが返されます。

- **ra\_downgrade\_prepare** コマンドが有効なのは、Replication Agent のインスタンスが Admin 状態または Replication Down 状態のときです。

### 参照：

- ra\_downgrade (90 ページ)
- ra\_downgrade\_accept (91 ページ)
- ra\_migrate (121 ページ)

## ra\_dump

Replication Server の **rs\_dumpdb** および **rs\_dumptran** システム・ファンクションをエミュレートします。

### 構文

```
ra_dump { database | transaction } , dbname, dump_label
```

### パラメータ

- **database** – **rs\_dumpdb** システム・ファンクションに関連付けられている関数文字列をプライマリ Replication Server に適用させるキーワード。
- **transaction** – **rs\_dumptran** システム・ファンクションに関連付けられている関数文字列をプライマリ Replication Server に適用させるキーワード。
- **dbname** – ダンプするデータベースの名前。
- **dump\_label** – データベース・ダンプの識別情報を格納している *varchar(30)* 値。

### 例

- 例 1 –

```
ra_dump database, MSSQL_source, rssddmp
```

ここで、MSSQL\_source はデータベース名、rssddmp はダンプ・ラベルです。

### 使用法

- **ra\_dump** が呼び出されると、Replication Agent は **dump** マーカを Replication Agent のトランザクション・ログに挿入してコーディネート・ダンプを容易化します。
- トランザクション・ログが存在しない場合、**ra\_dump** コマンドはエラー・メッセージを返します。
- **ra\_dump** コマンドが有効なのは、Replication Agent のインスタンスが Admin 状態、Replicating 状態、または Replication Down 状態のときです。



- Replication Server の **rs\_dumpdb** および **rs\_dumptran** システム・ファンクションの詳細については、『Replication Agent 管理ガイド』と『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。

**参照：**

- ra\_config (86 ページ)
- ra\_migrate (121 ページ)

## ra\_dumptran

(Oracle のみ) このコマンドは、指定されたデータベース・トランザクションのトラブルシューティングで使用する情報を返します。

---

**注意：** このコマンドは Oracle でのみ使用可能です。

---

### 構文

```
ra_dumptran "{ opid | locator | tranid }"
```

### パラメータ

- **opid** – データベース・オペレーションの Replication Agent オペレーション ID。
- **locator** – データベース・オペレーションの Replication Agent ロケータ。
- **tranid** – データベース・トランザクションの Oracle トランザクション ID。

### 例

• **例 1 –**

```
ra_dumptran
0x0000.01783d95.0000:0001.0000003fe.00000031.0010
go
```

このコマンドは、Replication Agent オペレーション ID によって指定されたオペレーションに関する情報を返します。

Name	Value
-----	-----
-	
BEGIN OPERATION ID	0x0000.01783d95.0000:0001.0000003fe.00000031.0010
BEGIN SCN	24657302
TRANSACTION ID	0004.0016.00000016
USERNAME	AUSER

## コマンド・リファレンス

```
EXECUTION TIME      2010-07-12 10:28:14.0
THREAD NUMBER      1
TRANSACTION SKIP
COMMAND            pdb_skip_op add, 24657301, 1, 1022.49.16
DUMP FILE NAME     C:\somepath\XID0004.0016.00000016.log
(8 rows affected)
```

### • 例 2 –

```
ra_dumptran
0000000001783d9600020001000003fe0000003400e8000001783d9500000000
go
```

このコマンドは、Replication Agent ロケータによって指定されたオペレーションに関する情報を返します。

```
Name                Value
-----
-
BEGIN OPERATION ID  0x0000.01783d95.0000:0001.000003fe.
00000031.0010
BEGIN SCN           24657302
TRANSACTION ID      0004.0016.00000016
USERNAME            AUSER
EXECUTION TIME      2010-07-12 10:28:14.0
THREAD NUMBER      1
TRANSACTION SKIP
COMMAND            pdb_skip_op add, 0.24657301.0, 1, 1022.49.16
DUMP FILE NAME     C:\somepath\XID0004.0016.00000016.log
(8 rows affected)
```

### • 例 3 –

```
ra_dumptran 0x0004.0016.00000016
go
```

このコマンドは、Oracle トランザクション ID によって指定されたトランザクションに関する情報を返します。

```
Name                Value
-----
-
```

```

BEGIN OPERATION ID 0x0000.01783d95.0000:0001.0000003fe.
00000031.0010

BEGIN SCN          24657302

TRANSACTION ID     0004.0016.000000016

USERNAME           AUSER

EXECUTION TIME     2010-07-12 10:28:14.0

THREAD NUMBER      1

TRANSACTION SKIP
COMMAND            pdb_skip_op add, 0.24657301.0, 1, 1022.49.16

DUMP FILE NAME     C:\somepath\XID0004.0016.000000016.log

(8 rows affected)

```

- 例 4**–次は、`ra_dumptran` コマンドからのログ・ファイル出力の例です。このログ・ファイルには、トランザクション情報とトランザクション内のすべてのオペレーションに関する情報の両方が格納されます。

```

File name: C:\somepath\XID0004.0016.000000016.log

File contents:

BEGIN OPERATION ID 0x0000.01783d95.0000:0001.0000003fe.
00000031.0010

BEGIN SCN          24657302

TRANSACTION ID     0004.0016.000000016

USER NAME          AUSER

EXECUTION TIME     2010-07-12 10:28:14.0

THREAD NUMBER      1

TRANSACTION SKIP
COMMAND            pdb_skip_op add, 24657301, 1, 1022.49.16

SCN      THREAD OPERATION ID          OBJECT
ID OBJECT NAME REPLICATE OPERATION SQL

-----
-----
-----

24657301 1          0x0000.01783d95.0000:0001.0000003fe.00000031.0010
0          NULL          NO          START          set transaction read
write;

24657301 1          0x0000.01783d95.0000:0001.0000003fe.00000031.0010
51809    BLL$TEST    YES          INSERT          insert into "QA7USER"."BLL
$TEST"("QUANTITY","ORDER_NUMBER") values ('85','1234567890');

```

```

24657302 1      0x0000.01783d96.0000:0001.000003fe.00000033.010c
51809      BLL$TEST  YES      DELETE  delete from "QA7USER"."BLL
$TEST" where "QUANTITY" = '85' and "ORDER_NUMBER" = '1234567890'
and ROWID = 'AAAMphAAEAAAYrWAAC';

24657303 1      0x0000.01783d97.0000:0001.000003fe.00000035.00c4
0          NULL      NO          COMMIT   Commit;

```

## 使用法

- ra\_dumptran** コマンドは、指定されたトランザクションのすべてのオペレーションを、失敗したオペレーションまたはトランザクションのトラブルシューティングで使用される排他的なログ・ファイルにダンプします。このログ・ファイルのヘッダは、**ra\_dumptran** によって返された結果セットから成り、次のローを含んでいます。
  - BEGIN OPERATION ID** – トランザクション **begin** オペレーションの Replication Agent オペレーション ID。このフィールドの形式は *wrap.scn.subscn.thread.lsn.block.offset* です。各パラメータの意味は次のとおりです。
    - wrap.scn.subscn* は、**begin** オペレーションの System Change Number (SCN) です。
    - thread* は、データベース・スレッド番号です。
    - lsn* は、**begin** オペレーションのログ・シーケンス番号です。
    - block* は、**begin** オペレーションが存在するブロックです。
    - offset* は、**begin** オペレーションが存在するオペレーションのオフセット値です。
  - BEGIN SCN – REDO** ログ・ファイルに記録されるトランザクション・オペレーションの SCN。
  - TRANSACTION ID** – このオペレーションを一部に含んでいるトランザクションの ID。
  - USERNAME** – トランザクションを実行したユーザの名前。
  - EXECUTION TIME** – トランザクションが実行された日付と時刻。
  - THREAD NUMBER** – トランザクションを実行した Oracle スレッド。
  - TRANSACTION SKIP COMMAND** – 複写時にトランザクションが Replication Agent によってスキップされる原因となる Replication Agent コマンド。
  - DUMP FILE NAME** – **ra\_dumptran** によってトランザクションが書き込まれるログ・ファイルの名前。
  - DUMP FILE ロー** によって指定されるログ・ファイルには、指定したトランザクションのオペレーション結果も格納されます。
    - SCN – REDO** ログ・ファイルに記録されるオペレーションの SCN。

- **THREAD** – オペレーションを実行したスレッド。
- **OPERATION ID** – トランザクション **begin** オペレーションの Replication Agent オペレーション ID。このフィールドの形式は *wrap.scn.subscn.thread.lsn.block.offset* です。
- **OBJECT ID** – 影響を受けるオブジェクトのオブジェクト ID。
- **OBJECT NAME** – 影響を受けるオブジェクトの名前。
- **REPLICATE** – オペレーションの影響を受けるオブジェクトが Replication Agent によって複製の対象としてマーク付けされているかどうか (YES または NO)。
- **OPERATION** – オペレーション・タイプ。
- **SQL** – オペレーションの SQL 文。
- Oracle LogMiner スクリプト `$ORACLE_HOME/rdbms/admin/dbmslm.sql` がプライマリ・データベースにインストールされていないと、**ra\_dumptran** コマンドは正しく動作しません。このスクリプトがインストールされていない場合、**ra\_dumptran** はエラーを返します。
- LogMiner がインストールされたら、次のように入力してパブリック・シノニムを作成します。こうすることで、所有者としてログインしなくても LogMiner 関数を実行できるようになります。

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

---

**注意：**これは、Oracle 10g を使用している場合に必要となります。

---

- **ra\_dumptran** コマンドを正しく機能させるには、以下の権限を **pds\_username** に付与する必要があります。
  - **EXECUTE\_CATALOG\_ROLE**
  - **SELECT ON V\_\$LOGMNR\_CONTENTS**
  - **SELECT ON V\_\$LOGMNR\_LOGS**
  - **SELECT ANY TRANSACTION**
- 指定された *opid* または *locator* 値に対して **ra\_dumptran** コマンドが結果を返さない場合は、対応するデータベース・オペレーションがデータベース・トランザクション内の多数のオペレーションのうちの1つである可能性があります。この場合は、データベース・オペレーションが所属するトランザクションの ID を代わりに指定する必要があります。

## ra\_help

Replication Agent コマンドのヘルプ情報を返します。

### 構文

```
ra_help [command]
```

## パラメータ

- **command** – ヘルプ情報を表示する Replication Agent コマンドの名前。

## 例

- **例 1 –**

```
ra_help
```

このコマンドは、Replication Agent のすべてのコマンドに関するヘルプを返します。

- **例 2 –**

```
ra_help pdb_gen_id
```

このコマンドは、**pdb\_gen\_id** コマンドに関するヘルプを返します。

## 使用法

- オプションを指定しないで **ra\_help** を呼び出すと、Replication Agent のすべてのコマンドに関するヘルプ情報が返されます。
- **command** オプションを指定して **ra\_help** を呼び出すと、指定したコマンドのみに関するヘルプ情報が返されます。
- **ra\_help** コマンドが有効なのは、Replication Agent インスタンスが Admin 状態、Replicating 状態、または Replication Down 状態のときです。

## 参照：

- ra\_config (86 ページ)

## ra\_helparchive

(Oracle のみ) 管理されているアーカイブ・ログのすべて、特定の redo ログ・スレッド、または特定の redo ログ・スレッドのアーカイブ・ログのメタデータ・リストを表示します。

## 構文

```
ra_helparchive [ redo_log_thread_id ]
```

## パラメータ

- **redo\_log\_thread\_id** – アーカイブ・ログまたはメタデータを表示する redo ログ・スレッドの ID です。

## 使用法

- スレッド ID が指定されていない場合、**ra\_helparchive** は、すべての管理されているアーカイブ・ログのメタデータ・リストを返します。

## 参照：

- `pdb_archive_path` (206 ページ)

## ra\_helparticle

プライマリ・データベース・アーティクルに関する情報を RASD から返します。

## 構文

```
ra_helparticle [article [, version]]
```

## パラメータ

- **article** – プライマリ・データベース内のアーティクル (テーブルまたはプロシージャ) の名前またはオブジェクト ID。アーティクル名は所有者名で次のように修飾できます。

```
owner.article
```

所有者名によるアーティクル名の修飾はオプションです。

- **version** – *article* オプションで指定されたアーティクルのバージョンを識別する、16 進数のロケータ値。

## 例

- **例 1 –**

```
ra_helparticle
```

このコマンドは、RASD 内のすべてのアーティクルの全バージョンに関する情報を返します。

- **例 2 –**

```
ra_helparticle table1
```

このコマンドは、RASD 内の “table1” という名前のアーティクルの現在のバージョンに関する情報を返します。

- **例 3 –**

```
ra_helparticle table1,  
00000000000210a400003334000700003334000699940000d413c50000000000
```

このコマンドは、RASD 内の “table1” という名前のアーティクルのバージョン **00000000000210a400003334000700003334000699940000d413c50000000000** に関する情報を返します。

### 使用法

- **ra\_helparticle** コマンドは、アーティクル (テーブルとプロシージャ) に関する以下の情報を返します。
  - アーティクルのオブジェクト ID
  - プライマリ・データベースの名前
  - アーティクル所有者の名前またはエイリアス
  - アーティクル名
  - アーティクルのタイプ (テーブルまたはプロシージャ)
  - アーティクルのステータス (Current、Archived、または Dropped)
  - アーティクルのバージョン番号アーティクルのタイプ、ステータス、バージョン番号以外のすべての情報は、**pdb\_xloginit** コマンドを使用して Replication Agent を初期化したときにプライマリ・データベースによって返される値です。
- オプションを指定しないで **ra\_helparticle** を呼び出すと、RASD 内のすべてのアーティクル (テーブルとプロシージャ) の全バージョンに関する情報が返されます。
- *article* オプションを指定して **ra\_helparticle** を呼び出すと、RASD 内の指定したアーティクルの現在のバージョンのみにに関する情報が返されます。
- *article* および *version* オプションを指定して **ra\_helparticle** を呼び出すと、RASD 内の指定したアーティクルの指定したバージョンのみにに関する情報が返されます。
- **ra\_helparticle** コマンドが有効なのは、Replication Agent インスタンスが Admin 状態、Replicating 状態、または Replication Down 状態のときです。

### 参照：

- ra\_helpdb (102 ページ)
- ra\_helpfield (107 ページ)
- ra\_helplocator (109 ページ)
- ra\_helpuser (113 ページ)

### ra\_helpdb

(Oracle と Microsoft SQL Server のみ) RASD からプライマリ・データベースに関する情報を返します。

---

**注意：** このコマンドは Oracle および Microsoft SQL Server でのみ使用可能です。

---



## 構文

```
ra_helppdb
```

## 使用法

- **ra\_helppdb** を呼び出すと、プライマリ・データベースに関する以下の情報が返されます。
  - データベース・オブジェクト ID
  - データベース名データベース ID とデータベース名は、**pdb\_xlog init** コマンドを使用して Replication Agent を初期化したときにプライマリ・データベースによって返される値です。
- **ra\_helppdb** コマンドが有効なのは、Replication Agent インスタンスが Admin 状態、Replicating 状態、または Replication Down 状態のときです。
- **ra\_helppdb** コマンドが有効なのは、RASD が初期化された後のみ、つまり、**pdb\_xlog init** を実行した後のみです。

## 参照：

- ra\_devicepath (89 ページ)
- ra\_helpparticle (101 ページ)
- ra\_helpdevice (103 ページ)
- ra\_helpfield (107 ページ)
- ra\_helplocator (109 ページ)
- ra\_helpuser (113 ページ)
- ra\_updatedevices (142 ページ)

## ra\_helpdevice

(Oracle と Microsoft SQL Server のみ) このコマンドは、RASD ログ・デバイス・レポジトリからプライマリ・データベースに関する情報を返します。

---

**注意：** このコマンドは Oracle および Microsoft SQL Server でのみ使用可能です。

---

## 構文

```
ra_helpdevice [device]
```

## パラメータ

- **device** – プライマリ・データベース・ログ・デバイスのデバイス ID。

## 例

### • 例 1 –

```
ra_helpdevice
```

このコマンドは、ログ・デバイス・レポジトリに記録されているすべてのプライマリ・データベース・ログ・デバイスに関する情報を返します。

### • 例 2 –

```
ra_helpdevice 1
```

このコマンドは、ログ・デバイス・レポジトリ内のプライマリ・データベース・ログ・デバイス ID “1” に関する情報を返します。

## 使用法

- **ra\_helpdevice** コマンドは、RASD に記録されている各プライマリ・データベース・ログ・デバイスに関する以下の情報を返します。
  - デバイス ID – プライマリ・データ・サーバによって定義されたログ・デバイス ID。

---

**注意：** Oracle の場合、デバイス ID はこのファイルが所属する Oracle REDO ログ・グループの値です。

---

- データベース名 – ログ・デバイスに関連付けられているプライマリ・データベースの名前。
- デバイス名 – プライマリ・データ・サーバによって定義されたログ・デバイスの論理名。
- サーバ・デバイス・パス – ログ・デバイスのマルチプレックス・バージョンのパス。
- ディスク・ミラー・パス – (スタンバイ・サイトにある) ログ・デバイスのパス。
- ディスク・デバイス・ステータス – サーバ・デバイス・パスの現在のステータス (**ACCESSIBLE**、**NOT\_VALID**、または **OPEN**)。
- ログ・デバイス ID、プライマリ・データベース名、ログ・デバイス名、およびサーバ・ログ・デバイス・パスは、**pdb\_xlog init** コマンドを使用して Replication Agent を初期化したとき、または、**ra\_updatedevices** コマンドを使用してログ・デバイス・レポジトリを更新したときにプライマリ・データ・サーバによって返される値です。
- ディスク・ミラー・パスは、RASD に記録されている現在の値です。Replication Agent は、RASD に記録されているディスク・ミラー・パスを使用して各ログ・デバイスを見つけます。RASD に記録された各ログ・デバイスの場合、ディスク・デバイス・パスを **ra\_devicepath** コマンドで設定または変更できます。

**ra\_devicepath** を使用してディスク・デバイス・パスを指定しない場合、ディスク・ミラー・パスについて記録される値は **DEFAULT** であり、Replication Agent はサーバ・デバイス・パスについて記録された値を使用してログ・デバイスを見つけます。

- ディスク・デバイスのステータスは、**ra\_helpdevice** コマンドを呼び出すたびに Log Reader コンポーネントによって更新されます。
- オプションを指定しないで **ra\_helpdevice** を呼び出すと、RASD ログ・デバイス・レポジトリに記録されているすべてのログ・デバイスに関する情報が返されます。
- **device** オプションを指定して **ra\_helpdevice** を呼び出すと、指定したログ・デバイスのみに関する情報が返されます。
- **ra\_helpdevice** コマンドが有効なのは、Replication Agent インスタンスが Admin 状態、Replicating 状態、または Replication Down 状態のときです。

**参照：**

- ra\_devicepath (89 ページ)
- ra\_helpdb (102 ページ)
- ra\_updatedevices (142 ページ)

## ra\_helpdeviceoffset

(Microsoft SQL Server のみ) プライマリ・データベース・ログ・デバイスに関するデバイス・オフセット情報を RASD ログ・デバイス・レポジトリから返します。

### 構文

```
ra_helpdeviceoffset [device]
```

### パラメータ

- **device** – プライマリ・データベース・ログ・デバイスのデバイス ID。

### 例

- **例 1 –**

```
ra_helpdeviceoffset
```

このコマンドは、ログ・デバイス・レポジトリに記録されているすべてのプライマリ・データベース・ログ・デバイスに関するデバイス・オフセット情報を返します。

- **例 2 –**

```
ra_helpdeviceoffset 1
```

このコマンドは、ログ・デバイス・レポジトリ内のプライマリ・データベース・ログ・デバイス ID 1 に関する情報を返します。

### 使用法

- **ra\_helpdeviceoffset** コマンドは、RASD に記録されている各プライマリ・データベース・ログ・デバイスに関する以下の情報を返します。
  - デバイス ID – プライマリ・データ・サーバによって定義されたログ・デバイス ID。
  - データベース名 – ログ・デバイスに関連付けられているプライマリ・データベースの名前。
  - デバイス名 – プライマリ・データ・サーバによって定義されたログ・デバイスの論理名。
  - サーバ・デバイス・パス – ログ・デバイスのマルチプレックス・バージョンのパス。
  - ディスク・デバイス・パス – (スタンバイ・サイトにある) ログ・デバイスのパス。
  - ディスク・デバイス・オフセット – Replication Agent によるログ・デバイスでのログ・スキャンの開始位置を示すオフセット値。
  - ディスク・デバイス・ステータス – サーバ・デバイス・パスの現在のステータス (ACCESSIBLE、NOT\_VALID、または OPEN)。
- ディスク・デバイス・パスは、RASD に記録されている現在の値です。Replication Agent は、RASD に記録されているディスク・デバイス・パスを使用して各ログ・デバイスを見つけます。
- RASD に記録されているログ・デバイスごとに、**ra\_devicepath** を使用してディスク・デバイス・パスを設定または変更したり、**ra\_deviceoffset** を使用してデバイス・オフセットを設定または変更したりできます。
- **ra\_devicepath** を使用してディスク・デバイス・パスを指定しない場合、そのディスク・デバイス・パスについて記録される値は DEFAULT であり、Replication Agent はサーバ・デバイス・パスについて記録された値を使用してログ・デバイスを見つけます。
- ディスク・デバイスのステータスは、**ra\_helpdeviceoffset** コマンドを呼び出すたびに Log Reader コンポーネントによって更新されます。
- オプションを指定しないで **ra\_helpdeviceoffset** を呼び出すと、RASD ログ・デバイス・レポジトリに記録されているすべてのログ・デバイスに関する情報が返されます。
- **device** オプションを指定して **ra\_helpdeviceoffset** を呼び出すと、指定したログ・デバイスのみに関する情報が返されます。

- **ra\_helpdeviceoffset** コマンドが有効なのは、Replication Agent インスタンスが Admin 状態、Replicating 状態、または Replication Down 状態のときです。

#### 参照：

- ra\_helpdevice (103 ページ)
- ra\_deviceoffset (88 ページ)
- ra\_devicepath (89 ページ)
- ra\_updatedevices (142 ページ)

## ra\_helpfield

プライマリ・データベース・フィールド (テーブル内のカラムまたはストアド・プロシージャ内の入力パラメータ) に関する情報を RASD から返します。

### 構文

```
ra_helpfield article [, version [, field]]
```

### パラメータ

- **article** – プライマリ・データベース内のアーティクル (テーブルまたはプロシージャ) の名前またはオブジェクト ID。アーティクル名は所有者名で次のように修飾できます。

```
owner.article
```

所有者名によるアーティクル名の修飾はオプションです。

- **version** – 指定されたアーティクルのバージョンを識別する、16進数のロケータ値。
- **field** – 指定されたアーティクル内のフィールド (カラムまたは入力パラメータ) の名前またはオブジェクト ID。

### 例

- **例 1 –**

```
ra_helpfield authors
```

このコマンドは、RASD 内の *authors* という名前のアーティクルの現在のバージョンに含まれているすべてのフィールドに関する情報を返します。

- **例 2 –**

```
ra_helpfield authors,  
00000000000210a400003334000700003334000699940000d413c50000000000
```

このコマンドは、RASD 内の “authors” という名前のアーティクルのバージョン **00000000000210a400003334000700003334000699940000d413c50000000000** に含まれているすべてのフィールドに関する情報を返します。

### • 例 3 –

```
ra_helpfield authors,  
00000000000210a400003334000700003334000699940000d413c50000000000,  
au_fname
```

このコマンドは、RASD 内の “authors” という名前のアーティクルのバージョン **00000000000210a400003334000700003334000699940000d413c50000000000** に含まれている *au\_fname* という名前のフィールドに関する情報を返します。

## 使用法

- **ra\_helpfield** コマンドは、フィールドに関する以下の情報を返します。
  - フィールド (カラムまたは入力パラメータ) のオブジェクト ID
  - フィールド名
  - フィールド・タイプ ID
  - フィールドのデータ型 (精度、長さ、位取りを指定)
  - フィールドの NULL モード
  - フィールドの IDENTITY ステータス
  - フィールドのプライマリ・キー・ステータスフィールドに関するすべての情報項目は、**pdb\_xlog init** コマンドを使用して Replication Agent を初期化したときにプライマリ・データベースによって返される値です。
- *article* オプションを指定して **ra\_helpfield** を呼び出すと、RASD 内の指定したアーティクルの現在のバージョンに含まれているすべてのフィールドに関する情報が返されます。
- *article* および *version* オプションを指定して **ra\_helpfield** を呼び出すと、RASD 内の指定したアーティクルの指定したバージョンに含まれているすべてのフィールドに関する情報が返されます。
- *article*、*version*、および *field* オプションを指定して **ra\_helpfield** を呼び出すと、RASD 内の指定したアーティクルの指定したバージョンに含まれている指定したフィールドに関する情報が返されます。
- **ra\_helpfield** コマンドが有効なのは、Replication Agent インスタンスが Admin 状態、Replicating 状態、または Replication Down 状態のときです。
- **pdb\_xlog init** コマンドを使用した RASD の初期化をまだ行っていない場合、**ra\_helpfield** は結果を返しません。

**参照：**

- ra\_config (86 ページ)
- ra\_help (99 ページ)
- ra\_helparticle (101 ページ)
- ra\_helppdb (102 ページ)
- ra\_helpdevice (103 ページ)
- ra\_helplocator (109 ページ)
- ra\_helpuser (113 ページ)

**ra\_helplocator**

LTM ロケータ値に含まれているフィールドに関する情報を返します。

**構文**

```
ra_helplocator [locator_value]
```

**パラメータ**

- **locator\_value** – LTM ロケータの 16 進文字列値。

**例**

- **例 1 –**

```
ra_helplocator
```

このコマンドは、現在の LTM ロケータ値に含まれているフィールドに関する情報を返します。

- **例 2 –**

```
ra_helplocator locator_value
```

このコマンドは、指定した LTM ロケータ値に含まれているフィールドに関する情報を返します。

**使用法**

- **ra\_helplocator** コマンドは、LTM ロケータ値に関する以下の情報を返します。
  - ロケータ・フィールドの名前
  - ロケータ・フィールドの 16 進数値
  - ロケータ・フィールドの 10 進数値

- オプションを指定しないで **ra\_helplocator** を呼び出すと、現在の LTM ロケータ値に含まれているフィールドに関する情報が返されます。
- *locator\_value* オプションを指定して **ra\_helplocator** を呼び出すと、指定した LTM ロケータ値に含まれているフィールドに関する情報が返されます。
- **ra\_helplocator** コマンドが有効なのは、Replication Agent インスタンスが Admin 状態、Replicating 状態、または Replication Down 状態のときです。

### 参照：

- ra\_config (86 ページ)
- ra\_help (99 ページ)
- ra\_locator (116 ページ)

## ra\_helpop

このコマンドは、指定されたデータベース・トランザクション・ログ・オペレーションのトラブルシューティングで使用する情報を返します。

---

**注意：** このコマンドは Oracle でのみ使用可能です。

---

### 構文

```
ra_helpop "{ opid | lsn | locator }"
```

### パラメータ

- **opid** – (Oracle の場合) データベース・オペレーションの Replication Agent オペレーション ID。
- **lsn** – (UDB の場合) ログ・レコードの System Change Number (LSN) キーワード。
- **locator** – データベース・オペレーションの Replication Agent ロケータ。

### 例

- **例 1 –**

```
ra_helpop
0x0000.01783d96.0000:0001.0000003fe.00000034.00e8
go
```

このコマンドは、Replication Agent オペレーション ID によって指定されたオペレーションに関する情報を返します。

Name	Value
OPERATION ID	0x0000.01783d96.0000:0001.0000003fe .00000034.00e8
SCN	24657302
THREAD	1



```

USERNAME          AUSER
EXECUTION TIME    2010-07-12 10:28:14.0
OBJECT ID         51809
OBJECT NAME       BLL$TEST
OPERATION         INSERT
REPLICATE        YES
TRANSACTION ID    0004.0016.00000016
REDO SQL          insert into "AUSER"."BLL$TEST" (
                  "QUANTITY","ORDER_NUMBER") values
                  ('85','1234567890');
UNDO SQL          delete from "AUSER"."BLL$TEST" where
                  "QUANTITY"='85' and "ORDER_NUMBER"
                  = '1234567890' and ROWID =
                  'AAAMphAAEAAAYrWAAC';
SKIP COMMAND      pdb_skip_op add, 24657302, 1,
                  1022.52.2322
(13 rows affected)

```

• **例 2 -**

```

ra_helpop
0000000001783d9600020001000003fe0000003400e8000001783d9500000000
go

```

このコマンドは、Replication Agent ロケータによって指定されたオペレーションに関する情報を返します。

Name	Value
OPERATION ID	0x0000.01783d96.0000:0001.000003fe .00000033.010c
SCN	24657302
THREAD	1
USERNAME	AUSER
EXECUTION TIME	2010-07-12 10:28:14.0
OBJECT ID	51809
OBJECT NAME	BLL\$TEST
OPERATION	INSERT
REPLICATE	YES
TRANSACTION ID	0004.0016.00000016
REDO SQL	insert into "AUSER"."BLL\$TEST" (                   "QUANTITY","ORDER_NUMBER") values                   ('85','1234567890');
UNDO SQL	delete from "AUSER"."BLL\$TEST" where                   "QUANTITY"='85' and "ORDER_NUMBER"                   = '1234567890' and ROWID =                   'AAAMphAAEAAAYrWAAC';
SKIP COMMAND	pdb_skip_op add, 24657302, 1,                   1022.52.232
(13 rows affected)	

**使用法**

- **ra\_helpop** コマンドは、失敗したオペレーションまたはトランザクションのラブルシューティングで使用する、指定されたオペレーションのデータベース

および Replication Agent に関する情報を表示します。 **ra\_helpop** によって返される結果セットには以下のローが含まれます。

- OPERATION ID – *wrap.scn.subscn.thread.lsn.block.offset* という形式の Replication Agent オペレーション ID。各パラメータの意味は次のとおりです。
  - *wrap.scn.subscn* は、指定されたオペレーションの System Change Number (SCN) です。
  - *thread* は、データベース・スレッド番号です。
  - *lsn* は、指定されたオペレーションのログ・シーケンス番号です。
  - *block* は、指定されたオペレーションが存在するブロックです。
  - *offset* は、指定されたオペレーションが存在するオペレーションのオフセット値です。
- SCN – REDO ログ・ファイルに記録されるオペレーションの SCN。
- THREAD – オペレーションを実行したスレッド。
- USERNAME – オペレーションを実行したユーザの名前。
- EXECUTION TIME – オペレーションが実行された日付と時刻。
- OBJECT ID – 影響を受けるオブジェクトのデータベース ID。
- OBJECT NAME – 影響を受けるオブジェクトの名前。
- OPERATION – オペレーション・タイプ。
- REPLICATE – オペレーションの影響を受けるオブジェクトが Replication Agent によって複製の対象としてマーク付けされているかどうか (YES または NO)。
- TRANSACTION ID – このオペレーションを一部に含んでいるトランザクションの ID。
- REDO SQL – オペレーションをやり直すために使用できる SQL。
- UNDO SQL – オペレーションを元に戻すために使用できる SQL。
- SKIP COMMAND – 複製時にオペレーションが Replication Agent によってスキップされる原因となる Replication Agent コマンド。
- **ra\_helpop** コマンドは、複数のオペレーションに関する情報を表示できます。オペレーションに関する情報は、*locator* または *opid* によって指定された SCN を持つオペレーションごとに返されます。
- Oracle LogMiner スクリプト `$ORACLE_HOME/rdbms/admin/dbmslm.sql` がプライマリ・データベースにインストールされていないと、**ra\_helpop** コマンドは正しく動作しません。このスクリプトがインストールされていない場合、**ra\_helpop** はエラーを返します。
- LogMiner がインストールされたら、次のように入力してパブリック・シノニムを作成します。こうすることで、所有者としてログインしなくても LogMiner 関数を実行できるようになります。

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

**注意：**これは、Oracle 10g を使用している場合に必要となります。

- **ra\_helpop** コマンドを正しく機能させるには、以下の権限を **pds\_username** に付与する必要があります。
  - **EXECUTE\_CATALOG\_ROLE**
  - **SELECT ON V\_\$LOGMNR\_CONTENTS**
  - **SELECT ON V\_\$LOGMNR\_LOGS**
  - **SELECT ANY TRANSACTION**
- Oracle REDO ログでは、単一の DML コマンドが 2 つ以上の連続したオペレーションとして表される場合があります。ただし、Oracle LogMiner に表示されるのは最初のオペレーションの SCN だけです。そのため、Oracle REDO ログ内の連続したオペレーションの途中で起こるオペレーションについては、**ra\_helpop** によって結果が返されない可能性があります。指定した **opid** または **locator** 値に関する結果が **ra\_helpop** によって返されない場合は、**ra\_dumptran** コマンドを使用して、データベース・オペレーションが所属するトランザクションのトランザクション ID を指定し、トランザクション全体を Oracle REDO ログからダンプします。その後で、Oracle LogMiner によって見つけられなかったオペレーションをダンプで検索します。

## ra\_helptran

(Oracle のみ) すべてのオープン・トランザクションのリストを返します。

### 構文

```
ra_helptran
```

### 使用法

オープン・トランザクションがない場合、**ra\_helptran** は空の結果セットを返します。

## ra\_helpuser

(Oracle と Microsoft SQL Server のみ) RASD からプライマリ・データベース・ユーザに関する情報を返します。

**注意：**このコマンドは Oracle および Microsoft SQL Server でのみ使用可能です。

### 構文

```
ra_helpuser [user [, version]]
```

## パラメータ

- **user** – プライマリ・データベース内のユーザの名前またはユーザ ID。
- **version** – RASD 内のデータベース・ユーザのバージョン番号。

## 例

- **例 1 –**

```
ra_helpuser
```

このコマンドは、RASD 内のすべてのユーザの全バージョンに関する情報を返します。

- **例 2 –**

```
ra_helpuser bob
```

このコマンドは、RASD 内のデータベース名 “bob” の現在のバージョンに関する情報を返します。

- **例 3 –**

```
ra_helpuser bob,  
00000000000210a400003334000700003334000699940000d413c50000000000
```

このコマンドは、RASD 内の “bob” という名前のデータベース・ユーザのバージョン **00000000000210a400003334000700003334000699940000d413c50000000000** に関する情報を返します。

## 使用法

- **ra\_helpuser** コマンドは、プライマリ・データベース・ユーザに関する以下の情報を返します。
  - ユーザ ID
  - ユーザ名
  - ユーザのステータス (Current、Archived、または Dropped)
  - プライマリ・データベースのバージョン (ロケータ値)ユーザ ID とユーザ名は、**pdb\_xlog init** コマンドを使用して Replication Agent を初期化したときにプライマリ・データベースによって返される値です。
- オプションを指定しないで **ra\_helpuser** を呼び出すと、RASD 内のプライマリ・データベースのすべてのバージョンに含まれている全ユーザに関する情報が返されます。
- **user** オプションを指定して **ra\_helpuser** を呼び出すと、RASD 内のプライマリ・データベースに含まれている、指定したユーザの現在のバージョンに関する情報が返されます。

- *user* および *version* オプションを指定して **ra\_helpuser** を呼び出すと、RASD 内のプライマリ・データベースの指定したバージョンに含まれている、指定したユーザに関する情報が返されます。
- **ra\_helpuser** コマンドが有効なのは、Replication Agent インスタンスが Admin 状態、Replicating 状態、または Replication Down 状態のときです。
- **pdb\_xlog init** コマンドを使用した RASD の初期化をまだ行っていない場合、**ra\_helpuser** は結果を返しません。

#### 参照：

- ra\_config (86 ページ)
- ra\_help (99 ページ)
- ra\_helparticle (101 ページ)
- ra\_helppdb (102 ページ)
- ra\_helpdevice (103 ページ)
- ra\_helpfield (107 ページ)
- ra\_helplocator (109 ページ)

## ra\_license

Replication Agent のライセンス情報とライセンスされている機能を返します。

### 構文

```
ra_license [param]
```

### パラメータ

- **param – ra\_license** にこのライセンスの設定パラメータに関する情報を返すように指示します。

### 例

- 例 1 –

```
ra_license
```

このコマンドは、以下のような基本的なライセンス情報を返します。

License Name	Version	Quantity	Status	Expiry Date
RAX_SERVER	2012.xxxx	2	expirable	Oct 10 2013 7:30AM

- 例 2 –

```
ra_license param
```

このコマンドは、以下のライセンスの設定パラメータに関する情報を返します。

```
Property          Value
-----          -
License Edition   Development and
                  Testing License
License Type      CP
Licensed to       Sybase, Inc.
Total Licenses    2
Total in Use      1
Email Severity    NONE
SMTP Host         smtp
SMTP Port         25
Email Sender      tomserve@sybase.com
Email Recipients  deep13@sybase.com
10 rows Affected.
```

### 使用法

- **ra\_helpuser** コマンドはキーワードがない場合は基本的なライセンス情報を返します。
- **ra\_helpuser** コマンドに **param** キーワードを指定すると、ライセンスのエディション、タイプ、SMTP ホストと SMTP ポートを含む設定可能なライセンス・パラメータ情報を返します。

### ra\_locator

Replication Agent によって管理されている LTM ロケータの現在の値を返すか、プライマリ Replication Server からの LTM ロケータ値を要求するか、Replication Agent によって管理されている LTM ロケータの値をゼロに設定します。

### 構文

```
ra_locator [ { update | zero | move_truncpt } ]
```

### パラメータ

- **update** – プライマリ Replication Server からの新しい LTM ロケータ値を要求するためのオプションのキーワード。
- **zero** – Replication Agent のトランザクション・ログに格納されている LTM ロケータの値をゼロに設定するためのオプションのキーワード。
- **move\_truncpt** – トランケーション・ポイントを移動するキーワード。

### 例

- 例 1 –

```
ra_locator
```

このコマンドは、Replication Agent によって管理されている LTM ロケータの現在の値を次のように返します。

```
Locator
-----
000000005200000000000000527FFFFFFFFFFFFFFFF0022FB3B
(1 row affected)
```

• **例 2-**

```
ra_locator update
```

このコマンドは、プライマリ Replication Server からの新しい LTM ロケータ値を要求します。

• **例 3-**

```
ra_locator zero
```

このコマンドは、Replication Agent によって管理されている LTM ロケータの値をすべてゼロに設定します。

• **例 4-**

```
ra_locator move_truncpt
```

このコマンドは、トランザクション・ログのトランケーション・ポイントを現在のトランザクション・ログの末尾に移動します。

**使用法**

- オプションを指定しないで **ra\_locator** を呼び出すと、Replication Agent のインスタンスによって管理されている LTM ロケータの現在の値が返されます。UDB の場合、Replication Agent は LTM ロケータの値をプライマリ・データベースのテーブルに格納します。Oracle および Microsoft SQL Server の場合、この値は RASD に格納されます。

---

**注意：** Replication Agent によって管理されている LTM ロケータの値は、オリジン・キュー ID とも呼ばれます。

---

- **update** キーワードを指定して **ra\_locator** を呼び出すと、プライマリ Replication Server からの新しい LTM ロケータの値が要求され、Replication Agent によってその値が保存されます。

---

**注意：** **update** キーワードを指定して **ra\_locator** を呼び出すと、Replication Agent のインスタンスが **Replicating** 状態の場合に限り変更が有効になります。

---

- **zero** キーワードを指定して **ra\_locator** を呼び出すと、Replication Agent によって管理されている LTM ロケータの値がゼロに設定されます。
- LTM ロケータには、Replication Agent がトランザクション・ログの読み取り開始位置を決定するために使用する情報が格納されます。

Replication Agent は起動後またはコネクション・エラーからの回復後すぐに、プライマリ Replication Server からの LTM ロケータ値を自動的に要求します。

- プライマリ Replication Server から返された LTM ロケータの値がゼロの場合、Replication Agent はトランザクション・ログ・システム・テーブルに格納されている LTM ロケータ値を使用します。
- トランザクション・ログ・システム・テーブルの LTM ロケータの値がゼロの場合、Replication Agent は、トランザクション・ログの読み取りをログの現在の先頭から開始するか、UDB の場合はログの末尾から開始します。
- オリジン・キュー ID の形式の詳細については、『Replication Agent プライマリ・データベース・ガイド』で特定のプライマリ・データ・サーバに関するセクションを参照してください。
- Replication Agent のトランザクション・ログが存在しない場合、**ra\_locator** コマンドはエラー・メッセージを返します。
- **zero** キーワードを指定した **ra\_locator** コマンドが有効なのは、Replication Agent インスタンスが Admin 状態または Replication Down 状態の場合のみです。
- **zero** キーワードを指定しない **ra\_locator** コマンドが有効なのは、Replication Agent インスタンスが Admin 状態、Replicating 状態、または Replication Down 状態のときです。
- **move\_truncpt** キーワードを指定して **ra\_locator** を呼び出すと、Replication Agent コンポーネントへの変更なしに、トランケーション・ポイントがログの末尾 (Oracle の場合は現在のオンライン REDO ログの末尾) に移動されます。Replication Agent が初期化されていない場合、**move\_truncpt** オプションの効果はありません。

---

**注意：** **move\_truncpt** オプションによって設定されたログ内のロケーションよりも前に発生するログ開始ポイントを Replication Server が要求することを回避するには、Replication Server のプライマリ・コネクションの LTM ロケータ値をゼロにする必要があります。プライマリ・データベース・コネクションに対して Replication Server システム・データベース (RSSD) の **rs\_zeroltm** コマンドを実行することにより、LTM ロケータの値をゼロにします。

