

SYBASE®

Reference Manual

Replication Agent™

15.5

[Linux, Microsoft Windows, and UNIX]

DOCUMENT ID: DC00268-01-1550-01

LAST REVISED: May 2010

Copyright © 2010 by Sybase, Inc. All rights reserved.

This publication pertains to Sybase software and to any subsequent release until otherwise indicated in new editions or technical notes. Information in this document is subject to change without notice. The software described herein is furnished under a license agreement, and it may be used or copied only in accordance with the terms of that agreement.

To order additional documents, U.S. and Canadian customers should call Customer Fulfillment at (800) 685-8225, fax (617) 229-9845.

Customers in other countries with a U.S. license agreement may contact Customer Fulfillment via the above fax number. All other international customers should contact their Sybase subsidiary or local distributor. Upgrades are provided only at regularly scheduled software release dates. No part of this publication may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without the prior written permission of Sybase, Inc.

Sybase trademarks can be viewed at the Sybase trademarks page at <http://www.sybase.com/detail?id=1011207>. Sybase and the marks listed are trademarks of Sybase, Inc. ® indicates registration in the United States of America.

Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

Unicode and the Unicode Logo are registered trademarks of Unicode, Inc.

All other company and product names mentioned may be trademarks of the respective companies with which they are associated.

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.

Contents

About This Book	ix
------------------------------	-----------

CHAPTER 1	Command Reference	1
	Command reference table	1
	log_system_name	6
	lr_dump_marker	7
	pdb_asmdiskmap	7
	pdb_capabilities	8
	pdb_commit_generation_filter	9
	pdb_commit_generation_status	10
	pdb_date	11
	pdb_execute_sql	12
	pdb_gen_id	13
	pdb_get_columns	14
	pdb_get_databases	16
	pdb_get_primary_keys	17
	pdb_get_procedure_parms	18
	pdb_get_procedures	20
	pdb_get_sql_database	22
	pdb_get_tables	22
	pdb_ownerfilter	24
	pdb_send_osuser_list	27
	pdb_set_sql_database	29
	pdb_setrepcol	30
	pdb_setrepddl	35
	pdb_setrepproc	43
	pdb_setrepseq	54
	pdb_setreptable	56
	pdb_skip_op	71
	pdb_truncate_xlog	73
	pdb_version	75
	pdb_xlog	75
	quiesce	80
	ra_config	81

ra_date	83
ra_deviceoffset.....	84
ra_devicepath.....	85
ra_downgrade_accept.....	86
ra_downgrade_prepare.....	87
ra_dump	87
ra_dumptran.....	88
ra_help	93
ra_helparticle.....	94
ra_helpdb	95
ra_helpdevice.....	96
ra_helpdeviceoffset.....	98
ra_helpfield.....	99
ra_helplocator	101
ra_helpop	102
ra_helpuser	106
ra_locator	107
ra_maintid	109
ra_marker.....	110
ra_migrate	111
ra_set_autocorrection	112
ra_set_login.....	114
ra_statistics	115
ra_status	125
ra_truncatearticles.....	127
ra_truncateddfilters.....	128
ra_truncateusers	128
ra_updatedevices.....	129
ra_version	131
ra_version_all.....	132
rasd_backup.....	133
rasd_helpbackup.....	133
rasd_removebackup.....	134
rasd_restore.....	135
rasd_trunc_schedule.....	136
resume	139
rs_create_repdef	141
rs_drop_repdef.....	144
rs_ticket.....	145
server_xlog.....	146
shutdown.....	147
suspend.....	148
test_connection	149
trace	152

CHAPTER 2	Configuration Parameters	157
	Configuration parameter overview	157
	Replication Agent configuration file	157
	Changing configuration parameters	158
	Copying a Replication Agent configuration	159
	Configuration parameter reference	159
	admin_port	166
	archive_read_block_count	167
	asm_password	167
	asm_tns_connection	168
	asm_tns_filename	168
	asm_username.....	169
	auto_adjust_block_count.....	170
	column_compression	170
	compress_ltl_syntax.....	171
	connect_to_rs.....	171
	ddl_password	172
	ddl_username.....	173
	dump_batch_timeout.....	174
	filter_maint_userid	174
	function_password	175
	function_username	175
	log_backup_files.....	176
	log_directory.....	176
	log_read_block_count	177
	log_trace_verbose	178
	log_wrap.....	178
	lr_max_scan_queue_size.....	179
	lr_max_op_queue_size	179
	lr_ntext_byte_order	179
	lr_parallel_scan	181
	lr_read_buffer_size.....	182
	lr_send_trunc_partition_ddl.....	182
	lti_batch_mode.....	183
	lti_max_buffer_size	184
	lti_update_trunc_point.....	184
	ltl_batch_size.....	185
	ltl_big_endian_unitext.....	186
	ltl_character_case	186
	ltl_origin_time_required	187
	ltl_send_only_primary_keys	188
	ltm_admin_pw	188
	ltm_admin_user.....	189
	max_ops_per_scan	190

pdb_archive_path.....	190
pdb_archive_remove.....	191
pdb_auto_create_repdefs	192
pdb_automark_tables.....	194
pdb_auto_run_scripts.....	195
pdb_commit_generation.....	196
pdb_commit_interval	197
pdb_convert_datetime.....	197
pdb_dflt_column_repl.....	199
pdb_dflt_object_repl.....	200
pdb_ignore_unsupported_anydata.....	201
pdb_include_archives.....	202
pdb_skip_missing_user.....	203
pdb_support_large_identifier.....	203
pdb_timezone_file	204
pdb_xlog_device	204
pdb_xlog_prefix.....	205
pdb_xlog_prefix_chars	205
pds_connection_type	206
pds_dac_port_number	207
pds_database_name.....	208
pds_datasource_name.....	208
pds_host_name.....	209
pds_integrated_security	209
pds_password	210
pds_port_number	210
pds_retry_count.....	210
pds_retry_timeout.....	211
pds_server_name.....	211
pds_tns_connection	212
pds_tns_filename	212
pds_username.....	213
ra_retry_count	213
ra_retry_timeout	214
rasd_backup_dir.....	214
rasd_database.....	215
rasd_mirror_tran_log.....	216
rasd_trace_log_dir.....	217
rasd_tran_log	218
rasd_tran_log_mirror	219
ra_standby.....	219
rman_enabled	220
rman_password.....	221
rman_username	221

rs_charset.....	222
rs_host_name.....	223
rs_packet_size	224
rs_password.....	224
rs_port_number	225
rs_replicate_owner_required.....	225
rs_retry_count	225
rs_retry_timeout	226
rs_source_db.....	226
rs_source_ds.....	226
rs_ticket_version	227
rs_username	227
rssd_charset.....	228
rssd_database_name.....	228
rssd_host_name.....	229
rssd_password	229
rssd_port_number	230
rssd_username.....	230
scan_eol_threshold	231
scan_sleep_increment	231
scan_sleep_max	232
skip_ltl_errors	232
structured_tokens.....	233
truncation_interval	234
truncation_type.....	234
use_rssd.....	235

Glossary	237
-----------------------	------------

Index	245
--------------------	------------

About This Book

Replication Agent™ extends the capabilities of Replication Server® to support the following non-Sybase® primary data servers in a Sybase replication system:

- DB2 Universal Database (on UNIX and Microsoft Windows platforms)
- Microsoft SQL Server
- Oracle Database Server

Audience

This book is for anyone who manages or administers a Sybase replication system with primary databases, or administers the primary databases in a Sybase replication system. This may include:

- Database administrators
- Network administrators
- System administrators

How to use this book

Use the *Replication Agent Reference Manual* to look up detailed information about Replication Agent commands and configuration parameters.

This book is organized as follows:

Chapter 1, “Command Reference,” describes all Replication Agent commands, including syntax, options, examples, and detailed command usage notes.

Chapter 2, “Configuration Parameters,” describes the Replication Agent configuration file, and provides a configuration parameter reference.

Related documents

Replication Agent Refer to the following documents to learn more about the Replication Agent:

- *Replication Agent Administration Guide* – for an overview of the Replication Agent, information about configuring and administering Replication Agent instances, and information about configuring the other components in a Sybase replication system.

-
- *Replication Agent Primary Database Guide* – for detailed, database-specific information about each non-Sybase database that is supported by the Replication Agent.
 - *Replication Agent Installation Guide* – for information about installing the Replication Agent software.
 - *Replication Agent Release Bulletin* – for last-minute information that was too late to be included in the books.

Note A more recent version of the *Replication Agent Release Bulletin* may be available on the World Wide Web. To check for critical product or document information that was added after the release of the product CD, use the Sybase Product Manuals Web site.