---

**ra\_locator move\_truncpt** を使用してセカンダリ・トランケーション・ポイントをプライマリ・データベース・トランザクション・ログの末尾に移動すると、ログに記録されている DDL コマンドがスキップされる危険性があります。DDL コマンドは、Replication Agent システム・データベース (RASD) 内に格納された情報の更新のために Replication Agent によって使用された可能性があります。一部のログ・レコードの処理をスキップしたために RASD の内容が正しくない場合は、**ra\_admin refresh** を使用して RASD 内のすべてのスキーマ情報を強制的に更新できます。RASD に格納された単一のオブジェクトのスキーマのみに関心がある場合は、そのオブジェクトのみのマークを外して、再度マーク



付けします。これにより、そのオブジェクトのスキーマが RASD に再度読み込まれます。

#### 参照：

- pdb\_gen\_id (15 ページ)
- pdb\_truncate\_xlog (73 ページ)
- pdb\_xlog (75 ページ)
- ra\_admin (81 ページ)

## ra\_maintid

プライマリ・データベース・メンテナンス・ユーザのログイン名を返します。

### 構文

```
ra_maintid
```

### 使用法

- Replication Server では、データベースに接続するたびにメンテナンス・ユーザのログイン名が必要になります。データベース・コネクション用のメンテナンス・ユーザ・ログイン名は、Replication Server の **create connection** または **alter connection** コマンドを使用して指定されます。

プライマリ・データベースのメンテナンス・ユーザ・ログイン名を Replication Server で変更すると (**alter connection** コマンドを使用)、Replication Agent が Replicating 状態の場合に、新しいメンテナンス・ユーザ・ログイン名が Replication Agent に自動的に送信されます。

Replication Agent が Replicating 状態になるたびに、プライマリの Replication Server からプライマリ・データベースのメンテナンス・ユーザ・ログイン名が自動的に取得され、キャッシュされます。

- **ra\_maintid** を呼び出すと、キャッシュされたプライマリ・データベース・メンテナンス・ユーザのログイン名が次のように返されます。

```
maintenance user
-----
SYS
(1 row affected)
```

- Replication Agent が Replicating 状態のときに **ra\_maintid** を呼び出すと、常に正しいメンテナンス・ユーザ・ログイン名が返されます。

Replication Agent が Admin 状態または Replication Down 状態のときに **ra\_maintid** を呼び出すと、正しいメンテナンス・ユーザ・ログイン名が返されない場合があります。これは、メンテナンス・ユーザのログイン名が Replication Agent に

## コマンド・リファレンス

よって前回取得され、格納された後で、Replication Server で変更されている可能性があるためです。

- **filter\_maint\_userid** 設定パラメータは、双方向複写をサポートするために用意されています。双方向複写では、プライマリ・データベースは、Replication Server によって適用されるトランザクションを含んだレプリケート・データベースとしても機能します。

**filter\_maint\_userid** パラメータの値が **true** の場合、メンテナンス・ユーザによって適用されるデータベース・オペレーションはプライマリ・データベースから複写されません。Replication Agent の Log Reader コンポーネントは、トランザクション・ログを読み取るときに、メンテナンス・ユーザによって適用されるデータ変更オペレーションを除外します。

- **ra\_maintid** コマンドが有効なのは、Replication Agent インスタンスが Admin 状態、Replicating 状態、または Replication Down 状態のときです。

### 参照：

- ra\_config (86 ページ)
- ra\_statistics (126 ページ)

## ra\_marker

プライマリ・データベースのトランザクション・ログにマーカを挿入します。

### 構文

```
ra_marker command_tag
```

### パラメータ

- **command\_tag** – サブスクリプション・マテリアライゼーションに使用される情報を格納する *varchar* 値。Microsoft SQL Server データ・サーバで使用される場合、この値は *varchar(8000)* になります。Oracle データ・サーバで使用される場合、この値は *varchar(4000)* になります。

### 例

- **例 1 –**

```
ra_marker 'activate subscription 309 0 with suspension'
```

このコマンドは、Replication Server の **activate subscription** コマンドを呼び出すマーカ・オブジェクトをプライマリ・データベースのトランザクション・ログに挿入します。

## 使用法

- **ra\_marker** を呼び出すと、プライマリ・データベース・トランザクション・ログに記録されているプライマリ・データベース内のトランザクションが Replication Agent によって実行されます。複製済みトランザクションは、マーカ・オブジェクトとしてプライマリ Replication Server に送信されます。
- Replication Agent のトランザクション・ログが存在しない場合、**ra\_marker** コマンドはエラー・メッセージを返します。
- **ra\_marker** コマンドが有効なのは、Replication Agent インスタンスが Admin 状態、Replicating 状態、または Replication Down 状態のときです。
- Replication Server の **rs\_marker** システム・ファンクションの詳細については、『Replication Server 管理ガイド』と『Replication Server リファレンス・マニュアル』を参照してください。

### 参照：

- **ra\_dump** (94 ページ)

## ra\_migrate

Replication Agent のリリース間のアップグレード・タスクとダウングレード・タスクに必要なマイグレーションを実行します。

このコマンドは、アップグレード・プロセスを完了するか、Replication Agent のダウングレード元インスタンス (後のバージョン) が実行する **ra\_downgrade** コマンドによって開始されたダウングレード・プロセスを完了するために使用されます。

---

**注意：** **ra\_downgrade\_prepare** コマンドおよび **ra\_downgrade\_accept** コマンドは廃止されました。できる限り **ra\_downgrade** コマンドと **ra\_migrate** コマンドを使用してください。『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。

---

### 構文

```
ra_migrate
```

### パラメータ

- **None** - パラメータはありません。

## 使用法

- 新しいリリースの Replication Agent にアップグレードした後は、まず **ra\_migrate** を実行して Replication Agent の最新バージョンに更新する必要があります。

- **ra\_migrate** コマンドが有効なのは、Replication Agent インスタンスが Admin 状態または Replication Down 状態のときだけです。
- (Oracle のみ) **ra\_migrate** コマンドは、以下の権限が **pds\_username** に付与されていることを確認します。
  - **EXECUTE\_CATALOG\_ROLE**
  - **select on V\_\$LOGMNR\_CONTENTS**
  - **select on V\_\$LOGMNR\_LOGS**これらの権限は、**ra\_dumptran** コマンドおよび **ra\_helpop** コマンドが正しく機能するために必要です。これらの権限は、複製には必要なく、デバッグおよびトラブルシューティングに使用する **ra\_dumptran** コマンドおよび **ra\_helpop** コマンドを使用する場合にのみ必要です。**ra\_migrate** を呼び出すときにこれらの権限が付与されていない場合は、警告メッセージが返され、Replication Agent ログ・ファイルに記録されます。
- **ra\_migrate** コマンドは、ダウングレード後にファイルから RASD を復元します。
- **ra\_migrate** を使用するには、Replication Agent がプライマリ・データベースと接続できなければなりません。

### 参照：

- **ra\_downgrade** (90 ページ)
- **ra\_downgrade\_accept** (91 ページ)
- **ra\_downgrade\_prepare** (93 ページ)

## ra\_purge\_first\_open

(Oracle のみ) オープン・トランザクションのリストから最初のオープン・トランザクションを削除します。

### 構文

```
ra_purge_first_open
```

### 使用法

- オープン・トランザクションがない場合、**ra\_purge\_first\_open** を呼び出すとエラーになります。
- **ra\_purge\_first\_open** は、問題のトランザクションに複製されるべきコンテンツが含まれている可能性があるときに、Sybase サポート・センタの指示を受けた場合にのみ使用します。

## ra\_set\_autocorrection

マーク付けされたテーブルのオートコレクションの有効化と無効化

### 構文

```
ra_set_autocorrection { all | tablename } [, { enable | disable } ]
```

### パラメータ

- **all** – マーク付けされたすべてのテーブルに対してオートコレクションを有効にするには、次のように **ra\_set\_autocorrection** コマンドの後に **all** および **enable** キーワードを続けます。

**ra\_set\_autocorrection all, enable**

マーク付けされたすべてのテーブルに対してオートコレクションを無効にするには、次のように **ra\_set\_autocorrection** コマンドの後に **all** および **disable** キーワードを続けます。

**ra\_set\_autocorrection all, disable**

- **tablename** – マーク付けされた単一のテーブルに対してオートコレクションを有効にするには、次のように **ra\_set\_autocorrection** コマンドの後に **tablename** パラメータと **enable** キーワードを続けます。

**ra\_set\_autocorrection tablename, enable**

マーク付けされた単一のテーブルに対してオートコレクションを無効にするには、次のように **ra\_set\_autocorrection** コマンドの後に **tablename** パラメータと **disable** キーワードを続けます。

**ra\_set\_autocorrection tablename, disable**

マーク付けされた単一のテーブルのオートコレクション・ステータスを表示するには、次のように **ra\_set\_autocorrection** コマンドの後に **tablename** パラメータのみを続けます。

**ra\_set\_autocorrection tablename**

- **enable** – マーク付けされた単一のテーブルまたはマーク付けされたすべてのテーブルに対してオートコレクションを有効にするには、**enable** キーワードを使用します。
- **disable** – マーク付けされた単一のテーブルまたはマーク付けされたすべてのテーブルに対してオートコレクションを無効にするには、**disable** キーワードを使用します。

## 例

- 例 1-

```
ra_set_autocorrection mytable, enable
```

このコマンドは、マーク付けされたテーブル `mytable` に対してオートコレクションを有効にします。

- 例 2-

```
ra_set_autocorrection all, enable
```

このコマンドは、マーク付けされたすべてのテーブルに対してオートコレクションを有効にします。

- 例 3-

```
ra_set_autocorrection mytable, disable
```

このコマンドは、マーク付けされたテーブル `mytable` に対してオートコレクションを無効にします。

- 例 4-

```
ra_set_autocorrection all, disable
```

このコマンドは、マーク付けされたすべてのテーブルに対してオートコレクションを無効にします。

- 例 5-

```
ra_set_autocorrection mytable
```

このコマンドは、マーク付けされたテーブル `mytable` のオートコレクション・ステータスを表示します。

## 使用法

- このコマンドは、Replication Server のオートコレクション機能をサポートするために使用されます。
- 複製の対象としてマーク付けされていないテーブルのオートコレクションは設定できません。マーク付けされたテーブルに対してオートコレクションを有効にした後で、マークが解除された場合、このテーブルのオートコレクションは自動的に無効になります。
- Replication Agent が複製しているマーク付きテーブルに対してオートコレクションが有効になっている場合、**column\_compression** および **ltl\_send\_only\_primary\_keys** 設定パラメータは無視されます。

- オートコレクションが有効になっている場合、Replication Server は各 **update** オペレーションまたは **insert** オペレーションを 2 つのオペレーションの組 (つまり、**delete** オペレーションの後に **insert** を続ける) に変換します。
- プライマリ・データベースが Oracle であり、テーブル・レベルのサブリメンタル・ロギングがまだ有効になっていない場合に、オートコレクションを有効にすると、指定したテーブルのすべてのカラムに対してサブリメンタル・ロギングが有効になります。
- Replication Agent インスタンスが最小カラム・データを送信するように設定されている (**column\_compression** および **ltl\_send\_only\_primary\_keys** が true に設定されている) 場合、複製定義でサーチャブルに指定されているカラムの一部のカラム・データが省略されることがあります。その結果として、サブスクリプションを作成するデータベースで挿入、サブスクリプション・マイグレーション、またはカスタム・ファンクション文字列に必要なデータがない場合、エラーが発生することがあります。このため、サーチャブル・カラムのある複製定義で参照されるテーブルのオートコレクションを有効にすることをおすすめします。

## ra\_set\_login

Replication Agent 管理者のログインとパスワードを設定します。

### 構文

```
ra_set_login username, password[, encryption ]
```

### パラメータ

- **username** – Replication Agent 管理者のログイン名。
- **password** – Replication Agent 管理者のパスワード。
- **encryption** – Replication Agent 管理者ログイン・パスワードの暗号化モード。次の値を指定できます。
  - 3 – クライアントは Tabular Data Stream™ (TDS) Extended Plus の暗号化パスワード・プロトコルを使用する必要があります。
  - 2 – クライアントは拡張パスワード暗号化によるネゴシエーションまたは TDS Extended Plus の暗号化パスワード・プロトコルを使用する必要があります。
  - 1 – クライアントは拡張パスワード暗号化によるネゴシエーションを使用する必要があります。
  - 0 – クライアントは暗号化モードを選択でき、暗号化を使用しなくても構いません。

## 例

### • 例 1 –

```
ra_set_login tom, S3Rv0
```

このコマンドは Replication Agent 管理者ログインを “tom” に設定し、パスワードを “S3Rv0” に設定します。

```
ra_set_login crow, Tr0b0t, 3
```

このコマンドは Replication Agent 管理者ログインを “crow” に設定し、パスワードを “Tr0b0t” に設定して、TDS Extended Plus の暗号化パスワード・プロトコルを使用することをクライアントに対して要求します。

## 使用法

- Replication Agent 管理者ログインには、管理ポートを使用して Replication Agent インスタンスにログインするためのパーミッションがあります。
- 同時に複数の Replication Agent 管理者ログイン名が有効になることはありません。
- Replication Agent 管理者ログインまたはパスワードに加えた変更は直ちに有効になります。Replication Agent インスタンスに次回ログインするときは、新しいログインとパスワードを使用する必要があります。
- 管理者ログインに指定されたパスワードは、Replication Agent の設定ファイルで暗号化されます。
- **ra\_set\_login** コマンドが有効なのは、Replication Agent インスタンスが Admin 状態、Replicating 状態、または Replication Down 状態のときです。
- このパラメータに対して暗号化パスワードを指定できます。

## 参照：

- ra\_config (86 ページ)

## ra\_statistics

Replication Agent のコンポーネントと Java 仮想マシン (Java VM) に関するパフォーマンス統計情報を返すか、統計情報カウンタをリセットします。

---

**注意：** 統計情報カウンタはプライマリ・データベースによって異なる場合があります。

---

## 構文

```
ra_statistics [component|reset]
```



## パラメータ

- **component** – Replication Agent コンポーネントまたは Java VM を識別するオプションのキーワード。有効な *component* キーワードは次のとおりです。
  - **LR** – Log Reader
  - **LTI** - Log Transfer Interface
  - **LTM** – Log Transfer Manager
  - **VM** - Java 仮想マシン
- **reset** – 統計カウンタをリセットするオプションのキーワード。

## 例

- **例 1 –**

```
ra_statistics
```

このコマンドは、Replication Agent インスタンスと Java VM のパフォーマンス統計情報を返します。

- **例 2 –**

```
ra_statistics reset
```

このコマンドは、Replication Agent インスタンスの統計カウンタをリセットします。

- **例 3 –**

```
ra_statistics VM
```

このコマンドは、Java VM 統計情報 (126 ページ) の表で説明されている Java VM の統計情報を返します。

## 使用法

- オプションを指定せずに **ra\_statistics** を呼び出した場合、このコマンドはすべての Replication Agent コンポーネントと Java VM の統計情報を返します。
- *component* オプションを指定して **ra\_statistics** を呼び出した場合、**ra\_statistics** コマンドは指定された Replication Agent コンポーネントまたは Java VM の統計情報を返します。
- 表 3: Java VM 統計情報 (128 ページ) に、Java VM について返される統計情報を示します。

表 3 : Java VM 統計情報

統計	説明
VM 最大メモリ	Java VM が使用できる最大メモリ (バイト単位)
割り付けられた VM 合計メモリ	起動時に Java VM に割り付けられた合計メモリ (バイト単位)
VM 空きメモリ	Java VM に割り付けられていて使用されていないメモリ (バイト単位)
VM メモリ使用量	Java VM に割り付けられていて使用されているメモリ (バイト単位)
VM 最大メモリ使用率 (%)	Java VM が使用可能な最大メモリに対して現在使用している割合 (パーセンテージ)

- 表 4 : Log Transfer Manager 統計情報 (128 ページ)に Log Transfer Manager コンポーネントについて返される統計情報を示します。

表 4 : Log Transfer Manager 統計情報

統計	説明
統計情報を取得した日時	<code>ra_statistics</code> が呼び出され、情報が返された曜日、日付、および時刻
複写を最後に開始した日時	Replicating 状態になった曜日、日付、および時刻
統計を最後にリセットした日時	統計カウンタがリセットされた曜日、日付、および時刻
グローバル LRUCache に保持されている項目数	最も長い間使用されていない (LRU) 内部キャッシュに保持されているオブジェクト参照の数

- 表 5 : UDB の Log Reader 統計情報 (128 ページ)に UDB の Log Reader 統計情報を示します。

表 5 : UDB の Log Reader 統計情報

統計	説明
スキャン済みトランザクション・ログ数	ログ・デバイスから読み込まれたオペレーションの数
トランザクション・ログ・スキャンあたりの平均未処理オペレーション数	トランザクション・ログ・スキャンごとの未処理オペレーションの平均数

統計	説明
トランザクション・ログ・スキャン平均時間	ログ・デバイスから読み込まれたオペレーションのトランザクション・ログ・スキャン平均時間
複写済みオペレーション数	正常に複写されたオペレーションの数
複写済みトランザクション数	正常に複写されたトランザクションの数
スキップされたトランザクション・ログ・オペレーション数 (maint_user、マーク解除テーブル)	スキップされたトランザクション・ログ・オペレーションの数
空のトランザクション・ログの平均待機時間	トランザクション・ログが使用されていない平均時間
平均PDB サービス時間/オペレーション	各データベースのサービスとオペレーションの平均時間
オペレーション・キュー・サイズ	オペレーションに使用されたキュー・サイズ
オペレーション・データ・ハッシュ・サイズ	オペレーションのデータ・ハッシュ・サイズ
トランケート済みトランザクション数	トランケートされたトランザクションの数

- 表 6 : Microsoft SQL Server の Log Reader 統計情報 (129 ページ)に Microsoft SQL Server の Log Reader 統計情報を示します。

**表 6 : Microsoft SQL Server の Log Reader 統計情報**

統計	説明
スキャン済み合計オペレーション数	前回のリセット以降にログ・デバイスから読み込まれたオペレーションの数
処理済み合計オペレーション数	前回のリセット以降にログ・デバイスから読み込まれ LTI に渡されたオペレーションの数
スキップされた合計オペレーション数	前回のリセット以降に、ログ・デバイスから読み込まれたが何らかの理由で処理されていないオペレーションの数
フィルタ済み合計メンテナンス・ユーザ・オペレーション数	前回のリセット以降に、ログ・デバイスから読み込まれスキップされたメンテナンス・ユーザ・オペレーションの数
オペレーション平均処理時間	前回のリセット以降の Log Reader オペレーション平均処理時間 (ミリ秒単位)

統計	説明
処理済み合計トランザクション数	前回のリセット以降にログ・デバイスから読み込まれたトランザクションの数
スキップされた合計トランザクション数	前回のリセット以降に、ログ・デバイスから読み込まれたが何らかの理由で処理されていないトランザクションの数
開かれた合計トランザクション数	前回のリセット以降に、ログ・デバイスから読み込まれた <b>begin transaction</b> コマンドの数
閉じられた合計トランザクション数	前回のリセット以降に、ログ・デバイスから読み込まれた <b>commit</b> コマンドと <b>rollback</b> コマンドの数
コミット済み合計トランザクション数	前回のリセット以降に、ログ・デバイスから読み込まれた <b>commit</b> コマンドの数
アボート (ロールバック) された合計トランザクション数	前回のリセット以降に、ログ・デバイスから読み込まれた <b>rollback</b> コマンドの数
スキップされた合計システム・トランザクション数	前回のリセット以降に、ログ・デバイスから読み込まれスキップされたシステム・トランザクションの数
トランザクションあたりの平均オペレーション数	前回のリセット以降に、ログ・デバイスから読み込まれた各トランザクションの平均オペレーション数
現在のスキャン・バッファ・サイズ	Log Reader スキャン・バッファの現在のサイズ (バイト単位)
現在のオペレーション・キュー・サイズ	Log Reader 入力キューの現在のサイズ (バイト単位)
現在のセッション・キャッシュ・サイズ	セッション・キャッシュの現在のサイズ (バイト単位)
ログ再配置ポイント・ロケータ	ログ・デバイスでの再配置ポイントのロケータ値
最後に処理されたオペレーション・ロケータ	ログ・デバイスから読み込まれた、最後に処理されたオペレーションのロケータ値
トランザクション・ログ・オペレーション平均待機時間 (ms)	前回のリセット以降に、新しい各オペレーションがログに記録されるまで Log Reader が待機した平均時間 (ミリ秒単位)
送信側オペレーション平均処理時間 (ms)	前回のリセット以降に、Log Reader の送信側が各オペレーションの処理に要した平均時間 (ミリ秒単位)

統計	説明
送信側オペレーション平均待機時間 (ms)	前回のリセット以降に、Log Reader の送信側が各処理済みオペレーションを LTI 入力キューに送信するのを待機した平均時間 (ミリ秒単位)
ChangeSet 平均送信時間 (ms)	前回のリセット以降に、Log Reader の送信側が各処理済みオペレーションを LTI 入力キューに送信するのに要した平均時間 (ミリ秒単位)
処理済み合計送信側オペレーション数	前回のリセット以降に、Log Reader の送信側が処理したオペレーションの数
現在のマーク付きオブジェクト・キャッシュ・サイズ	マーク付きオブジェクトのキャッシュ・サイズ

- 表 7: Oracle の Log Reader 統計情報 (131 ページ) に Oracle 用 Log Reader コンポーネントについて返された統計情報を示します。

表 7 : Oracle の Log Reader 統計情報

統計	説明
RBA 平均検索時間 (ms)	ログ・スキャナ配置中のレコード・バイト・アドレス (RBA) 平均検索時間
読み込まれた合計バイト数	プライマリ・データベース・トランザクション・ログから読み込まれた合計バイト数
読み込まれた合計ログ・レコード数	プライマリ・データベース・トランザクション・ログから読み込まれた合計ログ・レコード数
1 秒あたりの読み込まれた平均バイト数	プライマリ・データベース・トランザクション・ログから 1 秒あたりに読み込まれた平均バイト数
1 レコードあたりの平均バイト数	読み込まれた 1 ログ・レコードあたりの平均バイト数
1 ログあたりの平均読み込み時間 (ms)	1 プライマリ・データベース・トランザクション・ログあたりの平均読み込み時間
オンライン・ログ読み込み合計時間 (ms)	プライマリ・データベース・オンライン・トランザクション redo ログの読み込みに要した合計時間
アーカイブ・ログ読み込み合計時間 (ms)	プライマリ・データベース・トランザクション redo ログ・アーカイブの読み込みに要した合計時間
1 オンライン・ログ・デバイスあたりの平均読み込み時間 (ms)	1 オンライン・ログ・デバイスあたりの平均読み込み時間

統計	説明
1アーカイブ・ログ・デバイスあたりの平均読み込み時間 (ms)	1アーカイブ・ログ・デバイスあたりの平均読み込み時間
キュー内の合計ログ・レコード数	処理のキュー内にあるログ・レコードの合計数
フィルタされた合計ログ・レコード数	フィルタされたログ・レコードの合計数
ログ・スキャン・チェックポイント・セット・サイズ	現在、チェックポイント・セットにあるログ・レコードの数
1チェックポイントあたりの平均ログ・レコード数	読み込まれた各チェックポイント・ログ・レコードの平均ログ・レコード数
ログ・レコード・チェックポイント間の平均秒数	読み込んだログ・レコード・チェックポイントの間の平均秒数
スキャン済み合計オペレーション数	前回のリセット以降にログ・デバイスから読み込まれたオペレーションの数
処理済み合計オペレーション数	前回のリセット以降にログ・デバイスから読み込まれ LTI に渡されたオペレーションの数
スキップされた合計オペレーション数	前回のリセット以降に、ログ・デバイスから読み込まれたが何らかの理由で処理されていないオペレーションの数
フィルタ済み合計メンテナンス・ユーザ・オペレーション数	前回のリセット以降に、ログ・デバイスから読み込まれスキップされたメンテナンス・ユーザ・オペレーションの数
オペレーション平均処理時間	前回のリセット以降の Log Reader オペレーション平均処理時間 (ミリ秒単位)
処理済み合計トランザクション数	前回のリセット以降にログ・デバイスから読み込まれたトランザクションの数
スキップされた合計トランザクション数	前回のリセット以降に、ログ・デバイスから読み込まれたが何らかの理由で処理されていないトランザクションの数
開かれた合計トランザクション数	前回のリセット以降に、ログ・デバイスから読み込まれた <b>begin transaction</b> コマンドの数
閉じられた合計トランザクション数	前回のリセット以降に、ログ・デバイスから読み込まれた <b>commit</b> コマンドと <b>rollback</b> コマンドの数
コミット済み合計トランザクション数	前回のリセット以降に、ログ・デバイスから読み込まれた <b>commit</b> コマンドの数

統計	説明
アボート (ロールバック) された合計トランザクション数	前回のリセット以降に、ログ・デバイスから読み込まれた <b>rollback</b> コマンドの数
スキップされた合計システム・トランザクション数	前回のリセット以降に、ログ・デバイスから読み込まれスキップされたシステム・トランザクションの数
1 トランザクションあたりの平均オペレーション数	前回のリセット以降に、ログ・デバイスから読み込まれた各トランザクションの平均オペレーション数
現在のスキャン・バッファ・サイズ	Log Reader スキャン・バッファの現在のサイズ
現在のオペレーション・キュー・サイズ	Log Reader オペレーション・キューの現在のサイズ
現在のセッション・キャッシュ・サイズ	セッション・キャッシュの現在のサイズ
PDB からのクエリ・データで処理された合計 LOB オペレーション数	プライマリ・データベースから処理された LOB オペレーションの合計数
LOB オペレーション処理に関する PDB クエリ平均時間	LOB を 1 件処理するためプライマリ・データベースへのクエリに要した平均時間
現在のオペレーション・プロセッサ RASD がマーク付けしたオブジェクトのキャッシュ・サイズ	オペレーション・プロセッサがマーク付けしたオブジェクトのレポジトリ・キャッシュの現在のサイズ
オペレーション・プロセッサ RASD がマーク付けしたオブジェクトの合計キャッシュ・ヒット数	オペレーション・プロセッサがマーク付けしたオブジェクトのレポジトリ・キャッシュ・ヒットの合計数
オペレーション・プロセッサ RASD がマーク付けしたオブジェクトの合計キャッシュ・ミス数	オペレーション・プロセッサがマーク付けしたオブジェクトのレポジトリ・キャッシュ・ミスの合計数
ログ再配置ポイント・ロケータ	ログ・デバイスでの再配置ポイントのロケータ値
最後に処理されたオペレーション・ロケータ	ログ・デバイスから読み込まれた、最後に処理されたオペレーションのロケータ値
xlog オペレーション平均待機時間 (ms)	前回のリセット以降に、新しい各オペレーションがログに記録されるまで Log Reader が待機した平均時間 (ミリ秒単位)

統計	説明
送信側オペレーション平均処理時間 (ms)	前回のリセット以降に、Log Reader の送信側が各オペレーションの処理に要した平均時間 (ミリ秒単位)
送信側オペレーション平均待機時間 (ms)	前回のリセット以降に、Log Reader の送信側が各処理済みオペレーションを LTI 入力キューに送信するのを待機した平均時間 (ミリ秒単位)
変更セット平均送信時間 (ms)	前回のリセット以降に、Log Reader の送信側が各処理済みオペレーションを LTI 入力キューに送信するのに要した平均時間 (ミリ秒単位)
処理済み送信側オペレーション数	前回のリセット以降に、Log Reader の送信側が処理したオペレーションの数
現在のマーク付きオブジェクト・キャッシュ・サイズ	マーク付きオブジェクトのキャッシュ・サイズ

- 表 8 : Oracle RAC の追加統計情報 (134 ページ)にプライマリ・データベースが Oracle RAC である場合に返される統計情報を示します。これらの統計情報は、表 7 : Oracle の Log Reader 統計情報 (131 ページ)に示されている通常の Log Reader 統計情報とともに返されます。

**表 8 : Oracle RAC の追加統計情報**

統計	説明
ログ・スキャン・リーダの現在の LSN	各クラスタ・インスタンスについて読み込み中のログの現在のログ・シーケンス番号
ログ・スキャン・リーダのログ終わりステータス	各クラスタ・ログ・スキャナのログ終わりの現在のステータス
ログ・スキャン・リーダの最後の読み込み時間	各クラスタ・スキャナの最後の読み込み以降に経過した秒数
ログ・スキャン・レコード・セット分散統計	ログ・スキャン・スレッド全体でのログ・スキャン・チェックポイント・セットの分散統計
ログ・スキャン・リーダの最後のレコード SCN	各クラスタ・スキャナが最後に読み込んだログ・レコードの SCN
ログ・スキャン・リーダのチェックポイント	各クラスタ・スキャナが最後に読み込んだチェックポイント・ログ・レコードのチェックポイント SCN



統計	説明
ログ・スキャン・チェックポイント SCN	すべてのクラスタ・スキャナに基づく現在のチェックポイント SCN
ログ・スキャン・アクティブ・チェックポイント SCN	すべてのクラスタ・スキャナに基づくアクティブなチェックポイント SCN
1 redo ログ・スレッドあたりの合計ログ・レコード数	ログ・スキャン・スレッド全体で読み込んだ合計ログ・レコードの分散統計
ログ・スキャン・レコード・セット・サイズ	各クラスタ・ログ・スキャナの現在のスキャン・レコード・セット・サイズ
ログ・スキャン・チェックポイント・キュー・サイズ	各クラスタ・ログ・スキャナの現在のチェックポイント・キュー・サイズ

- 表 9 : Log Transfer Interface 統計情報 (135 ページ)に Log Transfer Interface コンポーネントについて返される統計情報を示します。

**表 9 : Log Transfer Interface 統計情報**

統計	説明
送信された LTL コマンド数	前回のリセット以降に、Replication Server に送信された LTL コマンドの合計数
LTL コマンドの平均サイズ	前回のリセット以降に、Replication Server に送信された各 LTL コマンドの平均サイズ (バイト単位)
1 秒あたりの平均 LTL コマンド数	前回のリセット以降に、Replication Server に送信された 1 秒あたりの LTL コマンドの平均数
送信された総バイト数	前回のリセット以降に、Replication Server に送信されたバイト数
転送中の 1 秒あたりの平均バイト数	前回のリセット以降に、コネクションを介して Replication Server に送信された 1 秒あたりの平均バイト数
LTL バッファ・キャッシュ平均時間	LTL コマンドが LTL バッファに入れられてから実際に Replication Server に送信されるまでの平均時間 (ミリ秒単位)
Rep Server 平均ターンアラウンド・タイム	前回のリセット以降に、Replication Server が送信された各 LTL コマンド・バッファを確認応答するまでの平均時間 (ミリ秒単位)
分配作成平均時間	前回のリセット以降に、LTI が変更セットを LTL に変換するのに要した平均時間 (ミリ秒単位)

統計	説明
平均 LTL バッファ・サイズ	前回のリセット以降に、Replication Server に送信された各 LTL バッファの平均サイズ (バイト単位)
LTM バッファ平均使用率 (%)	前回のリセット以降に、Replication Server に送信された各 LTL バッファの平均使用率 (LTL バッファ・サイズのパーセンテージ)
1 バッファあたりの平均 LTL コマンド数	前回のリセット以降に、Replication Server に送信された 1 バッファあたりの LTL コマンドの平均数
エンコード済みカラム名 キャッシュ・サイズ	現在のエンコード済みカラム名キャッシュ・サイズ
LTI キューの現在のコマンドの数	LTI キューの現在のコマンドの数
LTI キューの現在フォーマットされていないコマンドの数	LTI キューの現在フォーマットされていないコマンドの数
最後に送信された QID	Replication Server に最後に送信されたオリジン・キュー ID の 16 進数値
最後に送信されたトランザクション ID	Replication Server に最後に送信されたトランザクション ID の 16 進数値

- Replication Agent インスタンスが Replicating 状態になるたび、統計カウンタは自動的にリセットされます。
- **ra\_statistics** が **reset** キーワードを付けて呼び出されると、Replication Agent はすべての統計情報を直ちにリセットします。ただし、以下を除きます。
  - 統計情報を取得した日時 (LTM)
  - 複写を最後に開始した日時 (LTM)
  - 統計を最後にリセットした日時 (LTM)
  - 最後に送信された QID (LTI)
  - 最後に送信されたトランザクション ID (LTI)
  - すべての Java VM 統計情報

**注意：** **ra\_statistics** が呼び出されるたびに、Java VM 統計情報がすべて更新されます。キュー、バッファ、およびキャッシュ・サイズの統計情報もすべて更新されますが、その方法は平均値や合計値がリセットされる方法とは異なります。

- **ra\_statistics** コマンドが有効なのは、Replication Agent インスタンスが Admin 状態、Replicating 状態、または Replication Down 状態のときです。

**参照：**

- `ra_statrack_interval` (240 ページ)
- `ra_status` (138 ページ)

**ra\_statrack**

統計追跡スレッドの開始と停止を行います。

**構文**

```
ra_statrack { start | stop | status }
```

**パラメータ**

- **start** – 統計追跡スレッドを開始します。
- **stop** – 統計追跡スレッドを停止します。
- **status** – Replication Agent の稼働ステータスを表示します。

**使用法**

統計追跡スレッドは 60 秒のデフォルト間隔で統計を収集し、`STATRACK.log` に統計を記録します。また、**ra\_statrack** は Replication Agent の現在の稼働状況も表示します。

**参照：**

- `ra_statrack_interval` (240 ページ)
- `ra_statrack_list` (137 ページ)

**ra\_statrack\_list**

統計グループを追跡リストに追加またはリストから削除、追跡リストの交換、現在追跡されている統計リストの表示を行います。

**構文**

```
ra_statrack_list  
[ reset |  
  { {add | delete}, statistic_name } |  
  { replace, statistic_list } ]
```

**パラメータ**

- **reset** – 利用可能な統計をすべて追跡するようにリストをリセットします。
- **add** – 指定された統計を追跡されている統計のリストに追加します。

## コマンド・リファレンス

- **delete** – 指定された統計を追跡されている統計のリストから削除します。
- **statistic\_name** – 統計を追跡されている統計のリストに追加またはリストから削除します。
- **replace** – 追跡されている統計の現在のリストを指定されたリストに交換します。
- **statistic\_list** – 追跡されている統計の現在のリストにとって代わる統計リストです。

### 使用法

- *statistic\_name* の値は、VM、LTM、LTI または LR でなければなりません。
  - **LR** – Log Reader
  - **LTI** – Log Transfer Interface
  - **LTM** – Log Transfer Manager
  - **VM** – Java 仮想マシンこれらの値では大文字と小文字が区別されません。
- *statistic\_list* の値は、VM、LTM、LTI または LR でなければなりません。これらの値では大文字と小文字が区別されません。複数の値を指定する場合、リストは二重引用符で囲み、リストのアイテムはカンマで区切る必要があります。

### 参照：

- `ra_statrack` (137 ページ)
- `ra_statrack_interval` (240 ページ)

## ra\_status

Replication Agent インスタンスの現在の状態を返します。

### 構文

```
ra_status
```

### 使用法

- **ra\_status** は、呼び出されると、Replication Agent インスタンスの現在の状態とその短い説明を次のように返します。

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

---

**注意：**説明の最初の単語が "Transitioning" の場合、Replication Agent インスタンスは状態間を遷移中です。Replication Agent インスタンスが状態遷移中の場合、一部のコマンドは無効になります。

---

- Replication Agent の状態は、以下のとおりです。
  - Admin – この状態のとき、Replication Agent インスタンスは実行中ですが、有効なコネクションはありません。Replication Agent インスタンスが Admin 状態の場合、どの設定パラメータも変更できます。
  - Replicating – この状態のとき、Log Reader コンポーネントは、プライマリ・データベースから複製するオペレーションのトランザクション・ログをスキャンしています。複製するオペレーションがある場合、Log Transfer Interface コンポーネントは Replication Server に LTL コマンドを送信しています。
  - Replicating (Resynchronization) – この状態のとき、Replication Agent は、再起動が完了し、プライマリ・データベースおよびレプリケート・データベースを再同期しています。
  - Replication Down – この状態のときは、エラーにより複製が停止しています。エラーが解決すると、Replication Agent は Replicating 状態に復帰できます。

---

**注意：**Replication Agent の動作は Replication Down 状態と Admin 状態とで同じですが、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 の状態の詳細については、『Replication Agent 管理ガイド』を参照してください。

- **ra\_status** コマンドが有効なのは、Replication Agent インスタンスが Admin 状態、Replicating 状態、または Replication Down 状態のときです。

#### 参照：

- quiesce (80 ページ)
- ra\_statistics (126 ページ)
- resume (152 ページ)
- shutdown (161 ページ)
- suspend (162 ページ)

## **ra\_truncatearticles**

RASD で未使用のアーティクルをトランケートします。

### **構文**

```
ra_truncatearticles locator
```

## パラメータ

- **locator** – システム・データ・レポジトリから古いバージョンのアーティクルをトランケートするためのカットオフ・ポイントを識別するログ・ロケータ値 (LTM ロケータ)。

## 使用法

- **ra\_truncatearticles** は呼び出されると、*locator* 値で識別されるバージョンより古いバージョンのプライマリ・データベース・アーティクルを最新のバージョンを除いてすべてシステム・データ・レポジトリからトランケートします。現在 (最新) のバージョンのアーティクルが *locator* 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。
- プライマリ・データベースで実行されるほとんどの一般的な DDL コマンドとストアド・プロシージャ (**alter table** など) は、トランザクション・ログに記録され、スタンバイ・データベースに複写されます。これらの DDL トランザクションが複写用に処理されると、Replication Agent は RASD を自動的に更新し、影響を受けたプライマリ・データベース・アーティクルの新しいバージョンが作成されます。  
定期的なメンテナンス手順の一環として **ra\_truncatearticles** を使用して、RASD が無制限に大きくならないようにしてください。詳細については、『Replication Agent 管理ガイド』を参照してください。

---

**注意：** トランケートする前に、必ず **rasd\_backup** を使用して RASD をバックアップしてください。

---

- **ra\_truncatearticles** コマンドが有効なのは、Replication Agent インスタンスが Admin 状態、Replicating 状態、または Replication Down 状態のときです。

## 参照：

- [ra\\_truncateusers](#) (141 ページ)

## ra\_truncateddlfilters

(Oracle のみ) RASD でフィルタされる DDL コマンドの古いリストをトランケートします。

## 構文

```
ra_truncateddlfilters locator
```

## パラメータ

- **locator** – フィルタされた DDL コマンドの古いリストをシステム・データ・レポジトリからトランケートするためのカットオフ・ポイントを識別するログ・ロケータ値 (LTM ロケータ)。

## 使用法

- **ra\_truncateddlfilters** は呼び出されると、*locator* 値で識別されるリスト・バージョンより古いフィルタされた DDL コマンドのリストをシステム・データ・レポジトリからすべてトランケートします。

## ra\_truncateusers

(Oracle と Microsoft SQL Server のみ) RASD のシステム・データ・レポジトリの古いバージョンのプライマリ・データベース・ユーザをトランケートします。

---

**注意：** このコマンドは Oracle および Microsoft SQL Server でのみ使用可能です。

---

## 構文

```
ra_truncateusers locator
```

## パラメータ

- **locator** – システム・データ・レポジトリから古いバージョンのデータベース・ユーザをトランケートするためのカットオフ・ポイントを識別するログ・ロケータ値 (LTM ロケータ)。

## 使用法

- **ra\_truncateusers** は呼び出されると、*locator* 値で識別されるバージョンより古いバージョンのプライマリ・データベース・ユーザを最新のバージョンを除いてすべてシステム・データ・レポジトリからトランケートします。  
現在 (最新) のバージョンのユーザが *locator* 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。
- **ra\_truncateusers** コマンドが有効なのは、Replication Agent インスタンスが Admin 状態、Replicating 状態、または Replication Down 状態のときです。

## 参照：

- ra\_truncatearticles (139 ページ)

## ra\_updatedevices

(Oracle と Microsoft SQL Server のみ) RASD のプライマリ・データベース・ログ・デバイスに関する情報を更新します。

---

**注意：** このコマンドは Oracle および Microsoft SQL Server でのみ使用可能です。

---

### 構文

```
ra_updatedevices
```

### 使用法

- Oracle の場合、Automatic Storage Management (ASM) が redo ログ・ファイルを管理し、ディスク・グループがディスクの追加または削除で変更されたときは、**ra\_updatedevices** コマンドを呼び出してログ・デバイス・レポジトリを正しい ASM 記憶領域情報で更新する必要があります。
- **ra\_updatedevices** が呼び出されると、Replication Agent は以下の動作を行います。
  - アーカイブ・ログ情報を更新します。
  - ログ・デバイス・レポジトリにあるデータをすべて削除します。

---

**注意：** デバイスの位置が設定されている場合、その位置は上書きされません。

---

- すべてのログ・デバイスに関する情報についてプライマリ・データベースにクエリを実行します。
- RASD のログ・デバイス・レポジトリにプライマリ・データベースから返されたプライマリ・データベース・ログ・デバイスに関する最新の情報を再入力します。
- プライマリ・データベースに関連付けられたログ・デバイスが、プライマリ・データ・サーバで追加、削除、拡張、または移動された場合、以下を行ってください。
  - 複写を停止 (**quiesce** または **suspend** を使用) して、Replication Agent インスタンスを Admin 状態にします。
  - **ra\_updatedevices** を呼び出して、RASD のログ・デバイス・レポジトリを更新します。

詳細については、『Replication Agent 管理ガイド』を参照してください。

---

**注意：** 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。

---

- ログ・デバイス・レポジトリが更新される前にプライマリ・データ・サーバが新しい(または変更された)ログ・デバイスに書き込んだ場合、Replication Agent インスタンスは複写処理を停止し、Replication Down 状態になります。



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してください。

- **ra\_updatedevices** が呼び出されると、Replication Agent はログ・デバイス・レポジトリ全体を再作成するので、**ra\_devicepath** を使用して以前に変更したログ・デバイス・パスは、プライマリ・データベースからの最新のログ・デバイス情報で上書きされます。

例：

```
ID=1 serverpath=/dev1 mirror=/dev1a
```

server path が "dev44" に変更されると、上のパスは次のようになります。

```
ID=1 serverpath=/dev44 mirror=/dev1a
```

---

**注意：** ログ・デバイスの "デフォルト" パス (つまり、プライマリ・データベースから返されたログ・デバイス・パス) を変更する必要がある場合、**ra\_devicepath** コマンドを使用する前に、**ra\_updatedevices** を呼び出してください。

---

- RASD に記録された各ログ・デバイスの場合、ディスク・デバイス・パスを **ra\_devicepath** コマンドで設定または変更できます。  
**ra\_devicepath** を使用してディスク・デバイス・パスを指定しない場合、そのディスク・デバイス・パスについて記録される値は **DEFAULT** であり、Replication Agent はサーバ・デバイス・パスについて記録された値を使用して、ログ・デバイスを見つけます。
- **ra\_updatedevices** コマンドが有効なのは、Replication Agent インスタンスが Admin 状態または Replication Down 状態のときだけです。
- (Oracle の場合) トランザクション・ログの初期化中にログ・デバイスを作成する場合、または **ra\_updatedevices** コマンドを使用してデバイスを更新する場合、Replication Agent はディスク・マップ・ファイルを使用してミラー・ログ・デバイスを作成します。Replication Agent は、Replicating 状態のとき、マップ・ファイルで指定されたミラーリングされたディスクからデータを読み込みます。
- **ra\_updatedevices** コマンドは、ASM ディスク・マップ・ファイルを更新します。実行されると、ASM ディスク・マップ・ファイルは、以下のように更新されます。
  - redo ログの読み込みに必要なディスク・グループが変更されたかどうかを確認するため、ASM にクエリが実行されます。新しいディスクがいずれかの ASM ディスク・グループに追加された場合、その新しいディスクの ASM ディスク・マップ・ファイルにデフォルト・ミラー・エントリが追加されます。

- アーカイブ・ログ・パス・パラメータによって指定される ASM ディスク・グループに新しいディスクがあるかどうかと、ASM ディスク・グループに各オンライン redo ログがあるかどうかをチェックします。
- ASM ディスク・マップ・ファイルが更新されてからログ・デバイスが更新されることで、どの ASM ディスク・パス変更も更新されたログ・デバイスに確実に含まれます。
- 新しいディスク・エントリが ASM ディスク・マップ・ファイルに追加されても、レポジトリでログ・デバイスは更新されません。ファイルに新しいエントリがあり、デバイスが更新される前に変更される必要があることを示すメッセージが表示されます。**ra\_updatedevices** コマンドが次回実行されると、ログ・デバイスが更新されます。
- Replication Agent コマンド **ra\_helpdevice** は、デバイスが物理的に配置されている場所のログ・デバイス・ステータスのデバイス情報と、デバイスが別のデバイスにミラーリングされているかどうかに関する情報を提供します。この物理情報は、ファイルまたはロー・デバイスへの単純なパスです。ASM は多数のディスクを含む可能性があるディスク・グループを使用し、デバイスに必要なディスクをすべて示すために、ASM デバイスの物理情報が提供されます。デバイスが格納されているグループのディスクごとに、1 つのローが出力されます。

### 参照：

- ra\_devicepath (89 ページ)
- ra\_helpdevice (103 ページ)

## ra\_updateusers

プライマリ・データベースから RASD にユーザ情報を再ロードします。

---

**注意：** このコマンドは Oracle でのみ使用可能です。

---

### 構文

```
ra_updateusers
```

### 使用法

- **ra\_updateusers** は、プライマリ・データベースから RASD にユーザ情報を再ロードします。**ra\_updateusers** は、RASD のユーザ情報がプライマリ・データベースと同期しなくなった場合に使用します。
- **ra\_updateusers** を呼び出すと、Replication Agent は次の処理を行います。
  1. RASD からすべてのユーザ情報を削除する
  2. プライマリ・データベースにユーザ情報を問い合わせる

3. プライマリ・データベースから返されたユーザ情報で再度 RASD を指定する
- **ra\_updateusers** は、Replication Agent が Admin 状態または Replication Down 状態の場合にのみ使用できます。

## ra\_version

Replication Agent インスタンスのバージョン、ホスト・オペレーティング・システムのバージョン、および JRE バージョンを返します。

### 構文

```
ra_version
```

### 使用法

**ra\_version** は、呼び出されると、Replication Agent バージョン文字列を 1 つのローで返します。

```
Sybase Replication Agent for Unix &  
Windows/15.7.0.6100/P/generic/JDK  
7.0/main/6100/VM: Oracle Corporation  
1.7.0_02/OPT/Wed Apr 15 06:38:13 MST  
2012
```

### 参照：

- [pdb\\_version \(75 ページ\)](#)
- [ra\\_status \(138 ページ\)](#)
- [ra\\_version\\_all \(145 ページ\)](#)

## ra\_version\_all

Replication Agent インスタンスの名前、タイプ、バージョンと、プライマリ・データ・サーバ、プライマリ Replication Server、通信ドライバのバージョン情報を返します。

### 構文

```
ra_version_all
```

**注意：** (Oracle のみ) Replication Agent が ASM に接続するように設定されている (**asm\_tns\_connection** がデフォルト値でない null 以外の値で設定されている) 場合、**ra\_version\_all** には、接続されている ASM のバージョンを示す追加出力行が含まれます。**asm\_tns\_connection** が設定されていない場合、**ra\_version\_all** 出力に ASM エントリはリストされません。

## 使用法

- `ra_version_all` は、呼び出されると、以下の情報を返します。

Component	Version
Instance:	rao_rac11r2 - Oracle
RepAgent:	Sybase Replication Agent for Unix & Windows/15.7.1.6100/P/generic/JDK 7.0/main /6100/VM: Oracle Corporation 1.7.0_02/OPT/ Mon Mar 05 16:54:15 MST 2012
JRE:	Oracle Corporation Java(TM) SE Runtime Environment/1.7.0_02-b13/Windows XP 5.1/x86/32
RASD: Primary Data Server:	SQL Anywhere/11.0.0.1264/WindowsXP Oracle Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production With the Partitioning, Real Application Clusters, Automatic Storage Management, OLAP, Data Mining and Real Application Testing options
PDS JDBC Driver:	Oracle JDBC driver 11.2.0.3.0
ASM Server:	Oracle Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production With the Real Application Clusters and Automatic Storage Management options
RepServer:	Replication Server/15.6/P/NT (IX86)/Windows 2003/1/DEBUG/Thu Sep 16 14:03:14 2010
RSSD:	SQL Anywhere/11.0.0.1264/WindowsXP
Sybase JDBC Driver:	jConnect (TM) for JDBC(TM)/7.07 GA(Build 26714)/P/EBF19793/JDK 1.6.0/jdbcmain/Tue Feb 28 07:32:34 PST 2012

### 参照：

- [pdb\\_version \(75 ページ\)](#)
- [ra\\_status \(138 ページ\)](#)
- [ra\\_version \(145 ページ\)](#)

## rasd\_backup

(Oracle と Microsoft SQL Server のみ) Replication Agent システム・データベース (RASD) をバックアップします。

---

**注意：** このコマンドは Oracle および Microsoft SQL Server でのみ使用可能です。

---

### 構文

```
rasd_backup
```

## 使用法

- **rasd\_backup** は、呼び出されると、RASD のデータベース・バックアップ・プロセスを開始します。

---

**注意：** **ra\_truncatearticles** または **ra\_truncateusers** を使用したトランケートの前に、必ず RASD をバックアップしてください。

---

- 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 インスタンスを作成すると、RASD バックアップ・ディレクトリがインスタンスのディレクトリ構造の一部として自動作成されます。  
**rasd\_backup\_dir** パラメータのデフォルト値は、そのディレクトリを指します。
- **rasd\_backup** コマンドが有効なのは、Replication Agent インスタンスが Admin 状態、Replicating 状態、または Replication Down 状態のときです。

### 参照：

- rasd\_restore (149 ページ)
- ra\_truncatearticles (139 ページ)
- ra\_truncateusers (141 ページ)

## rasd\_helpbackup

(Oracle と Microsoft SQL Server のみ) RASD バックアップのリストを表示します。

---

**注意：** このコマンドは Oracle および Microsoft SQL Server でのみ使用可能です。

---

### 構文

```
rasd_helpbackup
```

### 使用法

- **rasd\_helpbackup** は、呼び出されると、**rasd\_backup\_dir** 設定パラメータで指定されるディレクトリに格納されている RASD バックアップのリストを表示します。
- これらのバックアップには、バックアップ作成日時を使用した名前が付けられます。

### 参照：

- rasd\_restore (149 ページ)
- rasd\_backup (146 ページ)
- rasd\_removebackup (148 ページ)

## **rasd\_removebackup**

(Oracle および Microsoft SQL Server のみ) RASD バックアップを削除します。

---

**注意：** このコマンドは Oracle および Microsoft SQL Server でのみ使用可能です。

---

### **構文**

```
rasd_removebackup [ backup_name | all ]
```

### **パラメータ**

- **backup\_name** – 削除するバックアップの名前。
- **all** – RASD バックアップをすべて削除できるキーワード。

### **例**

- **例 1 –**

```
rasd_removebackup 2008-07-24_15.41.10
```

```
This command causes the backup named  
2008-07-24_15.41.10 to be removed
```

```
rasd_removebackup 2008-07-24_15.41.10  
go
```

```
RASD Backups removed  
-----  
2008-07-24_15.41.10  
(1 row affected)
```

- **例 2 –**

```
rasd_removebackup all  
This command causes all of the backups to be removed.
```

```
rasd_removebackup all  
2> go
```

```
RASD Backups removed  
-----  
2008-07-28_10.08.27  
2008-07-28_10.09.29  
2008-07-28_10.11.31  
2008-07-28_10.20.55  
(4 rows affected)
```

## 使用法

- **rasd\_removebackup** は、呼び出されると、**rasd\_backup\_dir** 設定パラメータで指定されるディレクトリに格納されている RASD バックアップを削除します。
- **rasd\_removebackup** は、**all** キーワードを付けて呼び出されると、**rasd\_backup\_dir** 設定パラメータで指定されるディレクトリに格納されている RASD バックアップをすべて削除します。
- **rasd\_backup** は呼び出されると、削除されたすべてのバックアップの名前を表示します。
- これらのバックアップには、バックアップ作成日時を使用した名前が付けられます。

### 参照：

- rasd\_restore (149 ページ)
- rasd\_backup (146 ページ)
- rasd\_helpbackup (147 ページ)

## rasd\_restore

(Oracle および Microsoft SQL Server のみ) RASD をリストアできます。

---

**注意：** このコマンドは Oracle および Microsoft SQL Server でのみ使用可能です。

---

### 構文

```
rasd_restore [backup_name]
```

---

**注意：** **rasd\_restore** の実行後、**rasd\_restore** に成功すると、Replication Agent は自動的に停止します。

---

### パラメータ

- **backup\_name** – リストア元のバックアップの名前。バックアップ名を省略すると、最新のバックアップがリストアされます。

### 例

- 例 1 –

```
rasd_restore
```

パラメータなしでこのコマンドを実行すると、最新のバックアップから RASD がリストアされます。

- 例 2 –

```
rasd_restore 2008-07-24_15.41.10
```

このコマンドを実行すると、RASD が 2008-07-24\_15.41.10 バックアップからリストアされます。

### 使用法

- **rasd\_restore** は、呼び出されると、RASD のリストア・プロセスを開始します。
- パラメータを使用しない場合、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で最新の RASD バックアップを探します。
- パラメータとしてバックアップ名を指定した場合、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の指定されたバックアップからリストアします。
- Replication Agent インスタンスを作成すると、RASD バックアップ・ディレクトリがインスタンスのディレクトリ構造の一部として自動作成されます。**rasd\_backup\_dir** パラメータのデフォルト値は、そのディレクトリを指します。
- Replication Agent インスタンスが Replicating 状態のときに **rasd\_restore** を呼び出すと、エラーが返されます。
- **rasd\_restore** コマンドが有効なのは、Replication Agent インスタンスが Admin 状態または Replication Down 状態のときだけです。

### 参照：

- **rasd\_backup** (146 ページ)
- **rasd\_helpbackup** (147 ページ)
- **rasd\_removebackup** (148 ページ)

## rasd\_trunc\_schedule

(Oracle と Microsoft SQL Server のみ) トランケーション・スケジュールを管理します。**rasd\_trunc\_schedule** は、レポジトリ・トランケーション・スケジュールのリストを返します。特定のスケジュールを追加、削除することもできます。

---

**注意：** このコマンドは Oracle および Microsoft SQL Server でのみ使用可能です。

---

### 構文

```
rasd_trunc_schedule [ add, schedule | remove, schedule | clear | force ]
```

### パラメータ

- **schedule** – 制限された UNIX cron スタイルの形式の日付と時刻の文字列。自動レポジトリ・トランケーションがいつ実行されるのかを示します。

UNIX cron スタイル形式の有効なスケジュール文字列を以下に示します。

```
[mm] [HH] [DOM] [MON] [DOW]
```



- mm – 正時から経過した分数。
- HH – 24 時間表記の時間。
- DOM – 月の日付を表す **1** ~ **31** の 2 桁の数。
- MON – 1 年の 1 月 ~ 12 月の月を表す 3 文字形式の省略形 ("Jan" や "Feb" など) または 1 ~ 12 の 2 桁の数。
- DOW – 日曜日から土曜日までの曜日を表す、3 文字形式の省略形 ("Sun" や "Sat" など) または **1** ~ **7** の 2 桁の数。
- 任意の有効値に一致させるには、特定のスケジュール・フィールド ([mm], [HH],[DOM],[MON],[DOW]) にアスタリスクを使用します。
  - たとえば、"1720\*\*\*" は毎日のスケジュールの午後 8:17 を表します。
  - DOW と DOM を両方指定した場合、スケジュールは DOW または DOM の一方に一致する 2 日を表します。  
たとえば、"\*1216\*Mon" は毎月曜日の午前 12:00 か、毎月 16 日の午前 12:00 を表します。
  - 複数のエントリを指定するにはセミコロンを使用します。  
たとえば、"\*1216\*Mon" または "1720\*\*\*;\*1216\*Mon" です。
  - フィールド間にスペースを挿入しないでください。挿入すると、そのスケジュールは無効なスケジュール形式として拒否されます。たとえば、"\* 12 16 \*Mon" は無効なスケジュールです。
- 値の範囲を指定するには、ダッシュ "-" 演算子を使用します。たとえば、"1-6" は連続した "1,2,3,4,5,6" を表します。
- 特定の数の値をスキップするには、スラッシュ "/" 演算子を使用します。たとえば、時間のフィールドの "\* / 3" は連続した "0,3,6,9,12,15,18,21" を表します。

構文の説明は次のとおりです。

- **clear** – レポジトリ・トランケーション・スケジュールをすべて削除するには、以下のように入力します。

```
rasd_trunc_schedule clear
```

レポジトリ・トランケーション・スケジュール・リストが設定されていないか空の場合、スケジュールによるレポジトリ・トランケーションは無効になります。

- **force** – 自動トランケーション時刻スケジュールを無視して、レポジトリ・トランケーションを手動で直ちに実行するには、以下のように入力します。

```
rasd_trunc_schedule force
```

## 例

- 例 1 –

```
rasd_trunc_schedule
```

## コマンド・リファレンス

このコマンドは、レポジトリ・トランケーションが発生するすべてのレポジトリ・トランケーション・スケジュール時刻のリストを返します。

- **例 2-**

```
rasd_trunc_schedule add, 1720***
```

このコマンドは、スケジュール・リストに毎日午後 8:17 のレポジトリ・トランケーション・スケジュールを追加します。

- **例 3-**

```
rasd_trunc_schedule remove, 1720***
```

このコマンドは、スケジュール・リストから毎日午後 8:17 のレポジトリ・トランケーション・スケジュールを削除します。

- **例 4-**

```
rasd_trunc_schedule clear
```

このコマンドは、設定されているレポジトリ・トランケーション・スケジュールをすべて消去します。

- **例 5-**

```
rasd_trunc_schedule force
```

このコマンドは、トランケーション・スケジュールの有無にかかわらず、レポジトリを直ちにトランケートします。

### 使用法

- **rasd\_trunc\_schedule** は、呼び出されたときに、指定されたキーワードとオプションに従って機能が決まります。
- 複数のキーワードとオプションを指定する場合は、それぞれをカンマで区切ります。カンマの前後の空白は省略可能です。例：

```
rasd_trunc_schedule add, 1720***
```

### 参照：

- ra\_truncatearticles (139 ページ)
- ra\_truncateusers (141 ページ)

## resume

Replication Agent インスタンスで複製処理を開始します。

### 構文

```
resume [ resync[, init] | purge ]
```

## パラメータ

- **resynch** – (Oracle のみ) Replication Agent は、**resync database** マーカを Replication Server に送信します。
- **init** – (Oracle のみ) このキーワードは、**resync** キーワードと組み合わせる場合にのみ使用できます。Replication Agent は、**resync database** マーカと初期化コマンドの両方を Replication Server に送信します。
- **purge** – (Oracle と Microsoft SQL Server のみ) Replication Agent は、複製のレジューム時に **purge** コマンドを Replication Server に送信します。

## 使用法

- **resume** が呼び出されると、以下のように Replication Agent インスタンスは Replicating 状態になり、複製オペレーションを開始しようとします。
  - Replication Agent は、プライマリ・データベース、プライマリ Replication Server、および RSSD へのネットワーク・コネクションを開こうとします。コネクションの確立に失敗した場合、Replication Agent は警告メッセージをシステム・ログに記録し、このコネクションの設定パラメータに基づいてコネクションを再度確立しようとします。
  - 設定された再試行回数内にプライマリ・データベースへのコネクションを確立できなかった場合、Replication Agent はそれ以降の **resume** 処理をすべて中止し、Replication Down 状態に戻り、エラーをログに記録します。
  - Replication Agent は、プライマリ Replication Server に最新の LTM ロケータ値を要求し、その値を Replication Agent トランザクション・ログに格納します。
  - Log Reader コンポーネントは、トランザクション・ログのスキャンを開始し、複製するオペレーションがないか確認します。Log Reader は、LTM ロケータ値で指定されるポイントからログのスキャンを開始します。
  - Log Reader は、複製するトランザクションを見つけると、変更セット・データとして Log Transfer Interface コンポーネントの入力キューに渡します。
  - Log Transfer Interface コンポーネントは、入力キューから変更セット・データを読み込み、LTL コマンドを生成し、Replication Server に転送するためそれらの LTL コマンドを出力キューに入れます。
- **resume resync** が呼び出されると、Replication Agent インスタンスは Replicating (Resynchronization) 状態で起動しようとします。
  - Replication Agent は、**resync database** マーカを Replication Server に送信します。Replication Server はこの **resync database** マーカを処理し、Replication Agent からの **dump database** マーカを待機します。
  - **pdb\_xlog init, force** が呼び出されてから **resume resync** が実行されると、Replication Agent はインバウンド・キュー内のすべてのオープン・トランザ

クシオンをパージして重複の検出をリセットしてから、新しいインバウンド・トランザクションを受け取るよう、Replication Server に指示を送信します。Replication Server は、Replication Agent からの **dump database** マーカを待機します。

- **resume resync, init** が呼び出されると、Replication Agent は **resync database** マーカと初期化コマンドを送信して、インバウンド・キュー内のすべてのオープン・トランザクションをパージして重複の検出をリセットしてから、アウトバウンド DSI をサスペンドするよう、Replication Server に指示します。このオプションは、レプリケート・データベースと同じダンプからプライマリ・データベースを再ロードするときに使用します。

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。

- **resume purge** が呼び出されると、Replication Agent インスタンスは、複製のレジューム時に **purge** コマンドを Replication Server に送信します。これにより、Replication Server はこの Replication Agent インスタンスの接続先コネクションのインバウンド・キューからデータをパージします。**purge** キーワードは、以前のバージョンの Replication Agent にダウングレードした後、または Sybase テクニカル・サポートから推奨されたときだけ使用してください。
- 起動オペレーションに失敗した場合、Replication Agent インスタンスは Replication Down 状態に戻り、エラーをログに記録します。
- **resume** コマンドが正常に実行された場合、Replication Agent インスタンスは Replicating 状態になります。Replication Agent の現在の状態を判断するには、**ra\_status** コマンドを使用します。
- 以下のいずれかの状況では、**resume** コマンドがエラーを返します。
  - Replication Agent インスタンスが既に Replicating 状態である。
  - RASD のシステム・データ・レポジトリが存在しないか、初期化されていない (Oracle と Microsoft SQL Server)。
  - Replication Agent コネクション設定パラメータが正しく設定されていないか、正しく設定されていてもプライマリ・データベースまたはプライマリ Replication Server に接続できない。
  - プライマリ・データベースのデータベース・コネクションが、プライマリ Replication Server で正しく定義されていない。
- **resume** コマンドが正常に実行された場合、Replication Agent インスタンスは Replicating 状態になります。
- **resume** コマンドが有効なのは、Replication Agent インスタンスが Admin 状態または Replication Down 状態のときだけです。

### 参照：

- quiesce (80 ページ)
- ra\_status (138 ページ)

- shutdown (161 ページ)
- suspend (162 ページ)

## rs\_create\_repdef

(Oracle および Microsoft SQL Server のみ) マーク付けされた特定のテーブルとプロシージャ、またはマーク付けされたすべてのテーブルとプロシージャの複写定義を Replication Server で作成します。

---

**注意：** このコマンドは Oracle および Microsoft SQL Server でのみ使用可能です。

---

Replication Agent は、Replication Server 15.0 以降で使用できる複写定義データ型と一致するようあらかじめ設定されています。以前のバージョンの Replication Server について複写定義を生成する場合は、この設定を変更する必要があります。この調整を行う場合、Sybase 製品の保守契約を結んでいるサポート・センタに問い合わせてください。

### 構文

```
rs_create_repdef {all | name}
```

### パラメータ

- **all** - 複写するようマーク付けされたすべてのテーブルとストアド・プロシージャに対して、複写定義が作成されます。
- **name** - *name* で指定されたテーブルまたはプロシージャに対して、複写定義が作成されます。

---

**注意：** **rs\_create\_repdef** は、常にプライマリ・データベースのデータベース複写定義があると想定しています。

---

### 使用法

- **rs\_username** ユーザは、create object パーミッションを持っている必要があります。これがないと、Replication Agent はそれを使用して Replication Server から複写定義を作成できません。このパーミッションは、RSSD から手動で付与する必要があります。
- テーブルが複写するようマーク付けされ、所有者モードが **on** に設定されている場合、**rs\_create\_repdef** で作成された複写定義には、テーブル複写定義の "with primary table named" 句内にテーブル名の一部として所有者名が含まれます。
- このコマンドは、常にプライマリ・データベースのデータベース複写定義があると想定しています。**rs\_create\_repdef** で作成されたすべての複写定義には、**send standby** 句が含まれます。つまり、データベース・レベルの複写定義が既にある場合、その複写定義が Replication Server によってのみ使用されます。

**rs\_create\_repdef** で作成された複写定義を個別にサブスクライブすることはできません。データベース・レベルの複写定義が必要ない場合は、別のツールを使用するか、複写定義を手動で作成する必要があります。**rs\_create\_repdef** を使用しないでください。

- **rs\_create\_repdef** で作成された複写定義は、常に Replication Server にインストールされた使用可能なユーザ定義データ型を使用してデータ型を定義します。つまり、**rs\_create\_repdef** を使用する場合は、Replication Agent 設定パラメータ **pdb\_convert\_datetime** を **true** に設定しないでください。これを行うと、date および timestamp データ型が UDD 形式でなく Sybase 形式に変換されます。
- Replication Agent の設定パラメータ **pdb\_auto\_create\_repdefs** を使用すると、**rs\_create\_repdef** を実行した場合と同じ結果になります。
- **rs\_create\_repdef** を呼び出して “all” または “ALL” パラメータを指定すると、複写するようマーク付けされたすべてのテーブルまたはプロシージャに対して複写定義が作成されます。
- **rs\_create\_repdef** を呼び出して、複写するようマーク付けされたテーブルまたはプロシージャの名前を指定すると、そのテーブルまたはプロシージャの複写定義のみが作成されます。
- 複写定義の作成を試行した各テーブルまたはプロシージャに対して、結果セットが返されます。結果セットには、複写定義名および作成ステータスが含まれます。複写定義が作成された場合は、ステータスが “created” になります。エラーが発生すると、Replication Server からエラー・メッセージが返されます。
- 複写定義のオブジェクト名は、**ltl\_character\_case** 設定に従って大文字か小文字に設定されます。
- 複写定義のテーブルおよびプロシージャには、以下が当てはまります。
  - 英数字以外のすべての文字およびスペースは削除され、テーブル名またはプロシージャ名に使用されることはありません。
  - アンダースコアは英数字以外でも、名前に使用されます。
  - ピリオドはアンダースコアに置き換えられます。
- テーブルの複写定義名は、必ずプレフィクス “*ra*\$” から始まり、その後一意の英数字の識別子 (最大 8 文字) が続き、最後がテーブル名またはオブジェクト名になります。たとえば、複写名が “My Table” の場合、複写定義名は “*ra* \$0x7952\_mytable” になります。“mytable89012345678901234567890” のように特に長い複写名 (30 文字) の場合、複写定義名は “*ra* \$0x7952\_mytable8901234567890” になります (**pdb\_support\_large\_identifier** 設定パラメータが設定されているかどうかに応じて、最大 30 文字か 255 文字)。
- *date* カラムには、**rs\_create\_repdef** コマンドによって、Replication Agent の **pdb\_convert\_datetime** 設定パラメータが **false** に設定されていることを想定して、定義されたカラム・データ型の複写定義が作成されます。**pdb\_convert\_datetime** が **true** に設定されている場合、*date* 値の形式が Replication Server で予想される形式と一致しません。この問題を回避するには、

**pdb\_convert\_datetime** 設定パラメータを **false** に変更するか、(**rs\_create\_repdef** コマンドを使用せずに) 複写定義を手動で作成します。

#### 参照：

- **rs\_drop\_repdef** (157 ページ)

## rs\_drop\_repdef

(Oracle および Microsoft SQL Server のみ) 設定された Replication Server でテーブルとプロシージャの複写定義が削除されます。

---

**注意：** このコマンドは Oracle および Microsoft SQL Server でのみ使用可能です。

---

### 構文

```
rs_drop_repdef name
```

### パラメータ

- **name** – そのテーブルまたはプロシージャの複写定義が削除されます。

### 使用法

- **rs\_drop\_repdef** を呼び出すと、そのテーブルの複写定義が Replication Server で削除されます。
- **rs\_drop\_repdef** を呼び出して、複写するようマーク付けされたテーブルまたはプロシージャの名前を指定すると、そのテーブルまたはプロシージャの複写定義のみが作成されます。
- 複写定義が削除された各テーブルまたはプロシージャに対して、結果セットが返されます。結果セットには、テーブル名および作成ステータスが含まれます。複写定義が作成された場合は、ステータスが “dropped” になります。エラーが発生すると、Replication Server からエラー・メッセージが返されます。
- 複写定義のオブジェクト名は、**ltl\_character\_case** 設定に従って大文字か小文字に設定されます。
- 複写定義のテーブルおよびプロシージャには、以下が当てはまります。
  - 英数字以外のすべての文字およびスペースは削除され、テーブル名またはプロシージャ名に使用されることはありません。
  - アンダースコアは英数字以外でも、名前に使用されます。
  - ピリオドはアンダースコアに置き換えられます。
- テーブルの複写定義名は、必ずプレフィクス “*ra\$*” から始まり、その後一意の英数字の識別子 (最大 8 文字) が続き、最後がテーブル名またはオブジェクト名になります。たとえば、複写名が “My Table” の場合、複写定義名は “*ra* \$0x7952\_mytable” になります。“mytable89012345678901234567890” のように特に長い複写名 (30 文字) の場合、複写定義名は “*ra*

\$0x7952\_mytable8901234567890” になります (**pdb\_support\_large\_identifier** 設定パラメータが設定されているかどうかに応じて、最大 30 文字か 255 文字)。

**参照：**

- rs\_create\_repdef (155 ページ)

## **rs\_ticket**

プライマリ・データベースのトランザクション・ログに **rs\_ticket** マーカを設定して、Replication Server の **rs\_ticket** の処理をサポートします。このコマンドは、Replication Server の **rs\_ticket** 機能をサポートするために作成されました。

### **構文**

```
rs_ticket H1 [, H2[, H3 [, H4]]]
```

### **パラメータ**

- **H1**、**H2**、**H3** – 各パラメータには、1 ~ 10 文字が含まれます。これは自由形式で、識別子として使用します。
- **H4** – 1 ~ 50 文字が含まれます。これも自由形式で、識別子として使用します。

### **例**

- **例 1** –

以下は **rs\_ticket** を実行し、4つのパラメータ (1つのパラメータのみ必須) によって識別されるレコードの処理時間をモニタします。

```
rs_ticket test1, 1221, appxyz.monitoring_system
```

**注意：** パラメータはオプションで、**rs\_ticket** の実行の識別または区別で使用できません。

これは個別に使用するか、他の実行とグループ化して処理時間を比較することができます。

この例では、次の情報が Replication Server に送信されます。

```
rs_ticket 'V=1;H1=test1;H2=1221;H3=appxyz;  
H4=monitoring_system;PDB (name)=hh:mm:ss.ddd'
```

“name” はプライマリ・データベースの名前です。

**rs\_ticket** がレプリケート・データベースに到達すると、Replication Server によって、Replication Server の EXEC、DIST、および DSI コンポーネントに追加時間値が追加されます。レプリケート・データベースから見た最終結果は次のようになります。



```
rs_ticket 'V=1;H1=test1;H2=1221;H3=appxyz;
H4=monitoring_system;PDB(name)=hh:mm:ss.ddd;
EXEC=hh:mm:ss.ddd;DIST=hh:mm:ss.ddd;
DSI(name)=hh:mm:ss.ddd;RDB(name)=hh:mm:ss.ddd'
```

提供された結果を使用して、複製の遅延およびパフォーマンスをモニタできます。ユーザは H1 ~ H4 の異なるパラメータまたは記述的なパラメータを使用することで、プライマリ・データベースに入力したときにどの **rs\_ticket** データがコマンドのアクティビティまたはタイミングと一致するかを簡単に調べることができます。

#### • 例 2 -

バッチ作業のパフォーマンスを測定するには、次のシーケンスのように、作業の前後に **rs\_ticket** を実行することができます。

(Replication Agent で実行)

```
rs_ticket start
```

(プライマリ・データ・サーバで実行)

```
execute replication benchmarks
```

(Replication Agent で実行)

```
rs_ticket stop
```

### 使用法

- Replication Server の EXEC、DIST、および DSI モジュールは、**rs\_ticket** サブコマンドを解析および処理します。
- **rs\_ticket** についてのサブスクリプションはありません。レプリケート・サイトから少なくとも 1 つのサブスクリプションがないかぎり、DIST は DSI に対して **rs\_ticket** を送信しません。
- **rs\_ticket** では、**pds\_username** で指定したユーザ名を Replication Server への接続に指定したユーザ ID (メンテナンス・ユーザ) と別にする必要があります。メンテナンス・ユーザの名前は、**ra\_maintid** を実行することで取得できます。

Replication Server における **rs\_ticket** の使用方法の詳細については、Replication Server のマニュアルを参照してください。

### server\_xlog

(Microsoft SQL Server のみ) Microsoft SQL Server システム・リソース・データベースに Replication Agent システム・オブジェクトを作成または削除します。

このコマンドは、Microsoft SQL Server の Replication Agent を初めて初期化する際に使用します。**pdb\_xlog** init より前に実行してください。このコマンドは、Microsoft

SQL Server の Replication Agent を初めて初期化する際に使用します。**pdb\_xlog init** より前に実行してください。

---

**注意：** このコマンドは Microsoft SQL Server でのみ使用可能です。

---

### 構文

```
server_xlog {init | remove } [, force]
```

### パラメータ

- **init** – Microsoft SQL Server システム・リソース・データベースに Replication Agent システム・オブジェクトを作成するためのキーワード。
- **remove** – Microsoft SQL Server システム・リソース・データベースから Replication Agent システム・オブジェクトを削除するためのキーワード。
- **force – init** オペレーションまたは **remove** オペレーションを参照するキーワード。

### 例

#### 例 1 –

```
server_xlog init
```

このコマンドは Microsoft SQL Server システム・リソース・データベースに Replication Agent システム・オブジェクトを作成します。

#### 例 2 –

```
server_xlog remove, force
```

このコマンドは Microsoft SQL Server システム・リソース・データベースから Replication Agent システム・オブジェクトを削除します。

### 使用法

- トランザクション・ログを初期化する前に Microsoft SQL Server の **pdb\_xlog init** を実行しようとする、**pdb\_xlog init** が停止し、戻ってから最初に **server\_xlog init** を実行するようプロンプトが表示されます。

### 参照：

- [pdb\\_xlog \(75 ページ\)](#)

## **shutdown**

Replication Agent インスタンスを停止し、そのプロセスを終了します。

### **構文**

```
shutdown [immediate]
```

### **パラメータ**

- **immediate** – Replication Agent インスタンスを即時停止するためのオプションのキーワード。

### **使用法**

- オプションを指定せずに **shutdown** を呼び出すと、Replication Agent により通常 (適切な) 停止が開始されます。  
通常の停止では、Replication Agent が最初に静止し、その後プロセスが終了します。Replication Agent のクワイースの詳細については、「quiesce」を参照してください。
- **shutdown** を **immediate** キーワードを指定して呼び出すと、Replication Agent により即時停止が開始されます。  
即時停止では、Replication Agent が
  - 実行中または転送中のトランザクションを考慮することなく、すべての複製プロセスを停止します。
  - すべてのコネクションを削除します。
  - アプリケーション・プロセスを終了します。
- **immediate** キーワードを指定した **shutdown** コマンドは、状態の遷移中を含め、Replication Agent インスタンスがいかなる状態でも、常に有効です。
- キーワードを指定しない **shutdown** コマンド (通常の停止) は、状態の遷移中を除き、Replication Agent インスタンスが Admin、Replicating、または Replication Down 状態の場合に有効です。

### **参照：**

- quiesce (80 ページ)
- ra\_status (138 ページ)
- resume (152 ページ)
- suspend (162 ページ)

## **suspend**

現在のすべての複製プロセスを終了し、Replication Agent インスタンスを Admin 状態にします。

### **構文**

```
suspend
```

### **使用法**

- **suspend** を呼び出すと、Replication Agent インスタンス内にある現在のすべての複製プロセスを終了します。
  - 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 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) 内のあらゆるデータは、それ以上処理されずに削除されます。
  - 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接続を即時解放し、プライマリ Replication Server (および、接続されている場合は RSSD) への接続を削除します。
  - Replication Agent インスタンスは Replicating 状態から Admin 状態になります。

---

**注意：** **quiesce** コマンドのアクションは **suspend** コマンドと同様ですが、**quiesce** では、Replication Agent インスタンスを Admin 状態にする前に、最初に Replication Agent の内部キューにある保留中のトランザクションが処理されることが異なります。

---

- Replication Agent インスタンスが Admin 状態の場合に **suspend** コマンドを実行すると、エラーが返されます。
- **suspend** コマンドは Replication Agent インスタンスが Replicating 状態の場合にのみ有効です。

### **参照：**

- **quiesce** (80 ページ)
- **ra\_status** (138 ページ)
- **resume** (152 ページ)
- **shutdown** (161 ページ)

## test\_connection

Replication Agent のコネクション設定およびネットワーク接続をテストします。

### 構文

```
test_connection [conn_name]
```

**注意：**(Oracle のみ) Replication Agent が ASM に接続するように設定されている (**asm\_tns\_connection** がデフォルト値を除く null 以外の値で設定されている) 場合、**test\_connection** には、接続されている ASM のバージョンを示す追加出力行が含まれます。**asm\_tns\_connection** が設定されていない場合、**test\_connection** 出力に ASM エントリがリストされません。

### パラメータ

- **conn\_name** – テストする Replication Agent コネクションのキーワード。有効なキーワードは次のとおりです。
  - **PDS** – プライマリ・データ・サーバ
  - **RS** – プライマリ Replication Server (および接続されている場合は RSSD)

**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 への接続テスト時に、RSSD への Replication Agent の接続がテストされます。**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 の接続はテストされません。

### 例

- **例 1 –**

```
test_connection
```

このコマンドにより、プライマリ・データ・サーバ・コネクション、プライマリ Replication Server コネクション、RSSD コネクション (設定されている場合) を含むすべての Replication Agent コネクションがテストされます。

- **例 2 –**

```
test_connection PDS
```

このコマンドにより、プライマリ・データ・サーバの Replication Agent コネクションのみがテストされます。

## 使用法

- オプションを指定せずに **test\_connection** を呼び出すと、Replication Agent で設定ファイルに格納されたコネクション・パラメータを使用し、各コネクションに該当するサーバにログインを試行して、すべてのコネクションをテストします。
- **test\_connection** を **RS** キーワードまたは **PDS** キーワードを指定して呼び出すと、Replication Agent で指定されたコネクションがテストされます。
- **test\_connection** コマンドは、ネットワーク接続と、プライマリ・データベースの次の Replication Agent コネクション設定パラメータを検証します。
  - コネクションの種類 (コネクティビティ・ドライバおよびプロトコル) – **pds\_connection\_type**
  - データベース名 – **pds\_database\_name**
  - データ・サーバ名 – **pds\_server\_name**
  - データ・ソース名 (ODBC ドライバのみ) – **pds\_datasource\_name**
  - ホスト・マシン名 – **pds\_host\_name**
  - ポート番号 – **pds\_port\_number**
  - ユーザ・ログイン・アクセス – **pds\_password** および **pds\_username**

---

**注意：** **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・ログイン・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・ログイン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** コマンドは、ネットワーク接続と、プライマリ Replication Server (および接続されている場合は RSSD) の次の Replication Agent コネクション設定パラメータを検証します。
  - データベース名 – **rssd\_database\_name** (RSSD のみ)
  - Replication Server データ・ソース (Replication Server プライマリ・データベース・コネクションで指定されたもの) – **rs\_source\_db** および **rs\_source\_ds** (Replication Server のみ)
  - ホスト・マシン名 – **rs\_host\_name** (および **rssd\_host\_name**)
  - ネットワーク・パケット・サイズ – **rs\_packet\_size** (Replication Server のみ)
  - ポート番号 – **rs\_port\_number** (および **rssd\_port\_number**)
  - ユーザ・ログイン・アクセス – **rs\_password**、**rs\_username** (および **rssd\_password** および **rssd\_username**)

---

**注意：** **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。

- **test\_connection** コマンドは、次のように接続の種類およびそのステータスを返します。

```
Type Connection
-----
PDS succeeded
RS succeeded
```

(2 rows affected)

接続・ステータスが **failed** の場合、次のいずれかを示します。

- Replication Agent 接続設定パラメータが正しく設定されていません。
- ネットワーク障害または通信エラーのために、接続を確立できません。
- 接続に関連付けられたサーバが停止しています。
- 接続・ステータスが **failed** の場合、Replication Agent のシステム・ログをチェックして、障害の原因を確認してください。

**注意：** さらに、接続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。

- Replication Agent 接続設定パラメータの設定の詳細については、『Replication Agent 管理ガイド』を参照してください。
- 特定の接続設定パラメータの詳細については、「設定パラメータ」を参照してください。
- **ra\_truncatearticles** コマンドは、Replication Agent インスタンスが Admin、Replicating、または Replication Down 状態の場合に有効です。

#### 参照：

- 設定パラメータ (169 ページ)
- ra\_config (86 ページ)
- ra\_statistics (126 ページ)
- ra\_status (138 ページ)

## trace

Replication Agent インスタンスに現在のトレース・フラグの設定を返すか、トレース・フラグの設定を変更します。

### 構文

```
trace [ {flag | all }, switch ]
```

## パラメータ

- **flag** – トレース・フラグ名を設定対象に変更します。
- **all** – スイッチ値をすべてのトレース・フラグに同時に適用できるキーワードです。
- **switch** – ブール値 (**true** または **false**) で、*flag* オプション内で特定されたトレース・ポイントに対し、トレースを有効または無効にします。

## 使用法

- **trace** コマンドは Sybase サポート・センタのエンジニアが Replication Agent のトラブルシューティングを行うときに使用するためのものです。
- オプションを指定しないで **trace** を呼び出すと、すべての Replication Agent トレース・フラグに対する現在の設定が返されます。
- **trace** を *flag* オプションおよび *switch* オプションを指定して呼び出すと、特定されたトレース・フラグの設定を変更して、トレース・フラグの現在の (新しい) 設定を返します。
- **trace** を **all** キーワードおよび *switch* オプションを指定して呼び出すと、すべての Replication Agent トレース・フラグの設定を *switch* オプションで指定された値に変更して、すべてのトレース・フラグの現在の (新しい) 設定を返します。
- **trace** コマンドで行った変更は、ただちに有効になります。
- トレース・フラグが **true** に設定されると、フラグによって特定されたトレース・ポイントに対するトレースが有効になります。**false** に設定されると、そのトレース・ポイントに対するトレースが無効になります。
- すべてのトレース・ポイントからの出力は (LTITRACELTL を除き)、Replication Agent システムのログ・ファイルに送信されます。**log\_system\_name** コマンドを使用すると、Replication Agent システム・ログ・ファイルの名前とパスを確認できます。
- LTITRACELTL トレース・ポイントからの出力は、LTITRACELTL.log という名前の別のトレース出力ファイルに送信されます。LTITRACELTL.log ファイルの内容を表示するには、ファイル・ビューが非常に長い行を扱えなければなりません。