Replication Server Refer to the following documents for more information about transaction replication systems and the Replication Server software:

- *Replication Server Design Guide* – for an introduction to basic transaction replication concepts and Sybase replication technology.
- *Replication Server Heterogeneous Replication Guide* – for detailed information about configuring Replication Server and implementing a Sybase replication system with non-Sybase databases.

Primary data server Make sure that you have appropriate documentation for the non-Sybase primary data server that you use with the Sybase replication system.

Java environment The Replication Agent requires a Java Runtime Environment (JRE) on the Replication Agent host machine.

- The Replication Agent release bulletin contains the most up-to-date information about Java and JRE requirements.
- Java documentation available from your operating system vendor describes how to set up and manage the Java environment on your platform.

Other sources of information

Use the Sybase Getting Started CD and the Sybase Product Manuals Web site to learn more about your product:

- The Getting Started CD contains release bulletins and installation guides in PDF format. It is included with your software. To read or print documents on the Getting Started CD, you need Adobe Acrobat Reader, which you can download at no charge from the Adobe Web site using a link provided on the CD.

- The Sybase Product Manuals Web site is an online version of the books that you can access using a standard Web browser. In addition to product manuals, you will find links to EBFs/Maintenance, Technical Documents, Case Management, Solved Cases, newsgroups, and the Sybase Developer Network.

To access the Sybase Product Manuals Web site, go to Product Manuals at <http://www.sybase.com/support/manuals/>.

Sybase certifications on the Web

Technical documentation at the Sybase Web site is updated frequently.

❖ Finding the latest information on product certifications

- 1 Point your Web browser to Technical Documents at <http://www.sybase.com/support/techdocs/>.
- 2 Click Partner Certification Report.
- 3 In the Partner Certification Report filter select a product, platform, and timeframe and then click Go.
- 4 Click a Partner Certification Report title to display the report.

❖ Finding the latest information on component certifications

- 1 Point your Web browser to Availability and Certification Reports at <http://certification.sybase.com/>.
- 2 Either select the product family and product under Search by Base Product; or select the platform and product under Search by Platform.
- 3 Select Search to display the availability and certification report for the selection.

❖ Creating a personalized view of the Sybase Web site (including support pages)

Set up a MySybase profile. MySybase is a free service that allows you to create a personalized view of Sybase Web pages.

- 1 Point your Web browser to Technical Documents at <http://www.sybase.com/support/techdocs/>.
- 2 Click MySybase and create a MySybase profile.

Sybase EBFs and software maintenance

❖ Finding the latest information on EBFs and software maintenance

- 1 Point your Web browser to the Sybase Support Page at <http://www.sybase.com/support>.
- 2 Select EBFs/Maintenance. If prompted, enter your MySybase user name and password.
- 3 Select a product.
- 4 Specify a time frame and click Go. A list of EBF/Maintenance releases is displayed.

Padlock icons indicate that you do not have download authorization for certain EBF/Maintenance releases because you are not registered as a Technical Support Contact. If you have not registered, but have valid information provided by your Sybase representative or through your support contract, click Edit Roles to add the “Technical Support Contact” role to your MySybase profile.

- 5 Click the Info icon to display the EBF/Maintenance report, or click the product description to download the software.

Style conventions

The following style conventions are used in this book:

- In a sample screen display, commands that you should enter exactly as shown appear like this:

```
pdb_xlog
```

- In the regular text of this document, variables or user-supplied words appear like this:

Specify the *value* option to change the setting of the configuration parameter.

- In a sample screen display, variables or words that you should replace with the appropriate value for your site appear like this:

```
resume connection to pds.pdb
```

where *pds* and *pdb* are the variables you should replace.

- In the regular text of this document, names of programs, utilities, procedures, and commands appear like this:

Use the `pdb_xlog` command to initialize the primary database.

- In the regular text of this document, names of database objects (tables, columns, stored procedures, etc.) appear like this:
Check the price column in the widgets table.
- In the regular text of this document, names of datatypes appear like this:
Use the date or datetime datatype.
- In the regular text of this document, names of files and directories appear like this:
Log files are located in the `$SYBASE/RAX-15_5/inst_name/log` directory.

Syntax conventions

The following syntax conventions are used in this book:

Table 1: Syntax conventions

Key	Definition
{ }	Curly braces indicate that you must choose at least one of the enclosed options. Do not type the braces when you enter the command.
[]	Brackets mean that choosing one or more of the enclosed options is optional. Do not type the brackets when you enter the command.
()	Parentheses are to be typed as part of the command.
	The vertical bar means you can select only one of the options shown.
,	The comma means you can choose as many of the options shown as you like, separating your choices with commas that you type as part of the command.

In reference sections of this document, statements that show the syntax of commands appear like this:

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

The words *param* and *value* in the syntax are variables or user-supplied words.

Character case conventions

The following character case conventions are used in this book:

- All command syntax and command examples are shown in lowercase. However, Replication Agent command names are *not* case-sensitive. For example, RA_CONFIG, Ra_Config, and ra_config are equivalent.
- Names of configuration parameters are case-sensitive. For example, Scan_Sleep_Max is not the same as scan_sleep_max, and the former would be interpreted as an invalid parameter name.
- Database object names are *not* case-sensitive in Replication Agent commands. However, to use a mixed-case object name in a Replication Agent command (to match a mixed-case object name in the primary database), delimit the object name with quote characters. For example:

```
pdb_get_tables "TableName"
```

Accessibility features

This document is available in an HTML version that is specialized for accessibility. You can navigate the HTML with an adaptive technology such as a screen reader, or view it with a screen enlarger.

Replication Agent™ and the HTML documentation have been tested for compliance with U.S. government Section 508 Accessibility requirements. Documents that comply with Section 508 generally also meet non-U.S. accessibility guidelines, such as the World Wide Web Consortium (W3C) guidelines for Web sites.

The online help for this product is also provided in HTML, which you can navigate using a screen reader.

Note You might need to configure your accessibility tool for optimal use. Some screen readers pronounce text based on its case; for example, they pronounce ALL UPPERCASE TEXT as initials, and MixedCase Text as words. You might find it helpful to configure your tool to announce syntax conventions. Consult the documentation for your tool.

For information about how Sybase supports accessibility, see Sybase Accessibility at <http://www.sybase.com/accessibility>. The Sybase Accessibility site includes links to information on Section 508 and W3C standards.

For information about how Sybase supports accessibility, see Sybase Accessibility at <http://www.sybase.com/accessibility>. The Sybase Accessibility site includes links to information on Section 508 and W3C standards.

For a Section 508 compliance statement for Replication Agent 15.5, see Sybase Accessibility at http://www.sybase.com/detail_list?id=52484.

If you need help

Each Sybase installation that has purchased a support contract has one or more designated people who are authorized to contact Sybase Technical Support. If you cannot resolve a problem using the manuals or online help, have the designated person contact Sybase Technical Support or the Sybase subsidiary in your area.

This chapter describes the Replication Agent commands including targets, syntax, options, examples, and command usage.

Command reference table

This section describes the Replication Agent commands including targets, syntax, options, examples, and command usage. In Table 1-1, the targets column indicates when the command is used for “All” targets or specific targets; Oracle, Microsoft SQL Server (MSSQL), or DB2 UDB (UDB).

Table 1-1: Replication Agent commands

Command name	Target	Description
log_system_name	All	Returns the path to the Replication Agent system log file.
lr_dump_marker	Oracle	Returns or sets the system change number (SCN), indicating the point at which the primary database was last dumped.
pdb_asmdiskmap	Oracle	Creates a file with disk mirror paths for all ASM disks that online redo logs and archive logs are stored on.
pdb_capabilities	All	Returns a list of the Replication Agent capabilities.
pdb_commit_generation_filter	Oracle	Returns and modifies a list of Oracle instances that are filtered for commit log record generation.
pdb_commit_generation_status	Oracle	Returns a status for each Replication Agent thread that generates commit log records for an Oracle instance.
pdb_date	All	Returns the current date and time from the primary data server.
pdb_execute_sql	All	Executes the specified SQL statement in the current database.

Command name	Target	Description
pdb_gen_id	All	Returns the current value of the database generation ID; updates the value of the database generation ID.
pdb_get_columns	All	Returns a list of all the columns in the specified table.
pdb_get_databases	All	Returns a list of all the databases in the primary data server.
pdb_get_primary_keys	All	Returns a list of all the columns that make up the primary keys in the specified table.
pdb_get_procedure_parms	Oracle MSSQL	Returns a list of the properties for the specified procedure.
pdb_get_procedures	Oracle MSSQL	Returns a list of all the procedures in the specified database.
pdb_get_sql_database	All	Returns the name of the database specified for SQL statement execution.
pdb_get_tables	All	Returns a list of all the tables in the specified database.
pdb_ownerfilter	Oracle MSSQL	Returns a list of owners whose objects will be filtered for initialization; adds and removes owners to the list.
pdb_send_osuser_list	Oracle	Returns and modifies a list of database users whose primary database user names will be replaced with the corresponding operating system user name in LTL sent to Replication Server.
pdb_set_sql_database	All	Specifies the database to be used for SQL statement execution.
pdb_setrepcol	All	Returns replication marking status; enables or disables replication for all marked columns or a specified column.
pdb_setreppddl	Oracle MSSQL	Returns DDL replication status; enables or disables replication for DDL statements.
pdb_setrepproc	Oracle MSSQL	Changes and reports stored procedure marking status.
pdb_setreppseq	Oracle	Changes and reports sequence replication marking status.
pdb_setreptable	All	Changes and reports table replication marking status.

Command name	Target	Description
pdb_skip_op	Oracle MSSQL	Returns, adds, or removes record identifiers from a list of records to skip in processing.
pdb_truncate_xlog	All	Truncates the Replication Agent primary database transaction log.
pdb_version	All	Returns the type and version of the primary data server.
pdb_xlog	All	Returns names of transaction log objects; creates transaction log base objects in the primary database; removes transaction log base objects from the primary database.
quiesce	All	Stops current Log Reader activity after all data remaining in the transaction log and Replication Agent internal queues is processed and puts Replication Agent in <i>Admin</i> state.
ra_config	All	Returns help information for configuration parameters; sets the value of a configuration parameter.
ra_date	All	Returns the current date and time from the Replication Agent server.
ra_deviceoffset	Oracle MSSQL	Changes the raw disk device read offset for a log device recorded in the RASD.
ra_devicepath	Oracle MSSQL	Changes the disk device path for a log device recorded in the RASD.
ra_downgrade_accept	All	Completes the version downgrade process initiated by the instance from which Replication Agent is being downgraded.
ra_downgrade_prepare	All	Prepares Replication Agent for a version downgrade.
ra_dump	All	Records a dump marker in the primary database transaction log.
ra_dumptran	Oracle	Returns information for use in troubleshooting a specific database transaction.
ra_help	All	Returns help information for Replication Agent commands.
ra_helparticle	Oracle MSSQL	Returns information about articles from the RASD.

Command name	Target	Description
ra_helpdb	Oracle MSSQL	Returns information about the primary database from the RASD.
ra_helpdevice	Oracle MSSQL	Returns information about primary database log devices from the RASD.
ra_helpdeviceoffset	Oracle MSSQL	Returns device offset information about primary database log devices from the RASD log device repository.
ra_helpfield	Oracle MSSQL	Returns information about fields (columns in tables, or input parameters in stored procedures) from the RASD.
ra_helplocator	Oracle MSSQL	Returns LTM locator field values.
ra_helpop	Oracle	Returns information for use in troubleshooting a specific database transaction log operation.
ra_helpuser	Oracle MSSQL	Returns information about primary database users from the RASD.
ra_locator	All	Returns and changes the current value of the LTM Locator stored by Replication Agent.
ra_maintid	All	Returns the maintenance user for the Replication Agent connection.
ra_marker	All	Records a marker in the primary database transaction log.
ra_migrate	All	Performs migration tasks (as necessary) between releases of Replication Agent.
ra_set_autocorrection	All	Enables or disables autocorrection for marked tables.
ra_set_login	All	Sets the Replication Agent admin user login and password.
ra_statistics	All	Returns statistics for either a specified Replication Agent component or all components, and resets statistics for all components.
ra_status	All	Returns the current Replication Agent state.
ra_truncatearticles	Oracle MSSQL	Truncates older versions of primary database articles in the system data repository in the RASD.
ra_truncateddilters	Oracle	Truncates old lists of DDL commands that are filtered in the RASD.

Command name	Target	Description
ra_truncateusers	Oracle MSSQL	Truncates older versions of primary database users in the system data repository in the RASD.
ra_updatedevices	Oracle MSSQL	Updates the log device repository in the RASD.
ra_version	All	Returns the Replication Agent version.
ra_version_all	All	Returns Replication Agent, primary data server, Replication Server, and communications driver versions.
rasd_backup	Oracle MSSQL	Backs up the Replication Agent System Database (RASD).
rasd_helpbackup	Oracle MSSQL	Displays a list of RASD backups.
rasd_removebackup	Oracle MSSQL	Removes RASD backups.
rasd_restore	Oracle MSSQL	Restores the Replication Agent System Database (RASD).
rasd_trunc_schedule	Oracle MSSQL	Returns a list of the repository truncation weekly schedule; also adds or removes a specific schedule.
resume	All	Starts replication for the current active log and puts Replication Agent in <i>Replicating</i> state.
rs_create_repdef	Oracle MSSQL	Creates a replication definition at Replication Server for a marked table and procedure, or for all marked tables and procedures.
rs_drop_repdef	Oracle MSSQL	A replication definition for a table is dropped at the Replication Server.
rs_ticket	All	Supports Replication Server <code>rs_ticket</code> processing by placing an <code>rs_ticket</code> marker in the Primary database transaction log.
server_xlog	MSSQL	Creates or removes transaction log base objects in the Microsoft SQL Server system resource database. This command is used for first-time initialization of Replication Agent for Microsoft SQL Server and should be run before <code>pdb_xlog init</code> .
shutdown	All	Shuts down Replication Agent.

Command name	Target	Description
suspend	All	Immediately stops all Log Reader activity, drops connections, and puts Replication Agent in <i>Admin</i> state.
test_connection	All	Tests Replication Agent connectivity.
trace	All	Returns current trace flag settings; changes a specified trace flag.

log_system_name

Description	Returns the full path of the Replication Agent instance log file.
Syntax	log_system_name
Usage	<ul style="list-style-type: none">When you create a Replication Agent instance, a log directory is created automatically as part of the instance directory structure. The default value of the log_directory parameter points to that directory.The default path of the Replication Agent log directory is: <pre>%SYBASE%\RAX-15_5\inst_name\log\</pre>where:<ul style="list-style-type: none"><code>%SYBASE%</code> is the Replication Agent installation directory.<code>inst_name</code> is the name of the Replication Agent instance.If you specify a valid directory path as the value of the log_directory parameter, the Replication Agent instance places its system log file in the directory you specify. If you change the value of the log_directory parameter with the ra_config command, the new value is recorded in the configuration file immediately, but you must shut down and restart the Replication Agent instance to make the new value take effect. See the log_directory parameter in Chapter 2, “Configuration Parameters” for more information. <ul style="list-style-type: none">The log_system_name command is valid when the Replication Agent instance is in the <i>Admin</i>, <i>Replicating</i>, or <i>Replication Down</i> state.
See also	ra_config, trace

lr_dump_marker

Description Returns or sets the system change number (SCN), indicating the last committed transaction in the primary database dump. This value is sent to Replication Server when Replication Agent encounters a log record with an SCN greater than or equal to this value.

Note This command is available only for Oracle.

Syntax lr_dump_marker [*scn*]

Parameters *scn*
The SCN indicating the point at which the primary database was last dumped. This is the dump database marker and denotes the oldest committed transaction in the dump.

Usage

- lr_dump_marker invoked with no option returns the SCN marking the point at which the primary database was last dumped. If no SCN has yet been specified, lr_dump_marker returns 0.
- To set an SCN dump point, invoke lr_dump_marker with a valid SCN marking the point at which the primary database was last dumped. If you made the dump with the Oracle Recovery Manager (RMAN) utility, you can obtain this SCN by using the RMAN list backup command.
- The SCN set with the lr_dump_marker command takes effect when Replication Agent is in the *Replicating (Resynchronization)* state.

pdb_asmdiskmap

Description Creates a file with default disk paths for all disks in the Automatic Storage Management (ASM) disk groups that are used to store online redo logs and archive redo logs. The disks required for online logs are automatically selected by querying ASM. The disks required for archive redo logs are specified by the archive log path parameter and are queried by ASM for the disks in the specified group. An entry for each disk in online redo log disk group and archive redo log disk group is added to the ASM disk map file. The default entry is the same path as the system path. The user must edit the file to change the map path to the mirror device.

When `pdb_asmdiskmap` is executed and the ASM disk map file already exists, new disk paths are obtained from ASM and added to the file. In this case, new disk paths are added only when disks have been added to the ASM disk groups used by redo logs.

Note This command is available only for Oracle.

Syntax

`pdb_asmdiskmap`

Usage

- The disk map file is named `<instance_name>_ASMDisks.props` and is created in the Replication Agent instance directory.