---

**注意：** LTITRACELTL.log ファイルには、プライマリ Replication Server に実際に送信される LTL コマンドではなく、人間が判読できる形式の LTL が含まれます。

---

- 表 10 : Replication Agent トレース・フラグ (167 ページ) は、Replication Agent トレース・フラグをリストします。



表 10 : Replication Agent トレース・フラグ

トレース・フラグ	説明
BMGRTRACE	“true” に設定すると、このフラグは Bean 管理イベント・トレースを有効にします。
CACHETRC	“ture” に設定すると、このフラグは内部キャッシュ・イベントのトレースを有効にします。
DBCCONTEXT	“ture” に設定すると、このフラグはデータベース・コンテキスト・イベントのトレースを起動します。
LATRC	“true” に設定すると、このフラグは一般ログ管理者の操作をトレースします。
LATRCSQL	“true” に設定すると、このフラグはログ管理者とプライマリ・データベース間の SQL 対話をトレースします。
LICTRACE	“ture” に設定すると、このフラグは機能ライセンスのチェックイン/チェックアウト・イベントをトレースします。
LOGREADTRC	“ture” に設定すると、データベース・ログ読み込みのトレースをオンにします。
LRTRACE	“true” に設定すると、このフラグは Log Reader コンポーネントの通常の実行をトレースします。
LTITRACE	“true” に設定すると、このトレース・フラグは、Log Transfer Interface コンポーネントの操作のトレースを有効にします。
LTITRACELTL	“true” に設定すると、このトレース・フラグは、LTITRACELTL.log ファイル内の LTL 文のトレースを有効にします。
LTMCI	“true” に設定すると、LTM コンポーネント・インタフェース呼び出しと他のコンポーネント・インターフェースの LTM 呼び出しをトレースします。
LTLFMTRC	“ture” に設定すると、このトレース・フラグは、LTL フォーマッタのトレースを有効にします。
LTMHL	“true” に設定すると、LTM 実行パス内のハイライトが認識されます。
LTMSC	“true” に設定すると、Replication Agent のすべてのステータス変更がトレースされます。
RACONTRC	“true” に設定すると、コネクションおよびクエリの実行がトレースされます。

トレース・フラグ	説明
RACONTRCSQL	“true” に設定すると、SQL 文のトレースが実行されます。
RASDTRC	(Oracle および Microsoft SQL Server では) “true” に設定すると、Replication Agent システム・データ・レポジトリ・イベントのトレースを起動します。
RATRACE	“true” に設定すると、Replication Agent イベントがトレースされます。
RSTICKETTRC	“true” に設定すると、Replication Agent が <b>rs_ticket</b> 値を含むトレース・メッセージを LTL フォーマット中に Replication Agent システム・ログに記録します。
STMTRACE	“true” に設定すると、LTM ステータス・モニタ・イベントがトレースされます。
THREADTRC	“ture” に設定すると、ThreadPool トレース・イベントをログに記録します。

- SYSTEM トレース・フラグの設定は変更できません。

表 11 : Replication Agent SYSTEM トレース・フラグ (168 ページ)は、Replication Agent SYSTEM トレース・フラグをリストします。

**表 11 : Replication Agent SYSTEM トレース・フラグ**

トレース・フラグ	説明
CONFIG	設定変更イベントがログされました。
ERROR	重大なエラー；リカバリに手動の介入が必要になる場合があります。
FATAL	重大なエラー；アプリケーション停止；リカバリに手動の介入が必要です。
INFORMATION	情報のみ；何もする必要はありません。
WARNING	マイナー エラー；機能には影響しない、または問題はリカバリ可能です。

- **trace** コマンドが有効なのは、Replication Agent のインスタンスが Admin ステータス、Replicating ステータス、または Replication Down ステータスのときです。

**参照：**

- log\_system\_name (11 ページ)

# 設定パラメータ

設定パラメータは、Replication Agent インスタンスの動作方法を制御するユーザ設定可能な設定を記録します。すべての設定パラメータの現在の値は、各 Replication Agent インスタンスの設定ファイルに格納されます。

## Replication Agent 設定ファイル

---

Replication Agent インスタンスを作成すると、設定ファイルが自動的に作成されません。Replication Agent instance が起動するたびに、設定ファイルを読み取って実行に必要な設定情報を取得します。

起動後に Replication Agent が設定ファイルにアクセスするのは、設定パラメータの値を変更するために `ra_config` コマンドまたは `ra_set_login` コマンドを呼び出したときのみです。設定ファイルは、Replication Agent ベース・ディレクトリ内の instance サブディレクトリにあります。設定ファイルには、Replication Agent インスタンスの名前と `.cfg` 拡張子が付きます (たとえば、インスタンス名が “my\_ra” の場合、設定ファイルは `my_ra.cfg` になります)。

設定パラメータの値を変更すると、Replication Agent により新しい値が保存され、設定ファイル全体が上書きされます。

## 設定ファイル・フォーマット

設定ファイルは、1つのインスタンスの設定情報が格納された ASCII フラット・ファイルです。

設定ファイルの最初の2行は、Replication Agent 設定ファイルとしてファイルを識別し、ファイルの最終変更時刻を記録します。例：

```
#RAO Property File
#Fri Jan 12 07:33:18 MST 2008
```

各設定パラメータ名は個々の行に示され、その後に等号記号 (=) と現在のパラメータ値が続きます。例：

```
compress_ltl_syntax=true
```

Replication Agent インスタンスが実行されていない場合は、設定ファイルを表示して、現在の Replication Agent 設定を確認できます。

---

**注意：** Replication Agent では、パラメータ値を変更するために **ra\_config** コマンドまたは **ra\_set\_login** コマンドを呼び出すたびに、設定ファイル全体が上書きされるため、設定ファイルを編集しないでください。

---

Replication Agent インスタンスの実行中は、**ra\_config** コマンドを使用して現在の Replication Agent 設定を表示します。

## 設定パラメータの変更

---

Replication Agent 設定パラメータの現在の値を表示、設定、または変更するには、**ra\_config** コマンドを使用します。

Replication Agent の現在の管理者ログイン (**ltm\_admin\_user**) または管理者パスワード (**ltm\_admin\_pw**) を変更するには、**ra\_set\_login** コマンドを使用する必要があります。

---

**注意：** **ltm\_admin\_user** パラメータおよび **ltm\_admin\_pw** パラメータは、直接使用できず、**ra\_config** によって返されるパラメータ・リストに表示されません。

---

**ra\_config** コマンドおよび **ra\_set\_login** コマンドの使用方法の詳細については、「コマンド・リファレンス」を参照してください。

### 参照：

- コマンド・リファレンス (5 ページ)
- ra\_config (86 ページ)
- ra\_set\_login (125 ページ)

## Replication Agent 設定のコピー

---

新しい Replication Agent インスタンスを **ra\_admin** ユーティリティで作成するときには、新しいインスタンスで既存の Replication Agent インスタンスと同じ設定パラメータ値を使用することを指定できます。

---

**注意：** 新しい Replication Agent インスタンスを作成するときに既存の設定インスタンスをコピーすると、一部の設定パラメータ値が新しい設定にコピーされません。詳細については、『Replication Agent 管理ガイド』を参照してください。

---

新しい Replication Agent インスタンスを作成するときに既存の設定をコピーしない場合、**ra\_admin** ユーティリティによって、すべての設定パラメータをデフォルト値に設定したデフォルト設定ファイルが作成されます。

## 設定パラメータ・リファレンス

Replication Agent 設定パラメータのテーブルには、すべての Replication Agent 設定パラメータと各パラメータの簡単な説明を示しています。ターゲット列は、パラメータが「すべて」のターゲットに使用されるか、Oracle、Microsoft SQL Server (MSSQL)、または DB2 UDB (UDB) の特定のターゲットに使用されるかを示します。

表 12 : Replication Agent 設定パラメータ

パラメータ名	ターゲット	説明
admin_port (179 ページ)	ALL	Replication Agent が管理コネクションの受信に使用するポート番号。
asm_password (180 ページ)	Oracle	<b>asm_username</b> に指定したユーザの Oracle Automatic Storage Management (ASM) アクセス用のパスワード。
asm_tns_connection (181 ページ)	Oracle	tnsnames.ora ファイルで検出される Oracle ASM コネクション名を識別。
asm_tns_filename (181 ページ)	Oracle	ASM コネクション情報が格納された Oracle tnsnames.ora ファイル名を識別。
asm_username (182 ページ)	Oracle	ASM サーバへの接続に使用する Oracle ユーザ名を識別します。
auto_adjust_block_count (183 ページ)	MSSQL	読み取りの一部を廃棄する場合に、Replication Agent が <b>log_read_block_count</b> パラメータを自動的に調整するかどうかを決定。
column_compression (184 ページ)	ALL	LTL に最小カラム情報を使用。
compress_ltl_syntax (185 ページ)	ALL	省略された LTL 構文を使用。
connect_to_rs (185 ページ)	ALL	LTI から Replication Server へのコネクションを有効または無効にする。
ddl_password (186 ページ)	Oracle MSSQL	<b>ddl_username</b> のパスワード。

設定パラメータ

パラメータ名	ターゲット	説明
ddl_username (186 ページ)	Oracle MSSQL	DDL コマンドをレプリケート・データベースに複製するために LTL に含めるデータベース・ユーザ名。
dump_batch_timeout (187 ページ)	ALL	不完全な LTL バッファを Replication Server に送信するまで待機する秒数。
filter_maint_userid (188 ページ)	ALL	メンテナンス・ユーザ ID を指定した Log Reader フィルタ・オペレーション。
function_password (189 ページ)	Oracle MSSQL	複製ストアド・プロシージャを呼び出す LTL に渡されるユーザ ID のパスワード。
function_username (189 ページ)	Oracle MSSQL	複製ストアド・プロシージャを呼び出す LTL に渡されるユーザ ID。
log_backup_files (190 ページ)	ALL	ログ・ディレクトリに保存するログ・バックアップ・ファイルの数を決定。
log_directory (190 ページ)	ALL	Replication Agent システム・ログ・ファイルが格納されているディレクトリ。
log_read_block_count (191 ページ)	MSSQL	プライマリ・データベース・トランザクション・ログから一度に読み取られるブロックまたはページの数を決定。
log_trace_verbose (192 ページ)	ALL	トレース・ログ・ファイルの冗長モードのオン/オフを切り替え。
log_wrap (192 ページ)	ALL	ラッピングするまでにログ・ファイルに書き込む 1KB ブロックの数。
lr_max_lobdata_cache (193 ページ)	Oracle	ロー外 LOB データの LOB データ・キャッシュの最大サイズ。
lr_max_op_queue_size (193 ページ)	ALL	複製中に Log Reader オペレーション・キューに認められるオペレーションの最大数。
lr_max_scan_queue_size (193 ページ)	Oracle MSSQL	複製中に Log Reader ログ・スキャン・キューに認められるログ・レコードの最大数。
lr_nxt_byte_order (194 ページ)	Oracle MSSQL	Oracle 用の NCLOB および Microsoft SQL Server 用の NTEXT の複製時に使用するバイト順序を指定。

パラメータ名	ターゲット	説明
lr_read_buffer_size (196 ページ)	UDB	ログ読み取りバッファのバイト単位のサイズ。
lr_send_trunc_partition_ddl (196 ページ)	Oracle	レプリケート・データベースにパーティションのトランケート・コマンドを DDL または DML として送信するかどうかを決定。
lti_batch_mode (197 ページ)	ALL	LTI バッチ・モードのオン/オフを切り替え。
ltl_formatter_count (198 ページ)	ALL	LTI キューの項目で同時に動作する LTL フォーマッタのスレッドの数。
lti_max_buffer_size (198 ページ)	ALL	LTI 入力バッファに格納する変更セットの最大数。
lti_update_trunc_point (199 ページ)	ALL	LTI が新しい LTM ロケータを要求するまでに送信する LTL コマンドの数。
ltl_batch_size (200 ページ)	ALL	LTL バッチ・バッファのサイズ。
ltl_big_endian_unitext (200 ページ)	ALL	LTL を Replication Server に送信する前に、Unicode LOB データをリトル・エンディアンからビッグ・エンディアンに変換するかどうかを指定。
ltl_character_case (201 ページ)	ALL	Replication Server に送信するデータベース・オブジェクト名の大文字と小文字の区別。
ltl_origin_time_required (202 ページ)	ALL	LTL で origin_time コマンド・タグを送信するかどうかを指定する。
ltl_send_only_primary_keys (202 ページ)	ALL	Replication Agent で更新および削除オペレーションの更新前イメージに対するプライマリ・キー・カラム・データのみを送信するかどうかを制御。
ltm_admin_pw (204 ページ)	ALL	Replication Agent の管理ポートのパスワード。
ltm_admin_pw_min_length (204 ページ)	ALL	Replication Agent 管理者のログイン・パスワードの最小の長さ。
ltm_admin_user (205 ページ)	ALL	Replication Agent の管理ポートのユーザ ID。

パラメータ名	ターゲット	説明
max_ops_per_scan (205 ページ)	ALL	Log Reader が 1 回のログ・スキャンで読み取るオペレーションの最大数。
pdb_archive_path (206 ページ)	Oracle UDB	Replication Agent でアーカイブされた Oracle redo ログ・ファイルまたはアーカイブされた UDB トランザクション・ログ・ファイルの検出を予想するディレクトリ・パスを識別。
pdb_archive_remove (207 ページ)	Oracle UDB	<b>pdb_archive_path</b> により指定されたパスからのアーカイブされたトランザクション・ログ・ファイルの削除を有効または無効にする。
pdb_auto_create_repdefs (208 ページ)	Oracle MSSQL	true に設定すると、テーブルおよびプロシージャが複写するようマーク付けされている場合に、Replication Server でそのテーブルまたはプロシージャの複写定義が自動的に作成される。
pdb_automark_tables (210 ページ)	Oracle MSSQL	初期化または DDL 複写中に Replication Agent がテーブルを複写するよう自動的にマーク付けするかどうかを決定。
pdb_auto_run_scripts (211 ページ)	ALL	トランザクション・ログ・オブジェクトの作成/削除およびプライマリ・データベース・オブジェクトのマーク付け/マーク解除に使用する SQL スクリプトの自動実行。
pdb_convert_datetime (212 ページ)	ALL	ネイティブの日付/時刻フォーマットを Sybase の <i>datetime</i> フォーマットに変換。
pdb_dflt_column_repl (214 ページ)	ALL	テーブルがマーク付けされている場合に、デフォルトで LOB カラムの複写を有効にする。
pdb_dflt_object_repl (214 ページ)	Oracle UDB	オブジェクトがマーク付けされている場合に、デフォルトで複写を有効にする。
pdb_ignore_unsupported_anydata (215 ページ)	Oracle	Replication Agent が <i>ANYDATA</i> 型のカラムに格納されたサポート対象外のデータ型のデータを無視するかどうかを決定。
pdb_include_archives (217 ページ)	Oracle	Oracle アーカイブ・ログ・ファイルの使用を有効または無効にする。



パラメータ名	ターゲット	説明
pdb_skip_missing_user (217 ページ)	ALL	Replication Agent が一致するセッション情報またはユーザ情報のないコマンドの処理をスキップするかどうかを決定。
pdb_support_large_identifier (218 ページ)	ALL	Replication Server 12.6 以降で、長さ 255 文字までの長い識別子の複写をサポートする。
pdb_timezone_file (219 ページ)	Oracle	Oracle タイム・ゾーン情報を取得するために Replication Agent 初期化時に読み取るファイルを指定。
pdb_xlog_device (219 ページ)	ALL	プライマリ・データベース・デバイスの名前。
pdb_xlog_prefix (220 ページ)	ALL	トランザクション・ログ・オブジェクトを識別するために使用する文字列のプレフィクス。
pdb_xlog_prefix_chars (221 ページ)	ALL	<b>pdb_xlog_prefix</b> に許可される英字以外の文字。
pds_connection_type (222 ページ)	ALL	プライマリ・データ・サーバへの接続の種類。
pds_dac_port_number (223 ページ)	MSSQL	プライマリ・データベースのサーバ・レベルの初期化時に、Replication Agent が使用するプライマリ・データベースの専用管理接続のポート番号。
pds_database_name (223 ページ)	ALL	プライマリ・データ・サーバから複写するデータベースの名前。
pds_datasource_name (224 ページ)	MSSQL UDB	プライマリ・データ・サーバから複写するデータベースのデータ・ソース名またはデータベース・エイリアス。
pds_host_name (224 ページ)	ALL	プライマリ・データ・サーバのホスト・マシンの名前。
pds_integrated_security (225 ページ)	MSSQL	プライマリ SQL Server への接続時に Replication Agent で Windows 認証を使用するかどうかを決定。
pds_password (226 ページ)	ALL	Replication Agent がプライマリ・データ・サーバへのアクセスに使用するユーザ ID のパスワード。

## 設定パラメータ

パラメータ名	ターゲット	説明
pds_port_number (226 ページ)	ALL	プライマリ・データ・サーバのポート番号。
pds_retry_count (227 ページ)	ALL	プライマリ・データ・サーバへの接続をリトライする回数。
pds_retry_timeout (227 ページ)	ALL	接続をリトライするまでに待機する秒数。
pds_server_name (227 ページ)	ALL	プライマリ・データ・サーバのサーバ名。
pds_tns_connection (228 ページ)	Oracle	プライマリ・データベースの接続情報を識別する tnsnames.ora ファイルで検出された Oracle 接続名。
pds_tns_filename (228 ページ)	Oracle	プライマリ・データベースの接続情報の識別に使用する Oracle tnsnames.ora ファイルを識別する Oracle ファイル名。
pds_username (229 ページ)	ALL	Replication Agent がプライマリ・データ・サーバへのアクセスに使用するユーザ ID。
ra_admin_device (230 ページ)	ALL	Replication Agent システム・オブジェクトを作成するプライマリ・データベース・デバイス。
ra_admin_instance_prefix (230 ページ)	ALL	1 つの Replication Agent インスタンスに固有の Replication Agent システム・オブジェクトを識別するために使用するプレフィクス文字列。
ra_admin_prefix (231 ページ)	ALL	共有 Replication Agent システム・オブジェクトを識別するために使用するプレフィクス文字列。
ra_admin_prefix_chars (232 ページ)	ALL	Replication Agent システム・オブジェクトを識別するデータベース・オブジェクト名のプレフィクス文字列に許可される英字以外の文字。
ra_admin_owner (233 ページ)	ALL	共有システム・オブジェクトおよびインスタンス固有のシステム・オブジェクトを含むすべての Replication Agent システム・オブジェクトの所有者。

パラメータ名	ターゲット	説明
ra_retry_count (234 ページ)	ALL	障害発生後に LTM が <i>Replicating</i> 状態への復帰を試みる回数。
ra_retry_timeout (234 ページ)	ALL	LTM が <i>Replicating</i> 状態への復帰をリトライするまでに待機する秒数。
rasd_backup_dir (235 ページ)	Oracle MSSQL	Replication Agent システム・データベース (RASD) のバックアップ・ファイルのディレクトリ・パス。
rasd_database (235 ページ)	Oracle MSSQL	Replication Agent システム・データベース (RASD) のデータベース・ファイルのディレクトリ・パス。
rasd_mirror_tran_log (236 ページ)	Oracle MSSQL	Replication Agent システム・データベース (RASD) のトランザクション・ログのミラーリングを有効または無効にする。
rasd_trace_log_dir (237 ページ)	Oracle MSSQL	Replication Agent システム・データベース (RASD) のトレース・ログ・ファイルのディレクトリ・パス。
rasd_tran_log (238 ページ)	Oracle MSSQL	Replication Agent システム・データベース (RASD) のトランザクション・ログ・ファイルのディレクトリ・パス。
rasd_tran_log_mirror (239 ページ)	Oracle MSSQL	Replication Agent システム・データベース (RASD) のトランザクション・ログ・ファイル・ミラーのディレクトリ・パス。
ra_standby (239 ページ)	Oracle	Replication Agent ファンクションがスタンバイ・モードかどうかを判断。
ra_statrack_interval (240 ページ)	ALL	統計をサンプリングする秒単位の間隔を決定。
rman_enabled (241 ページ)	Oracle	Replication Agent が Oracle <b>RMAN</b> ユーティリティを使用して古いアーカイブ・ログ・ファイルをトランケートするかどうかを示す。
rman_password (242 ページ)	Oracle	<b>rman_username</b> を使用して Oracle <b>RMAN</b> ユーティリティに接続するためのパスワード。

## 設定パラメータ

パラメータ名	ターゲット	説明
rman_username (242 ページ)	Oracle	<b>rman_password</b> を使用して Oracle <b>RMAN</b> ユーティリティに接続するためのログイン名。
rs_charset (243 ページ)	ALL	Replication Server との通信に使用する文字セット。
rs_host_name (244 ページ)	ALL	プライマリ Replication Server ホスト・マシンの名前。
rs_packet_size (245 ページ)	ALL	Replication Server に送信するデータのネットワーク I/O パケット・サイズ。
rs_password (246 ページ)	ALL	Replication Agent が Replication Server へのアクセスに使用するユーザ ID のパスワード。
rs_port_number (246 ページ)	ALL	プライマリ Replication Server のポート番号。
rs_replicate_owner_required (246 ページ)	ALL	複写定義を生成するときに、レプリケート・テーブル句に所有者が常に含まれているかを示す。
rs_retry_count (247 ページ)	ALL	プライマリ Replication Server への接続をリトライする回数。
rs_retry_timeout (247 ページ)	ALL	接続をリトライするまでに待機する秒数。
rs_source_db (248 ページ)	ALL	Replication Server によって識別されるプライマリ・データベースの名前。
rs_source_ds (248 ページ)	ALL	Replication Server によって識別されるプライマリ・データ・サーバの名前。
rs_ticket_version (249 ページ)	ALL	Replication Agent がプライマリ・データベースの時刻またはプライマリ・データベースの日付と時刻を <b>rs_ticket</b> マーカに書き込むかどうかを決定。
rs_username (249 ページ)	ALL	Replication Agent がプライマリ Replication Server へのアクセスに使用するユーザ ID。
rssd_charset (250 ページ)	ALL	RSSD との通信に使用する文字セット。
rssd_database_name (251 ページ)	ALL	RSSD データベースの名前。

パラメータ名	ターゲット	説明
rssd_host_name (251 ページ)	ALL	RSSD ホスト・マシンの名前。
rssd_password (251 ページ)	ALL	Replication Agent が RSSD へのアクセスに使用するユーザ ID のパスワード。
rssd_port_number (252 ページ)	ALL	RSSD のポート番号。
rssd_username (252 ページ)	ALL	Replication Agent が RSSD へのアクセスに使用するユーザ ID。
scan_sleep_increment (253 ページ)	ALL	複製するオペレーションがないことを検出した後、次のスキャンまでの Log Reader の待機時間を増加する秒数。
scan_sleep_max (254 ページ)	ALL	複製するオペレーションがないことを検出した後、次のスキャンまで Log Reader が待機する最大秒数。
skip_lr_errors (254 ページ)	Oracle UDB	Replication Agent がログ・レコード処理エラーを無視するかどうかを決定。
skip_ltl_errors (255 ページ)	ALL	LTI が Replication Server により返されるエラー・メッセージを無視。
structured_tokens (255 ページ)	ALL	LTI が LTL 出力の生成時に構造化トークンを使用。
truncation_interval (256 ページ)	ALL	自動ログ・トランケーションをリトライするまでに待機する分数。
truncation_type (257 ページ)	ALL	許可されるログ・トランケーション方法。
use_rssd (258 ページ)	ALL	複製定義のための RSSD へのアクセスのオン/オフを切り替え。

## admin\_port

Replication Agent のクライアント・ソケット・ポート番号。

デフォルト値  
10000

## 設定パラメータ

### 値

Replication Agent ホスト・マシンの有効なポート番号。

### コメント

- Replication Agent インスタンスを作成する場合、インスタンスの管理ポートのクライアント・ソケット・ポート番号を指定する必要があります。クライアント・アプリケーションはこのポート番号を使用して、Replication Agent インスタンスに接続します。
- Replication Agent ホスト・マシンで既に使用されているポート番号と競合しないポート番号を指定する必要があります。
- **admin\_port** パラメータの値を **ra\_config** コマンドで変更した場合、新しい値は設定ファイルにすぐに記録されますが、新しいポート番号を有効にするために Replication Agent インスタンスを停止し、再起動する必要があります。
- **admin\_port** パラメータの値を **ra\_config** コマンドで変更した場合は、次回 Replication Agent 管理ポートにログインするときに新しいポート番号を使用する必要があります。

## asm\_password

(Oracle のみ) Oracle Automatic Storage Management (ASM) サーバ・インスタンスにアクセスするための設定パラメータ **asm\_username** ユーザに関連付けられたパスワード。

---

**注意：** このパラメータは Oracle でのみ使用可能です。

---

### デフォルト値

"" (空の文字列)

### 値

有効なパスワード。

### コメント

- Oracle オンライン redo またはアーカイブ redo の記憶域を管理するために Oracle ASM 製品を使用する場合にのみ、この設定パラメータを設定する必要があります。
- **asm\_password** パラメータの値は、Replication Agent インスタンス設定ファイルで暗号化されています。

### 参照：

- [asm\\_tns\\_connection \(181 ページ\)](#)
- [asm\\_tns\\_filename \(181 ページ\)](#)

- asm\_username (182 ページ)

## asm\_tns\_connection

(Oracle のみ) tnsnames.ora ファイルで Oracle Automatic Storage Management (ASM) コネクションのコネクション・パラメータを識別する Oracle コネクション名。設定パラメータ **asm\_tns\_filename** を設定していない場合、**pds\_tns\_filename** によって識別される tnsnames.ora ファイルが使用されます。

---

**注意：** このパラメータは Oracle でのみ使用可能です。

---

デフォルト値  
<設定なし>

値  
Oracle tnsnames.ora ファイルからの、**asm\_tns\_filename** 設定パラメータまたは **pds\_tns\_filename** 設定パラメータで指定された有効な ASM コネクション名。

コメント  
Oracle オンライン redo またはアーカイブ redo の記憶域を管理するために Oracle ASM 製品を使用する場合にのみ、この設定パラメータを設定する必要があります。

### 参照：

- asm\_tns\_filename (181 ページ)
- asm\_password (180 ページ)
- asm\_username (182 ページ)
- pds\_tns\_filename (228 ページ)

## asm\_tns\_filename

(Oracle のみ) Oracle ASM コネクション・パラメータが格納された Oracle tnsnames.ora ファイルを識別する完全修飾ファイル名。ASM に必要なコネクション・パラメータ情報が **pds\_tns\_filename** 設定パラメータによって示された tnsnames.ora ファイルに存在しない場合にのみ、この設定パラメータが必要になります。

---

**注意：** このパラメータは Oracle でのみ使用可能です。

---

デフォルト値  
<設定なし>

## 設定パラメータ

### 値

Oracle ASM サーバへの接続・パラメータが格納された有効な Oracle `tnsnames.ora` ファイル。このファイルは通常、ASM インスタンスの `ORACLE_HOME¥network¥admin` ディレクトリに格納されています。

### コメント

- Oracle オンライン redo またはアーカイブ redo の記憶域を管理するために Oracle ASM 製品を使用する場合にのみ、この設定パラメータを設定する必要があります。
- `pds_tns_filename` 設定パラメータの値が設定されていないか、使用する ASM コネクションのコネクション定義が設定パラメータ `pds_tns_filename` によって示された `tnsnames.ora` ファイルに含まれない場合にのみ、このパラメータが必要になります。これが発生するのは、プライマリ Oracle インスタンスと ASM インスタンスが異なる `ORACLE_HOME` ロケーションにインストールされ、各サーバに異なる `tnsnames.ora` ファイルが存在する場合です。

---

**警告！** Replication Agent プロセスは、このファイルの `read` パーミッションを持つ必要があります。アクセスできない場合は、Replication Agent が Oracle ASM サーバに接続できません。

---

### 参照：

- `pds_tns_filename` (228 ページ)
- `asm_password` (180 ページ)

## asm\_username

(Oracle のみ) Replication Agent が Oracle ASM サーバにアクセスするときに使用するログイン名。

---

**注意：** このパラメータは Oracle でのみ使用可能です。

---

### デフォルト値

<設定なし>

### 値

有効な Oracle ASM ユーザ名。



### コメント

- Oracle オンライン redo またはアーカイブ redo の記憶域を管理するために Oracle ASM 製品を使用する場合にのみ、この設定パラメータを設定する必要があります。
- **asm\_username** パラメータの値は、Oracle オンラインまたはアーカイブ redo ログ用の記憶域を管理する Oracle ASM サーバへのログインに Replication Agent が使用するログイン名です。
- ASM システム・ビューのクエリのために、適切な権限を指定してこのログイン名を ASM サーバに定義する必要があります。
- Replication Agent はこのログインを使用して、Oracle オンラインまたはアーカイブ redo ログのロケーションを判断するために ASM にクエリを行います。
- 複製に必要なくなったアーカイブ・ログ・ファイルを ASM から削除するには、ユーザが ASM の更新権限を持つ必要があります。

### 参照：

- **asm\_tns\_connection** (181 ページ)
- **asm\_tns\_filename** (181 ページ)
- **pdb\_archive\_remove** (207 ページ)
- **asm\_password** (180 ページ)

## auto\_adjust\_block\_count

(Microsoft SQL Server のみ) 読み取りの一部を廃棄する場合に、Replication Agent が **log\_read\_block\_count** パラメータを自動的に調整するかどうかを判断します。

### デフォルト値

**true**

### 値

**true** – ログ読み取りブロック・カウントの自動調整を有効にします。

**false** – ログ読み取りブロック・カウントの自動調整を無効にします。

### コメント

- **auto\_adjust\_block\_count** は、**log\_read\_block\_count** が 1 より大きい値に設定されている場合にのみ、有効性を持ちます。
- Replication Agent は、**log\_read\_block\_count** 設定パラメータによって指定されたブロック・カウントを超えるブロック・カウントに自動調整することはありません。

## column\_compression

Log Transfer Interface コンポーネントがローの更新後イメージ内のすべてのカラムを送信するか、**更新**オペレーションで変更されたカラムのみを送信するかを決定します。

デフォルト値

**true**

値

**true** – 最小カラム情報 (ローの更新後イメージの変更カラムのみ) を**更新**オペレーションの Log Transfer Language (LTL) で有効にします。

**false** – **更新**オペレーションの LTL で最小カラム情報を無効にします。

コメント

- **column\_compression** パラメータを **false** に設定した場合、LTI コンポーネントは LTL 内の完全なローの更新後イメージ (**更新**オペレーションでデータが変更されていないカラムを含む) を送信します。
- **column\_compression** パラメータを **true** に設定した場合、LTI コンポーネントは LTL 内のローの更新後イメージの最小カラム情報 (**更新**オペレーションで変更されたカラムのみ) を送信します。**更新**によりデータが変更されていないカラムは LTL に送信されません。
- 一般に、**column\_compression** パラメータの値を **true** に設定すると、Replication Agent のスループットが向上します。
- Replication Agent インスタンスが最小カラム・データを送信するように設定されている (**column\_compression** および **ltl\_send\_only\_primary\_keys** が **true** に設定されている) 場合、複写定義でサーチャブルに指定されているカラムの一部のカラム・データが省略されることがあります。その結果として、サブスクリプションを作成するデータベースで挿入、サブスクリプション・マイグレーション、またはカスタム・ファンクション文字列に必要なデータがない場合、エラーが発生することがあります。このため、サーチャブル・カラムのある複写定義で参照されるテーブルのオートコレクションを有効にすることをおすすめします。

参照：

- [ltl\\_send\\_only\\_primary\\_keys \(202 ページ\)](#)

## compress\_ltl\_syntax

Log Transfer Interface コンポーネントが省略形構文を使用して Log Transfer Language (LTL) コマンドを圧縮するかどうかを決定します。

デフォルト値  
**true**

値  
**true** – 省略された LTL 構文を使用した LTL 圧縮を有効にします。  
**false** – LTL 圧縮を無効にします。

コメント

- **compress\_ltl\_syntax** パラメータの値を **true** に設定すると、Replication Agent のスループットが向上します。
- LTL コマンドおよび省略された LTL 構文の詳細については、『Replication Agent 管理ガイド』を参照してください。

## connect\_to\_rs

プライマリ Replication Server へのネットワーク・コネクションを有効または無効にします。

デフォルト値  
**true**

値  
**true** – Replication Server へのネットワーク・コネクションを有効にします。  
**false** – Replication Server へのネットワーク・コネクションを無効にします。

コメント

- **connect\_to\_rs** パラメータの値を **false** に設定すると、Replication Agent から Replication Server へのネットワーク・コネクションが無効になり、複写を行うことができません。
- **connect\_to\_rs** パラメータにより Replication Server へのネットワーク・コネクションを無効にした場合でも、Replication Agent インスタンスを *Replicating* 状態にすることができますが、次の制限事項があります。
  - Replication Agent のダミー・コネクションが Replication Server への真のコネクションをエミュレートします。

## 設定パラメータ

- Replication Agent トランザクション・ログに格納された LTM ロケータの値が 0 に設定されます。
- メンテナンス・ユーザ名が無効なユーザ ID に設定されます。

---

**注意：** `connect_to_rs` パラメータの値が `false` の場合は、メンテナンス・ユーザ・オペレーションをフィルタできません。

---

- `connect_to_rs` パラメータを使用して、Replication Server へのネットワーク・コネクションをテストのために一時的に無効にすることができます。
- `connect_to_rs` パラメータの値が `false` の場合、Replication Agent インスタンスを *Replicating* 状態にし、LTITRACELTL トレース・フラグの値を `true` に設定して、コネクションが無効化されていない場合に Replication Server に送信される、判読可能な形式の LTL を表示することができます。
- 通常の Replication Agent オペレーションでは、`connect_to_rs` パラメータの値を `true` にする必要があります。

## ddl\_password

(Oracle および Microsoft SQL Server のみ) RASD のログ・デバイス・レポジトリを更新します。`ddl_username` のパスワードを識別します。

---

**注意：** このパラメータは Oracle および Microsoft SQL Server でのみ使用可能です。

---

デフォルト値  
"" (空の文字列)

値  
有効なパスワード。

コメント

- `ddl_password` パラメータの値は、30 文字以内で指定します。
- `ddl_password` パラメータの値は、`ddl_username` パラメータで指定したデータベース・ユーザ名のパスワードです。
- `ddl_password` パラメータの値は、Replication Agent 設定ファイルで暗号化されています。

## ddl\_username

(Oracle および Microsoft SQL Server のみ) DDL コマンドをスタンバイ・データベースに複製するために LTL に含めるデータベース・ユーザ名。このユーザは、スタンバイ・データベースで複製されたすべての DDL コマンドを実行するパーミッションを持つ必要があります。

---

**注意：** このパラメータは Oracle および Microsoft SQL Server でのみ使用可能です。

---

デフォルト値  
設定なし。

値  
スタンバイ・データベースで有効なユーザ名。

コメント

- **ddl\_username** の値を、Replication Server でスタンバイ・コネクション用に定義されているメンテナンス・ユーザの値と同じにすることはできません。異なる名前を指定しない場合は、Replication Server エラーが発生します。
- **ddl\_username** パラメータの値は、複製されたすべての DDL 文に対して LTL で送信されます。
- **ddl\_password** パラメータの値は、**ddl\_username** パラメータで指定したデータベース・ユーザ名のパスワードです。
- DDL を複製する場合、Replication Server は **ddl\_username** および **ddl\_password** を使用してレプリケート・データベースに接続します。
- Oracle では、Replication Server から次のメッセージが出されます。

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

ここで、*user* はプライマリ・データベースで DDL 操作を生成したユーザ ID です。実際の DDL コマンドがレプリケート・データベースに対して実行されます。**ddl\_username** に **ALTER SESSION SET CURRENT\_SCHEMA** を発行するパーミッションがない場合や、ユーザ・スキーマに対して DDL コマンドを実行するパーミッションがない場合は、コマンドが失敗します。

- Microsoft SQL Server の場合、Replication Agent は次のメッセージを送信します。

```
execute as login = user
```

ここで、*user* はプライマリ・データベースで DDL 操作を生成したユーザ ID です。実際の DDL コマンドがレプリケート・データベースに対して実行されます。**ddl\_username** に **execute as login** を発行するパーミッションがない場合や、*user* スキーマに対して DDL コマンドを実行するパーミッションがない場合は、コマンドが失敗します。

## dump\_batch\_timeout

バッファが満杯でない場合でも、Log Transfer Interface (LTI) バッファの内容を Replication Server に送信するのを待機する秒数を指定します。

デフォルト値  
5

## 設定パラメータ

### 値

1 ~ 60 の整数。

### コメント

- **dump\_batch\_timeout** パラメータの値は、前の LTI バッファが Replication Server に送信されてから次のバッファが送信されるまでの時間 (秒数) です。
- **lti\_batch\_mode** パラメータの値が **false** の場合、**dump\_batch\_timeout** パラメータの影響はありません。

## filter\_maint\_userid

メンテナンス・ユーザが適用したオペレーションが無視されるかどうかを指定します。

### デフォルト値

**true**

### 値

**true** – Log Reader がメンテナンス・ユーザ・オペレーションを無視できます。

**false** – Log Reader フィルタでメンテナンス・ユーザ・オペレーションを複製できなくなります。

### コメント

- **filter\_maint\_userid** 設定パラメータは、プライマリ・データベースが Replication Server メンテナンス・ユーザによって適用されるトランザクションを含んだレプリケート・データベースとしても機能する双方向複製をサポートするために用意されています。
- **filter\_maint\_userid** パラメータの値が **true** の場合、メンテナンス・ユーザによって適用されるデータベース・オペレーションは複製されません。Log Reader コンポーネントは、トランザクション・ログを読み取ると、メンテナンス・ユーザによって適用されるオペレーションを除外 (無視) します。
- **filter\_maint\_userid** パラメータの値が **false** の場合、メンテナンス・ユーザによって適用されるデータベース・オペレーションは複製されます。Log Reader コンポーネントは、オペレーションを適用したユーザに関係なく、マーク付けされたオブジェクトですべてのオペレーションを複製します。
- メンテナンス・ユーザのログインは、プライマリ・データベースのデータベース・コネクションが Replication Server で作成されるときに指定されます。

## function\_password

(Oracle および Microsoft SQL Server のみ) 要求ストアド・プロシージャ複製用のログ転送言語に含まれるパスワード。

---

**注意：** このパラメータは Oracle および Microsoft SQL Server でのみ使用可能です。

---

デフォルト値

"" (空の文字列)

値

有効なパスワード。

コメント

- **function\_password** パラメータの値は、30 文字まで指定できます。
- **function\_password** パラメータの値は、**function\_username** パラメータで指定したデータベース・ユーザ名のパスワードです。
- **function\_password** パラメータの値は、Replication Agent 設定ファイルで暗号化されています。
- 要求ストアド・プロシージャの詳細については、Replication Server のマニュアルを参照してください。

## function\_username

(Oracle および Microsoft SQL Server のみ) 要求ストアド・プロシージャ複製用のログ転送言語 (LTL) に含まれるデータベース・ユーザ名。

---

**注意：** このパラメータは Oracle および Microsoft SQL Server でのみ使用可能です。

---

デフォルト値

sa

値

プライマリ・データベースで有効なユーザ名。

コメント

- **function\_username** パラメータの値は、プライマリ・データベース内のすべての複製ストアド・プロシージャの LTL で送信されます。
- **function\_password** パラメータの値は、**function\_username** パラメータで指定したデータベース・ユーザ名のパスワードです。

## 設定パラメータ

- 要求ストアド・プロシージャの詳細については、Replication Server のマニュアルを参照してください。

### log\_backup\_files

Replication Agent のインスタンス log ディレクトリに格納されるバックアップ・ログ・ファイルの数。

デフォルト値  
3

値  
1 以上の整数。

#### コメント

- システム・ログを循環させる場合、Replication Agent は現在のログ・ファイルをバックアップ・ファイルにコピーし、生成した番号がファイル名に追加されます。  
たとえば、システム・ログ・ファイル名が my\_ra.log の場合、システム・ログの循環時に作成した最初のバックアップ・ファイルの名前は my\_ra1.log になります。2 番目に作成したバックアップ・ファイルの名前は my\_ra2.log となり、以降同様に続きます。
- バックアップ・ファイルの数が **log\_backup\_files** パラメータの値を超えると、最も古いバックアップ・ファイル (生成した番号が最も小さいファイル) は、次のバックアップ・ファイルが作成される前に log ディレクトリから削除されます。