For example:

```
<install_root>\<instance>
\<instance_name>_ASMDisks.props
```

If `<install_root>` is `%SYBASE%\sybase\MA-15_5` the map file is created in:

```
%SYBASE%\sybase\MA-15_5\mra_ASMDisks.props
```

- The Replication Agent uses the disk map file to identify the mirror log devices when log devices are created during transaction log initialization and when devices are updated using the `ra_updatedevices` command. When Replication Agent is in *Replicating* state, it reads data from the mirrored disks specified in the map file. For more detailed information, see the `ra_updatedevices` command.

pdb_capabilities

Description

Returns a list of Replication Agent capabilities, which is used by the replication management tools.

Syntax

`pdb_capabilities`

Usage

- When `pdb_capabilities` is invoked, it returns a list of the capabilities of the Replication Agent instance.
- The purpose of the `pdb_capabilities` command is to support the replication management tools.
- The `pdb_capabilities` command is valid when the Replication Agent instance is in the *Admin*, *Replicating*, or *Replication Down* state.

pdb_commit_generation_filter

Description Returns and modifies a list of Oracle RAC database instances (or nodes) that are filtered for commit log record generation.

Note This command is available only for Oracle and is only applicable to Oracle RAC databases.

Syntax `pdb_commit_generation_filter`
`{ [add, inst_name] | [remove, [inst_name | all]] }`

Parameters `add`
 To add an Oracle instance name to the list of instances filtered for commit log record generation, follow the `pdb_commit_generation_filter` command with the `add` keyword and the `inst_name` parameter specifying the Oracle instance name:

```
pdb_commit_generation_filter add, inst_name
```

`remove`
 To remove an Oracle instance from the list of instances filtered for commit log record generation, follow the `pdb_commit_generation_filter` command with the `remove` keyword and the `inst_name` parameter specifying the Oracle instance name:

```
pdb_commit_generation_filter remove, inst_name
```

To remove all Oracle instances from the list of instances filtered for commit log record generation, follow the `pdb_commit_generation_filter` command with the `remove` keyword and `all` keyword:

```
pdb_commit_generation_filter remove, all
```

To list the Oracle instances that are currently being filtered for commit log record generation, enter the `pdb_commit_generation_filter` command alone:

```
pdb_commit_generation_filter
```

Examples **Example 1**

```
pdb_commit_generation_filter
```

This command lists Oracle RAC database instances that are currently being filtered for commit log record generation.

Example 2

```
pdb_commit_generation_filter add, ora_inst1
```

This command adds the Oracle instance named ora_inst1 to the list of instances filtered for commit log record generation.

Example 3

```
pdb_commit_generation_filter remove, ora_inst1
```

This command removes the Oracle instance named ora_inst1 from the list of instances filtered for commit log record generation.

Example 4

```
pdb_commit_generation_filter remove, all
```

This command removes all Oracle instance names from the list of instances filtered for commit log record generation.

Usage

- The instance name specified in the `pdb_commit_generation_filter` command represents the primary database instance being filtered for commit log record generation.
- Filtering should only be used in an Oracle RAC environment to allow the user to control which nodes generate commit log records. For example, in a database with both active and inactive nodes, commit log record generation has no effect on an active node, but it is desirable for passive nodes to reduce replication latency.

See also

`pdb_commit_generation_status`

pdb_commit_generation_status

Description

Returns a status for each Replication Agent thread that generates commit log records for an Oracle instance.

Note This command is available only for Oracle and is only applicable to Oracle RAC databases.

Syntax

```
pdb_commit_generation_status
```

Examples

```
pdb_commit_generation_status
```

This command returns status for four threads that generate commit log records for the oradb1, oradb2, oradb3, and oradb4 Oracle instances:

```
Instance Name State  
-----
```



```

oradb1      RUNNING
oradb2      RUNNING
oradb3      RUNNING
oradb4      RUNNING

```

Usage

There are seven thread states:

Table 1-2: Commit-generation thread states

State	Description
ERRORED	The commit-generation thread encountered an error while generating commit log records in the primary database and is no longer running.
FILTERED	The commit-generation thread is filtered and is not generating generating commit log records in the primary database
REQUESTED TO FILTER	The running commit-generation thread received a filter request and is transitioning to the FILTERED state.
RUNNING	The commit-generation thread is generating commit log records in the primary database.
STARTING	The commit-generation thread is in the process of starting to generate commit log records.
STOPPED	The commit-generation thread has stopped generating commit log records after receiving a request to stop.
STOP REQUESTED	The commit-generation thread is generating commit log records but has received a request to stop. The thread is transitioning to the STOPPED state.

See also

`pdb_commit_generation_filter`

pdb_date

Description

Returns the current date and time from the primary data server.

Syntax

`pdb_date`

Usage

- When `pdb_date` is invoked, it returns the current date and time from the primary data server in the form of a Sybase datetime datatype, as follows:

```

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

```

- The `pdb_date` command is valid when the Replication Agent instance is in the *Admin*, *Replicating*, or *Replication Down* state.

See also `ra_date`

pdb_execute_sql

Description Executes a SQL statement in the current database at the primary data server.

Syntax `pdb_execute_sql statement`

Parameters *statement*

A string in the form of a SQL statement enclosed in double quotes.

Usage

- The Replication Agent instance executes the specified SQL statement against the “current” database.

The current database is either:

- The *default* current database, which is the primary database specified in the Replication Agent `pds_database_name` configuration parameter, or
- The database specified in the `pdb_set_sql_database` command (to which the Replication Agent instance is currently connected).
- To set or change the current database, use the `pdb_set_sql_database` command.
- To find the name of the current database, use the `pdb_get_sql_database` command.

Note If the `pdb_set_sql_database` command has not been invoked to set or change the current database, the `pdb_get_sql_database` command returns the name of the default current database.

- The SQL statement specified in the `pdb_execute_sql` command must be a single SQL command enclosed in double quotes. For example:

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

The string is passed directly to the database for execution. No command to terminate is required and no syntax or other validation is performed.

- Any results returned from execution of the SQL statement are passed to the Replication Agent administrative client, by way of the Replication Agent administration port.
- The `pdb_execute_sql` command is valid when the Replication Agent instance is in the *Admin*, *Replicating*, or *Replication Down* state.

See also

`pdb_get_sql_database`, `pdb_set_sql_database`

pdb_gen_id

Description	Returns the current value of the database generation ID, or updates the value of the database generation ID.
Syntax	<code>pdb_gen_id [number]</code>
Parameters	<i>number</i> The value of the new database generation ID to be used when the database generation ID is updated. It must be a number between 0 and 32767.
Examples	<p>Example 1</p> <pre>pdb_gen_id</pre> <p>This command returns the current value of the database generation ID.</p> <p>Example 2</p> <pre>pdb_gen_id 10</pre> <p>This command updates the database generation ID to the value 10.</p>
Usage	<ul style="list-style-type: none"> • When <code>pdb_gen_id</code> is invoked with no option, it returns the current value of the database generation ID stored in the Replication Agent transaction log system table (UDB) or in the RASD (Oracle or Microsoft SQL Server). • When <code>pdb_gen_id</code> is invoked with the <code>number</code> option, it updates the value of the database generation ID in the Replication Agent transaction log system table (UDB) or in the RASD (Oracle or Microsoft SQL Server). Changing the database generation ID takes effect immediately. • The database generation ID is the first 2 bytes of the origin queue ID. The database generation ID is used by Replication Server to support recovery operations, which may require the Replication Agent to re-send transactions.

During recovery, if the Replication Agent must re-send operations that the Replication Server has already processed, you can change the database generation ID to prevent the Replication Server from recognizing the operations as already processed.

- For more information about the origin queue ID, see `ra_helplocator` on page 101, or refer to the chapter for your specific primary data server in the *Replication Agent Primary Database Guide*.
- If the Replication Agent transaction log (UDB) or the RASD (Oracle or Microsoft SQL Server) does not exist, the `pdb_gen_id` command returns an error.
- The `pdb_gen_id` command with parameters is valid when the Replication Agent instance is in the *Admin* or *Replication Down* state.

See also `ra_locator`, `pdb_truncate_xlog`

pdb_get_columns

Description Returns a list of columns in tables in the current database at the primary data server.

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

Parameters *ownername*

The user name of the owner of the table specified in the *tablename* option. This option can be delimited with quote characters to specify character case.

tablename

The name of the table in the current database for which information is returned. This option can be delimited with quote characters to specify character case.

colname

The name of the column for which information is returned. This option can be delimited with quote characters to specify character case.

Examples

Example 1

```
pdb_get_columns
```

This command returns a list of all of the columns in all of the user tables in the current database.

Example 2

```
pdb_get_columns bob, authors
```

This command returns a list of all of the columns in the table `authors`, owned by the user “bob” in the current database.