### log\_directory

Replication Agent システム・ログ・ファイルのディレクトリ。

デフォルト値  
Replication Agent インスタンスの作成時に作成された log ディレクトリのパス。  
例：

- Microsoft Windows プラットフォームの場合：

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

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 プラットフォームの場合：

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



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。

#### 値

Replication Agent ホスト・マシンの有効なパス。

#### コメント

- Replication Agent インスタンスを作成すると、log ディレクトリがインスタンスのディレクトリの一部として作成されます。`log_directory` パラメータのデフォルト値は、そのディレクトリを指します。
- `log_directory` パラメータの値として有効なパスを指定すると、Replication Agent のインスタンスは、そのシステム・ログ・ファイルを次回起動するときに指定するディレクトリに配置します。
- `log_directory` パラメータのデフォルト値を `default` キーワード (`ra_config` コマンド内) を使って指定すると、次回起動時に、Replication Agent は、Replication Agent のインスタンスの作成時に作成された log ディレクトリにそのシステム・ログ・ファイルを配置します。
- `ra_config` コマンドで `log_directory` パラメータの値を変更すると、新しい値は設定ファイルにすぐに記録されますが、新しい値を有効にするには Replication Agent のインスタンスを停止して再起動する必要があります。

## log\_read\_block\_count

(Microsoft SQL Server のみ) プライマリ・データベースのトランザクション・ログから一度に読み込むブロックまたはページの数指定します。

#### デフォルト値

64

#### 値

1 ~ 8192 の整数

#### コメント

- Microsoft SQL Server では、トランザクション・ログのページ・サイズは 2K、4K、8K、または 16K に指定できます。

## log\_trace\_verbose

Replication Agent システム・ログ・ファイルで追加の診断情報を有効または無効にします。

デフォルト値  
**false**

値  
**true** – ログ・ファイルの詳細な診断情報を有効にします。  
**false** – ログ・ファイルの詳細な診断情報を無効にします。

コメント  
詳細な診断情報は、Sybase サポート・センタのサポートに基づくトラブルシューティングを目的としたものです。

## log\_wrap

Replication Agent システム・ログ・ファイルの循環前の最大サイズ (1K ブロック単位)。

デフォルト値  
**1000**

値  
1000 以上の整数。

コメント

- **log\_wrap** パラメータの値は、Replication Agent がシステム・ログ・ファイルを循環させるまでに書き込む 1KB ブロックの数です。
- **log\_wrap** パラメータの値が大きいほど、各ファイルに多くのログ履歴を含めることができます。値が小さいほど、ログ・ファイルは小さくなります。
- ログ・ファイルを循環させる場合、Replication Agent は現在のログ・ファイルをバックアップ・ファイルにコピーし、生成した番号がファイル名に追加されます。  
たとえば、システム・ログ・ファイル名が `my_ra.log` の場合、システム・ログの循環時に作成した最初のバックアップ・ファイルの名前は `my_ra1.log` になります。2 番目に作成したバックアップ・ファイルの名前は `my_ra2.log` となり、以降同様に続きます。
- バックアップ・ファイルの数が **log\_backup\_files** パラメータの値を超えると、最も古いバックアップ・ファイル (生成した番号が最も小さいファイル) は、次

のバックアップ・ファイルが作成される前に log ディレクトリから削除されます。

### lr\_max\_lobdata\_cache

(Oracle のみ) ロー外 LOB データの LOB データ・キャッシュの最大サイズをバイト数で示したものの。

デフォルト値

**8192**

値

8192 ~ 2147483647 の整数。

コメント

- **lr\_max\_lobdata\_cache** を使用すると、LOB データの複製に対する Replication Agent のパフォーマンスを微調整できます。
- 最高のパフォーマンスを得るために、**lr\_max\_lobdata\_cache** を複製される LOB データの最大サイズよりも大きい値に設定します。たとえば、LOB データが 10 ~ 120KB の範囲にある場合には、**lr\_max\_lobdata\_cache** を 120KB より大きい値に設定します。

### lr\_max\_op\_queue\_size

(Oracle および Microsoft SQL Server のみ) 複製中に Log Reader のオペレーション・キューで使用できるオペレーションの最大数。

デフォルト値

**1000**

値

25 ~  $2^{31} - 1$  の整数。

コメント

- **lr\_max\_op\_queue\_size** パラメータを使用すると、Replication Agent のパフォーマンスを微調整できます。

### lr\_max\_scan\_queue\_size

(Oracle および Microsoft SQL Server のみ) 複製中に Log Reader のログ・スキャン・キューで使用できるログ・レコードの最大数。

**注意：** このパラメータは Oracle および Microsoft SQL Server でのみ使用可能です。

## 設定パラメータ

デフォルト値

**1000**

値

25 ~  $2^{31} - 1$  の整数。

コメント

- **lr\_max\_scan\_queue\_size** パラメータを使用すると、Replication Agent のパフォーマンスを微調整できます。

### lr\_ntext\_byte\_order

(Oracle および Microsoft SQL Server のみ) NCLOB データ (Oracle 用) または NTEXT データ (Microsoft SQL Server 用) のバイト順序がレプリケート・データベースに正しく送信されます。

異なるプライマリ・データベースとレプリケート・データベースのタイプ (Oracle と Microsoft SQL Server など)、または異なるプラットフォーム (リトル・エンディアン Windows とビッグ・エンディアン UNIX など) にあるプライマリ・データベースとレプリケート・データベースを使用して複製する場合は、バイト順序が正しい必要があります。

---

**注意：** このパラメータは Oracle および Microsoft SQL Server でのみ使用可能です。

---

デフォルト値

**big**

値

**big** – ビッグ・エンディアン

**little** – リトル・エンディアン

例

- Microsoft SQL Server の場合  
Microsoft SQL Server 間でデータを複製する場合、Microsoft SQL Server は Windows でのみ動作するため、プライマリ・データベースとレプリケート・データベースのどちらもリトル・エンディアンです。パラメータはリトル・エンディアンに設定する必要があります。レプリケート・データベースが Microsoft SQL Server でない場合は、リトル・エンディアンとビッグ・エンディアンのどちらなのかを確認し、パラメータが一致するように設定する必要があります。
- Oracle の場合

Oracle 間で複写する場合は、レプリケート・データベースのエンディアンを確認し、正しいエンディアンがレプリケートに送信されるようにパラメータを設定する必要があります。Oracle 以外のデータベースに複写する場合も同様です。

## コメント

- **lr\_ntext\_byte\_order** が **little** に設定されている場合は、**ltl\_big\_endian\_uni**text を **false** に設定する必要があります。
- ビッグ・エンディアンは左から右のバイト順序アーキテクチャを示し、リトル・エンディアンは右から左のバイト順序アーキテクチャを示します。
- Microsoft SQL Server は、リトル・エンディアンのバイト順序でダブルバイトの ntext データ型値を格納します。デフォルトで、ntext データのバイト順序は複写中にビッグ・エンディアンに変換されるため、一般的なネットワーク・バイト順序、またはビッグ・エンディアンを使用してネットワークを介してデータを転送できます。ターゲット・データベースも Microsoft SQL Server である場合、SQL Server は送信したビッグ・エンディアン順序から SQL Sever で必要なリトル・エンディアン順序に複写したデータを自動的に変換しません。ntext データの Microsoft SQL Server (または、必要な変換を提供できない他のレプリケート・サーバ) への複写をサポートするには、**lr\_ntext\_byte\_order** パラメータを使用してバイト順序を強制的に送信します。それには、レプリケート・データベースの要求を満たすため、必要に応じて、**big** または **little** を指定します。

---

**注意：** Replication Agent のデフォルトの動作では、設定パラメータ **ltl\_big\_endian\_uni**text の定義に従って、ユニコード・データが強制的にビッグ・エンディアン順序になります。また、設定パラメータ **lr\_ntext\_byte\_order** で SQL Server バイト順序を正常に上書きするには、**lr\_next\_byte\_order** パラメータを使用するたびに **ltl\_big\_endian\_uni**text 設定パラメータを **false** に設定する必要があります。

---

- 以下に、**ltl\_big\_endian\_uni**text 設定パラメータと **lr\_ntext\_byte\_order** 設定パラメータの関係について説明します。
  - デフォルトで、**ltl\_big\_endian\_uni**text が **true** に設定されている場合、Replication Agent ではすべてのユニコード・データがビッグ・エンディアン順序で送信されます。
  - **false** に設定されている場合に **ltl\_big\_endian\_uni**text を使用すると、データがトランザクション・ログ・ファイルに格納されるときに使用されるバイト順序でユニコード・データを送信できます。
  - 一方、**lr\_ntext\_byte\_order** を使用すると、トランザクション・ログから読み込んだユニコード・データの結果は、トランザクション・ログ・ファイル内で通常どのような順序かに関係なく、要求したバイト順序になります。

参照：

- `ltl_big_endian_unitext` (200 ページ)

## lr\_read\_buffer\_size

(UDB のみ) ログ読み込みバッファのサイズ (バイト単位)。

---

**注意：** このパラメータは UDB でのみ使用可能です。

---

デフォルト値

**16384**

値

10000 ~  $2^{31} - 1$  の整数。

コメント

- Replication Agent for UDB の Log Reader コンポーネントは、`lr_read_buffer_size` パラメータの値を使用して、各スキャン中にトランザクション・ログから読み込まれる最大バイト数を決定します。読み込みバッファ・サイズが小さすぎて 1 つのオペレーションを読み込めない場合、Log Reader コンポーネントは Replication Agent インスタンスを停止し、UDB **-2650** または **-30081** エラーをレポートします。ただし、バッファ・サイズが不十分な場合だけでなく、一般的な通信エラーもこれらの UDB エラー・メッセージの対象に含まれます。
- 読み込みバッファ・サイズが大きすぎると、Replication Agent インスタンスがメモリ不足になる可能性があります。

## lr\_send\_trunc\_partition\_ddl

(Oracle のみ) レプリケート・データベースに `truncate partition` コマンドを DDL または DML のどちらとして送信するかを指定します。

---

**注意：** このパラメータは Oracle でのみ使用可能です。

---

デフォルト値

true

値

true – `truncate partition` コマンドが DDL コマンド (`alter table`) として送信されます。

false – `truncate partition` が DML オペレーションとして送信されます。

### コメント

- true に設定すると、**truncate partition** コマンドは DDL コマンドとして送信されます。通常は、Oracle に複写するように設定されます。
- **truncate partition** コマンドを DML として扱うデータベースに複写する場合は、値を false に設定します。

## lti\_batch\_mode

Log Transfer Interface コンポーネントの LTL バッチ・モードを有効または無効にします。

### デフォルト値

**true**

### 値

**true** – LTL バッチ・モードを有効にします。

**false** – LTL バッチ・モードを無効にします。

### コメント

- **lti\_batch\_mode** パラメータの値が **true** の場合、LTI コンポーネントは LTI コマンドを Replication Server に 1 コマンドずつではなくバッチで送信します。
  - LTI コンポーネントは、LTL バッチ・モード・バッファにできるかぎり多くの LTL コマンドを含めてから、コマンドを Replication Server に送信します。
  - **dump\_batch\_timeout** パラメータで指定した時間間隔を経過すると、LTI コンポーネントは、バッファが満杯でなくても、その時点の LTL バッチ・モードのバッファの内容を Replication Server に送信します。
- **lti\_batch\_mode** パラメータの値が **false** の場合、LTI コンポーネントは入力キューで設定した変更ごとに、個々の LTL コマンドを Replication Server に送信します。
- Replication Agent を Replication Server に接続すると、Replication Server のバージョンを特定します。
  - Replication Server のバージョンが 12.5 より前の場合、LTL バッチ・モードのバッファのサイズは自動的に 16KB に設定されます。
  - Replication Server のバージョンが 12.5 以降である場合は、Replication Agent は LTL バッチ・モードのバッファのサイズを **ltl\_batch\_size** パラメータで指定したサイズに設定します。
- Replication Server のバージョンが 12.5 以降である場合は、Replication Agent **ltl\_batch\_size** パラメータを使用して、LTI コンポーネントの LTL バッチ・モードのバッファのサイズを設定できます。

---

**注意：**LTL バッチ・モードのバッファのサイズを調整することで、複製システムのパフォーマンスを最適化できます。

---

- Replication Server のバージョンが 12.5 より前で、**lti\_batch\_mode** パラメータの値が **true** の場合に、1 つの LTL **distribute** コマンドが LTL バッチ・モードのバッファのサイズ (16K) を超えていると、Replication Server はエラーを返し、Replication Agent は *Replication Down* 状態になります。
- 一般に、**lti\_batch\_mode** パラメータの値を **true** に設定すると、Replication Agent のスループットが向上します。

### **lti\_formatter\_count**

LTI キューの項目で同時に動作する LTL フォーマッタのスレッドの数。パラメータの値は、マシンにインストールされた Replication Agent で使用可能なパラメータ数に応じて調整する必要があります。

デフォルト値

3

値

1-200

コメント

- **lti\_formatter\_count** によって指定された各スレッドは、LTI キューの個々の項目で動作します。
- **lti\_formatter\_count** の値を増加させるとパフォーマンスが向上するかどうかを判断するには、LTI 統計値で、「LTI キューの現在のコマンドの数」および「LTI キューの現在フォーマットされていないコマンドの数」を調べます。LTI キューのコマンド数がほぼ上限で、フォーマットされていないコマンドの数がゼロよりも上限に近い場合、**lti\_formatter\_count** の値を増加させると、Replication Agent のパフォーマンスが向上する場合があります。
- **lti\_formatter\_count** は、Replication Agent が Admin 状態または Replication Down 状態の場合にのみ変更できます。
- 並列で動作する LTL フォーマッタ・スレッド数を調整することで、マルチプロセッサ・マシン上の Replication Agent のパフォーマンスを向上させることができます。

### **lti\_max\_buffer\_size**

LTL キューに格納できる項目の最大数。このパラメータは、LTL フォーマッタ・プロセス・スレッド数と使用可能なパラメータ数に応じて設定します。

デフォルト値

5000



**値**

1000 ~ 100000 の整数。

**コメント**

- **lti\_max\_buffer\_size** は、Log Reader によって LTI キューに配置できる項目の最大数を指定します。
- **lti\_max\_buffer\_size** に設定された値が非常に大きい場合、使用可能なメモリが不足すると、パフォーマンスが低下する可能性があります。
- **lti\_max\_buffer\_size** は、Replication Agent が Admin 状態または Replication Down 状態の場合にのみ変更できます。
- **lti\_max\_buffer\_size** を使用して LTI キューのサイズを調整することで、マルチプロセッサがインストールされているマシンの Replication Agent のパフォーマンスを向上させることができます。

**lti\_update\_trunc\_point**

新しい LTM ロケータを要求する前に送信されるログ転送言語 (LTL) コマンドの数。

**デフォルト値**

**10000**

**値**

1 ~ 100000 の整数。

**コメント**

- **lti\_update\_trunc\_point** パラメータの値は、Replication Agent が新しい LTM ロケータ (セカンダリ・トランケーション・ポイント) を要求する前に送信する LTL コマンドの数です。
- 数が小さいほど、Replication Agent が新しい LTM ロケータを要求する頻度は高くなります。
- **truncation\_type** パラメータの値が **locator\_update** の場合に **lti\_update\_trunc\_point** パラメータの値を低い数に設定すると、自動ログ・トランケーションの発生頻度が高くなります。
- **lti\_update\_trunc\_point** パラメータの値は、システム・パフォーマンスが向上するほどリカバリ時間が長くなる両者の間のトレードオフを示します。
  - 値が小さいと複製エラーからリカバリするために要する時間が短縮されますが、システム・スループット全体に悪影響を与える可能性があります。
  - 値が大きいとシステム・スループット全体が向上しますが、複製エラーからリカバリするために要する時間が長くなる可能性があります。

## 設定パラメータ

- Replication Agent が信頼できないネットワーク環境で稼働している場合は、リカバリ時間を短縮できるように **lti\_update\_trunc\_point** パラメータに小さい値を設定することをおすすめします。

### lti\_batch\_size

Log Transfer Interface コンポーネントのログ転送言語 (LTL) バッチ・モードのバッファのサイズ (バイト単位)。

デフォルト値  
**40000**

値  
16384 ~ 10485760 の整数。

#### コメント

- **lti\_batch\_size** パラメータの値は、LTI コンポーネントの LTL バッチ・モードのバッファのサイズ (バイト単位) です。
- Replication Agent を Replication Server に接続すると、Replication Server のバージョンを特定します。
  - Replication Server のバージョンが 12.5 より前である場合は、LTL バッチ・モードのバッファのサイズは自動的に 16K に設定され、**lti\_batch\_size** パラメータの値は無視されます。
  - Replication Server のバージョンが 12.5 以降である場合は、Replication Agent は LTL バッチ・モードのバッファのサイズを **lti\_batch\_size** パラメータで指定したサイズに設定します。
- Log Transfer Interface コンポーネントは、**lti\_batch\_mode** パラメータの値が **true** の場合にのみ、LTL バッチ・モードのバッファを使用します。**lti\_batch\_mode** パラメータの値が **false** の場合、LTL バッチ・モードのバッファは使用されません。

### lti\_big\_endian\_unitext

LTL を Replication Server に送信する前に、"unitext" データをリトル・エンディアンからビッグ・エンディアンに変換するかどうかを指定します。

デフォルト値  
**true**

値  
**true** – Unitext データはリトル・エンディアンのバイト順序からビッグ・エンディアンのバイト順序に変更されます。

**false** – Unitext データのバイト順序は変更されません。

### コメント

このパラメータを設定する場合は、`lr_ngettext_byte_order` の設定方法を理解している必要があります。パラメータ `lr_ngettext_byte_order` がレプリケート・データベースの正しいバイト順序を送信するように設定されている場合は、バイト順序が変更されないように `ltl_big_endian_unitext` を `false` に設定する必要があります。

### 参照：

- `lr_ngettext_byte_order` (194 ページ)

## ltl\_character\_case

Replication Server に送信するログ転送言語 (LTL) のデータベース・オブジェクト名に使用される大文字と小文字。

### デフォルト値

**asis**

### 値

**asis** – データベース・オブジェクト名は、プライマリ・データベースから返されたものと同じ大文字と小文字で送信されるか、(`use_rssd` パラメータの値が `true` の場合は) 複写定義で指定されたものと同じ大文字と小文字で送信されます。

**lower** – データベース・オブジェクト名は、プライマリ・データベースからどのように返されるか、または複写定義でどのように指定されるかに関係なく、「すべて小文字」で送信されます。

**upper** – LTL のデータベース・オブジェクト名は、プライマリ・データベースからどのように返されるか、または複写定義でどのように指定されるかに関係なく、「すべて大文字」で送信されます。

### コメント

- `ltl_character_case` 設定パラメータを使用すると、プライマリ・データベースが返すものとは異なるオブジェクト名を指定する複写定義を使用するように LTL のデータベース・オブジェクト名の処理をカスタマイズできます。
- `ltl_character_case` パラメータの値が `asis` で、`use_rssd` パラメータの値が `true` の場合、データベース・オブジェクト名は複写定義で指定されているものと同じ大文字と小文字で送信されます。
- `ltl_character_case` パラメータの値が `asis` で、`use_rssd` パラメータの値が `false` の場合、データベース・オブジェクト名はプライマリ・データベースから返されたものと同じ大文字と小文字で送信されます。

## 設定パラメータ

- 複写定義でデータベース・オブジェクト名をすべて小文字で指定する場合は、**ltl\_character\_case** パラメータの値を **lower** に設定します。
- 複写定義でデータベース・オブジェクト名をすべて大文字で指定する場合は、**ltl\_character\_case** パラメータの値を **upper** に設定します。
- 大文字と小文字が「混在する」データベース・オブジェクト名 (*MyTable* など) を送信する場合は、**ltl\_character\_case** パラメータの値を **asis** に設定します。

### ltl\_origin\_time\_required

ログ転送言語 (LTL) の **origin\_time** コマンド・タグを有効または無効にします。

デフォルト値

**false**

値

**true** – LTL の **origin\_time** コマンド・タグを有効にします。

**false** – LTL の **origin\_time** コマンド・タグを無効にします。

コメント

- **ltl\_origin\_time\_required** パラメータの値が **true** の場合、Log Transfer Interface コンポーネントには生成される LTL の **origin\_time** コマンド・タグが含まれます。
- Replication Server ファンクション文字列で **origin\_time** コマンド・タグをチェックする場合は、**ltl\_origin\_time\_required** パラメータの値を **true** に設定します。
- LTL の **origin\_time** コマンド・タグに格納された *datetime* 値は、元のプライマリ・データベース・オペレーションが Log Reader コンポーネントによってスキャンおよび処理された時間ではなく、トランザクション・ログに記録された時間です。
- **ltl\_origin\_time\_required** パラメータの値を **false** に設定すると、Replication Agent のスループットが向上します。
- Replication Manager を使用して遅延時間をレポートする場合は、**ltl\_origin\_time\_required** パラメータの値を **true** に設定する必要があります。

### ltl\_send\_only\_primary\_keys

Replication Agent がレプリケート・データベースに対する更新および削除オペレーションを行うため Replication Server に「更新前」イメージのプライマリ・キー・カラムのみを送信するか、すべての「更新前」イメージのカラムを送信するかを指定します。

デフォルト値

**true**

## 値

**true** – 「更新前」イメージのプライマリ・キー・カラムのみを Replication Server に送信します。

**false** – すべての「更新前」イメージのカラムを Replication Server に送信します。

## コメント

- **true** に設定し、テーブルのプライマリ・キー・カラムを識別する複写定義が存在する場合は、更新および削除オペレーションで「更新前」イメージのプライマリ・キー・カラム値のみが送信されます。プライマリ・キー・カラムのデータのみを送信すると、Replication Server に送信されるデータ量が少なくなります。これは、更新および削除オペレーションの "where" 句を作成する際にプライマリ・キーのみが使用されるためです。
- **false** に設定した場合、プライマリ・キーの定義に関係なく、使用できるすべてのカラムの「更新前」イメージの値が送信されます。
- レプリケート・データベースでカスタム・ファンクション文字列をサポートする場合や、追加のカラム・データがあるとその他の問題の解決に役立つ場合など、「更新前」イメージの値を追加することでメリットがある場合のみ、**ltl\_send\_only\_primary\_keys** を **false** に設定してください。
- プライマリ・キーは、テーブル・レベルの複写定義内で指定されます。設定で **use\_rssd** が **false** の場合、複写定義に関する情報が Replication Server システム・データベース (RSSD) から収集されないため、**ltl\_send\_only\_primary\_keys** の設定は影響しません。
- 値を **false** に設定すると、パフォーマンスが低下します。
- Replication Agent インスタンスが最小カラム・データを送信するように設定されている (**column\_compression** および **ltl\_send\_only\_primary\_keys** が **true** に設定されている) 場合、複写定義でサーチャブルに指定されているカラムの一部のカラム・データが省略されることがあります。その結果として、サブスクリプションを作成するデータベースで挿入、サブスクリプション・マイグレーション、またはカスタム・ファンクション文字列に必要なデータがない場合、エラーが発生することがあります。このため、サーチャブル・カラムのある複写定義で参照されるテーブルのオートコレクションを有効にすることをおすすめします。

## 参照：

- [column\\_compression \(184 ページ\)](#)

## ltm\_admin\_pw

Replication Agent の管理者ログイン・パスワード。

デフォルト値  
"" (空の文字列)

値  
有効なパスワード。

コメント

- **ltm\_admin\_pw** パラメータの値は、Replication Agent にログインする許可を得ているユーザ名のパスワードです。
- **ltm\_admin\_pw** パラメータの値は Replication Agent 設定ファイルで暗号化されています。
- **ltm\_admin\_pw** パラメータの値を変更するには、**ra\_set\_login** を使用します。
- **ltm\_admin\_pw** パラメータの値を **ra\_set\_login** を使用して変更すると、新しい値が設定ファイルにすぐに記録されます。ただし、新しいパスワードを有効にするには、Replication Agent インスタンスを停止してから再起動する必要があります。  
**ltm\_admin\_pw** パラメータの値を **ra\_set\_login** を使用して変更した後、Replication Agent に次回ログインするときには新しいパスワードを使用する必要があります。

参照：

- ltm\_admin\_pw\_min\_length (204 ページ)

## ltm\_admin\_pw\_min\_length

Replication Agent 管理者のログイン・パスワードの最小の長さ。

デフォルト値  
-1 (無効)

値  
3 ~ 12 の整数。

### コメント

- Replication Agent の管理者ログイン用のパスワードを変更しようとする、新しいパスワードは `ltm_admin_pw_min_len` に照合され、パスワードの長さが3文字以上12文字以下であることが確認されます。

### 参照：

- `ltm_admin_pw` (204 ページ)

## ltm\_admin\_user

Replication Agent の管理者ログイン名。

### デフォルト値

**sa**

### 値

Replication Agent ホスト・マシンの有効なユーザ名。

### コメント

- `ltm_admin_user` パラメータの値は、Replication Agent にログインする許可を得ているユーザ名です。
- `ltm_admin_user` パラメータの値を変更するには、`ra_set_login` コマンドを使用します。
- `ltm_admin_user` パラメータの値を `ra_set_login` コマンドを使用して変更すると、新しい値が設定ファイルにすぐに記録されます。ただし、新しい管理者の名前を有効にするには、Replication Agent インスタンスを停止してから再起動する必要があります。
- `ltm_admin_user` パラメータの値を `ra_set_login` を使用して変更した後、Replication Agent に次回ログインするときには新しい管理者の名前を使用する必要があります。

## max\_ops\_per\_scan

`max_ops_per_scan` パラメータは今後廃止される予定で、下位互換性を保つためだけにのみ提供されています。`max_ops_per_scan` パラメータに行った変更は、Replication Agent の動作には影響しません。

### デフォルト値

**1000**

### 値

25 ~  $2^{31} - 1$  の整数。

参照：

- `lr_read_buffer_size` (196 ページ)

## **pdb\_archive\_path**

(Oracle および UDB のみ) Replication Agent がアーカイブされた redo ログ・ファイル (Oracle 用) またはアーカイブされたトランザクション・ログ・ファイル (UDB 用) の検出を想定しているディレクトリ・パスを識別します。アーカイブされた redo ログ・ファイルまたはアーカイブされたトランザクション・ログ・ファイルがファイル・システムに格納されると、設定パラメータはファイル・システムのパスに設定されます。

---

**注意：** このパラメータは Oracle および UDB でのみ使用可能です。

---

デフォルト値  
<設定なし>

### 値

Oracle の場合、Oracle がアーカイブされた redo ログ・ファイルを配置する場所を指す、Replication Agent をホストするマシン上の有効なディレクトリ・パス。次に例を示します。

```
ORACLE_HOME¥oradata¥orcl¥archive
```

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

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

**pdb\_truncate\_xlog** を設定して、アーカイブ・ログを手動で削除することもできます。**pdb\_archive\_path** を ASM ディスク・グループ名に設定し、手動で削除するアーカイブ・ログの前にプラス "+" 記号を付けます。

UDB の場合、DISK に設定されている UDB LOGARCHMETH1 または LOGARCHMETH2 データベース設定パラメータによって定義されているディスク上の有効なディレクトリ・パス。例を示します。

```
DISK: <path>
```

### コメント

- Oracle では、設定パラメータ **pdb\_include\_archives** が **true** に設定されている場合は **pdb\_archive\_path** を設定する必要があり、Replication Agent が「複写」状態になるには、有効な場所に設定する必要があります。



- Replication Agent は Oracle オンライン redo ログで必要なログ・レコードを見つけることができない場合、必要なレコードを含んでいるアーカイブ・ログ・ファイルについてこのディレクトリを検索します。
- UDB では、**pdb\_archive\_path** の設定は、**pdb\_archive\_remove** を有効にする (**true** に設定する) 前に行う必要があります。**pdb\_archive\_remove** パラメータが **true** に設定されている場合、Replication Agent for UDB は **pdb\_archive\_path** により指定された場所からアーカイブされたログ・ファイルを削除します。

**参照：**

- **pdb\_archive\_remove** (207 ページ)
- **pdb\_include\_archives** (217 ページ)

## **pdb\_archive\_remove**

(Oracle および UDB のみ) **pdb\_archive\_path** により指定されたパスからのアーカイブされた Oracle の redo ログ・ファイルまたはアーカイブされた UDB のトランザクション・ログの削除を有効または無効にします。

---

**注意：** このパラメータは Oracle および UDB でのみ使用可能です。

---

デフォルト値

**false**

値

**true** – **pdb\_archive\_path** により指定されたパスからのアーカイブされた Oracle の redo ログおよびアーカイブされた UDB のトランザクション・ログ・ファイルの削除を許可します。削除は、コマンド **pdb\_truncate\_xlog** の実行、またはパラメータ **truncation\_type** および **truncation\_interval** で指定された自動トランザクションのタイミングに基づいて行われます。

**false** – アーカイブされた Oracle の redo ログ・ファイルまたはアーカイブされた UDB のトランザクション・ログ・ファイルの削除を無効にします。

コメント

- Oracle では、**pdb\_archive\_path** で指定されたパスが複写機能をサポートするためにのみ設定されていて、アーカイブされた不要なログ・ファイルを自動削除する必要がある場合、この設定に **true** を指定します。
- UDB では、ログをアーカイブするため **LOGARCHMETH1** 設定パラメータを **DISK:<パス>** に設定していて、アーカイブされたログ・ファイルを自動削除する必要がある場合、この設定に **true** を指定します。
- **pdb\_archive\_path** により指定されたパスを他のプロセスで共有する場合、またはアーカイブされたログ・ファイルの削除が Replication Agent 以外のプロセス

## 設定パラメータ

で実行されると想定される場合、このパラメータは **false** である必要があります。

- 設定パラメータ **truncation\_type** および **truncation\_interval** とコマンド **pdb\_truncate\_xlog** は、この設定パラメータが **false** に設定されている場合は、影響を与えません。

### 参照：

- **pdb\_archive\_path** (206 ページ)
- **truncation\_interval** (256 ページ)
- **truncation\_type** (257 ページ)
- **pdb\_truncate\_xlog** (73 ページ)

## pdb\_auto\_create\_repdefs

(Oracle および Microsoft SQL Server のみ) テーブルまたはプロシージャが初期化の後に複製するようマーク付けされている場合、Replication Server で複製定義を自動的に作成するように Replication Agent を設定します。パフォーマンスを向上させるため、**pdb\_auto\_create\_repdefs** は初期化中は無視されます。

---

**注意：** このパラメータは Oracle および Microsoft SQL Server でのみ使用可能です。

---

**注意：** **pdb\_xlog create** コマンドは、初期化中に **pdb\_auto\_create\_repdefs** の設定をチェックしません。**pdb\_xlog create** を実行した後でマーク付けされたすべてのテーブルの複製定義を作成するには、コマンド **rs\_create\_repdef all** を実行します。

---

デフォルト値  
false

値  
true – テーブルまたはプロシージャが初期化後にマーク付けされている場合、Replication Agent は Replication Server で複製定義を自動的に作成します。

false – テーブルまたはプロシージャがマーク付けされている場合、複製定義は作成されません。

コメント

---

**注意：** Replication Agent では、プライマリ・データベースのデータベース複製定義が必ず存在すると想定しています。

---

- **rs\_username** ユーザは、create object パーミッションを持っている必要があります。これがないと、Replication Agent はそれを使用して Replication Server

から複写定義を作成できません。このパーミッションは、RSSD から手動で付与する必要があります。

- Replication Agent が作成するテーブルとプロシージャの複写定義では、プライマリ・データベースのデータベース・レベルの複写定義が Replication Server に既に存在すると想定しています。pdb\_auto\_create\_repdefs を使用して作成された複写定義にはすべて、send standby 句が含まれます。これは、データベース・レベルの複写定義がある場合、またはプライマリ Replication Server コネクションがウォーム・スタンバイ設定用である場合、その複写定義が Replication Server 専用であることを意味します。rs\_create\_repdef で作成された複写定義を個別にサブスクライブすることはできません。データベース・レベルの複写定義またはウォーム・スタンバイ設定を必要としない場合は、別のツールを使用するか、複写定義を手動で作成する必要があります。rs\_create\_repdef は使用しないでください。
- rs\_create\_repdef で作成された複写定義では、常に Replication Server にインストールされた使用可能なユーザ定義データ型 (UDD) を使用してデータ型を定義します。つまり、rs\_create\_repdef を使用する場合は、Replication Agent 設定パラメータ pdb\_convert\_datetime を true に設定しないでください。これを行うと、date および timestamp データ型が UDD 形式でなく Sybase 形式に変換されます。
- このパラメータを true に設定し、プロシージャ (複数も可) をマーク付けするために pdb\_setrepproc を呼び出すと、複写のマークが付けられるプロシージャごとに Replication Server で複写定義が作成されます。
- このパラメータを true に設定し、テーブル (複数も可) をマーク付けするために pdb\_setreptable を呼び出すと、複写のマークが付けられるテーブルごとに Replication Server で複写定義が作成されます。
- このパラメータを true に設定し、テーブル (複数も可) のマークを解除するために pdb\_setreptable を呼び出すと、複写のマークを解除されるテーブルごとに Replication Server で複写定義が削除されます。
- このパラメータを true に設定し、プロシージャ (複数も可) のマークを解除するために pdb\_setrepproc を呼び出すと、複写のマークを解除されるプロシージャごとに Replication Server で複写定義が削除されます。
- 複写定義のテーブルおよびプロシージャには、以下が当てはまります。
  - 英数字以外のすべての文字およびスペースは削除され、テーブル名またはプロシージャ名に使用されることはありません。
  - アンダースコアは英数字以外ではなくても、名前に使用されます。
  - ピリオドはアンダースコアに置き換えられます。
- テーブルの複写定義名は、必ずプレフィクス "ra\$" から始まり、その後一意の英数字の識別子 (最大 8 文字) が続き、最後がテーブル名またはオブジェクト名になります。たとえば、複写名が "My Table" の場合、複写定義名は "ra \$0x7952\_mytable" になります。"mytable89012345678901234567890" のように特に長い複写名 (30 文字) の場合、複写定義名は "ra

\$0x7952\_mytable8901234567890" になります (**pdb\_support\_large\_identifier** 設定パラメータが設定されているかどうかに応じて、最大 30 文字か 255 文字)。

- プロシージャの複写定義名は、プロシージャの名前と同じです。

### **pdb\_automark\_tables**

(Oracle および Microsoft SQL Server のみ) DDL 複写中に Replication Agent が自動的にテーブルに複写のマーク付けをするかどうかを指定します。

---

**注意：** このパラメータは Oracle および Microsoft SQL Server でのみ使用可能です。

---

デフォルト値

**true**

値

**true** – ユーザ・テーブルは DDL 複写中に自動的にマーク付けされます。

**false** – ユーザ・テーブルは DDL 複写中に自動的にマーク付けされません。常に **pdb\_setreptable** コマンド (デフォルト) を使用してユーザ・テーブルにマーク付けする必要があります。

コメント

- **pdb\_automark\_tables** のデフォルト値は、Replication Agent インスタンスの作成時に **true** に設定されます。このデフォルト設定では、**pdb\_xlog** コマンドが **init** キーワードを使用して実行される場合、すべてのユーザ・テーブル (所有者が **owner\_filter\_list** に含まれていないもの) に複写のマークが付けられます。さらに、DDL コマンドの複写が有効である場合 (**pdb\_setrepddl** 設定はデフォルトで有効)、ユーザ・テーブル (所有者が **owner\_filter\_list** に含まれていないもの) の **create table** コマンドには自動的に複写のマークが付けられます。テーブルに自動的にマークを付ける必要がない場合は、この設定パラメータの値を **false** に変更します。
- 新しいテーブル (**create table** コマンドを使用してプライマリ・データベースで作成したもの) への自動的なマーク付けは、DDL コマンドの複写が有効であり (**pdb\_setrepddl** が **enable** に設定されている)、テーブルがユーザ・テーブル (所有者が **owner\_filter\_list** に含まれていないもの) で、**pdb\_automark\_tables** が **true** に設定されている場合にのみ行われます。初期化の後に **owner\_filter\_list** を変更すると、矛盾が生じる場合があります。
- **drop table** コマンドがプライマリで発行され、トランザクション・ログに記録されると、**pdb\_setrepddl** または **pdb\_automark\_tables** の設定に関係なく、テーブルは自動的に複写のマークを解除されます。これは、削除されたテーブルを複写できないことが原因です。
- ユーザ・テーブルの自動的なマーク付けは、**pdb\_setreptable** コマンドを使用したテーブルの手動によるマーク付けとは別のものです。つまり、

**pdb\_setreptable** コマンドを使用すると、**pdb\_automark\_tables** の設定に関係なく、個々のテーブルまたはすべてのテーブルについて複製のマーク付けまたはマーク解除をいつでも行うことができます。

## **pdb\_auto\_run\_scripts**

Replication Agent がプライマリ・データベースでトランザクション・ログの作成と削除、およびオブジェクトのマーク付けとマーク解除のスクリプトを自動的に実行するかどうかを指定します。

デフォルト値

**true**

値

**true** – Replication Agent はスクリプトを自動的に実行します。

**false** – Replication Agent はスクリプトを生成し保存しますが、プライマリ・データベースで自動的に実行しません。

コメント

- **pdb\_xlog** コマンドがトランザクション・ログを作成または削除するために呼び出されると、Replication Agent は Replication Agent システム・オブジェクトを作成または削除するスクリプトを生成します。
- **pdb\_setrepproc** コマンドがプライマリ・データベースでオブジェクトをマーク付けまたはマーク解除するために呼び出されると、Replication Agent はオブジェクトのマーク付けに必要なトランザクション・ログ・オブジェクトを作成または削除するスクリプトを生成します。
- Replication Agent は常にスクリプトをファイルに保存します。ログの作成および削除のスクリプトは、partinit.sql および partdeinit.sql という名前のファイルに保存されます。オブジェクトのマーク付けおよびマーク解除のスクリプトは、partmark.sql および unmark.sql という名前のファイルに保存されます。
- **pdb\_auto\_run\_scripts** パラメータが **false** に設定されている場合、スクリプトは作成されますが、何も実行されません。これにより、どのような操作が実行されるかを実行前にスクリプトで確認できます。スクリプトを実行することはできません。必要な操作を実行するには、**pdb\_auto\_run\_scripts** パラメータを **true** に設定し、コマンドを再実行する必要があります。
- **pdb\_xlog** および **pdb\_setrepproc** コマンドについて上記で説明したように、Oracle および Microsoft SQL Server は partinit、partdeinit、partmark、および partunmark スクリプトを作成します。Oracle および Microsoft SQL Server の場合、これらのスクリプトは実行されず (RASD を更新しないため)、参照目的にのみ使用されます。

## 設定パラメータ

- 初期化を行うには、このパラメータを **true** に設定する必要があります。

### **pdb\_convert\_datetime**

Replication Agent が Sybase 以外の時間データ型の値を Sybase *datetime* 形式に変換するかどうかを指定します。

---

**注意：** このパラメータは廃止されました。Replication Server バージョン 12.0 以降を使用している場合は、すべてのデータ型変換に Replication Server 異機種データ型サポート (HDS) 機能を使用します。

---

#### デフォルト値

false

#### 値

true – Replication Agent はプライマリ・データベースのネイティブな date/time データ型のすべてデータを Sybase *datetime* 形式に変換します。

false – Replication Agent はプライマリ・データベースのネイティブな date および time データ型のデータを文字列として複製します。

#### コメント

- **pdb\_convert\_datetime** パラメータは、以前のバージョンの Replication Agents および Replication Server の下位互換性を保つために提供されています。Replication Server バージョン 12.0 以降を使用している場合は、すべてのデータ型変換に Replication Server 異機種データ型サポート (HDS) 機能を使用します。
- **rs\_create\_repdef** コマンドで作成された複製定義では、常に Replication Server にインストールされた使用可能なユーザ定義データ型 (UDD) を使用してデータ型が定義されます。**rs\_create\_repdef** コマンドを使用する場合は、Replication Agent 設定パラメータ **pdb\_convert\_datetime** を true に設定しないでください。このように設定すると、**date** および **timestamp** データ型が UDD 形式でなく Sybase 形式に変換されます。
- Replication Agent は、オブジェクトに複製のマークが付けられるとき、**pdb\_convert\_datetime** パラメータの値をチェックします。マーク付けされたオブジェクトの複製をサポートするトランザクション・ログ・オブジェクトが、必要な日付形式を提供するため作成されます。オブジェクトをマーク付けした後に **pdb\_convert\_datetime** パラメータの値を変更しても、マーク付けしたオブジェクトに影響はありません。マーク付けしたオブジェクトの *datetime* データ型変換を変更するには、オブジェクトのマークを解除し、**pdb\_convert\_datetime** パラメータの値を変更してから、オブジェクトに再度マーク付けする必要があります。

- ログベースの Replication Agents では、ログ・レコードが読み込まれてから Replication Server に送信する LTL が生成されるまでの間に変換が行われます。
- Sybase 以外の date/time データ型形式で欠落しているコンポーネントは、Sybase の *datetime* 形式に変換されるときに暗黙に 0 (ゼロ) として扱われます。
- **pdb\_convert\_datetime** パラメータの値が true の場合、各テーブルの複写定義では、すべての date/time カラムの宣言したデータ型が *datetime* であることを指定する必要があります。
- **pdb\_convert\_datetime** パラメータの値が false の場合、Replication Agent によって date/time データがプライマリ Replication Server に文字列として送信されます。文字列のサイズは、データベースとデータ型によって次のように変化します。
  - Oracle : *DATE* = *char*(19)
  - Microsoft SQL Server : *datetime* または *smalldatetime* = *char*(23)、*timestamp* = *binary*(8)
  - DB2 Universal Database : *DATE* = *char*(10)、*TIME* = *char*(8)、*TIMESTAMP* に対しては、文字列サイズはユーザ入力と同じになります。
- **pdb\_convert\_datetime** パラメータの値を true に設定します。これは、すべてのプライマリ・データベースからレプリケートされた date/time 値が Sybase *datetime* データ型としてレプリケートされる場合に必要な設定です。
- **pdb\_convert\_datetime** は、レプリケートされた LOB カラムのあるテーブルに、プライマリ・キーで *datetime* データ型がある場合には false に設定されている必要があります。
- Replication Agent date/time データ型変換は、次のいずれかの条件が存在しない限り LOB カラム複写では使用できません (これらの条件は Oracle および Microsoft SQL Server では必須ではありません)。
  - LOB カラムの複写が有効になったテーブル内に date/time カラムがないこと
  - LOB カラムの複写が有効になったテーブル内のプライマリ・キーに date/time データ型が含まれないこと

それ以外の場合には、**pdb\_setrepcol** コマンドを LOB カラム複写を有効にするために使用する場合は、**pdb\_convert\_datetime** パラメータの値を false に設定する必要があります。

Replication Agent for UDB は、プライマリ・データベースから LOB データを直接問い合わせる唯一の Replication Agent です。プライマリ・データベース・テーブルで LOB カラム値を正常に問い合わせるには、データ値のクエリ内にデータ値が適切に表示されるよう、データ・カラム値がプライマリ・データベースの形式および構造を維持する必要があります。日付値の形式と構造は Sybase の *datetime* 形式に変換できません。

- Replication Agent のスループット・パフォーマンスを向上し、最適なデータ型処理を実現するには、**pdb\_convert\_datetime** パラメータの値を false に設定します。

## 設定パラメータ

- **pdb\_convert\_datetime** が true の場合に DB2 Universal Database でパラメータ化された *TIMESTAMP* 値がある場合、結果として得られる *datetime* 値は、*TIMESTAMP* 精度パラメータの値に関わらず、精度 3 となります。たとえば、*TIMESTAMP(0)* が値 2012-05-20 10:15:45 に対して使用された場合、*datetime* 値は 2012-05-20 10:15:45:000 となります。*TIMESTAMP(12)* が値 2012-05-20 10:15:45:123456789012 に対して使用された場合、*datetime* 値は 2012-05-20 10:15:45:123 となります。

### **pdb\_dflt\_column\_repl**

テーブルがマーク付けされている場合に、LOB カラムの複写をデフォルトで有効化するかどうかを決定します。

デフォルト値  
**false**

値

**true** – テーブルがマーク付けされている場合に、LOB カラムの複写がデフォルト (自動で) 有効となります。

**false** – テーブルがマーク付けされている場合に、LOB カラムの複写をデフォルトで無効化されます。

コメント

- テーブルに複写のマーク付けがあるときに **pdb\_dflt\_column\_repl** パラメータの値が **false** の場合、テーブル内の LOB カラムに影響を及ぼすトランザクションはまったく複写できません。複写できるようにするには、**pdb\_setrepcol** コマンドで明示的に有効にする必要があります。
- **pdb\_setrepcol** コマンドを使用すると、すべてのマーク付きテーブル内のすべての LOB カラムの複写を一度に有効または無効にできます。
- 複写が LOB カラムに対して無効になっている場合、複写が有効化されているそれ以外のカラムに影響が及ぶ操作の場合であっても、LOB カラムに影響を及ぼす操作はトランザクション・ログにまったく記録されません。

### **pdb\_dflt\_object\_repl**

(Oracle および UDB のみ) オブジェクト (テーブルまたはストアド・プロシージャ) がマーク付けされている場合に複写をデフォルトで有効にするかどうかを決定します。

---

**注意：** このパラメータは Oracle および UDB でのみ使用可能です。

---



### デフォルト値

**true**

### 値

**true** – オブジェクトがマーク付けされている場合に、複製がデフォルト (自動で) 有効になります。

**false** – オブジェクトがマーク付けされている場合に、デフォルトで複製を無効にします。

### コメント

- テーブルに複製のマーク付けがあるときに **pdb\_dflt\_object\_repl** パラメータの値が **false** の場合、そのテーブルのトランザクションはまったく複製できません。複製できるようにするには、**pdb\_setreptable** コマンドで明示的に有効にする必要があります。
- ストアド・プロシージャに複製のマーク付けがあるときに **pdb\_dflt\_object\_repl** パラメータの値が **false** の場合、そのストアド・プロシージャの呼び出しはまったく複製できません。複製できるようにするには、**pdb\_setrepproc** コマンドで明示的に有効にする必要があります。
- **pdb\_setrepproc** コマンドまたは **pdb\_setreptable** コマンドを使用すると、すべてのマーク付きのストアド・プロシージャまたはテーブルの複製を一度に有効または無効にできます。
- テーブルに対して複製が無効になっている場合、そのテーブルに影響を及ぼす操作はトランザクション・ログにまったく記録されません。
- ストアド・プロシージャに対して複製が無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はトランザクション・ログにまったく記録されません。
- Microsoft SQL Server の場合、**pdb\_setrepproc** コマンドは **pdb\_dflt\_object\_repl** パラメータを無視します。その結果、**pdb\_setrepproc** は、オブジェクトがマーク付けされているときに、複製はデフォルトで常に有効になります。

## pdb\_ignore\_unsupported\_anydata

(Oracle のみ) Replication Agent が **ANYDATA** 型のカラムに格納された対応していないデータ型のデータを無視するかどうかを決定します。

### デフォルト値

**false**

### 値

**true** – Replication Agent は、**ANYDATA** 型のカラムに格納された対応していないデータ型のデータを Replication Server に送信せず無視します。

false – Replication Agent は、対応していないデータ型を文字列で Replication Server に送信します (*ANYDATA* 型のカラムに格納された対応していないデータ型のデータがある場合)。これによって Replication Server では障害が発生します。この障害が発生したら、Replication Server またはレプリケート・データベースで、対応していないデータ型のデータを格納している各テーブル行を修正する必要があります。

### コメント

- Replication Agent は、*ANYDATA* 型のカラムに格納された次の Oracle データ型のデータは複製しません。
  - *BFILE*
  - *NESTED TABLE*
  - *REF*
  - *UROWID*
  - *VARRAY*
- Replication Agent は、オブジェクトに複製のためのマークが付けられているときのみに **pdb\_ignore\_unsupported\_anydata** の設定をチェックします。**pdb\_ignore\_unsupported\_anydata** をリセットし、Replication Agent のマーク付けされたオブジェクトに対する動作を変更する場合、**pdb\_ignore\_unsupported\_anydata** をリセットする前にオブジェクトに付けられたマークを外す必要があります。**pdb\_ignore\_unsupported\_anydata** への変更は、オブジェクトに再度マーク付けすると有効になります。
- **pdb\_ignore\_unsupported\_anydata** が true に設定され、複写テーブルに *ANYDATA* カラムに対応するデフォルトのカラム値がある場合、プライマリ・テーブルおよび複写テーブルに矛盾が発生します。複写テーブルに *ANYDATA* カラムに対応するデフォルトのカラム値がない場合は、**pdb\_ignore\_unsupported\_anydata** が true に設定されている場合でも、Replication Server では障害が発生します。
- *ANYDATA* カラムにあるサポート対象外のデータ型のデータが原因で発生した Replication Server の障害からリカバリするには、以下のいずれかを実行してください。
  - 対応する複写テーブルを変更し、テーブル内の *ANYDATA* 型のカラムがデフォルト値を持つようにする。この手法は、**pdb\_ignore\_unsupported\_anydata** が true に設定されている場合のみに使用できます。
  - 対応する複写テーブル内にトリガを作成し、*ANYDATA* 型のカラムにデフォルト値を指定するようにする。この手法は、**pdb\_ignore\_unsupported\_anydata** が true に設定されている場合のみに使用できます。

- Replication Server の関数文字列をカスタマイズし、複製テーブル内の `ANYDATA` 型のカラムにデフォルト値を指定するようにする。

## pdb\_include\_archives

(Oracle のみ) Oracle アーカイブ・ログ・ファイルの使用を有効または無効にします。

---

**注意：** このパラメータは Oracle でのみ使用可能です。

---

デフォルト値

**true**

値

**true** – `pdb_archive_path` により指定されたパスからアーカイブされた Oracle の redo ログ・ファイルの読み取りを許可します。このモードでは、Oracle の自動アーカイブの設定がサポートされています。`pdb_archive_remove` を使用すると、複製をサポートする必要がなくなった古いアーカイブ・ログを削除できます。

**false** – オンライン redo ログ・ファイルのみが読み込まれます。Oracle の自動アーカイブを無効にする必要があります。Replication Agent は、Oracle のアーカイブ・コマンドを実行して、複製のために必要でなくなった redo ログをアーカイブしません。

コメント

- アーカイブ・ログの使用が優先する場合、または Oracle で自動アーカイブが実行されるように設定しなければならない場合は、**true** に設定します。オンライン redo ログのみへのアクセスが優先する場合は、この値を **false** に設定します。
- オンライン redo ログのみの使用が優先する場合は、この値を **false** に設定します。

参照：

- `pdb_archive_path` (206 ページ)
- `truncation_interval` (256 ページ)
- `truncation_type` (257 ページ)

## pdb\_skip\_missing\_user

Replication Agent が一致するセッション情報またはユーザ情報のないコマンドの処理をスキップするかどうかを決定します。

---

**注意：** このパラメータは Oracle でのみ使用可能です。

---

デフォルト値

**false**

値

**true** – Replication Agent は一致するセッション情報またはユーザ情報のないコマンドの処理をスキップします。スキップされた記録を識別するメッセージがログに記録されます。

**false** – Replication Agent は一致するセッション情報またはユーザ情報のないコマンドの処理を続行します。デフォルトのセッションユーザは、「欠落」として送信されます。

### **pdb\_support\_large\_identifier**

Replication Server 12.6 以降で、最高 255 文字までの長さの長い識別子の複写をサポートします。

デフォルト値

**false**

値

**true** – 長い識別子のあるオブジェクトに複写のためのマーク付けができます。

**false** – 長い識別子のあるオブジェクトは、複写のためのマークを付けることができません。

コメント

- オブジェクト (テーブル/プロシージャ/関数) に複写のマーク付けがされているときに **pdb\_support\_large\_identifier** の値が **false** の場合、オブジェクトに 30 文字を超える識別子があるかどうかチェックされます。オブジェクトに 30 文字を超える識別子がある場合、エラーが返され、オブジェクトの複写のためのマークがはずされます。
- このパラメータは、使用されている Replication Server がバージョン 12.6 以降の場合でレプリケート・データベースが長い識別子をサポートできる場合に **true** に設定できます。
- **pdb\_support\_large\_identifier** が **true** に設定されている場合、複写用のマークが付けられているオブジェクトに 30 文字を超える識別子があるかどうかはチェックされません。

## pdb\_timezone\_file

(Oracle のみ) Oracle タイム・ゾーン情報を取得するために Replication Agent 初期化時に読み取るファイルを指定します。

---

**注意：** このパラメータは Oracle でのみ使用可能です。

---

デフォルト値  
<設定なし>

値  
timezone ファイル名を含む Oracle タイム・ゾーン・ファイルへの有効なパス。

コメント

- 値が指定されていない場合、デフォルトの Oracle のインストール `oracore/zoneinfo/timezone.dat` ファイルが使用されます。次に例を示します。  
`$ORACLE_HOME/oracore/zoneinfo/timezone.dat`
- 指定する `timezone` ファイルは、プライマリ Oracle データベースと同じリリースおよびプラットフォームのものを指定する必要があります。たとえば、Oracle 9i `timezone` ファイルは、Oracle 10g プライマリ・データベースとの互換性はありません。また、Windows `timezone` ファイルは、Oracle の UNIX バージョンとの互換性はありません。

## pdb\_xlog\_device

Replication Agent トランザクション・ログ・オブジェクトを作成するプライマリ・データベース・デバイス。

---

**注意：** このパラメータは廃止されました。代わりに `ra_admin_device` パラメータを使用してください。

---

デフォルト値  
NULL

値  
有効なプライマリ・データベース・デバイス名または NULL。

コメント

- `pdb_xlog_device` パラメータの値は、トランザクション・ログ・オブジェクトを作成するために Replication Agent によって生成された SQL スクリプトで使用されるプライマリ・データベース・デバイスのデバイス指定です。

## 設定パラメータ

- **pdb\_xlog\_device** パラメータを使用すると、データベースが複数のデバイスを使用する場合でも、すべての Replication Agent トランザクション・ログ・オブジェクトを作成する単一のデバイスを指定できます。
- **pdb\_xlog\_device** パラメータの値が **NULL** の場合は、SQL **create** 文で指定されるデバイスはなく、Replication Agent のトランザクション・ログ・オブジェクトは、プライマリ・データ・サーバのシステム定義によるデフォルトのデバイス上に配置されます。

### 参照：

- ra\_admin\_device (230 ページ)

## pdb\_xlog\_prefix

Replication Agent トランザクション・ログ・オブジェクトを識別するためにデータベース・オブジェクト名に使用されているプレフィクス文字列。

---

**注意：** このパラメータは廃止されました。代わりに **ra\_admin\_instance\_prefix** パラメータを使用してください。

---

### デフォルト値

ra\_

### 値

1～3 文字の文字列。

### コメント

- Replication Agent は、プライマリ・データベースにトランザクション・ログ・コンポーネントのデータベース・オブジェクト名を生成するときに、**pdb\_xlog\_prefix** パラメータの値をオブジェクト名プレフィクスとして使用します。
- Replication Agent は、**pdb\_xlog\_prefix** パラメータの値を使用してプライマリ・データベース内のトランザクション・ログ・オブジェクトを認識します。したがって、トランザクション・ログ・オブジェクトの作成後に **pdb\_xlog\_prefix** パラメータの値を変更すると、Replication Agent はそのトランザクション・ログ・オブジェクトを検出できなくなります。
- **pdb\_xlog\_prefix\_chars** パラメータの値は、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を指定します。

### 参照：

- ra\_admin\_instance\_prefix (230 ページ)

## pdb\_xlog\_prefix\_chars

Replication Agent のトランザクション・ログ・オブジェクトを識別するデータベース・オブジェクト名の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。

---

**注意：** このパラメータは廃止されました。代わりに **ra\_admin\_prefix\_chars** パラメータを使用してください。

---

デフォルト値

**#@&1234567890**

値

セパレータがない文字列。

コメント

- **pdb\_xlog\_prefix\_chars** パラメータのデフォルト値は、Replication Agent インスタンスが作成されたプライマリ・データベースのタイプによって異なります。デフォルト値は、Sybase 以外の各データベースによって使用が許可されている標準のアルファベット以外の文字に基づきます。
- **pdb\_xlog\_prefix\_chars** パラメータの値を設定または変更すると、以前の値に新しい値を追加または付属させるのではなく、既存の値は新しい値によって置き換えられます。
- **ra\_config** コマンドを使って **pdb\_xlog\_prefix** パラメータの値を設定する場合、コマンド行で指定されているアルファベット以外のすべての文字は **pdb\_xlog\_prefix\_chars** パラメータの値で検証されます。
- アルファベット文字 **a ~ z** は、**pdb\_xlog\_prefix** パラメータで常に有効であるため、指定する必要はありません。
- Replication Agent は区切られたトランザクション・ログ・オブジェクト名をサポートしないので、**pdb\_xlog\_prefix** パラメータの値にスペース文字を使用することはできません。
- **pdb\_xlog\_prefix\_chars** パラメータに指定された値が検証されていません。含めることができる文字に制限はありません。

---

**注意：** プライマリ・データ・サーバによって、データベース・オブジェクト名の特定の位置に使用できる文字が制限されることがあります。詳細については、お使いのプライマリ・データ・サーバのマニュアルを参照してください。

---

参照：

- **ra\_admin\_prefix\_chars** (232 ページ)

## **pds\_connection\_type**

プライマリ・データベース・コネクションで使用するコネクティビティ・ドライバのタイプ。

### デフォルト値

Replication Agent のインスタンスが作成されると、以下のいずれかの値が自動的に設定されます。

### 値

**MSSQLJDBC** – Replication Agent は、プライマリ Microsoft SQL Server データベースへの接続に Microsoft SQL Server JDBC™ ドライバを使用します。

**ORAJDBC** – Replication Agent は、プライマリ Oracle データベースへの接続に Oracle JDBC ドライバを使用します。

**UDBJDBC** – Replication Agent は、DB2 Universal Database のプライマリ・データベースへの接続に DB2 Universal Database JDBC ドライバを使用します。

### コメント

- **pds\_connection\_type** パラメータの値は、Replication Agent インスタンスの作成時に自動的に設定されます。指定される値は、作成された Replication Agent インスタンスの種類によって異なります。

---

**注意：** **pds\_connection\_type** パラメータのデフォルト値は変更しないでください。

---

- **pds\_connection\_type** パラメータの値は、そのプライマリ・データベース・コネクションに関連付けられている他の複数の Replication Agent の設定パラメータのうちどのパラメータの値が指定されていないかにも決定します。
  - **MSSQLJDBC** は、以下のパラメータの対応する値を必要とします。
    - **pds\_server\_name**
    - **pds\_port\_number**
    - **pds\_database\_name**
  - **UDBJDBC** は、以下のパラメータの対応する値を必要とします。
    - **pds\_host\_name**
    - **pds\_port\_number**
    - **pds\_database\_name**
    - **pds\_datasource\_name**
  - **ORAJDBC** は、以下のパラメータの対応する値を必要とします。
    - **pds\_host\_name**



- **pds\_port\_number**
- **pds\_database\_name**
- tnsnames.ora ファイルでは、以下のパラメータが必要となります。
  - **pds\_tns\_filename**
  - **pds\_tns\_connection**
- **pds\_connection\_type** パラメータの値は、Replication Agent インスタンスの作成時に自動的に設定されます。

### **pds\_dac\_port\_number**

(Microsoft SQL Server のみ) Microsoft SQL Server エラー・ログの専用管理コネクション (DAC) ポート番号。

---

**注意：** このパラメータは Microsoft SQL Server でのみ使用可能です。

---

デフォルト値  
1111

値  
Microsoft SQL Server 用の有効なポート番号。

コメント

- このパラメータは、Microsoft SQL Server に対して **pdb\_xlog init** コマンドを初めて実行するときが必要となります。mssqlsystemresource データベースに対する変更のインストールをサポートしている必要があります。
- **pds\_dac\_port\_number** パラメータの使用の詳細については、『Replication Agent プライマリ・データベース・ガイド』の「プライマリ・データ・サーバと Replication Agent の初期化」および「初めての初期化」にある「Microsoft SQL Server」の章を参照してください。

参照：

- pdb\_xlog (75 ページ)

### **pds\_database\_name**

プライマリ・データベースの名前です。

デフォルト値  
<設定なし>

値  
有効なデータベース名。

---

**注意：** Oracle では、 **pds\_tns\_connection** が設定されている場合、**pds\_database\_name** は設定できません。

---

#### コメント

- **pds\_database\_name** パラメータの値は、プライマリ・データ・サーバ上のプライマリ・データベースの名前です。

---

**注意：** プライマリ・データ・サーバの中には、データ・サーバの単一のインスタンス内での複数データベースをサポートしていないものもあります。この場合、**pds\_database\_name** パラメータの値は、データ・サーバ・インスタンスの名前である必要があります。

---

- Replication Agent コネクション設定パラメータの詳細については、『Replication Agent 管理ガイド』を参照してください。

### pds\_datasource\_name

(Microsoft SQL Server および UDB のみ) プライマリ・データベース・コネクションで使用されるコネクティビティ・ドライバに対して指定されるデータ・ソース名 (DSN)。

---

**注意：** このパラメータは UDB でのみ使用可能です。

---

#### デフォルト値

<設定なし>

#### 値

有効な ODBC データ・ソース名。

#### コメント

- **pds\_datasource\_name** パラメータの値は、プライマリ・データベース・コネクション上のコネクティビティ・ドライバのデータ・ソース名 (DSN) です。
- **pds\_connection\_type** パラメータの値は **UDBJDBC**、**pds\_datasource\_name** パラメータの値は DB2 Universal Database サーバのプライマリ・データベースのデータベース・エイリアスである必要があります。
- Replication Agent コネクション設定パラメータの詳細については、『Replication Agent 管理ガイド』を参照してください。

### pds\_host\_name

プライマリ・データ・サーバのホスト・マシンの名前。

#### デフォルト値

<設定なし>

*値*

有効なホスト名。

---

**注意：** `pds_host_name` は、`pds_tns_connection` が設定されている場合は設定できません。

---

*コメント*

- `pds_host_name` パラメータの値は、プライマリ・データ・サーバが常駐するホスト・マシンのネットワーク名です。
- Replication Agent コネクション設定パラメータの詳細については、『Replication Agent 管理ガイド』を参照してください。

## pds\_integrated\_security

(Microsoft SQL Server のみ) プライマリ SQL Server への接続時に Replication Agent で Windows 認証を使用するかどうかを決定します。

---

**注意：** このパラメータは Microsoft SQL Server でのみ使用可能です。

---

*デフォルト値*

**false**

*値*

**true** – Replication Agent は Windows 認証を使用してプライマリ SQL Server に接続する必要があります。

**true** – Replication Agent が SQL Server 認証を使用してプライマリ SQL Server に接続する必要があります (デフォルト)。

*コメント*

- `pds_integrated_security` のデフォルト値は、Replication Agent インスタンスの作成時に **false** に設定されます。このデフォルトの設定では、`pds_password` パラメータを設定する必要があります。また、プライマリ SQL Server が SQL Server 認証を許容するように設定されている必要があります。
- この値を **true** に設定すると、Replication Agent は Windows 認証を使用してプライマリ SQL Server に接続します。Windows 環境およびプライマリ SQL Server は、Windows 認証を使用するように設定されている必要があります。詳細については、『Replication Agent プライマリ・データベース・ガイド』の「Microsoft SQL Server」のセクションを参照してください。

## **pds\_password**

Replication Agent がプライマリ・データ・サーバへのアクセスに使用するパスワード。

デフォルト値  
"" (空の文字列)

値  
有効なパスワード。

コメント

- **pds\_password** パラメータの値は、Replication Agent がプライマリ・データ・サーバにアクセスするために使用するユーザ・ログイン名のパスワードです。
- **pds\_password** パラメータの値は、Replication Agent インスタンス設定ファイルで暗号化されています。
- Replication Agent コネクション設定パラメータの詳細については、『Replication Agent 管理ガイド』を参照してください。

## **pds\_port\_number**

プライマリ・データ・サーバのクライアント・ポート番号。

デフォルト値  
1111

値  
プライマリ・データ・サーバ・ホスト・マシン上の有効なポート番号。

---

**注意：** **pds\_port\_number** は、**pds\_tns\_connection** が設定されている場合は設定できません。

---

コメント

- **pds\_port\_number** パラメータの値は、プライマリ・データ・サーバのクライアント・ポート番号です。
- Replication Agent コネクション設定パラメータの詳細については、『Replication Agent 管理ガイド』を参照してください。

## pds\_retry\_count

Replication Agent がプライマリ・データベースへのコネクション確立を試行する回数。

デフォルト値

5

値

0 ～ 2,147,483,647 の整数。

コメント

- **pds\_retry\_count** パラメータの値は、Replication Agent が、コネクションの障害発生後に、プライマリ・データベースへのネットワーク接続を試行する回数です。
- このパラメータを **5** に設定することをおすすめします。
- Replication Agent コネクション設定パラメータの詳細については、『Replication Agent 管理ガイド』を参照してください。

## pds\_retry\_timeout

Replication Agent がプライマリ・データベースへの接続を試行する間隔 (秒数)。

デフォルト値

10

値

0 ～ 3600 の整数。

コメント

- **pds\_retry\_timeout** パラメータの値は、Replication Agent が、コネクションの障害発生後に、プライマリ・データベースへのネットワーク接続の確立を再試行する間隔を示す秒数です。
- Replication Agent コネクション設定パラメータの詳細については、『Replication Agent 管理ガイド』を参照してください。

## pds\_server\_name

プライマリ・データ・サーバのサーバ名。

---

**注意：** このパラメータは Microsoft SQL Server でのみ使用可能です。

---

## 設定パラメータ

*デフォルト値*  
設定なし。

*値*  
有効なサーバ名。

*コメント*

- **pds\_server\_name** パラメータの値は、プライマリ・データ・サーバのサーバ名です。
- Replication Agent コネクション設定パラメータの詳細については、『Replication Agent 管理ガイド』を参照してください。

### pds\_tns\_connection

(Oracle のみ) Oracle `tnsnames.ora` ファイルでプライマリ・データベース・コネクションを識別する Oracle コネクション名。

*デフォルト値*  
設定なし。

*値*  
Oracle `tnsnames.ora` ファイルからの有効なプライマリ・データベース・コネクション名。このファイルは、**pds\_tns\_filename** 設定パラメータで指定されます。

*コメント*

- この設定パラメータを設定すると、設定パラメータ **pds\_host\_name**、**pds\_database\_name**、および **pds\_port\_number** の設定がオーバーライドされません。
- この設定パラメータは、複製される Oracle データ・サーバ・インスタンスが Real Application Cluster (RAC) 設定の一部である場合に必要となります。

参照：

- `pds_tns_filename` (228 ページ)

### pds\_tns\_filename

(Oracle のみ) プライマリ Oracle データ・サーバのコネクション・パラメータを含む Oracle `tnsnames.ora` ファイルを識別する完全修飾ファイル名。

---

**注意：** このパラメータは Oracle でのみ使用可能です。

---

デフォルト値  
設定なし。

値  
プライマリ Oracle データ・サーバへの接続・パラメータが含まれた有効な Oracle `tnsnames.ora` ファイル。このファイルは、通常、`ORACLE_HOME` `¥network¥admin` ディレクトリに常駐しています。

コメント

- Replication Agent は、設定されると、`tnsnames.ora` ファイルで指定されている接続情報を使用してプライマリ・データベースに接続し、**`pds_host_name`** および **`pds_port_number`** は無視されます。**`pds_tns_connection`** 名は、**`pds_tns_filename`** の設定時に Sybase インターフェイス・ファイル内のエントリ名に設定する必要があります。
- この設定パラメータは、複製される Oracle データ・サーバ・インスタンスが Real Application Cluster (RAC) 設定の一部である場合に設定が必要となります。

---

**警告！** Replication Agent プロセスは、このファイルの `read` パーミッションを持つ必要があります。アクセスできない場合は、Replication Agent が Oracle サーバに接続できません。

---

参照：

- `pds_tns_connection` (228 ページ)

## **pds\_username**

Replication Agent がプライマリ・データ・サーバへのアクセスに使用するユーザ・ログイン名。

デフォルト値  
<設定なし>

値  
有効なユーザ名。

コメント

- **`pds_username`** パラメータの値は、プライマリ・データ・サーバにログインするために Replication Agent が使用するユーザ・ログイン名です。  
このログイン名は、プライマリ・データベースの適切な権限または認証を使用してプライマリ・データ・サーバ内で定義されている必要があります。

## 設定パラメータ

- Replication Agent は、このログインを使用して、プライマリ・データベース・オブジェクトにアクセスしたり、そのトランザクション・ログ・オブジェクトをプライマリ・データベース内で作成、削除、管理したりします。
- **rs\_ticket** では、**pds\_username** で指定したユーザ名を Replication Server への接続に指定したユーザ ID (メンテナンス・ユーザ) と別にする必要があります。メンテナンス・ユーザの名前は、**ra\_maintid** を実行することで取得できます。
- Replication Agent コネクション設定パラメータの詳細については、『Replication Agent 管理ガイド』を参照してください。

### ra\_admin\_device

Replication Agent システム・オブジェクトを作成するプライマリ・データベース・デバイス。

デフォルト値

NULL

値

有効なプライマリ・データベース・デバイス名または NULL。

コメント

- **ra\_admin\_device** パラメータの値は、システム・オブジェクトを作成するために Replication Agent によって生成された SQL スクリプトで使用されるプライマリ・データベース・デバイスのデバイス仕様です。
- **ra\_admin\_device** パラメータを使用すると、データベースが複数のデバイスを使用する場合でも、すべての Replication Agent システム・オブジェクトを生成する単一のデバイスを指定できます。
- **ra\_admin\_device** パラメータの値が **NULL** の場合は、SQL **create** 文で指定されるデバイスはなく、Replication Agent システム・オブジェクトは、プライマリ・データ・サーバのシステム定義によるデフォルトのデバイス上に配置されます。

参照：

- **pdb\_xlog\_device** (219 ページ)

### ra\_admin\_instance\_prefix

1 つの Replication Agent インスタンスに固有の Replication Agent システム・オブジェクトを識別するために使用するプレフィクス文字列。

---

**注意：** このパラメータは Replication Agent for Oracle でのみ使用されます。

---



**デフォルト値****ra\_****値**

1～3文字の文字列。

**コメント**

- Replication Agent はプライマリ・データベースにシステム・オブジェクトを生成する際に、**ra\_admin\_instance\_prefix** パラメータをオブジェクト名プレフィクスとして使用します。
- Replication Agent はプライマリ・データベース内のシステム・オブジェクトを認識するために、**ra\_admin\_instance\_prefix** パラメータの値を使用します。したがって、システム・オブジェクトの作成後に **ra\_admin\_instance\_prefix** パラメータの値を変更すると、Replication Agent はそのオブジェクトを検出できなくなります。
- **ra\_admin\_instance\_prefix** パラメータの値は、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を指定します。
- Replication Agent グループの各 Replication Agent インスタンスは、**ra\_admin\_instance\_prefix** に対してユニークな値で設定される必要があります。

**参照：**

- pdb\_xlog\_prefix (220 ページ)
- ra\_admin\_prefix (231 ページ)

**ra\_admin\_prefix**

共有 Replication Agent システム・オブジェクトを識別するために使用するプレフィクス文字列。

---

**注意：** このパラメータは Replication Agent for Oracle でのみ使用されます。

---

**デフォルト値****ra\_****値**

1～3文字の文字列。

**コメント**

- Replication Agent はプライマリ・データベースにトランザクション・ログ・コンポーネントの共有データベース・オブジェクト名を生成する際に、

## 設定パラメータ

**ra\_admin\_instance\_prefix** パラメータの値をオブジェクト名プレフィクスとして使用します。

- **ra\_admin\_prefix** パラメータの値は、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を指定します。
- Replication Agent は、**ra\_admin\_prefix** パラメータの値を使用してプライマリ・データベース内のシステム・オブジェクトを認識します。したがって、Replication Agent インスタンスの作成後に **ra\_admin\_prefix** パラメータの値を変更すると、Replication Agent はそのオブジェクトを検出できなくなります。
- Replication Agent グループの各 Replication Agent インスタンスは、**ra\_admin\_prefix** パラメータに対して同じ値で設定される必要があります。

### 参照：

- [ra\\_admin\\_instance\\_prefix \(230 ページ\)](#)
- [ra\\_admin\\_owner \(233 ページ\)](#)

## ra\_admin\_prefix\_chars

Replication Agent システム・オブジェクトを識別するデータベース・オブジェクト名のプレフィクス文字列に許可される英字以外の文字。

### デフォルト値

**\_#@&1234567890**

### 値

セパレーターがない文字列。

### コメント

- **ra\_admin\_prefix\_chars** パラメータのデフォルト値は、Replication Agent インスタンスが設定されたプライマリ・データベースのタイプによって異なります。デフォルト値は、Sybase 以外の各データベースによって使用が許可されている標準のアルファベット以外の文字に基づきます。
- **ra\_admin\_prefix\_chars** パラメータの値を設定または変更すると、以前の値に新しい値を追加または付属させるのではなく、既存の値は新しい値によって置き換えられます。
- **ra\_config** コマンドを使って **ra\_admin\_instance\_prefix** パラメータの値を設定する場合、コマンド・ラインで指定されているアルファベット以外のすべての文字は **ra\_admin\_prefix\_chars** パラメータの値で検証されます。
- アルファベット文字 **a ~ z** は常に、**ra\_admin\_instance\_prefix** パラメータで有効とされ、指定される必要はありません。

- Replication Agent は区切られたシステム・オブジェクト名をサポートしないので、**ra\_admin\_instance\_prefix** パラメータの値にスペース文字を使用することはできません。
- **ra\_admin\_prefix\_chars** パラメータに指定された値が無効です。含めることができる文字に制限はありません。

---

**注意：** プライマリ・データ・サーバによって、データベース・オブジェクト名の特定の位置に使用できる文字が制限されることがあります。詳細については、お使いのプライマリ・データ・サーバのマニュアルを参照してください。

---

**参照：**

- `pdb_xlog_prefix_chars` (221 ページ)

## **ra\_admin\_owner**

共有システム・オブジェクトおよびインスタンス固有のシステム・オブジェクトを含むすべての Replication Agent システム・オブジェクトの所有者。

---

**注意：** このパラメータは Replication Agent for Oracle でのみ使用されます。

---

*デフォルト値*

<設定なし>

*値*

有効なユーザ名。

*コメント*

- **ra\_admin\_owner** が設定されていない場合は、**pds\_username** パラメータの値が使用されます。
- **ra\_admin\_owner** で指定されるユーザ名は、プライマリ・データ・サーバで定義されなければなりません。
- Replication Agent インスタンスが作成された後に、**ra\_admin\_owner** パラメータの値を変更することはできません。
- Replication Agent グループの各 Replication Agent インスタンスは、**ra\_admin\_owner** パラメータの値に対して同じ値で設定される必要があります。

**参照：**

- `ra_admin_instance_prefix` (230 ページ)
- `ra_admin_prefix` (231 ページ)

## ra\_retry\_count

障害発生後に、Replication Agent が複製の再起動を試みる回数。

デフォルト値

2

値

0 より大きい整数。

コメント

- **ra\_retry\_count** パラメータの値は、Log Transfer Manager コンポーネントが Replication Agent のインスタンスを *Replicating* 状態に戻すことを試みる回数 (*Replication Down* 状態になる原因となった障害またはエラーの発生後)。
- ネットワーク接続に障害が発生すると、Replication Agent は、その接続の接続設定パラメータ内に格納されている値を使用して接続の再確立を試みます。
- Replication Agent が **pds\_retry\_count** パラメータまたは **rs\_retry\_count** パラメータで指定されている回数の再試行を行っても再度接続を確立できない場合、Replication Agent のインスタンスは *Replication Down* の状態に移行し、Log Transfer Manager コンポーネントによって Replication Agent のインスタンスを *Replicating* 状態に戻すための試みが **ra\_retry\_count** パラメータおよび **ra\_retry\_timeout** パラメータの設定に基づいて実行されます。

## ra\_retry\_timeout

障害発生後に、Replication Agent が複製の再起動を行う場合に再起動の試行と試行の間隔の秒数。

デフォルト値

10

値

0 より大きい整数。

コメント

**ra\_retry\_timeout** パラメータの値は、Log Transfer Manager コンポーネントが Replication Agent のインスタンスを *Replicating* 状態に戻すための試行と試行の間の秒数で示された間隔 (*Replication Down* 状態になる原因となった障害またはエラーの発生後)。

## rasd\_backup\_dir

(Oracle と Microsoft SQL Server のみ) Replication Agent システム・データベース (RASD) バックアップ・ファイルへのディレクトリ・パス。

---

**注意：** このパラメータは Oracle および Microsoft SQL Server でのみ使用可能です。

---

### デフォルト値

Replication Agent インスタンスの作成時に自動的に作成された RASD backup ディレクトリへのパス。例：

- Microsoft Windows プラットフォームの場合：

```
%SYBASE%\RAX-15_5\inst_name\repository\backup
```

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 プラットフォームの場合：

```
$(SYBASE)/RAX-15_5/inst_name/repository/backup
```

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。

### 値

Replication Agent ホスト・マシンの有効なパス。

### コメント

- Replication Agent のインスタンスを作成すると、RASD backup ディレクトリがインスタンスのディレクトリ構造の一部として自動的に作成されます。  
`rasd_backup_dir` パラメータのデフォルト値は、そのディレクトリを指します。
- `rasd_backup_dir` パラメータの値として有効なパスを指定すると、Replication Agent は、RASD のバックアップ操作中に RASD バックアップ・ファイルをそのディレクトリ内に配置します。また、リストア操作中には RASD バックアップ・ファイルをそのディレクトリに探しに行きます。

## rasd\_database

(Oracle と Microsoft SQL Server のみ) Replication Agent システム・データベース (RASD) データベース・ファイルへのディレクトリ・パス。

---

**注意：** このパラメータは Oracle および Microsoft SQL Server でのみ使用可能です。

---

## 設定パラメータ

### デフォルト値

Replication Agent インスタンスの作成時に自動的に作成された RASD データベース・ファイルへのパス。例：

- Microsoft Windows プラットフォームの場合：

```
%SYBASE%\RAX-15_5\inst_name\repository\inst_name.db
```

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 プラットフォームの場合：

```
$(SYBASE)/RAX-15_5/inst_name/repository/inst_name.db
```

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。

### 値

Replication Agent ホスト・マシン上の有効なパスおよび RASD データベース・ファイル名。

### コメント

- Replication Agent インスタンスを作成すると、`repository` ディレクトリおよび RASD データベース・ファイルが自動的に作成されます。`rasd_database` パラメータのデフォルト値は、そのディレクトリ内の RASD データベース・ファイルを指します。
- `rasd_database` パラメータの値として有効なパスおよび RASD データベース・ファイル名を指定すると、Replication Agent インスタンスは、そのディレクトリで指定されたファイル名のある RASD データベース・ファイルを探します。

## rasd\_mirror\_tran\_log

(Oracle と Microsoft SQL Server のみ) Replication Agent システム・データベース (RASD) のトランザクション・ログのミラーリングを有効または無効にします。

---

**注意：** このパラメータは Oracle および Microsoft SQL Server でのみ使用可能です。

---

### デフォルト値

**false**

**値**

**true** – RASD トランザクション・ログの別のファイルへのミラーリングを有効にします。

**false** – RASD トランザクション・ログのミラーリングを無効にします。

**コメント**

**rasd\_mirror\_tran\_log** パラメータの値を **true** に設定すると、Replication Agent ホスト・マシン上のデバイスで障害が発生した場合に追加のリカバリ・オプションが使用できます。

**rasd\_trace\_log\_dir**

(Oracle と Microsoft SQL Server のみ) Replication Agent システム・データベース (RASD) トレース・ログ・ファイルへのディレクトリ・パス。

**注意：** このパラメータは Oracle および Microsoft SQL Server でのみ使用可能です。

**デフォルト値**

Replication Agent インスタンスの作成時に自動的に作成された RASD repository ディレクトリへのパス。例：

- Microsoft Windows プラットフォームの場合：

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

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 プラットフォームの場合：

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

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。

**値**

Replication Agent ホスト・マシンの有効なパス。

**コメント**

- Replication Agent インスタンスを作成すると、repository ディレクトリがインスタンスのディレクトリ構造の一部として自動的に作成されます。

## 設定パラメータ

**rasd\_trace\_log\_dir** パラメータのデフォルト値は、そのディレクトリを指します。

- **rasd\_trace\_log\_dir** パラメータの値として任意の有効なディレクトリ・パスが指定されている場合、Replication Agent インスタンスは、そのインスタンスの RASD トレース・ログ・ファイルをそのディレクトリに書き込みます。

## rasd\_tran\_log

(Oracle と Microsoft SQL Server のみ) Replication Agent システム・データベース (RASD) トランザクション・ログ・ファイルへのディレクトリ・パス。

---

**注意：** このパラメータは Oracle および Microsoft SQL Server でのみ使用可能です。

---

### デフォルト値

Replication Agent インスタンスの作成時に自動的に作成された RASD トランザクション・ログ・ファイルへのパス。例：

- Microsoft Windows プラットフォームの場合：

```
%SYBASE%\RAX-15_5\inst_name\repository\inst_name.log
```

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 プラットフォームの場合：

```
$(SYBASE)/RAX-15_5/inst_name/repository/inst_name.log
```

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。

### 値

Replication Agent ホスト・マシンの有効なパス。

### コメント

- Replication Agent インスタンスを作成すると、repository ディレクトリおよび RASD トランザクション・ログ・ファイルが自動的に作成されます。  
**rasd\_tran\_log** パラメータのデフォルト値は、そのトランザクション・ログ・ファイルを指します。
- **rasd\_tran\_log** パラメータの値として有効なパスおよび RASD トランザクション・ログ・ファイル名を指定すると、Replication Agent インスタンスは、そのディレクトリで指定されたファイル名のある RASD トランザクション・ログ・ファイルを探します。



## rasd\_tran\_log\_mirror

(Oracle と Microsoft SQL Server のみ) Replication Agent システム・データベース (RASD) トランザクション・ログ・ファイル・ミラーへのディレクトリ・パス。

---

**注意：** このパラメータは Oracle および Microsoft SQL Server でのみ使用可能です。

---

### デフォルト値

Replication Agent インスタンスの作成時に tran\_log\_mirror ディレクトリに自動的に作成された RASD トランザクション・ログ・ファイル・ミラーへのパス。  
例：

- Microsoft Windows プラットフォームの場合：

```
%SYBASE%\RAX-15_5\inst_name\repository\tran_log_mirror
\inst_name.log
```

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 プラットフォームの場合：

```
$SYBASE/RAX-15_5/inst_name/repository/tran_log_mirror/
inst_name.log
```

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。

### 値

Replication Agent ホスト・マシンの有効なパス。

### コメント

`rasd_tran_log_mirror` パラメータの値として有効なパスおよび RASD トランザクション・ログ・ファイル名を指定すると、Replication Agent インスタンスは、そのディレクトリで指定されたファイル名のある RASD トランザクション・ログ・ファイル・ミラーを探します。