Example 3

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

This command returns information about the column `au_fname` in the table `authors`, owned by the user “bob” in the current database.

Usage

Note (*For Oracle and Microsoft SQL Server*) Results from these commands are taken from the Replication Agent System database (RASD). (*For UDB*) Results from these commands are taken directly from the primary database.

- When `pdb_get_columns` is invoked with no option, it returns a result set that lists all of the columns in all of the user tables in the current database.
- When `pdb_get_columns` is invoked with the *ownername* and *tablename* options, it returns a result set that lists all of the columns in the specified table with the specified owner in the current database.
- When `pdb_get_columns` is invoked with the *ownername*, *tablename*, and *colname* options, it returns a result set with information about the specified column in the specified table with the specified owner in the current database.
- The `pdb_get_columns` command accepts the % wildcard character in the *ownername*, *tablename*, and *colname* options.
- The current database is either:
 - The *default* current database, which is the primary database specified in the Replication Agent `pds_database_name` configuration parameter, or
 - The database specified in the `pdb_set_sql_database` command (to which the Replication Agent instance is currently connected).

- To set or change the current database, use the `pdb_set_sql_database` command.

Note If the `pdb_set_sql_database` command has not been invoked to set or change the current database, the `pdb_get_columns` command returns information from the current database.

- To find the name of the current database, use the `pdb_get_sql_database` command.
- The `pdb_get_columns` command returns 0 rows if the specified table (with the specified owner) does not exist in the current database or if the specified column does not exist in the specified table.
- The `pdb_get_columns` command is valid when the Replication Agent instance is in the *Admin*, *Replicating*, or *Replication Down* state.

See also

`pdb_get_databases`, `pdb_get_primary_keys`, `pdb_get_procedure_parms`, `pdb_get_procedures`, `pdb_get_tables`

pdb_get_databases

Description

Returns a list of all user databases in the primary data server.

Note The Oracle data server does not support multiple user databases. The `pdb_get_databases` command returns the name of the database instance.

Syntax

`pdb_get_databases`

Usage

- When `pdb_get_databases` is invoked, it returns a result set that lists all of the user databases in the primary data server.

Note Depending on the type of system database, the result set may or may not include the user database in the primary data server. See the chapter for your specific primary data server in the *Replication Agent Primary Database Guide*.

- The `pdb_get_databases` command is valid when the Replication Agent instance is in the *Admin*, *Replicating*, or *Replication Down* state.

See also `pdb_get_columns`, `pdb_get_primary_keys`, `pdb_get_procedure_parms`,
`pdb_get_procedures`, `pdb_get_tables`

pdb_get_primary_keys

Description Returns a list of primary key columns in a specified table in the current database at the primary data server.

Syntax `pdb_get_primary_keys ownername, tablename`

Parameters

ownername
 The user name of the owner of the table specified in *tablename*. This option can be delimited with quote characters to specify character case.

tablename
 The name of the table in the current database for which primary key column information is returned. This option can be delimited with quote characters to specify character case.

Usage **Note** (For Oracle and Microsoft SQL Server) Results from these commands are from the Replication Agent System database (RASD). (For UDB) Results from these commands are directly from the primary database.

- When `pdb_get_primary_keys` is invoked, it returns a result set that lists all of the columns that are defined as primary keys in the specified table with the specified owner in the current database.
- The `pdb_get_primary_keys` command accepts the % wildcard character in the *ownername* option, but not in the *tablename* option.
- The current database is either:
 - The *default* current database, which is the primary database specified in the Replication Agent `pds_database_name` configuration parameter, or
 - The database specified in the `pdb_set_sql_database` command to which the Replication Agent instance is currently connected. (This is not valid for Oracle.)

- To set or change the current database, use the `pdb_set_sql_database` command.

Note In Oracle, you cannot change the current database.

- To find the name of the current database, use the `pdb_get_sql_database` command.
- The `pdb_get_primary_keys` command returns 0 rows if the specified table with the specified owner does not exist in the current database.
- The `pdb_get_primary_keys` command is valid when the Replication Agent instance is in the *Admin*, *Replicating*, or *Replication Down* state.

See also

`pdb_get_columns`, `pdb_get_databases`, `pdb_get_procedure_parms`,
`pdb_get_procedures`, `pdb_get_tables`

pdb_get_procedure_parms

Description Returns a list of input parameters for procedures in the current database at the primary data server.

Note This command is available only for Oracle and Microsoft SQL Server.

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

Parameters

ownername

The user name of the owner of the procedure specified in *procname*. This option can be delimited with quote characters to specify character case.

procname

The name of the procedure in the current database for which information is returned. This option can be delimited with quote characters to specify character case.

paramname

The name of the input parameter for which information is returned. This option can be delimited with quote characters to specify character case.

Examples

Example 1

```
pdb_get_procedure_parms
```


This command returns a list of all of the input parameters for all of the procedures in the current database.

Example 2

```
pdb_get_procedure_parms bob, sp_foo
```

This command returns a list of all of the input parameters for the procedure named `sp_foo`, owned by the user “bob” in the current database.

Example 3

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

This command returns information about the input parameter `foo_count` for the procedure `sp_foo`, owned by the user “bob” in the current database.

Usage

Note (*For Oracle and Microsoft SQL Server*) Results from these commands are from the Replication Agent System database (RASD). (*For UDB*) Results from these commands are directly from the primary database.

- When `pdb_get_procedure_parms` is invoked with no option, it returns a result set that lists all of the input parameters for all the procedures in the current database.
- When `pdb_get_procedure_parms` is invoked with the *ownername* and *procname* options, it returns a result set that lists all of the input parameters for the specified procedure with the specified owner in the current database.
- When `pdb_get_procedure_parms` is invoked with the *ownername*, *procname*, and *paramname* options, it returns a result set with information about the specified input parameter for the specified procedure with the specified owner in the current database.
- The `pdb_get_procedure_parms` command accepts the % wildcard character in both the *ownername* and *procname* options.
- The current database is either:
 - The *default* current database, which is the primary database specified in the Replication Agent `pds_database_name` configuration parameter, or
 - The database specified in the `pdb_set_sql_database` command to which the Replication Agent instance is currently connected. (This is not valid for Oracle.)

- To set or change the current database, use the `pdb_set_sql_database` command.

Note In Oracle, you cannot change the current database.

- To find the name of the current database, use the `pdb_get_sql_database` command.
- The `pdb_get_procedure_parms` command returns 0 rows if the specified procedure (with the specified owner) does not exist in the current database.
- The `pdb_get_procedure_parms` command is valid when the Replication Agent instance is in the *Admin*, *Replicating*, or *Replication Down* state.

See also

`pdb_get_columns`, `pdb_get_databases`, `pdb_get_primary_keys`,
`pdb_get_procedures`, `pdb_get_tables`

pdb_get_procedures

Description Returns a list of procedures in the current database at the primary data server.

Note This command is available only for Oracle and Microsoft SQL Server.

Syntax `pdb_get_procedures [ownername, procname]`

Parameters *ownername*

The user name of the owner of the procedure specified in *procname*. This option can be delimited with quote characters to specify character case.

procname

The name of the procedure in the current database for which information is returned. This option can be delimited with quote characters to specify character case.

Examples

Example 1

```
pdb_get_procedures
```

This command returns a list of all of the procedures in the current database.

Example 2

```
pdb_get_procedures bob, sp_foo
```

This command returns information about the procedure named `sp_foo`, owned by the user “bob” in the current database.

Usage

Note (For Oracle and Microsoft SQL Server) Results from these commands are from the Replication Agent System database (RASD). (For UDB) Results from these commands are directly from the primary database.

- When `pdb_get_procedures` is invoked with no option, it returns a result set that lists all of the procedures in the current database.
- When `pdb_get_procedures` is invoked with the *ownername* and *procname* options, it returns a result set with information about the specified procedure with the specified owner in the current database.
- The `pdb_get_procedures` command accepts the % wildcard character in both the *ownername* and *procname* options.
- The current database is either:
 - The *default* current database, which is the primary database specified in the Replication Agent `pds_database_name` configuration parameter, or
 - The database specified in the `pdb_set_sql_database` command to which the Replication Agent instance is currently connected. (This is not valid for Oracle.)
- To set or change the current database, use the `pdb_set_sql_database` command.

Note In Oracle, you cannot change the current database.

- To find the name of the current database, use the `pdb_get_sql_database` command.
- The `pdb_get_procedures` command returns 0 rows if the specified procedure (with the specified owner) does not exist in the current database.
- The `pdb_get_procedures` command is valid when the Replication Agent instance is in the *Admin*, *Replicating*, or *Replication Down* state.