## ra\_standby

(Oracle のみ) Replication Agent がスタンバイ・モードで機能するかどうかを決定します。

---

**注意：** このパラメータは Oracle でのみ使用可能です。

---

デフォルト値

**false**

値

**true** – Replication Agent はスタンバイ・モードで機能します。

**false** – Replication Agent はノーマル・モードで機能します。

コメント

- スタンバイ・モードでは、Replication Agent は次のように動作します。
  - トランザクション・ログをスキャンし、Replication Agent システム・データベース (RASD) を最新の状態に維持します。
  - ログ転送言語 (LTL: Log Transfer Language) を Replication Server に一切送信しません。
  - ログ・トランケーションの実行を継続します。
- Replication Agent がスタンバイ・モードで機能するための条件は以下のとおりです。
  - **rs\_source\_ds** パラメータおよび **rs\_source\_db** パラメータが Replication Server への物理コネクションとして設定されている必要があります。
  - DDL 文の複写を必要に応じて **pdb\_setrep\_ddl** コマンドを使用して有効または無効にします。
  - **pdb\_auto\_create\_repdefs** パラメータ、**pdb\_dflt\_column\_repl** パラメータ、**pdb\_dflt\_object\_repl** パラメータ、および **pdb\_automark\_tables** パラメータが **true** に設定されている必要があります。

参照：

- **rs\_source\_ds** (248 ページ)
- **rs\_source\_db** (248 ページ)
- **pdb\_auto\_create\_repdefs** (208 ページ)
- **pdb\_dflt\_column\_repl** (214 ページ)
- **pdb\_dflt\_object\_repl** (214 ページ)
- **pdb\_automark\_tables** (210 ページ)

## ra\_statrack\_interval

**ra\_statrack** が統計をサンプリングする秒単位の間隔を決定します。

デフォルト値

**60**

値

5 ~ 86400 の整数。

参照：

- ra\_statistics (126 ページ)
- ra\_statrack (137 ページ)
- ra\_statrack\_list (137 ページ)

## rman\_enabled

(Oracle のみ) Replication Agent が、Oracle **RMAN** ユーティリティを使用して、古いアーカイブ・ログ・ファイルをトランケートするかどうかを指定します。

---

**注意：** このパラメータは Oracle でのみ使用可能です。

---

デフォルト値

**false**

値

**true** – Oracle **RMAN** ユーティリティを使用したアーカイブ・ログ・ファイルのトランケーションを有効にします。

**false** – Oracle **RMAN** ユーティリティを使用したアーカイブ・ログ・ファイルのトランケーションを無効にします。

コメント

- **rman\_enabled** と **pdb\_archive\_remove** の両方のパラメータが **true** に設定されている場合、Replication Agent は、Oracle **RMAN** ユーティリティを使用して、**pdb\_archive\_path** パラメータによって指定されているパスの古いアーカイブ redo ログ・ファイルを削除します。**rman\_enabled** パラメータが **false** に設定されている一方、**pdb\_archive\_remove** パラメータが **true** に設定されている場合、Replication Agent は、ファイル・システム操作を使用して、古いアーカイブ redo ログ・ファイルを **pdb\_archive\_path** パラメータによって指定されたパスから削除します。
- **rman\_enabled** パラメータが **true** に設定されている場合、**ORACLE\_HOME** 環境変数が Replication Agent プロセスのランタイム・コンテキストに設定され、かつ **rman\_username** パラメータおよび **rman\_password** パラメータが適切に設定されている必要があります。

## 設定パラメータ

- Oracle **RMAN** ユーティリティは、Replication Agent と同じマシン上にインストールされ、トランケートするアーカイブ・ログ・ファイルを格納している Oracle データベースと互換性がある必要があります。
- **rman\_enabled** パラメータに対して加えられた変更は、Replication Agent がサスペンド後に再開されるまで有効になりません。

### 参照：

- `pdb_archive_remove` (207 ページ)
- `rman_password` (242 ページ)
- `rman_username` (242 ページ)

## rman\_password

(Oracle のみ) **rman\_username** で Oracle **RMAN** ユーティリティに接続するために使用されるパスワードが格納されています。

---

**注意：** このパラメータは Oracle でのみ使用可能です。

---

デフォルト値  
"" (空の文字列)

値  
有効なパスワード。

### コメント

- **rman\_password** 設定パラメータの設定は、**rman\_enabled** が **true** に設定されている場合のみ必要となります。
- **rman\_password** 設定パラメータの値は、Replication Agent インスタンスの設定ファイルで暗号化されています。

### 参照：

- `rman_enabled` (241 ページ)
- `rman_username` (242 ページ)

## rman\_username

(Oracle のみ) **rman\_password** で Oracle **RMAN** ユーティリティに接続するために使用されるログイン名が格納されています。

---

**注意：** このパラメータは Oracle でのみ使用可能です。

---

デフォルト値  
設定なし。

**値**

有効な Oracle ユーザ名 (**sysdba** 権限を持つ必要あり)。

**コメント**

- **rman\_username** 設定パラメータの設定は、**rman\_enabled** が **true** に設定されている場合のみ必要となります。
- **rman\_username** の値は、Replication Agent が Oracle **RMAN** ユーティリティに接続し、アーカイブ・ログ・ファイルを管理するために使用するログイン名です。このログイン名は、Oracle のプライマリ・データ・サーバで定義され、**sysdba** 権限が与えられている必要があります。**sysdba** 権限がないログイン名が **ra\_config** によって設定された場合、Replication Agent はエラーを返します。

**参照：**

- **rman\_enabled** (241 ページ)
- **rman\_password** (242 ページ)

**rs\_charset**

プライマリ Replication Server との通信で使用される文字セット。

Replication Agent のデフォルトの文字セットは、プライマリ・データベースの文字セットと一致するように設定されている必要があります。**rs\_charset** パラメータの値は、Replication Server の文字セットと一致するように設定されている必要があります。一致しない場合、Replication Agent は、Replication Server にデータを送信する前に文字セットの変換を実行します。

---

**注意：** Replication Agent が Replication Server 15.0.1 以降に接続できる場合、Replication Agent 内の **rs\_charset** は無視され、Replication Server 内の **RS\_charset** が使用されます。

---

Replication Agent の文字セットがプライマリ・データベースの文字セットと異なる場合は、**RA\_JAVA\_DFLT\_CHARSET** 環境変数を設定する必要があります。

Replication Agent の文字セットは、プライマリ・データベースの文字セットと同じである必要があります。**RA\_JAVA\_DFLT\_CHARSET** 環境変数設定の詳細については、『Replication Agent 管理ガイド』の第 2 章を参照してください。

---

**注意：** このパラメータをプライマリ Replication Server の文字セット以外の値に設定すると、Replication Agent が Replication Server へのデータ送信前に文字セット変換を実行するようになります。この変換処理は、Replication Agent のパフォーマンスを低下します。

---

### デフォルト値

デフォルトで空の文字列になります ("")。

### 値

Replication Agent ホスト・マシン上の Java VM によってサポートされている任意の有効な Sybase 文字セット。

### コメント

- 次のロケーションにある Replication Server 設定ファイルの **rs\_charset** パラメータの値と正確に同じ値を使用します (Replication Server 設定ファイルの拡張子は、.cfg です)。\$SYBASE/REP-15\_0/install/<インスタンス>.cfg。たとえば、iso\_1 と設定します。
- プライマリ・データ・サーバおよびプライマリ Replication Server を同じ文字セットを使用するように設定します。

---

**注意：** rs\_charset が複製の再開を試みる時点で設定されていない場合、Replication Agent は、エラーを返します。

---

Replication Agent インスタンスの作成時には、**rs\_charset** パラメータにはそのデフォルト値 "" (空の文字列) が設定されます。

- **rs\_charset** パラメータの値として有効な文字セットを指定した場合、Replication Agent インスタンスは、複製されたトランザクション・データをプライマリ・データベースからプライマリ Replication Server にその文字セットで送信します。
- 有効な文字セット名を **rs\_charset** パラメータの値として指定しないと (デフォルトの **rs\_charset** 値である "" を含む)、Replication Agent インスタンスによる複製は再開できません。
- **rs\_charset** の値およびシステムのデフォルトの文字セットが有効な場合でも同じ値でない場合、Replication Agent は、プライマリ Replication Server に複製されたトランザクション・データを送信する前に、そのデータをシステム定義のデータベース文字セットから Replication Server の文字セットに変換します。
- Replication Agent コネクション設定パラメータの詳細については、『Replication Agent 管理ガイド』を参照してください。

## rs\_host\_name

プライマリ Replication Server のホスト・マシンの名前。

### デフォルト値

<設定なし>

**値**

有効なホスト名。

**コメント**

- **rs\_host\_name** パラメータの値は、プライマリ Replication Server 用のホスト・マシンの名前です。
- Replication Agent コネクション設定パラメータの詳細については、『Replication Agent 管理ガイド』を参照してください。

**rs\_packet\_size**

プライマリ Replication Server へのコネクション上のネットワーク・パケット・サイズ。

**デフォルト値**

**2048**

**値**

2048 ~ 65536 の整数。

**コメント**

- **rs\_packet\_size** パラメータの値は、TCP/IP ネットワーク・プロトコルで処理できるネットワーク・パケットの最大サイズ(バイト単位)です。
- Replication Agent の **rs\_packet\_size** パラメータは、Replication Server の **rs\_packet\_size** パラメータと同等です。
- ネットワーク・パケット・サイズが小さい場合、Replication Server に指定されたデータの量を送信するため、より多くのパケットを処理する必要があります。ネットワーク・パケット・サイズが大きい場合、パケットの処理により多くのシステム・リソースが消費されます。
- **rs\_packet\_size** パラメータの最適値は、複製される標準データの性質に基づいています。標準オペレーションが大きい場合、パケット・サイズが大きい方が効率的です。
- **rs\_packet\_size** パラメータの値が大きいと、**lri\_batch\_mode** パラメータ値が **true** の場合にはより効率的です。
- Replication Agent コネクション設定パラメータの詳細については、『Replication Agent 管理ガイド』を参照してください。

## rs\_password

Replication Agent が Replication Server にアクセスするときに使用するパスワード。

デフォルト値  
"" (空の文字列)

値  
有効なパスワード。

コメント

- **rs\_password** パラメータの値は、プライマリ Replication Server にログインするために Replication Agent が使用するユーザ・ログイン名のパスワードです。
- **rs\_password** パラメータの値は、Replication Agent インスタンス設定ファイルで暗号化されています。
- Replication Agent コネクション設定パラメータの詳細については、『Replication Agent 管理ガイド』を参照してください。

## rs\_port\_number

プライマリ Replication Server のクライアント・ポート番号。

デフォルト値  
1111

値  
Replication Server ホスト・マシン上の有効なポート番号。

コメント

- **rs\_port\_number** パラメータの値は、プライマリ Replication Server のクライアント・ポート番号です。
- Replication Agent コネクション設定パラメータの詳細については、『Replication Agent 管理ガイド』を参照してください。

## rs\_replicate\_owner\_required

複写定義を生成するときに、レプリケート・テーブル句に所有者が常に含まれているかを示します。

デフォルト値  
true



値

**true** –所有者はレプリケート・テーブル句に常に含まれます。

**false** –テーブルが **on** に設定された所有者モードでマーク付けされていない限り、所有者はレプリケート・テーブル句に含まれません。

コメント

詳細については、**rs\_create\_repdef** コマンドを参照してください。

## **rs\_retry\_count**

Replication Agent がプライマリ Replication Server への接続の確立を再試行する回数。

デフォルト値

**5**

値

0 より大きい整数。

コメント

- **rs\_retry\_count** パラメータの値は、接続の失敗後に Replication Agent が Replication Server へのネットワーク・接続の確立を試行する回数です。
- このパラメータを **5** に設定することをおすすめします。
- Replication Agent 接続設定パラメータの詳細については、『Replication Agent 管理ガイド』を参照してください。

## **rs\_retry\_timeout**

Replication Agent によるプライマリ Replication Server への接続試行間の待機時間の秒数。

デフォルト値

**10**

値

0 より大きい整数。

コメント

- **rs\_retry\_timeout** パラメータの値は、接続の失敗後、Replication Agent によるプライマリ Replication Server へのネットワーク・接続確立の再試行間の待機時間です。

## 設定パラメータ

- Replication Agent コネクション設定パラメータの詳細については、『Replication Agent 管理ガイド』を参照してください。

### rs\_source\_db

Replication Server プライマリ・データベース・コネクションで識別されたデータベースの名前。

デフォルト値  
<設定なし>

値  
有効なデータベース名。

#### コメント

- **rs\_source\_db** パラメータの値は、プライマリ Replication Server がプライマリ・データベース・トランザクション・ログを認識するプライマリ・データベースの名前です。
- **rs\_source\_db** パラメータの値は、Replication Server の **create connection** コマンドでプライマリ・データベースに指定されたデータベースの名前と一致する必要があります。
- Replication Agent コネクション設定パラメータの詳細については、『Replication Agent 管理ガイド』を参照してください。

### rs\_source\_ds

Replication Server プライマリ・データベース・コネクションで識別されたデータ・サーバの名前。

デフォルト値  
<設定なし>

値  
有効なサーバ名。

#### コメント

- **rs\_source\_ds** パラメータの値は、プライマリ Replication Server がプライマリ・データベース・トランザクション・ログを識別するプライマリ・データ・サーバの名前です。
- **rs\_source\_ds** パラメータの値は、Replication Server の **create connection** コマンドでプライマリ・データベースに指定されたデータ・サーバの名前と一致する必要があります。

- **rs\_source\_ds** パラメータの値は、Replication Agent インスタンスの名前と同じ値にしないでください。
- Replication Agent コネクション設定パラメータの詳細については、『Replication Agent 管理ガイド』を参照してください。

### rs\_ticket\_version

Replication Agent がプライマリ・データベースの時刻またはプライマリ・データベースの日付と時刻を **rs\_ticket** マーカに書き込むかどうかを決定します。

デフォルト値

1

値

- 1 – Replication Agent は、プライマリ・データベースの時刻のみを記録します。
- 2 – Replication Agent は、プライマリ・データベースの日付と時刻の両方を記録します。

コメント

- 値が1に設定されている場合、Replication Agent は、プライマリ・データベースの時刻のみを **rs\_ticket** マーカに記録します。たとえば、13:20:19.368。
- 値が2に設定されている場合、Replication Agent は、プライマリ・データベースの日付と時刻の両方を **rs\_ticket** マーカに記録します。たとえば、12/14/07 13:20:19.368。

参照：

- rs\_ticket (158 ページ)

### rs\_username

Replication Agent が Replication Server アクセスに使用するユーザ・ログイン名。

デフォルト値

<設定なし>

値

有効なユーザ名。

コメント

- **rs\_username** パラメータの値は、プライマリ Replication Server にログインするために Replication Agent が使用するユーザ・ログイン名です。

## 設定パラメータ

- **rs\_password** パラメータの値は、**rs\_username** パラメータによって指定されたログイン名のパスワードです。
- Replication Agent が Replication Server にログインするために使用するユーザ・ログイン名には、Replication Server の **connect source** パーミッションが必要です。
- Replication Agent コネクション設定パラメータの詳細については、『Replication Agent 管理ガイド』を参照してください。
- **rs\_username** ユーザは、create object パーミッションを持っている必要があります。これがないと、Replication Agent はそれを使用して Replication Server から複写定義を作成できません。このパーミッションは、RSSD から手動で付与する必要があります。

### rssd\_charset

プライマリ Replication Server の RSSD との通信に使用される文字セット。

デフォルト値  
"" (空の文字列)

値  
Replication Agent ホスト・マシン上の Java VM によってサポートされている任意の有効な Sybase 文字セット。

#### コメント

- **rssd\_charset** パラメータの値は、RSSD 文字セットと一致する (または互換性がある) 必要があります。RSSD 文字セットは通常、Replication Server **rs\_charset** 設定パラメータによって識別される Replication Server のデフォルト文字セットと同じです。
- **rssd\_charset** パラメータの値に有効な文字セットを指定する場合、Replication Agent インスタンスは、その文字セットを使用して RSSD と通信します。
- 有効な文字セット名を **rssd\_charset** パラメータ (デフォルトの **rssd\_charset** 値 "" を含む) の値に指定しない場合、Replication Agent は RSSD 文字セットを使用して RSSD と通信します。
- **rssd\_charset** パラメータは、Replication Agent **use\_rssd** パラメータが **false** に設定されている場合、設定する必要がありません。
- Replication Agent コネクション設定パラメータの詳細については、『Replication Agent 管理ガイド』を参照してください。

## rssd\_database\_name

プライマリ Replication Server の RSSD のデータベース名。

デフォルト値

<設定なし>

値

有効なデータベース名。

コメント

- **rssd\_database\_name** パラメータの値は、プライマリ Replication Server の RSSD のデータベース名です。
- **rssd\_database\_name** パラメータは、Replication Agent **use\_rssd** パラメータが **false** に設定されている場合、設定する必要はありません。
- Replication Agent コネクション設定パラメータの詳細については、『Replication Agent 管理ガイド』を参照してください。

## rssd\_host\_name

プライマリ Replication Server の RSSD が存在するマシンの名前。

デフォルト値

<設定なし>

値

有効なホスト名。

コメント

- **rssd\_host\_name** パラメータの値は、プライマリ Replication Server の RSSD が存在するホスト・マシンの名前です。
- **rssd\_host\_name** パラメータは、Replication Agent の **use\_rssd** パラメータが **false** に設定されている場合、設定する必要はありません。
- Replication Agent コネクション設定パラメータの詳細については、『Replication Agent 管理ガイド』を参照してください。

## rssd\_password

Replication Agent がプライマリ Replication Server の RSSD へのアクセスに使用するパスワード。

デフォルト値

"" (空の文字列)

## 設定パラメータ

### 値

有効なパスワード。

### コメント

- **rssd\_password** パラメータの値は、Replication Agent がプライマリ Replication Server の RSSD にアクセスするために使用するユーザ・ログイン名のパスワードです。
- **rssd\_password** パラメータの値は、Replication Agent インスタンス設定ファイルで暗号化されています。
- **rssd\_password** パラメータは、Replication Agent の **use\_rssd** パラメータが **false** に設定されている場合、設定する必要はありません。
- Replication Agent コネクション設定パラメータの詳細については、『Replication Agent 管理ガイド』を参照してください。

## rssd\_port\_number

プライマリ Replication Server の Replication Server システム・データベース (RSSD) のクライアント・ポート番号。

### デフォルト値

1111

### 値

RSSD ホスト・マシン上の有効なポート番号。

### コメント

- **rssd\_port\_number** パラメータの値は、RSSD データ・サーバのクライアント・ポート番号です。
- **rssd\_port\_number** パラメータは、Replication Agent の **use\_rssd** パラメータが **false** に設定されている場合、設定する必要はありません。
- Replication Agent コネクション設定パラメータの詳細については、『Replication Agent 管理ガイド』を参照してください。

## rssd\_username

Replication Agent がプライマリ Replication Server の RSSD にアクセスするときに使用するユーザ・ログイン名。

### デフォルト値

<設定なし>

**値**

RSSD データ・サーバの有効なユーザ・ログイン名。

**コメント**

- **rssd\_username** パラメータの値は、RSSD にアクセスするために Replication Agent が使用するユーザ・ログイン名です。
- **rssd\_username** パラメータは、Replication Agent の **use\_rssd** パラメータが **false** に設定されている場合、設定する必要はありません。
- Replication Agent コネクション設定パラメータの詳細については、『Replication Agent 管理ガイド』を参照してください。

**scan\_sleep\_increment**

前の複製トランザクションのスキャンに失敗した後、トランザクション・ログをスキャンする前の待機間隔に追加する秒数。

**デフォルト値**

5

**値**

0 ~ 3600 の整数。

**コメント**

- **scan\_sleep\_increment** パラメータの値は、前のトランザクションのスキャンに失敗した後、Log Reader コンポーネントが複製されるトランザクションのログをスキャンする前の待機間隔に追加される秒数です。
- **scan\_sleep\_increment** パラメータによって指定された秒数は、待機間隔が **scan\_sleep\_max** パラメータによって指定された値に到達するまで、各待機間隔に追加されます。
- Replication Agent の最適パフォーマンスのためには、**scan\_sleep\_increment** パラメータの値は、ある特定の期間でプライマリ・データベースの平均オペレーション数とのバランスがとられていることが必要です。通常、各 Log Reader のスキャン中、トランザクション・ログからのオペレーションをより多く読み取ることにより、パフォーマンスが向上します。
- 頻繁に更新されないプライマリ・データベースでは、**scan\_sleep\_increment** パラメータの値を上げることにより、全体的なパフォーマンスが向上する可能性があります。
- データベースが継続的に更新される場合、**scan\_sleep\_increment** パラメータの値は、Replication Agent のパフォーマンスにとって重要でない場合があります。

## scan\_sleep\_max

Log Reader トランザクション・ログのスキャン間の最大待機間隔

デフォルト値

**60**

値

1 ~ 3600 の整数。

コメント

- **scan\_sleep\_max** パラメータの値は、前のトランザクションのスキャンに失敗した後、Log Reader コンポーネントが複製するトランザクションのトランザクション・ログをスキャンする間の最大秒数です。
- 頻繁に更新されないデータベースで複製遅延時間を短縮するためには、**scan\_sleep\_max** パラメータに対して、低い数字を設定します。
- プライマリ・データベースが継続的に更新される場合、**scan\_sleep\_max** パラメータの値は、Replication Agent のパフォーマンスにとって重要でない場合があります。

## skip\_lr\_errors

(Oracle および UDB) Replication Agent がログ・レコード処理エラーを無視するかどうかを決めます。

デフォルト値

**false**

値

**true** – Replication Agent はログ・レコード処理エラーをスキップし、複製を続けます。

**false** – Replication Agent はログ・レコード処理エラーをスキップしません。

コメント

- **skip\_lr\_errors** を true に設定すると、Replication Agent は発生したログ・レコード処理エラーと、エラーがスキップされたことを示す警告を記録します。エラー発生時にログ・レコードのトランザクション ID、オペレーション ID、ロケータが使用可能な場合は、それらも記録されます。Replication Agent は、トランザクション・ログ・レコードの処理を続けます。



- **skip\_lr\_errors** を false に設定すると、Replication Agent は例外を返し、すべての複製処理を停止し、Replication Down 状態に移行します。
- **skip\_lr\_errors** は、Sybase® サポート・センタの支援に基づくトラブルシューティングのみを対象としています。
- **skip\_lr\_errors** は、Replication Agent が Admin 状態または Replication Down 状態の場合にのみ変更できます。

---

**警告！** このパラメータを使用する場合、ログ・レコード処理エラーがスキップされたときに、トランザクションが失われたり、RASD とプライマリ・データベースが同期しなくなったりする恐れがあります。

---

## **skip\_ltl\_errors**

Replication Agent が、ログ転送言語 (LTL) のエラー・メッセージを無視するかを決定します。

---

**警告！** **skip\_ltl\_errors** パラメータを不適切に使用すると、プライマリ・データベースおよびレプリケート・データベースの間のデータの矛盾の原因になります。

---

デフォルト値

**false**

値

**true** – 省略 LTL エラーを有効にして、複製を継続する。

**false** – 省略 LTL エラーを無効にする。

コメント

- **skip\_ltl\_errors** 設定パラメータが **true** に設定されている場合、Replication Agent のインスタンスは、Replication Server によって返された LTL エラー・メッセージを offending LTL コマンドと共に記録し、トランザクション・ログ・レコードの処理を継続します。
- **skip\_ltl\_errors** 設定パラメータが **false** に設定されている場合、Replication Agent のインスタンスは、LTL エラー・メッセージを受け取り、エラーがリカバリできないと、すべての複製処理を停止し、*Replication Down* 状態になります。
- **skip\_ltl\_errors** パラメータは、Sybase サポート・センタの支援に基づくトラブルシューティングのみを対象としています。

## **structured\_tokens**

Replication Agent が LTL 構造化トークンを使用するかを決定します。

デフォルト値

**true**

## 設定パラメータ

### 値

**true** – LTL 構造化トークンを有効にする。

**false** – LTL 構造化トークンを無効にする。

### コメント

- **structured\_tokens** 設定パラメータが **true** に設定されている場合、ログ転送言語 (LTL) コンポーネントは、LTL コマンドを生成する際に LTL 構造化トークンを使用します。
- LTL で構造化トークンを使用すると、全般的な複製システムのパフォーマンスの大幅な向上をもたらします。
- LTL で構造化トークンを使用すると、特にプライマリ・データベースの Sybase 以外のデータ型が Replication Server によって変換された場合は、Replication Server パフォーマンスの大幅な向上をもたらします。
- カラム名に 1 つ以上のスペースのあるカラムを複製するには、**structured\_tokens** パラメータの値を **true** に設定する必要があります。

## truncation\_interval

Replication Agent トランザクション・ログの自動トランケーション間の時間間隔を指定します。

---

**警告！** UDB に自動トランケーションを設定する場合、Replication Agent は暗黙的にプライマリ・データベースのログ・ファイルを削除します。Oracle では、Replication Agent は必要のないアーカイブ・ログ・ファイルを削除します。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。

---

### デフォルト値

**0**

### 値

0 ~ 720 の整数。

### コメント

- **truncation\_interval** パラメータの値は、自動トランザクション・ログ・トランケーション間の分数です。
- **truncation\_interval** パラメータ値に基づいた自動トランザクション・ログ・トランケーションは、**truncation\_type** パラメータの値が **interval** である場合にのみ実行されます。

- トランケーション間隔の最大値は 720 分、または 12 時間です。
- **truncation\_interval** パラメータの値が 0 (ゼロ) で、**truncation\_type** パラメータの値が **interval** の場合、自動トランケーションは無効になります。
- トランザクション・ログを手動でトランケートするには、**pdb\_truncate\_xlog** コマンドを使用します。

**参照：**

- **pdb\_archive\_path** (206 ページ)
- **truncation\_type** (257 ページ)
- **pdb\_archive\_remove** (207 ページ)

## **truncation\_type**

Replication Agent のトランザクション・ログ・トランケーション動作を設定します。

---

**警告！** UDB では、Replication Agent は、暗黙的に UDB のプライマリ・データベース・ログ・ファイルを削除します。Oracle では、Replication Agent は必要のないアーカイブ・ログ・ファイルを削除します。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。

---

### デフォルト値 **locator\_update**

#### 値

**command** – Replication Agent は、**pdb\_truncate\_xlog** コマンドが呼び出された場合にのみトランザクション・ログをトランケートします。

**truncation\_type** パラメータの値が **command** の場合、トランザクション・ログをトランケートできる唯一の方法は、**pdb\_truncate\_xlog** コマンドを呼び出すことです。

**truncation\_type** パラメータの値が **command** の場合、自動トランケーションは実行されません。

**interval** – Replication Agent は、設定可能な間隔時間による場合、自動的にトランザクション・ログをトランケートします。

**locator\_update** – Replication Agent は、プライマリ Replication Server から新しい LTM ロケータの値を受け取るたびに、トランザクション・ログを自動的にトランケートします。

**truncation\_type** パラメータの値が **locator\_update** の場合、Replication Agent が、プライマリ Replication Server から新しい LTM ロケータの値を受け取ると、トランザクション・ログが自動的にトランケートされます。

### コメント

---

**注意：** Oracle および UDB では、Replication Agent に不必要なアーカイブ・ログ・ファイルのランケーションが **pdb\_archive\_path** ディレクトリから実行されますが、**pdb\_archive\_remove** パラメータの値が **true** である場合のみです。

---

- **truncation\_type** パラメータの値に関わらず、**pdb\_truncate\_xlog** コマンドを呼び出すことによって、Replication Agent トランザクション・ログをいつでも手動でランケートできます。
- **truncation\_interval** パラメータの値が **0** (ゼロ) で、**truncation\_type** パラメータの値が **interval** (両パラメータのデフォルト値) の場合、自動ランケーションは無効になります。
- Replication Agent は、**lti\_update\_trunc\_point** パラメータの値に基づいて、新しい LTM ロケータを受け取ります。

### 参照：

- **pdb\_archive\_path** (206 ページ)
- **pdb\_archive\_remove** (207 ページ)
- **truncation\_interval** (256 ページ)

## use\_rssd

Replication Agent が複写定義を使用するかどうかを判断します。

### デフォルト値

**true**

### 値

**true** – 複写定義の使用可

**false** – 複写定義の使用不可

### コメント

- **use\_rssd** パラメータの値が **true** の場合、Replication Agent のインスタンスは、*Replication Down* 状態から *Replicating* 状態 (たとえば、**resume** コマンドが呼び出される際) に移動するたびに、Replication Server System Database (RSSD) に接続して、プライマリ・データベースの複写定義を自動的に受け取ります。
- 複写定義を受け取るたびに、Replication Agent は情報をキャッシュに保存します。Replication Agent は、ログ転送言語 (LTL) のコマンドを生成する際、キャッシュに保存された複写定義を使用します。

- ログ転送インタフェース (LTI) コンポーネントが、キャッシュされた複写定義のないデータベース・オブジェクト上のオペレーションに遭遇した場合、Replication Agent は RSSD に再接続し、複写定義キャッシュを更新します。
- そのオペレーションの複写定義が見つからない場合、Replication Agent のインスタンスは、すべての複写を中止して、*Replication Down* 状態に移行します。
- Replication Agent は、RSSD に保存されるテーブルおよび関数の複写定義にある情報を使用して (つまり、個別のプライマリ・データベース・オブジェクトの複写定義)、より効率的な LTL を生成し、LTI コンポーネントおよび Replication Server のスループットを向上させます。

RSSD の複写定義にアクセスすると、以下によって、LTI コンポーネントのパフォーマンスが向上します。

- LTL 内のカラム名の省略。複写定義で指定された順序でカラムが送信される際、カラム・イメージをカラム名 (見出し) なしで送信すると、LTL のオーバーヘッドを削減できます。
- LTL 内の不必要なカラムの省略。複写定義で指定されたカラムが送信される際、変更されないカラムのイメージは送信の必要がないため、LTL のオーバーヘッドを削減できます。
- 複写定義で指定されたデータ型の各カラムのデータの送信。これにより、複写システム全体で、データがより効率的に処理されます。
- 複写定義で定義されているものと同じ大文字と小文字でのデータベース・オブジェクト名の送信。
- **use\_rssd** パラメータの値が **false** の場合、前に説明されたパフォーマンスの向上は実行不可能です。その場合、Replication Agent は、LTL の *char* データ型として、すべてのデータを送信します。

## 設定パラメータ

## 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](http://www.sybase.com/detail_list?id=9784) にアクセスします。
- プラットフォームの動作確認については、<http://certification.sybase.com/ucr/search.do> にアクセスします。

## MySybase プロファイルの作成

---

MySybase は無料サービスです。このサービスを使用すると、Sybase Web ページの表示方法を自分専用カスタマイズできます。

1. <http://www.sybase.com/mysybase> を開きます。
2. [Register Now (今すぐ登録)] をクリックします。



## アクセシビリティ機能

---

アクセシビリティ機能を使用すると、身体障害者を含むすべてのユーザーが電子情報に確実にアクセスできます。

Sybase 製品のマニュアルには、アクセシビリティを重視した HTML 版もあります。

オンライン・マニュアルは、スクリーン・リーダーで読み上げる、または画面を拡大表示するなどの方法により、視覚障害を持つユーザがその内容を理解できるよう配慮されています。

Sybase の HTML マニュアルは、米国のリハビリテーション法第 508 条のアクセシビリティ規定に準拠していることがテストにより確認されています。第 508 条に準拠しているマニュアルは通常、World Wide Web Consortium (W3C) の Web サイト用ガイドラインなど、米国以外のアクセシビリティ・ガイドラインにも準拠しています。

---

**注意：**アクセシビリティ・ツールを効率的に使用するには、設定が必要な場合もあります。一部のスクリーン・リーダーは、テキストの大文字と小文字を区別して発音します。たとえば、すべて大文字のテキスト (ALL UPPERCASE TEXT など) はイニシャルで発音し、大文字と小文字の混在したテキスト (Mixed Case Text など) は単語として発音します。構文規則を発音するようにツールを設定すると便利かもしれませんが、詳細については、ツールのマニュアルを参照してください。

---

Sybase のアクセシビリティに対する取り組みについては、Sybase Accessibility サイト (<http://www.sybase.com/products/accessibility>) を参照してください。このサイトには、第 508 条と W3C 標準に関する情報へのリンクもあります。

製品マニュアルには、アクセシビリティ機能に関する追加情報も記載されています。

追加の説明や情報の入手

## 用語解説

この用語解説では Replication Server Options で使用されている用語について説明します。

- **Adaptive Server** – Sybase リレーショナル・データベース管理システム (RDBMS) ソフトウェア製品のブランド名。
  - Adaptive Server Enterprise は、高容量オンライン・トランザクション処理 (OLTP: Online Transaction Processing) システムとクライアント・アプリケーション向けの大規模なリレーショナル・データベースを複数管理します。
  - Sybase®IQ は、特殊なインデックス・アルゴリズムで大規模なリレーショナル・データベースを複数管理して、高速、高容量のビジネス・インテリジェンス、意思決定サポート、レポート・クライアント・アプリケーションなどをサポートします。
  - SQL Anywhere® (旧称 Adaptive Server Anywhere) は、埋め込みアプリケーションやモバイル・デバイス・アプリケーションに最適な小型 DBMS によってリレーショナル・データベースを管理します。

「DBMS」と「RDBMS」参照。

- **アトミック・マテリアライゼーション** – マテリアライゼーション・メソッドの1つで、サブスクリプション・データをプライマリ・データベースからレプリケート・データベースに1回のアトミック・オペレーションでコピーします。プライマリ・データベースでサブスクリプション・データが取り込まれるまで、プライマリ・データの変更はできません。「バルク・マテリアライゼーション」と「ノンアトミック・マテリアライゼーション」参照。
- **BCP ユーティリティ** – ターゲット・データベースのテーブルに複数ローのデータをロードできるバルク・コピー転送ユーティリティ。「バルク・コピー」参照。
- **バルク・コピー** – データベース・テーブルとプログラム変数との間の高速データ転送に使用される Open Client™ インタフェース。バルク・コピーは、SQL **insert** コマンドと **select** コマンドを使用するデータ転送の代替となります。
- **バルク・マテリアライゼーション** – マテリアライゼーション・メソッドの1つで、レプリケート・データベースのサブスクリプション・データを複写システム外で初期化します。バルク・マテリアライゼーションは、テーブル複写定義とファンクション複写定義のどちらのサブスクリプションにも使用できます。「アトミック・マテリアライゼーション」と「ノンアトミック・マテリアライゼーション」参照。
- **クライアント** – クライアント/サーバ・システムでは、サーバに要求を送信して、その結果を処理する部分。「クライアント・アプリケーション」参照。

- **クライアント・アプリケーション** – メニュー、データ入力画面、レポート形式などのユーザ・インタフェースを制御するソフトウェア。「クライアント」参照。
- **コミット** – トランザクションで要求された変更を永続化する DBMS への命令。「トランザクション」参照。「ロールバック」と対比。
- **データベース** – ユーザのためにデータの受け入れ、格納、提供などを行う特定の構造 (スキーマ) を備えたデータの集まり。「データ・サーバ」、「DBMS」、「RDBMS」参照。
- **データベース・コネクション** – Replication Server がデータベースを管理し、トランザクションをデータベースに分配するためのコネクション。複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写では、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写にかかる時間。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・データベースで複写トランザクションをコミットするまでの経過時間です。

ディスクの複写では、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・デバイスで複写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写」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写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 にデータを複写するために 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写トランザクションを適用するとき使用するレプリケート・データベースの特別なユーザ・ログイン名。「レプリケート・データベース」と「Replication Server」参照。

- **マテリアライゼーション** – プライマリ・データベースからレプリケート・データベースにデータをコピーして、複写システムがトランザクションの複写を開始できるようにレプリケート・データベースを初期化するプロセス。「アトミック・マテリアライゼーション」、「バルク・マテリアライゼーション」、「ノンアトミック・マテリアライゼーション」参照。
- **Multi-Path Replication™** – 送信元データベースからターゲット・データベースへのデータの並列パスを有効にすることによってパフォーマンスを向上させる Replication Server の機能。これらの複数のパスではデータが個別に処理され、それらのパス間のトランザクションの一貫性を必要とせずにデータ・セットを並列処理できる場合に適用されます。
- **ノンアトミック・マテリアライゼーション** – マテリアライゼーション・メソッドの1つで、プライマリ・データベースをロックせずにサブスクリプション・データをコピーします。データの転送中もプライマリ・データを変更できるので、プライマリ・データベースとレプリケート・データベース間で一時的に不一致が生じる場合があります。「アトミック・マテリアライゼーション」と対比。「バルク・マテリアライゼーション」参照。
- **ODBC** – Open Database Connectivity の略語。クライアントがデータ・サーバに接続するための業界標準通信プロトコル。「クライアント」、「データ・サーバ」、「JDBC」参照。
- **Open Client** – カスタム・アプリケーション、サードパーティ製品、他の Sybase 製品が Open Server アプリケーションと通信するために必要なインタフェースを提供する Sybase 製品。「Open Server」参照。
- **Open Client アプリケーション** – Sybase Open Client ライブラリを使用して Open Client 通信プロトコルを実装するアプリケーション。「Open Client」と「Open Server」参照。
- **Open Server** – カスタム・サーバの作成に必要なツールとインタフェースを提供する Sybase 製品。「Open Client」参照。
- **Open Server アプリケーション** – Sybase Open Server ライブラリを使用して Open Server 通信プロトコルを実装するサーバ・アプリケーション。「Open Client」と「Open Server」参照。
- **アウトバウンド・キュー** – レプリケート・データベースにメッセージをスプールするために Replication Server が管理するステープル・キュー。「インバウンド・キュー」、「レプリケート・データベース」、「ステープル・キュー」参照。
- **プライマリ・データ** – 複写に使用されるデータ・ソース。プライマリ・データはプライマリ・データベースによって保存および管理されます。「プライマリ・データベース」参照。
- **プライマリ・データベース** – 複写システムを使用して別のデータベース(レプリケート・データベース)に複写するデータが格納されているデータベース。複写システムではプライマリ・データベースが複写データのソースです。アク

ティブ・データベースと呼ばれることもあります。「レプリケート・データベース」と対比。「プライマリ・データ」参照。

- **プライマリ・キー** – テーブル内の各ローをユニークに識別するカラムまたはカラムのセット。
- **プライマリ・サイト** – 通常の業務処理をサポートするためにプライマリ・データ・サーバとプライマリ・データベースが配備されている場所または施設。アクティブ・サイトまたはメイン・サイトと呼ばれることもあります。「プライマリ・データベース」と「レプリケート・サイト」参照。
- **プライマリ・テーブル** – 複写のソースとして使用されるテーブル。プライマリ・テーブルはプライマリ・データベース・スキーマで定義されます。「プライマリ・データ」と「プライマリ・データベース」参照。
- **プライマリ・トランザクション** – プライマリ・データベースでコミットされ、プライマリ・データベースのトランザクション・ログに記録されたトランザクション。「プライマリ・データベース」、「複写トランザクション」、「トランザクション・ログ」参照。
- **クワイス** – システムを、これ以上データ変更ができない状態に切り替えること。「クワイス状態」参照。
- **クワイス状態** – 複写システムでは、すべての更新がその送信先に反映された状態。Replication Agent と Replication Server の一部のコマンドでは、最初に複写システムをクワイスする必要があります。

データベースでは、トランザクションがデータを変更できないようにデータの更新がすべて停止し、データ・デバイスとログ・デバイスが静止している状態。

この用語は「クワイスされている」または「クワイス」と同義です。「クワイス」参照。

- **RASD** – Replication Agent システム・データベース (Replication Agent System Database) の略語。RASD 内の情報は、トランザクション・ログでデータベース構造やスキーマ・オブジェクトを認識するためにプライマリ・データベースによって使用されます。
- **RCL** – 複写コマンド言語 (Replication Command Language) の略語。Replication Server の管理に使用されるコマンド言語。「*Replication Server*」参照。
- **RDBMS** – リレーショナル・データベース管理システム (Relational Database Management System) の略語。リレーショナル・データベースを管理および制御するアプリケーション。「*DBMS*」と対比。「*リレーショナル・データベース*」参照。
- **リレーショナル・データベース** – カラム (データ項目) とロー (情報の単位) から成るテーブルにデータを格納して表示するデータの集まり。リレーショナル・データベースは SQL 要求によってアクセスできます。「データベース」と対比。「*SQL*」参照。



- **レプリケート・データ** - 複写システムによってプライマリ・データベースからレプリケート・データベースに複写されたデータ・セット。「プライマリ・データベース」、「複写システム」、「レプリケート・データベース」参照。
- **レプリケート・データベース** - 複写システムによって別のデータベース(プライマリ・データベース)から複写されたデータが格納されているデータベース。レプリケート・データベースは複写システムで複写されたデータを受け取るデータベースです。「プライマリ・データベース」と対比。「レプリケート・データ」、「複写トランザクション」、「複写システム」参照。
- **複写トランザクション** - トランザクション複写システムによってプライマリ・データベースからレプリケート・データベースに複写されたプライマリ・トランザクション。「プライマリ・データベース」、「プライマリ・トランザクション」、「レプリケート・データベース」、「トランザクション複写」参照。
- **レプリケート・サイト** - プライマリ・サイトでのスケジュールされているダウン時間中の通常の業務処理をサポートするためにレプリケート・データ・サーバとレプリケート・データベースが配備されている場所または施設。「プライマリ・サイト」と対比。「レプリケート・データベース」参照。
- **Replication Agent** - プライマリ・データベース・トランザクションのログを読み取ってプライマリ・データベースのデータ変更トランザクションに関する情報を取得し、ログ情報を処理してから、それをレプリケート・データベースに分配する目的で Replication Server に送信するアプリケーション。「プライマリ・データベース」と「Replication Server」参照。
- **複写定義** - サブスクリプションを作成できるプライマリ・データベースのテーブルまたはストアド・プロシージャの記述。Replication Server によって管理される複写定義には、複写されるカラムとプライマリ・テーブルまたはストアド・プロシージャの場所に関する情報が含まれています。「Replication Server」と「サブスクリプション」参照。
- **Replication Server** - トランザクション複写システムのインフラストラクチャを提供する Sybase ソフトウェア製品。「Replication Agent」参照。
- **複写システム** - データを別の場所に複写するデータ処理システム。データは1つのサイトの異なるシステム間、またはローカル・システムとリモート・システム間で複写できます。「トランザクションの複写」参照。
- **ロールバック** - 作業単位(すなわちトランザクション)で要求された変更を取り消すデータベースへの命令。「コミット」と対比。「トランザクション」参照。
- **ルート** - プライマリ Replication Server からレプリケート Replication Server への一方方向のメッセージ・ストリーム。ルートは、データ変更コマンド(RSSDのコマンドも含む)と、Replication Server 間で複写されたファンクション(データベース・プロシージャ)を転送します。「Replication Server」参照。