See also

`pdb_get_columns`, `pdb_get_databases`, `pdb_get_primary_keys`,
`pdb_get_procedure_parms`, `pdb_get_tables`

pdb_get_sql_database

Description	Returns the name of the current database, if any.
Syntax	<code>pdb_get_sql_database</code>
Usage	<ul style="list-style-type: none">• When <code>pdb_get_sql_database</code> is invoked, it returns the name of the current database.• If the <code>pdb_set_sql_database</code> command has not been invoked to set the current database, it returns the default current database.• The current database is either:<ul style="list-style-type: none">• The default current database, which is the primary database specified in the Replication Agent <code>pds_database_name</code> configuration parameter, or• The database specified in the command to which the Replication Agent instance is currently connected. (This is not valid for Oracle.)• To set or change the current database, use the <code>pdb_set_sql_database</code> command.

Note In Oracle, you cannot change the current database.

- The `pdb_get_sql_database` command is valid when the Replication Agent instance is in the *Admin*, *Replicating*, or *Replication Down* state.

See also `pdb_execute_sql`, `pdb_set_sql_database`

pdb_get_tables

Description	Returns a list of user tables in the current database at the primary data server.
Syntax	<code>pdb_get_tables [ownername, tablename]</code>
Parameters	<p><i>ownername</i></p> <p>The user name of the owner of the table specified in <i>tablename</i>. This option can be delimited with quote characters to specify character case.</p> <p><i>tablename</i></p> <p>The name of the table in the current database for which information is returned. This option can be delimited with quote characters to specify character case.</p>

Examples

Example 1

```
pdb_get_tables
```

This command returns a list of all of the user tables in the current database.

Example 2

```
pdb_get_tables bob, authors
```

This command returns information about the table authors, owned by the user “bob” in the current database.

Usage

Note (*For Oracle and Microsoft SQL Server*) Results from these commands are from the Replication Agent System database (RASD). (*For UDB*) Results from these commands are taken directly from the primary database.

- When `pdb_get_tables` is invoked with no option, it returns a result set that lists all of the user tables in the current database.

Note System tables may or may not be returned by some primary data servers when the `pdb_get_tables` command is invoked.

- When `pdb_get_tables` is invoked with the *ownername* and *tablename* options, it returns a result set with information about the specified table with the specified owner in the current database.
- The `pdb_get_tables` command accepts the % wildcard character in the both the *ownername* and *tablename* options.
- The current database is either:
 - The *default* current database, which is the primary database specified in the Replication Agent `pds_database_name` configuration parameter, or
 - The database specified in the `pdb_set_sql_database` command to which the Replication Agent instance is currently connected. (This is not valid for Oracle.)

- To set or change the current database, use the `pdb_set_sql_database` command.

Note In Oracle, you cannot change the current database.

- To find the name of the current database, use the `pdb_get_sql_database` command.
- The `pdb_get_tables` command returns 0 rows if the specified table (with the specified owner) does not exist in the current database.
- The `pdb_get_tables` command is valid when the Replication Agent instance is in the *Admin*, *Replicating*, or *Replication Down* state.

See also

`pdb_get_columns`, `pdb_get_databases`, `pdb_get_primary_keys`,
`pdb_get_procedure_parms`, `pdb_get_procedures`

pdb_ownerfilter

Description Returns a list of the owners whose objects will be filtered for initialization; adds, removes owners to the list.

Note This command is available only for Oracle and Microsoft SQL Server.

Syntax To return a list of the owners whose objects will be filtered for initialization:
`pdb_ownerfilter`

To add or remove an owner whose objects will be filtered for initialization:
`pdb_ownerfilter {add | remove}, owner`

Parameters

add

The `add` keyword filters out any objects that are owned by the owner you specify. Any objects that are owned by this owner cannot be marked for initialization.

remove

The `remove` keyword removes the filter for the owner you specify. Any objects that are owned by this owner can be marked for initialization.

For Oracle, you cannot remove the “SYS” owner.

owner

The name of the owner that is used for filtering.

The *owner* option can be delimited with quote characters to specify the character case.

If mixed case (uppercase and lowercase) is required, the name must be delimited. This parameter can be delimited with quotes to specify the character case. If mixed case is required, the name must be delimited. For example:

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

Examples

Example 1

```
pdb_ownerfilter
```

This command returns a list of all owners whose objects will be filtered for initialization.

Example 2

```
pdb_ownerfilter add, SYSTEM
```

This command adds the “system” user to the list of owners whose objects will be filtered for replication.

Example 3

```
pdb_ownerfilter remove, SYSTEM
```

This command removes the “system” user from the list of owners whose objects will be filtered for initialization.

Usage

- `pdb_ownerfilter` can be used to limit the number of objects that are loaded into the Replication Agent System Database during initialization (see `ra_init` command). When `ra_init` is processed, the objects and owners in the `pdb_ownerfilter` list will not be loaded. You can reduce the size of the RASD and reduce the time to perform initialization by adding owners to the list whose objects are not to be replicated, or for owners where the majority of objects are not to be replicated.

Note Any object marked for replication (using commands `pdb_setreptable`, `pdb_setrepproc`), is loaded into the RASD, even if the owner is not on the list. This list affects initialization processing, but not replication (replication occurs based on marking status, not owner filtering).

- When `pdb_ownerfilter` is invoked, its function is determined by the keywords and options you specify.
- When multiple keywords and options are specified, each must be separated by a comma. Blank space before or after a comma is optional. For example:

```
    pdb_ownerfilter add, system
```
- When `pdb_ownerfilter` is invoked with no keyword, it returns a list of users whose objects will be filtered.
- The `pdb_ownerfilter` command is valid only when the Replication Agent instance is in the *Admin* or *Replication Down* state.
- For Oracle, you cannot remove the “SYS” owner.
- For Oracle, after initialization you can replicate any object with `pdb_setreptable` and `pdb_setrepproc`, *except* for the following objects which *cannot* be replicated at any time:
 - Objects that are owned by “SYS” owner.
 - Any system table whose name begins with V\$.
 - Any system procedure or package whose name begins with DBMS.

See also

`pdb_setrepproc`, `pdb_setreptable`, `ra_config`

pdb_send_osuser_list

Description Returns and modifies a list of database users whose primary database user names will be replaced with the corresponding operating system user name in the LTL sent to Replication Server.

Note This command is available only for Oracle.

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

Parameters

add

Use the add keyword to add primary database user names to the list of users whose primary database user names will be replaced with operating system user names in LTL sent to Replication Server. To add one user name, follow the add keyword with the *user* parameter:

```
pdb_send_osuser_list user add, user
```

To add all valid primary database user names to the list, follow the add keyword with the all keyword:

```
pdb_send_osuser_list user add, all
```

remove

Use the remove keyword to remove primary database user names from the list of users whose primary database user names will be replaced with operating system user names in LTL sent to Replication Server. To remove one user name, follow the remove keyword with the *user* parameter:

```
pdb_send_osuser_list user remove, user
```

To remove all user names from the list, follow the remove keyword with the all keyword:

```
pdb_send_osuser_list remove, all
```

To display a list of all user names in the list of users whose primary database user names will be replaced with operating system user names in LTL sent to Replication Server., use the `pdb_send_osuser_list` command alone:

```
pdb_send_osuser_list
```

Examples

Example 1

```
pdb_send_osuser_list add, dbuser1
```

This command adds the primary database user name `dbuser1` to the list. If the operating system user name corresponding to `dbuser1` is `osuser1`, the LTL that Replication Agent sends to Replication Server will contain the user name `osuser1` instead of `dbuser1`.

Example 2

```
pdb_send_osuser_list add, all
```

This command adds all valid primary database user names to the list. The LTL that Replication Agent sends to Replication Server will contain the operating system user names corresponding to all valid primary database user names in the list.

Example 3

```
pdb_send_osuser_list remove, dbuser1
```

This command removes the primary database user name `dbuser1` from the list. If the operating system user name corresponding to `dbuser1` is `osuser1`, the LTL that Replication Agent sends to Replication Server will contain the primary database user name `dbuser1` instead of `osuser1`.

Example 4

```
pdb_send_osuser_list remove, all
```

This command removes all primary database user names from the list. The LTL that Replication Agent sends to Replication Server will contain primary database user names, not operating system user names.

Example 5

```
pdb_send_osuser_list
```

This command lists all database users whose primary database user names will be replaced with a corresponding operating system user name in the LTL sent to Replication Server.

Usage

- The `pdb_send_osuser_list` command with parameters is valid only when the Replication Agent instance is in the *Admin* or *Replication Down* state.
- The `pdb_send_osuser_list` command with parameters affects only database users who are logged in to the primary database at the time this command is invoked.

pdb_set_sql_database

Description	Sets the current database to be used for SQL statement execution.
Syntax	<code>pdb_set_sql_database database</code>
Parameters	<p><i>database</i></p> <p>The name of the database in the primary data server against which the Replication Agent can execute SQL statements (queries). To specify character case, delimit this parameter with quote characters.</p>
Usage	<ul style="list-style-type: none"> When <code>pdb_set_sql_database</code> is invoked, it sets the “current” database, in which the Replication Agent can execute SQL queries. <hr/> <p>Note The <code>pdb_set_sql_database</code> command has no effect in Oracle, but it is included to provide continuity with other Replication Agents that support database servers with multiple databases.</p> <hr/> <ul style="list-style-type: none"> The Replication Agent does not validate the database name you specify with <code>pdb_set_sql_database</code>. <p>If you specify an invalid database name, no error is returned until one of the following Replication Agent commands is invoked:</p> <ul style="list-style-type: none"> <code>pdb_execute_sql</code> <code>pdb_get_columns</code> <code>pdb_get_primary_keys</code> <code>pdb_get_procedure_parms</code> <code>pdb_get_procedures</code> <code>pdb_get_tables</code> <ul style="list-style-type: none"> To find the name of the current database, use <code>pdb_get_sql_database</code>. <hr/> <p>Note If the <code>pdb_set_sql_database</code> command has not been invoked to set the current database, the <code>pdb_get_sql_database</code> command returns the default current database, which is the primary database specified in the Replication Agent <code>pds_database_name</code> configuration parameter.</p> <hr/> <ul style="list-style-type: none"> The <code>pdb_set_sql_database</code> command is valid when the Replication Agent instance is in the <i>Admin</i>, <i>Replicating</i>, or <i>Replication Down</i> state.
See also	<code>pdb_execute_sql</code> , <code>pdb_get_sql_database</code>

pdb_setrepcol

Description	Returns LOB column replication status; enables or disables replication for LOB columns within marked tables.
Syntax	<ul style="list-style-type: none"> • To return replication status of all LOB columns in all tables or all LOB columns in a specific table, or to enable or disable all LOB columns in a table: <pre style="margin-left: 40px;">pdb_setrepcol [<i>tablename</i> [enable disable]]</pre> • To return replication status of a specific LOB column in a specific table: <pre style="margin-left: 40px;">pdb_setrepcol <i>tablename</i>, <i>colname</i></pre> • To enable or disable all LOB columns in all marked tables: <pre style="margin-left: 40px;">pdb_setrepcol all [, {enable disable} [, force]]</pre> • To enable or disable replication for a specified LOB column: <pre style="margin-left: 40px;">pdb_setrepcol <i>tablename</i>, <i>colname</i> [, {enable disable} [, force]]</pre>
Parameters	<p><i>tablename</i></p> <p>The name of the user table in the primary database that contains the column specified in the <i>colname</i> option.</p> <p>The <i>tablename</i> option can be owner-qualified (include the owner name), with each element separated by a period. For example:</p> <pre style="margin-left: 40px;">owner.table</pre> <p>The <i>tablename</i> option can be delimited with quote characters to specify the character case.</p> <p>If mixed case (uppercase and lowercase) is required, the name must be delimited. For example:</p> <pre style="margin-left: 40px;">"Owner".table "Owner"."Table"</pre> <p>Each mixed-case element of the <i>tablename</i> option must be delimited separately, as shown in the previous example.</p> <hr/> <p>Note If you must use an object name case that does not match the value of the <code>ttl_character_case</code> parameter, the object name must be delimited.</p> <hr/> <p>If an object name contains any non-alphanumeric characters, such as spaces or periods, it must be delimited with quote characters. For example:</p> <pre style="margin-left: 40px;">"table name"</pre>

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

If an object name contains a period, it must be *both* owner-qualified and delimited with quote characters. For example:

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

colname

The name of a LOB column in the user table specified in the *tablename* option.

The *colname* option can be delimited with quote characters to specify the character case.

If mixed character case (both uppercase and lowercase) is required, the name must be delimited. For example:

```
"Colname"  
"COLname"
```

Note If you must use a column name case that does not match the value of the `tl_character_case` parameter, the column name must be delimited. See “`tl_character_case`” on page 186 for more information.

all

A keyword that refers to all LOB columns in marked tables in the primary database. By using the *all* keyword, you can apply an enable or disable operation to all LOB columns in marked tables.

enable

A keyword that refers to enabling replication for LOB columns.

disable

A keyword that refers to disabling replication for LOB columns.

force

A keyword that refers to forcing replication to be disabled for LOB columns.

When the *force* keyword follows the *disable* keyword, the `pdb_setrepcol` command immediately disables replication for the specified LOB column. When the *force* keyword follows the *disable* keyword and the *all* keyword, the `pdb_setrepcol` command immediately disables replication for all marked LOB columns in marked tables in the primary database.

Examples

Example 1

```
pdb_setrepcol
```

This command returns replication information for all enabled LOB columns in marked tables in the primary database.

Example 2

```
pdb_setrepcol authors
```

This command returns replication information for all LOB columns defined for the table authors in the primary database.

Example 3

```
pdb_setrepcol authors, picture
```

This command returns replication information for the column called “picture” in the table authors in the primary database.

Example 4

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

This command enables replication for the column “picture” in the table “authors” in the primary database.

Example 5

```
pdb_setrepcol all, disable
```

This command disables replication for all LOB columns in all marked tables in the primary database.

Usage

- If a column is renamed or dropped and a new column with the original name is created, you must explicitly enable or disable replication from this new column because the new column has no replication status related information from the original column.

The marking information is maintained internally based on column number, not column name.

- When `pdb_setrepcol` is invoked, its function is determined by the keywords and options you specify.
- When multiple keywords or options are specified, each must be separated by a comma. Blank space before or after a comma is optional. For example:

```
pdb_setrepcol all, disable
```

- When you specify a column name in the `pdb_setrepcol` command, you must use the name of a valid LOB column.

- You cannot specify the following items as a table name in the `pdb_setrepcol` command:
 - Primary database system tables
 - Aliases or synonyms
 - Views
 - Replication Agent transaction log objects
- If a column name in the primary database is the same as a keyword, it can be identified by adding the string `col=` to the beginning of the column name. For example:

```
    pdb_setrepcol tablename, col=enable, disable
```

- If you enable LOB column replication with the `pdb_setrepcol` command do not configure the Replication Agent to convert date or time datatypes in the primary database. See “`pdb_convert_datetime`” on page 197 for more information.
- When `pdb_setrepcol` is invoked with either no option or a single option, it returns information about the enabled status of LOB columns in the primary database.
 - If `pdb_setrepcol` is invoked with no option, it returns a list of all LOB columns for which replication is enabled in the primary database.

Note Invoking the `pdb_setrepcol` command with no option produces the same result as invoking the `pdb_setrepcol` command with the `enable` keyword.

- If `pdb_setrepcol` is invoked with a table name, it returns information about the enabled status of all the LOB columns in the specified primary table.
- If `pdb_setrepcol` is invoked with the `enable` keyword, it returns a list of all LOB columns for which replication is enabled in the primary database.
- If `pdb_setrepcol` is invoked with the `disable` keyword, it returns a list of all LOB columns for which replication is disabled in the primary database.

For LOB columns listed as disabled, transactions are not captured for replication.

- When `pdb_setrepcol` is invoked with a valid primary table name and valid LOB column name, with no keywords, it returns information about the enabled status of the specified LOB column in the specified table in the primary database.
- When `pdb_setrepcol` is invoked with the `all` keyword, the operation specified by the following keyword (`enable` or `disable`) is applied to all LOB columns in marked tables in the primary database.
 - If `pdb_setrepcol` is invoked with the `all` keyword and the `enable` keyword, it enables replication for all LOB columns in marked tables in the primary database.
 - If `pdb_setrepcol` is invoked with the `all` keyword and the `disable` keyword, it disables replication for all LOB columns in marked tables in the primary database.
- When `pdb_setrepcol` is invoked with a valid primary table name and valid LOB column name followed by one or more keywords, the operation specified by the keyword (`enable` or `disable`) is applied to the specified LOB column in the specified primary table.
 - If `pdb_setrepcol` is invoked with a table name and LOB column name and the `enable` keyword, it enables replication for the specified LOB column in the primary database.
 - If `pdb_setrepcol` is invoked with a table name and LOB column name and the `disable` keyword, it disables replication for the specified LOB column in the primary database.

If the table name and LOB column name combination you specify does not exist in the primary database, the `pdb_setrepcol` command returns an error.

- If the Replication Agent transaction log does not exist in the primary database (UDB) or the RASD is not initialized (Oracle or Microsoft SQL Server), the `pdb_setrepcol` command returns an error.
- If `pdb_setrepcol` is invoked with a table containing a “DATE” column, the primary key in the primary table must *not* include the “DATE” column. This is true for Replication Agent for UDB but not for Replication Agent for Oracle or Microsoft SQL Server.