- **RSSD** – Replication Server システム・データベース (Replication Server System Database) の略語。Replication Server の複写システム情報を管理します。  
「*Replication Server*」参照。
- **SQL** – 構造化問合せ言語 (Structured Query Language) の略語。リレーショナル・データベースのデータ処理に使用される非手続き型プログラミング言語。ANSI SQL は業界標準の1つです。「*トランザクション*」参照。
- **ステーブル・キュー** – Replication Server が管理するディスク・デバイスベースの蓄積転送キュー。ステーブル・キューに書き込まれたメッセージは、適切なプロセスまたはレプリケート・データベースに配信できる時まで、このキューに格納されます。Replication Server は受信メッセージ (インバウンド・キュー) と送信メッセージ (アウトバウンド・キュー) の両方にステーブル・キューを提供します。「*データベース・コネクション*」、「*Replication Server*」、「*ルート*」参照。
- **ストアド・プロシージャ** – 1つまたは一連のオペレーションを表すデータ・サーバ・オブジェクト。この用語は、「*ファンクション*」と同じ意味で使用される場合があります。
- **サブスクリプション** – 指定した場所のレプリケート・データベース内のテーブルの複写コピーやテーブルのローを Replication Server に管理させる要求。「*レプリケート・データベース*」、「*複写定義*」、「*Replication Server*」参照。
- **テーブル** – リレーショナル DBMS では、テーブルに固有のカラム・グループから成る順不同のローが特定数含まれている、2次元のデータ配列または名前付きデータ・オブジェクト。「*データベース*」参照。
- **トランザクション** – データベースの作業単位。0、1、または多数のオペレーション (*insert*、*update*、*delete* など) を含むことができ、全体として適用または拒否されます。データベースの設定によっては、データを変更する各 SQL ステートメントを別々のトランザクションとして処理できます。「*SQL*」参照。
- **トランザクションの一貫性** – プライマリ・データベースのすべてのトランザクションが、プライマリ・データベースと同じ順序でレプリケート・データベースに適用される状態。
- **トランザクション・ログ** – 通常は、データ・サーバが管理しているデータに影響するトランザクションのログ。Replication Agent はトランザクション・ログを読み取って、複写するトランザクションを識別し、プライマリ・データベースから取得します。「*Replication Agent*」、「*プライマリ・データベース*」、「*Replication Server*」参照。
- **トランザクションの複写** – データ複写方式の1つで、データ変更オペレーションをプライマリ・データベースからレプリケート・データベースにコピーします。「*データの複写*」参照。
- **UDB** – IBM DB2 Universal Database (旧称 IBM DB2 for Linux, UNIX, and Windows)

- **WAN**-広域ネットワーク (Wide Area Network) の略語。ローカル・エリア・ネットワーク (LAN) をデータ通信回線で接続しているシステム。「LAN」と対比。



## 索引

## A

admin\_port 設定パラメータ 179  
 Admin 状態  
 asm\_password 設定パラメータ 180  
 asm\_tns\_connection 設定パラメータ 181  
 asm\_tns\_filename 設定パラメータ 181  
 asm\_username 設定パラメータ 182  
 auto\_adjust\_block\_count 設定パラメータ 183

## C

column\_compression 設定パラメータ 184  
 compress\_ltl\_syntax 設定パラメータ 185  
 connect\_to\_rs 設定パラメータ 185

## D

datetime Sybase データ型 212  
 ddl\_password 設定パラメータ 186  
 ddl\_username 設定パラメータ 186  
 DSN (データ・ソース名) 224  
 dump\_batch\_timeout 設定パラメータ 187

## F

filter\_maint\_userid 設定パラメータ 188  
 function\_password 設定パラメータ 189  
 function\_username 設定パラメータ 189

## J

Java Runtime Environment (JRE)  
   バージョン 145  
   文字セット 243, 250  
 JDBC ドライバ  
   Oracle データベース・サーバ 222  
   UDB 222  
   バージョン 75, 145

## L

LOB カラム  
 名前 34

日時変換 212  
 複写ステータス 34  
 複写の無効化 31  
 複写の有効化 31

## Log Reader コンポーネント

filter\_maint\_userid パラメータ 188  
 max\_ops\_per\_scan パラメータ 205  
 scan\_sleep\_increment パラメータ 253  
 scan\_sleep\_max パラメータ 254  
 オペレーション・キュー 205  
 クワイス処理 80  
 スキャン当たりのオペレーション 205  
 統計情報 126

## Log Transfer Interface (LTI) コンポーネント 184

## Log Transfer Interface コンポーネント

column\_compression パラメータ 184  
 compress\_ltl\_syntax パラメータ 185  
 connect\_to\_rs parameter 185  
 dump\_batch\_timeout パラメータ 187  
 lti\_update\_trunc\_point パラメータ 199  
 LTL バッチ・モード・バッファ 198, 200  
 ltl\_batch\_size パラメータ 200  
 ltl\_character\_case パラメータ 201  
 ltl\_origin\_time\_required パラメータ 202  
 クワイス処理 80  
 バッチ・タイムアウト 187  
 バッチ・モード 197, 200  
 バッファ・サイズ 200  
 統計情報 126

## ログ転送言語 (LTL) 45, 59

LTL トレース・ログ 165  
 LTL バッチ・モード・バッファ 198, 200  
 origin\_time コマンド・タグ 202  
 エラー・メッセージ 255  
 オブジェクト名の太文字と小文字 45, 56,  
   201  
 圧縮構文 185  
 構造化トークン 255  
 送信されたカラム 184

## 索引

Log Transfer Manager コンポーネント  
統計情報 126

log\_backup\_files 設定パラメータ 190

log\_directory 設定パラメータ

log\_read\_block\_count 設定パラメータ 191

log\_system\_name コマンド 11

log\_trace\_verbose 設定パラメータ 192

log\_wrap 設定パラメータ 192

lr\_dump\_marker コマンド 12

lr\_max\_lobdata\_cache 設定パラメータ 193

lr\_max\_op\_queue\_size 設定パラメータ 193

lr\_max\_scan\_queue\_size 設定パラメータ 193

lr\_read\_buffer\_size 設定パラメータ 196

lti\_batch\_mode 設定パラメータ 197

lti\_max\_buffer\_size 設定パラメータ 198

lti\_update\_trunc\_point 設定パラメータ 199

LTL の大文字と小文字  
カラム名 201  
ストアド・プロシージャ名 45, 201  
テーブル名 56, 201  
次も参照：ログ転送言語 (LTL)

ltl\_batch\_size 設定パラメータ 200

ltl\_big\_endian\_unitext 設定パラメータ 200

ltl\_character\_case 設定パラメータ 201

ltl\_origin\_time\_required 設定パラメータ 202

ltl\_send\_only\_primary\_keys 設定パラメータ 202

LTL の省略形 185

ltm\_admin\_pw\_min\_length 設定パラメータ 204

ltm\_admin\_pw 設定パラメータ

ltm\_admin\_user 設定パラメータ 170, 205

LTM ロケータ  
オリジン・キュー ID 15  
更新 199  
トランザクション・ログでの位置 116  
ヘルプ・コマンド 109

## M

max\_ops\_per\_scan 設定パラメータ 205

Microsoft SQL Server  
コネクション・タイプ 222

## O

ODBC ドライバ  
データ・ソース名 (DSN) 224

Oracle RMAN ユーティリティ  
アーカイブ・ログ・ファイルのトランケーション 241, 242

Oracle アーカイブ・ログ・ファイルのトランケーション  
RMAN ユーティリティの使用 241, 242

Oracle データベース・サーバ  
コネクション・タイプ 222

origin\_time LTL コマンド・タグ 202

## P

pdb\_archive\_path 設定パラメータ 206

pdb\_archive\_remove 設定パラメータ 207

pdb\_auto\_create\_repdefs 設定パラメータ 208

pdb\_auto\_run\_scripts 設定パラメータ 211

pdb\_automark\_tables 設定パラメータ 210

pdb\_capabilities コマンド 13

pdb\_convert\_datetime 設定パラメータ 212

pdb\_date コマンド 13

pdb\_dflt\_column\_repl 設定パラメータ 214

pdb\_dflt\_object\_repl 設定パラメータ 214

pdb\_execute\_sql コマンド 14

pdb\_gen\_id コマンド 15

pdb\_get\_columns コマンド 16

pdb\_get\_databases コマンド 18

pdb\_get\_primary\_keys コマンド 19

pdb\_get\_procedure\_parms コマンド 20

pdb\_get\_procedures コマンド 22

pdb\_get\_sql\_database コマンド 24

pdb\_get\_tables コマンド 24

pdb\_ignore\_unsupported\_anydata 設定パラメータ 215

pdb\_include\_archives 設定パラメータ 217

pdb\_ownerfilter 設定パラメータ 26

pdb\_send\_osuser\_list コマンド 28

pdb\_set\_sql\_database コマンド 30

pdb\_setrepcol コマンド 31

pdb\_setrepddl コマンド 36

pdb\_setrepproc コマンド 45

pdb\_setrepseq コマンド 54

pdb\_setreptable コマンド 56

pdb\_skip\_missing\_user 設定パラメータ 217

pdb\_skip\_op コマンド 69

- pdb\_support\_large\_identifier 設定パラメータ  
218  
 pdb\_thread\_filter 構成パラメータ 72  
 pdb\_timezone\_file 設定パラメータ 219  
 pdb\_truncate\_xlog コマンド 73  
 pdb\_version コマンド 75  
 pdb\_xlog\_device 設定パラメータ 219  
 pdb\_xlog\_prefix\_chars 設定パラメータ 221  
 pdb\_xlog\_prefix 設定パラメータ  
 pdb\_xlog コマンド 75  
 pds\_connection\_type 設定パラメータ 222  
 pds\_dac\_port\_number 設定パラメータ 223  
 pds\_database\_name 設定パラメータ 223  
 pds\_datasource\_name 設定パラメータ 224  
 pds\_host\_name 設定パラメータ 224  
 pds\_integrated\_security 設定パラメータ 225  
 pds\_password 設定パラメータ 226  
 pds\_port\_number 設定パラメータ 226  
 pds\_retry\_count 設定パラメータ 227  
 pds\_retry\_timeout 設定パラメータ 227  
 pds\_server\_name 設定パラメータ 227  
 pds\_tns\_connection 設定パラメータ 228  
 pds\_tns\_filename 設定パラメータ 228  
 pds\_username 設定パラメータ 229
- Q**
- quiesce コマンド 80
- R**
- ra\_admin\_device 設定パラメータ 81  
 ra\_admin\_instance\_prefix 設定パラメータ 81  
 ra\_admin\_prefix\_chars 設定パラメータ 81, 232  
 ra\_config コマンド  
 ra\_date コマンド 88  
 ra\_deviceoffset コマンド 88  
 ra\_devicepath コマンド 89  
 ra\_downgrade\_accept コマンド  
 ra\_downgrade\_prepare コマンド  
 ra\_downgrade コマンド  
 ra\_dumptran コマンド 95  
 ra\_dump コマンド 94  
 ra\_helparchive 設定パラメータ 100  
 ra\_helparticle コマンド 101  
 ra\_helppdb コマンド 102  
 ra\_helpdeviceoffset コマンド 105  
 ra\_helpdevice コマンド 103  
 ra\_helpfield コマンド 107  
 ra\_helpop コマンド 110  
 ra\_helprtran 設定パラメータ 113  
 ra\_helpuser コマンド 113  
 ra\_help コマンド 99  
 ra\_license コマンド 115  
 ra\_locator コマンド 116  
 ra\_maint\_id コマンド 119  
 ra\_marker システム・ファンクション 120  
 ra\_marker コマンド 120  
 ra\_migrate コマンド 121  
 ra\_migrate システム・ファンクション 121  
 ra\_purge\_first\_open 設定パラメータ 122  
 ra\_retry\_count 設定パラメータ 234  
 ra\_retry\_timeout 設定パラメータ 234  
 ra\_set\_autocorrection コマンド 123  
 ra\_set\_login コマンド  
 ra\_standby 設定パラメータ 239  
 ra\_statistics コマンド 126  
 ra\_statrack\_interval 設定パラメータ 240  
 ra\_statrack\_list 設定パラメータ 137  
 ra\_statrack 設定パラメータ 137  
 ra\_status コマンド 138  
 ra\_truncatearticles コマンド 139  
 ra\_truncateddlfilters コマンド 140  
 ra\_truncateusers コマンド 141  
 ra\_updatedevices コマンド 142, 186  
 ra\_updateusers コマンド 144  
 ra\_version\_all コマンド 145  
 ra\_version コマンド 145
- RASD**
- rasd\_backup\_dir パラメータ 235  
 rasd\_database パラメータ 235  
 rasd\_mirror\_tran\_log パラメータ 236  
 rasd\_trace\_log\_dir パラメータ 237  
 rasd\_tran\_log パラメータ 238  
 rasd\_tran\_log\_mirror パラメータ 239  
 アーティクル 101  
 データベース・バックアップ・ファイル  
235  
 データベース・ファイル 235

## 索引

- データベースのバックアップ 146
- トランケート 141
- トランザクション・ログ・ファイル 238
- バックアップからのリストア 149
- フィールド 107
- プライマリ・データベース・オブジェクト 101
- プライマリ・データベース・ユーザ 141
- プライマリ・データベース 102
- ミラー・ログ・デバイスの更新 186
- ミラーリングされた RASD ログ 239
- ミラー・ログ・デバイス、プライマリ・データベース 103
- ログ・デバイスの更新 142
- ログ・デバイス、プライマリ・データベース 89
- rasd\_backup\_dir 設定パラメータ 235
- rasd\_backup コマンド 146
- rasd\_database 設定パラメータ 235
- rasd\_helpbackup コマンド 147
- rasd\_mirror\_tran\_log 設定パラメータ 236
- rasd\_removebackup コマンド 148
- rasd\_restore コマンド 149
- rasd\_trace\_log\_dir 設定パラメータ 237
- rasd\_tran\_log\_mirror 設定パラメータ 239
- rasd\_tran\_log 設定パラメータ 238
- rasd\_trunc\_schedule コマンド 150
- RASD 内のアーティクル 101
  - トランケート 139
- RASD のトランケート 139, 141
- RASD のバックアップ
- Replicating 状態 138, 152
- Replication Agent
  - Admin 状態 138, 162
  - asm\_password 180
  - Log Reader コンポーネント 80, 253
  - Log Transfer Interface コンポーネント 80
  - LTL トレース・ログ 165
  - LTL バッチ・サイズ 197, 200
  - LTL 構造化トークン 255
  - LTM ロケータ 116, 199
  - pds\_tns\_connection 228
  - RASD 149, 235
  - RASD のリストア 149
  - RASD 内のアーティクル 101
  - RASD のバックアップ 146
  - Replicating 状態 138, 152
  - Replication Server ユーザ・ログイン 249
  - rs\_create\_repdef 155
  - rs\_drop\_repdef 157
  - RSSD コネクション 250, 252
  - RSSD ユーザ・ログイン 251, 252
  - アーティクルのフィールド 107
  - インスタンスの静止 80
  - インスタンスの停止 161
  - オリジン・キュー ID 116
  - 管理者ログイン 125, 204
  - 管理者ログイン・パスワード長 204
  - コネクションのテスト 163
  - システム・オブジェクト所有者 233
  - システム・オブジェクト名プレフィクス 230, 231
  - システム・ログ・ファイル 11, 190
  - スクリプト・ディレクトリ 75, 78, 81, 84
  - ステータス 138
  - 即時停止 161
  - データベースの世代 ID 15
  - 統計情報、パフォーマンス 126
  - トラブルシューティング 165, 191, 192, 254, 255
  - トランザクション・ログ・プレフィクス 75, 81, 220
  - トランザクション・ログの作成 75
  - トランザクション・ログの削除 75
  - バージョン 145
  - パフォーマンス統計情報 126
  - プライマリ・データベース・ユーザ・ログイン 226, 229
  - ヘルプ・コマンド 99
  - ログ・デバイス・レポジトリの更新 142
  - 管理ポート 179
  - 設定ファイル 169
  - 返された日付と時刻 88
- Replication Agent の状態
  - Admin 状態 138, 162
  - Replicating 状態 138, 152
  - 変更 80, 152, 162
- Replication Agent コマンドでヘルプ表示 99



- Replication Agent のクワイース 80
  - Replication Agent のステータス
  - Replication Agent の停止
  - Replication Server
    - LTL エラー 255
    - LTL バッチ・サイズ 200
    - LTM ロケータ 116, 199
    - ra\_marker システム・ファンクション 120
    - ra\_migrate システム・ファンクション 121
    - Replication Agent からのコネクション 163
    - Replication Agent ユーザ・ログイン 249
    - rs\_dumpdb マーカ 94
    - rs\_dumptran マーカ 94
    - コネクションのテスト 163
    - ソース・データベース 248
    - データベース・コネクション 119
    - データベースの世代 ID 15
    - テーブル複写定義 68
    - ネットワーク・パケット・サイズ 245
    - バージョンおよび LTL バッチ・サイズ 200
    - ファンクション複写定義 45
    - 複写定義 45, 56, 68
    - ポート番号 246
    - ホスト・マシン名 244
    - 文字セット 243
  - Replication Server でのデータベース・コネクション 68
  - Replication Server へのデータベース・コネクション 56
  - resume コマンド 152
  - rman\_enabled 設定パラメータ 241
  - rman\_password 設定パラメータ 242
  - rman\_username 設定パラメータ 242
  - rs\_charset 設定パラメータ 243
  - rs\_host\_name 設定パラメータ 244
  - rs\_packet\_size 設定パラメータ 245
  - rs\_password 設定パラメータ 246
  - rs\_port\_number 設定パラメータ 246
  - rs\_replicate\_owner\_required 設定パラメータ 246
  - rs\_retry\_count 設定パラメータ 247
  - rs\_retry\_timeout 設定パラメータ 247
  - rs\_source\_db 設定パラメータ 248
  - rs\_source\_ds 設定パラメータ 248
  - rs\_ticket\_version 設定パラメータ 249
  - rs\_ticket コマンド 158
  - rs\_username 設定パラメータ 249
  - RSSD
    - Replication Agent からのコネクション 250, 252
    - Replication Agent ユーザ・ログイン 251, 252
    - データベース名 251
    - 複写定義 258
    - ポート番号 252
    - ホスト・マシン名 251
    - 文字セット 250
  - rssd\_username 設定パラメータ 252
  - rssd\_charset 設定パラメータ 250
  - rssd\_database\_name 設定パラメータ 251
  - rssd\_host\_name 設定パラメータ 251
  - rssd\_password 設定パラメータ 251
  - rssd\_port\_number 設定パラメータ 252
- S**
- scan\_sleep\_increment 設定パラメータ 253
  - scan\_sleep\_max 設定パラメータ 254
  - shutdown コマンド 161
  - skip\_lr\_errors 設定パラメータ 254
  - skip\_ltl\_errors 設定パラメータ 255
  - SQL コマンドの実行
  - structured\_tokens 設定パラメータ 255
  - suspend コマンド 162
- T**
- test\_connection コマンド 163
  - truncation\_interval 設定パラメータ 256
  - truncation\_type 設定パラメータ 257
- U**
- UDB
    - DSN 内のデータベース・エイリアス 224
    - コネクション型 222
  - use\_rssd 設定パラメータ 258

## 索引

## あ

### 値

LTM ロケータ 116

## い

インスタンス、Replication Agent

管理者ログイン 125

ステータス 138

レジューム 152

静止 80

設定ファイル 169

停止 161

## う

ウォーム・スタンバイ

ra\_standby パラメータ 239

## え

エイリアス、データベース・オブジェクト 34,  
45, 63

エラー 254, 255

エラー、ログ・レコード処理 254

エラー、ログ転送言語 (LTL) 255

## お

オブジェクト、データベース

カラム 16

ストアド・プロシージャ 20, 22

テーブル 24

プライマリ・キー 19

ユーザ 113, 141

オブジェクトの所有者

ストアド・プロシージャ 45

オブジェクト所有者

プライマリ・テーブル 56

オブジェクト所有者名 59

オプションの強制 46

オペレーティング・システム

バージョン 145

オリジン・キュー ID

データベースの世代 ID 15

次も参照：LTM ロケータ

## か

### 開始

複写 152

### カラム

LOB カラムでの日時変換 212

LOB カラムの名前 34

LTL での名前 201

LTL への送信 184

RASD のフィールド 107

データベースによって返された名前 16

プライマリ・キー 19

複写ステータス 34

複写の有効化 214

複写の有効化と無効化 31

カラムの複写の有効化

すべての LOB カラム 35

カラムの複写の有効化

すべての LOB カラム 35

カラム複写の有効化 31

カラム複写の有効化

デフォルト 214

管理者ログイン

管理者ログイン・パスワード長 204

## き

### キュー

Log Reader 205

Log Transfer Interface (ログ転送インタフェ  
ース) 80, 187, 200

LTM ロケータ 116

オリジン・キュー ID 15

サスペンド処理 162

処理の静止 80

## く

クライアント・ポート

Replication Server 246

RSSD 252

プライマリ・データ・サーバ 226

## こ

## 更新

LTM ロケータ 199

RASD のログ・デバイス 142

## 構文、LTL 圧縮 185

## コネクション

pds\_connection\_type パラメータ 222

pds\_dac\_port\_number パラメータ 223

pds\_database\_name パラメータ 223

pds\_datasource\_name パラメータ 224

pds\_host\_name パラメータ 224, 225

pds\_port\_number パラメータ 226

pds\_retry\_count パラメータ 227

pds\_retry\_timeout パラメータ 227

pds\_server\_name パラメータ 227

Replication Server 文字セット 243

rs\_charset パラメータ 243

rs\_host\_name パラメータ 244

rs\_packet\_size パラメータ 245

rs\_password パラメータ 246

rs\_port\_number パラメータ 246

rs\_replicate\_owner\_required パラメータ 246

rs\_retry\_count パラメータ 247

rs\_retry\_timeout パラメータ 247

rs\_source\_db パラメータ 248

rs\_source\_ds パラメータ 248

rs\_username パラメータ 249

RSSD 文字セット 250

rssd\_charset パラメータ 250

rssd\_database\_name パラメータ 251

rssd\_host\_name パラメータ 251

rssd\_port\_number パラメータ 252

ダミー・コネクション 185

テスト 163

プライマリ・データ・サーバ文字セット  
243

文字セット 243, 250

## コネクションのテスト

## コマンド

log\_system\_name 11

lr\_dump\_marker 12

pdb\_capabilities 13

pdb\_date 13

pdb\_execute\_sql 14

pdb\_gen\_id 15

pdb\_get\_columns 16

pdb\_get\_databases 18

pdb\_get\_primary\_keys 19

pdb\_get\_procedure\_parms 20

pdb\_get\_procedures 22

pdb\_get\_sql\_database 24

pdb\_get\_tables 24

pdb\_send\_osuser\_list 28

pdb\_set\_sql\_database 30

pdb\_setrepcol 31

pdb\_setrepddl 36

pdb\_setrepproc 45

pdb\_setrepseq 54

pdb\_setreptable 56

pdb\_skip\_op 69

pdb\_truncate\_xlog 73

pdb\_version 75

pdb\_xlog 75

quiesce 80

ra\_config 86

ra\_date 88

ra\_deviceoffset 88

ra\_devicepath 89

ra\_downgrade 90

ra\_downgrade\_accept 91

ra\_downgrade\_prepare 93

ra\_dump 94

ra\_dumptran 95

ra\_help 99

ra\_helpparticle 101

ra\_helppdb 102

ra\_helpdevice 103

ra\_helpdeviceoffset 105

ra\_helpfield 107

ra\_helplocator 109

ra\_helppop 110

ra\_helpuser 113

ra\_license 115

ra\_locator 116

ra\_maint\_id 119

ra\_marker 120

ra\_migrate 121

ra\_set\_autocorrection 123

ra\_set\_login 125

ra\_statistics 126

ra\_status 138

ra\_truncatearticles 139

ra\_truncateddlfilters 140

ra\_truncateusers 141

## 索引

ra\_updatedevices 142, 186  
ra\_updateusers 144  
ra\_version 145  
ra\_version\_all 145  
rasd\_backup 146  
rasd\_helpbackup 147  
rasd\_removebackup 148  
rasd\_restore 149  
rs\_ticket 158  
suspend 162  
test\_connection 163  
ヘルプ情報 99  
レジューム 152  
停止 161

## さ

### 削除

トランザクション・ログ 75

### 作成

トランザクション・ログ 75

## し

時間データ型の変換 212  
システム・データ・レポジトリ  
バックアップ 146, 147  
リストア 149  
システム・ログ・ファイル 11  
実行中 SQL の現在のデータベース 14, 24, 30  
詳細なログ出力  
情報の取得  
Replication Agent ステータス 138  
Replication Agent のパフォーマンス 126  
Replication Agent の日付と時刻 88  
Replication Agent バージョン 145  
プライマリ・データベース・オブジェクト  
16, 24  
プライマリ・データベースのバージョン  
75  
プライマリ・データベースの日付と時刻  
13  
診断、詳細なロギング

## す

スクリプト  
ディレクトリ 75, 78, 81, 84

トランザクション・ログ作成 75  
自動的な実行 211  
スクリプトの自動的な実行 211  
ストアード・プロシージャ  
名前 22  
ストアード・プロシージャの複写の無効化  
すべてのストアード・プロシージャ 45  
ストアード・プロシージャの複写の有効化  
すべてのストアード・プロシージャ 51  
ストアード・プロシージャ 46  
RASD 内のアーティクル 139  
オブジェクト所有者 45  
マーク解除の強制 45  
マーク解除 45  
マーク付け 45  
複写の無効化 45  
複写の有効化 45  
返されたパラメータ 20  
名前の大文字と小文字 45  
名前の複写 45  
ストアード・プロシージャのマーク解除  
オプションの強制 45  
スクリプトの自動的な実行 211  
すべてのストアード・プロシージャ 45  
ストアード・プロシージャのマーク付け  
スクリプトの自動的な実行 211  
マーク付けステータス 50  
使用できない項目 45  
ストアード・プロシージャ・レプリケーション  
の無効化 45  
ストアード・プロシージャ・レプリケーション  
の有効化 45  
スレッド  
フィルタリング 72

## せ

### 設定

プライマリ・データベース・ログ・デバ  
イスのパス 89

### 設定パラメータ 169

admin\_port 179  
asm\_password 180  
asm\_tns\_connection 181  
asm\_tns\_filename 181  
asm\_username 182

auto\_adjust\_block\_count 183  
column\_compression 184  
compress\_ltl\_syntax 185  
connect\_to\_rs 185  
ddl\_password 186  
ddl\_username 186  
dump\_batch\_timeout 187  
filter\_maint\_userid 188  
function\_password 189  
function\_username 189  
log\_backup\_files 190  
log\_directory 190  
log\_read\_block\_count 191  
log\_trace\_verbos 192  
log\_wrap 192  
lr\_max\_lobdata\_cache 193  
lr\_max\_op\_queue\_size 193  
lr\_max\_scan\_queue\_size 193  
lr\_read\_buffer\_size 196  
lti\_batch\_mode 197  
lti\_max\_buffer\_size 198  
lti\_update\_trunc\_point 199  
ltl\_batch\_size 200  
ltl\_big\_endian\_unitext 200  
ltl\_character\_case 201  
ltl\_origin\_time\_required 202  
ltl\_send\_only\_primary\_keys 202  
ltm\_admin\_pw 204  
ltm\_admin\_pw\_min\_length 204  
ltm\_admin\_user 170, 205  
max\_ops\_per\_scan 205  
pdb\_archive\_path 206  
pdb\_archive\_remove 207  
pdb\_auto\_create\_repdefs 208  
pdb\_auto\_run\_scripts 211  
pdb\_automark\_tables 210  
pdb\_convert\_datetime 212  
pdb\_dflt\_column\_repl 214  
pdb\_dflt\_object\_repl 214  
pdb\_ignore\_unsupported\_anydata 215  
pdb\_include\_archives 217  
pdb\_ownerfilter 26  
pdb\_skip\_missing\_user 217  
pdb\_support\_large\_identifier 218  
pdb\_timezone\_file 219  
pdb\_xlog\_device 219  
pdb\_xlog\_prefix 220  
pdb\_xlog\_prefix\_chars 221  
pds\_connection\_type 222  
pds\_dac\_port\_number 223  
pds\_database\_name 223  
pds\_datasource\_name 224  
pds\_host\_name 224, 225  
pds\_integrated\_security 225  
pds\_password 226  
pds\_port\_number 226  
pds\_retry\_count 227  
pds\_retry\_timeout 227  
pds\_server\_name 227  
pds\_tns\_connection 228  
pds\_tns\_filename 228  
pds\_username 229  
ra\_admin\_device 81, 230  
ra\_admin\_instance\_prefix 81, 230  
ra\_admin\_owner 233  
ra\_admin\_prefix 231  
ra\_admin\_prefix\_chars 81, 232  
ra\_helparchive 100  
ra\_helptran 113  
ra\_purge\_first\_open 122  
ra\_retry\_count 234  
ra\_retry\_timeout 234  
ra\_standby 239  
ra\_statrack 137  
ra\_statrack\_interval 240  
ra\_statrack\_list 137  
rasd\_backup\_dir 235  
rasd\_database 235  
rasd\_mirror\_tran\_log 236  
rasd\_trace\_log\_dir 237  
rasd\_tran\_log 238  
rasd\_tran\_log\_mirror 239  
rman\_enabled 241  
rman\_password 242  
rman\_username 242  
rs\_charset 243  
rs\_host\_name 244  
rs\_packet\_size 245  
rs\_password 246  
rs\_port\_number 246  
rs\_replicate\_owner\_required 246  
rs\_retry\_count 247  
rs\_retry\_timeout 247  
rs\_source\_db 248  
rs\_source\_ds 248  
rs\_ticket\_version 249  
rs\_username 249  
rssd\_charset 250

## 索引

- rssd\_database\_name 251
- rssd\_host\_name 251
- rssd\_password 251
- rssd\_port\_number 252
- rssd\_username 252
- scan\_sleep\_increment 253
- scan\_sleep\_max 254
- skip\_lr\_errors 254
- skip\_ltl\_errors 255
- structured\_tokens 255
- truncation\_interval 256
- truncation\_type 257
- use\_rssd 258
- 設定 86
- 設定ファイルのフォーマット 169
  
- そ**
- 即時停止 161
- ソケット・ポート番号
  - Replication Server 246
  - RSSD 252
  - プライマリ・データ・サーバ 226
  
- た**
- ダウングレード 90, 91, 93
- ダミー・コネクション 185
  
- つ**
- 通信
  - JDBC ドライバ 222
  - ODBC ドライバ 222
  - Replication Server パラメータ 243, 249
  - RSSD パラメータ 252
  - コネクションのテスト 163
  - コネクティビティ・ドライバ 224
  - ドライバのバージョン 75, 145
  - ネットワーク・パケット・サイズ 245
  - プライマリ・データ・サーバ・パラメータ 229
  
- て**
- データ型
  - char (Sybase) 212
  - Sybase 以外の日/時の変換 212
  - 日時 (Sybase) 212
- データ・ソース名 (DSN) 224
- データベース 11
- データベース・コネクション
  - Replication Server 内 119
- データベース・オブジェクト 16
  - LOB カラム 31
  - LTL での名前の大文字と小文字 201
  - pdb\_xlog\_prefix 設定パラメータ 220
  - ra\_admin\_instance\_prefix configuration parameter 231
  - ra\_admin\_instance\_prefix 設定パラメータ 230
  - ra\_admin\_owner 設定パラメータ 233
- RASD のアーティクル 101
- system object name prefix 230, 231
- アーティクルのフィールド 107
- エイリアス、同義語、およびビュー 56
- カラム 16, 107
- システム・オブジェクト所有者 233
- ストアド・プロシージャ 45, 101
- テーブル 24, 101
- トランザクション・ログ・プレフィクス 220
- プライマリ・キー 19
- ユーザ 113, 141
- データベース・オペレーション
  - ヘルプ・コマンド 110
- データベース世代 ID
- データベース・デバイス
  - プライマリ・データベース・ログ・デバイス 89
  - プライマリ・データベースのミラー・ログ・デバイス 186
  - ヘルプ・コマンド 103
- データベース操作
  - トラブルシューティング 95
- テーブル、プライマリ・データベース
  - 次を参照：プライマリ・テーブル
- テーブルの複写の無効化
  - すべてのテーブル対象 66
- テーブルの複写の有効化
  - すべてのテーブル対象 66
- テーブル複写定義 56, 68

無効化テーブルの複写 56  
 テーブル複写の有効化 56  
 デフォルト 214

## と

同義語、データベース・オブジェクト  
 統計

追跡 137  
 追跡スレッド 137

統計情報、パフォーマンス 126

トラブルシューティング  
 LTL エラー 165, 255  
 ダミー・コネクション 185  
 ログ・レコード処理エラー 254  
 詳細なロギング 192

トラブルシューティング・コマンド  
 データベース・オペレーション 95

トランザクション  
 オープン 113

削除 122  
 トラブルシューティング 95  
 ヘルプ・コマンド 110

トランザクション・ログでのマーカ  
 ra\_marker オブジェクト 120

トランザクション・ログ  
 DDL オペレーション 139, 141  
 LTL でのオリジン時間 202  
 LTM ロケータ 116  
 ra\_marker オブジェクト 120  
 Replication Agent 75  
 Replication Agent オブジェクト 34, 35  
 rs\_dumpdb マーカ 94  
 rs\_dumptran マーカ 94

削除 75  
 作成 75  
 スキャン 254  
 データベースの世代 ID 15  
 トランケート 256  
 プライマリ・データベース・デバイス 89,  
 103, 142  
 プレフィクス 75, 81, 219  
 隠しテーブル 45, 63  
 作成スクリプト 75

トランザクション・ログのダンブ・マーカ 94

トランザクション・ログのマーカ

rs\_dumpdb マーカ 94  
 rs\_dumptran マーカ 94

トレース・ログ・ファイル

LTL 出力 165  
 RASD 237

次も参照： システム・ログ・ファイル

## な

名前

RASD データベース名 235  
 RSSD データベース名 251  
 ストアド・プロシージャ 22  
 ストアド・プロシージャの所有者 45  
 データベースによって返された名前 16  
 プライマリ・データ・サーバ 227  
 プライマリ・テーブルの所有者 56  
 プライマリ・データベース 18  
 ホスト・マシン 224, 244, 251

## ね

ネットワーク・パケット・サイズ 245

## は

バージョン

RASD のアーティクル 101, 107  
 RASD のプライマリ・データベース・ユー  
 ザ 113  
 Replication Agent 145  
 Replication Server 200  
 プライマリ・データ・サーバ 75

パス

RASD データベース・ファイル 235  
 RASD トランザクション・ログ 238  
 RASD トランザクション・ログ・ミラー  
 239  
 RASD トレース・ログ 237  
 RASD バックアップ・ディレクトリ 235  
 Replication Agent システム・ログ 166, 190  
 Replication Agent スクリプト・ディレク  
 トリ 75, 78, 81, 84  
 ミラー・ログ・デバイス 103, 186

## 索引

ログ・デバイス 89, 102, 103, 142  
パスワード  
Replication Agent の管理者 125  
Replication Server ユーザ・ログイン 246  
RSSD ユーザ・ログイン 251  
プライマリ・データベース・ユーザ・ログイン 226  
パスワードの長さ  
Replication Agent 管理者 204  
バックアップからの RASD のリストア 149  
バッチ・モード、LTL 197, 200  
バッファ、Log Transfer Interface  
パフォーマンス統計情報 126  
リセット 136  
パラメータ  
Replication Agent 設定 86  
ストアド・プロシージャ 20

## ひ

ビュー、データベース・オブジェクト  
表記規則  
スタイル 1  
構文 1

## ふ

ファイル  
LTL トレース・ログ 165  
RASD データベース・ファイル 235  
RASD トレース・ログ 237  
RASD バックアップ 235  
Replication Agent スクリプト・ディレクトリ 75, 78, 81, 84  
システム・ログ 11  
ミラーリングされた RASD トランザクション・ログ 239  
ファンクション複写定義 45  
複写定義  
オブジェクト名の大文字と小文字 45, 56, 201  
テーブル 56, 68  
関数 (ストアド・プロシージャ) 45  
プライマリ・キー・カラム 19  
プライマリ・データベース  
RASD でのユーザ・ログイン 113, 141

RASD 内のアーティクル 101  
Replication Agent からのコネクション 163  
Replication Agent ユーザ・ログイン 229  
Replication Server データベース・コネクション 119  
Replication Server のソース定義 248  
Replication Server でのデータベース・コネクション 45, 68  
Replication Server へのデータベース・コネクション 56  
SQL コマンド 14, 24, 30  
ゲートウェイ 75  
コネクションのテスト 163  
サーバ・ポート番号 226  
サーバのバージョン 75  
サーバの日付と時刻 13  
サーバ名 227  
ストアド・プロシージャ 20, 22  
データベース名 18  
デバイス名 219  
バージョン 145  
プライマリ・キー 19  
ホスト・マシンの名前 224  
文字セット 243  
ログ・デバイスの更新 142  
ログ・デバイス 103, 142  
世代 ID 15  
通信ドライバ 145  
返されたオブジェクト名 16  
返されたコラム名 16  
プライマリ・データベースのゲートウェイ 75  
プライマリ・データベースの世代 ID  
プライマリ・データベースのデバイス名 219  
プライマリ・テーブル  
LOB カラム 31  
RASD 内のアーティクル 139  
オブジェクト所有者 56  
データベースからリストを取得 24  
テーブル名 24, 56  
プライマリ・キー 19  
マーク解除の強制 56, 64, 66  
マーク解除 56  
マーク付け 56  
マーク付けステータス 63



複写の無効化 56  
 複写の有効化 214  
 名前の大文字と小文字 56  
 プライマリ・テーブルのマーク解除  
   スクリプトの自動的な実行 211  
   すべてのテーブル 66  
   強制オプション 56, 64, 66  
 プライマリ・テーブルのマーク付け  
   スクリプトの自動的な実行 211  
   すべてのユーザ・テーブル 66  
   マーク付けステータス 63  
   使用できない項目 56  
 プレフィクス、トランザクション・ログ 75,  
   81  
**へ**  
 ヘルプ  
   コマンド 99  
 ヘルプ・コマンド  
   LTM ロケータ 109  
   RASD のプライマリ・データベース・ユー  
   ザ 113  
   RASD 内のアーティクル 101  
   アーティクルのフィールド 107  
   データベース・オペレーション 110  
   プライマリ・データベース・ログ・デバ  
   イス 103  
   プライマリ・データベース 102  
 変更  
   プライマリ・データベース・ログ・デバ  
   イスのパス 89  
**ほ**  
 ポート番号  
   Replication Agent 179  
   Replication Server 246  
   RSSD 252  
   プライマリ・データ・サーバ 226  
 ホスト・マシン  
   Replication Agent 145, 179  
   Replication Server 244  
   RSSD 251  
   プライマリ・データ・サーバ 224  
**ま**  
 マーク解除の強制 46  
   ストアド・プロシージャ 45

テーブル 56, 64, 66

## み

ミラーリングされた RASD トランザクション・  
   ログ  
 ミラー・ログ・デバイス、プライマリ・デー  
   タベース  
   RASD での更新 186  
   ロケーションのパス 103

## め

メンテナンス・ユーザ  
   Log Reader によるフィルタ 188

## も

文字セット  
   Replication Server 243  
   RSSD 250  
   プライマリ・データ・サーバ 243

## ゆ

ユーザ ID  
   RASD のプライマリ・データベース・ユー  
   ザ 113  
   Replication Agent の管理者 125, 204  
   Replication Server 249  
   RSSD ユーザ・ログイン 251, 252  
   プライマリ・データベース 226, 229

## ら

ライセンス情報 115

## れ

レプリケート・テーブル  
   複写定義で指定された名前 68  
 レポジトリ  
   システム・データ 149, 235  
   プライマリ・データベース・ログ・デバ  
   イス 89, 103, 142

## 索引

### ろ

- ログ・デバイス
  - ヘルプ・コマンド 103
  - ミラー・ログ・デバイスのパス 103
  - ログ・デバイス・レポジトリの更新 142
  - ログ・デバイスへのパス 89
- ログ・デバイス、プライマリ・データベース
  - RASD での更新 142
  - ロケーションへのパス 89
- ログ・ファイル
  - RASD トランザクション・ログ 236
  - RASD トレース・ログ 237
  - Replication Agent システム・ログ 11, 190
  - トランザクション・ログのトランケート 73
  - ラッピング 192
- ログ・ファイルのサイズ 192
- ログ・ファイルのラッピング 192
- ログ・メタデータ
  - 表示 100
- ログ・レコード処理
  - エラー・メッセージ 254