See also

`pdb_setrepproc`, `pdb_setreptable`, `ra_config`

pdb_setrepddl

Description Returns DDL replication status and enables or disables replication for DDL statements.

Note This command is available only for Oracle and Microsoft SQL Server.

Syntax

For Replication Agent for Oracle:

```

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

```

For Replication Agent for Microsoft SQL Server:

```

pdb_setrepddl {enable|disable}

```

Parameters

override (Oracle only)

To enable the replication of DDL statements and override any existing filtering rules, follow the `pdb_setrepddl` command with the `enable` and `override` keywords:

```

pdb_setrepddl enable, override

```

tablename (Oracle only)

The name of a user table in the primary database. To enable or disable the replication of DDL involving a table, use the `tablename` parameter:

```

pdb_setrepddl tablename [[, enable[, override]] | [, disable] | [default]]

```

To list the current filter setting for a table, enter the `tablename` parameter alone:

```

pdb_setrepddl tablename

```

procname (Oracle only)

The name of a procedure in the primary database. To enable or disable the replication of DDL involving a procedure name, use the *procname* parameter:

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

To list the current filter setting for a procedure, enter the *procname* parameter alone:

```
pdb_setrepddl procname
```

sequence_name (Oracle only)

The name of a user sequence in the primary database. To enable or disable the replication of DDL involving a sequence, use the *sequence_name* parameter:

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

To list the current filter setting for a sequence, enter the *sequence_name* parameter alone:

```
pdb_setrepddl sequence_name
```

objects, all (Oracle only)

The objects keyword must be used with the keyword all and allows you to enable or disable the replication of DDL statements for all objects:

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

To list all objects for which DDL statements are filtered, follow the *pdb_setrepddl* command with the objects, all keywords:

```
pdb_setrepddl objects, all
```

user (Oracle only)

The **user** keyword allows you to enable or disable the replication of DDL statements executed by primary database users. To enable or disable the replication of DDL from a specified user, use the *user* parameter.

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

To list the current filter setting for a user, follow the **user** keyword with the *user* parameter:

```
pdb_setrepddl user, user
```

To list database users whose DDL statements will be filtered from replication, enter the **user** keyword alone:

```
pdb_setrepddl user
```

To enable or disable the replication of DDL statements for all users, follow the **user** keyword with the **all** keyword:

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

stmt (Oracle only)

The **stmt** keyword allows you to enable or disable the replication of DDL statements. To enable or disable the replication of DDL for a particular statement, use the *ddl_statement* parameter, which contains a string in the form of a DDL statement enclosed in single or double quotes:

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

To list the current filter setting for a particular DDL statement, follow the **stmt** keyword with the *ddl_statement* parameter:

```
pdb_setrepddl stmt, ddl_statement
```

To enable or disable the replication of DDL for an entire set of statements, use the *ddl_statement_keyword* parameter, which contains a string in the form of a DDL statement keyword:

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

Table 1-3 shows which DDL statements are filtered together by a DDL statement keyword.

Table 1-3: The ddl_statement_keyword parameter

Keyword value	DDL statements filtered
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
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

To enable or disable the replication of all DDL statements, follow the stmt keyword with the all keyword:

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

To list DDL statements that will be filtered from replication, enter the stmt keyword alone:

```
pdb_setrepddl stmt
```

owner (Oracle only)

The **owner** keyword allows you to enable or disable the replication of DDL statements affecting an object owned by a particular user. To enable or disable the replication of DDL statements affecting objects owned by a particular user, use the *ownername* parameter to specify the user:

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

To list the current filter setting for an object owner, follow the **owner** keyword with the *ownername* parameter:

```
pdb_setrepddl owner, ownername
```

To list the object owners for which DDL statements will be filtered from replication, enter the **owner** keyword alone:

```
pdb_setrepddl owner
```

To enable or disable the replication of DDL statements affecting objects for all owners, follow the **owner** keyword with the **all** keyword:

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

enable

For Replication Agent for Oracle, the **enable** keyword allows you to enable the replication of DDL statements as specified by other keywords and parameters in the `pdb_setrepddl` command. To override any previous filtering of DDL statements, follow the **enable** keyword with the **override** keyword.

For Replication Agent for Microsoft SQL Server, the **enable** keyword allows you to enable the replication of DDL statements for the primary database.

disable

For Replication Agent for Oracle, the **disable** keyword allows you to disable the replication of DDL statements as specified by other keywords and parameters in the `pdb_setrepddl` command.

For Replication Agent for Microsoft SQL Server, the **disable** keyword allows you to disable the replication of DDL statements for the primary database. Replication of DDL statements is disabled by default in Replication Agent for Microsoft SQL Server.

To return the status of DDL replication in Replication Agent for Microsoft SQL Server, use the `pdb_setrepddl` command alone, without the **disable** or **enable** keywords.

default (Oracle only)

If replication is not explicitly enabled or disabled for a particular owner, object, or DDL statement, the default keyword results in DDL statement filtering being enabled or disabled according to the following hierarchy:

- 1 Object: Any DDL filtering rules for an object will be observed, but not rules involving the object owner, statements affecting the object, or the user.
- 2 Owner: Any DDL filtering rules for an object owner will be observed, but not rules involving statements affecting the object or the user.
- 3 Statement: Any DDL filtering rules for DDL statements will be observed, but not rules involving the user.
- 4 User: Any DDL filtering rules involving the user will be observed.

Examples

Example 1

```
pdb_setrepddl stmt
```

(Oracle only) This command lists DDL statements that are filtered from replication.

Example 2

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

(Oracle only) This command lists the current filter setting for the create index statement.

Example 3

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

(Oracle only) This command causes Replication Agent to filter the create index statement from replication.

Example 4

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

(Oracle only) This command causes Replication Agent to filter the alter index, create index, and drop index statements from replication.

Example 5

```
pdb_setrepddl owner
```

(Oracle only) This command lists the object owners for which DDL statements will be filtered from replication.

Example 6

```
pdb_setrepddl owner, myuser
```

(*Oracle only*) This command lists the current filter setting for an object owner.

Example 7

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

(*Oracle only*) This command causes Replication Agent to filter DDL statements affecting objects owned by the user named myuser.

Example 8

```
pdb_setrepddl myuser.mytable
```

(*Oracle only*) This command lists the current filter setting for the table named myuser.mytable.

Example 9

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

(*Oracle only*) This command causes Replication Agent to filter all DDL statements that affect the table named myuser.mytable.

Example 10

```
pdb_setrepddl owner, myuser, disable  
pdb_setrepddl myuser.mytable, enable
```

(*Oracle only*) These commands cause Replication Agent to filter all DDL statements affecting objects owned by the user named myuser except the table named myuser.mytable.

Example 11

```
pdb_setrepddl owner, all, disable  
pdb_setrepddl owner, myuser, enable
```

(*Oracle only*) These commands cause Replication Agent to replicate only DDL statements affecting objects owned by the user named myuser.

Example 12

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

(*Oracle only*) This command causes Replication Agent to filter DDL from the user named myuser.

Example 13

```
pdb_setrepddl enable, override
```

(*Oracle only*) This command enables DDL replication and overrides any existing filter settings.

Example 14

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

(*Oracle only*) This command enables replication of DDL from the user named myuser and overrides any existing filter settings.

Example 15

```
pdb_setrepddl
```

This command returns the current DDL replication status for the primary database.

Example 16

```
pdb_setrepddl enable
```

This command enables replication of DDL commands issued into the primary database after this point in time.

Example 17

```
pdb_setrepddl disable
```

This command disables replication of DDL commands issued into the primary database after this point in time.

Usage

- In addition to enabling DDL replication using `pdb_setrepddl` command, you must set the Replication Agent `ddl_username` and `ddl_password` parameters.
- A database replication definition that enables DDL is required for DDL replication. If the `use_rssd` configuration parameter is set to true, the database replication definition must exist in the RSSD of the primary Replication Server before the Replication Agent resume command is invoked.
- Only DDL statements that have identical syntax in the primary and replicate databases can be replicated. Replication Agent sends DDL statements using the syntax of the statements in the primary database.
- If the Replication Agent has not been initialized, the `pdb_setrepddl` command returns an error.
- The `pdb_setrepddl` command can only use the `enable` and `disable` options when the Replication Agent instance is in the *Admin* or *Replication Down* state.

See also See configuration parameters: `ddl_password`, `asm_username`

pdb_setrepproc

Description Returns stored procedure replication marking status; marks specified procedures for replication; unmarks all marked procedures or a specified procedure; enables or disables replication for all marked procedures or a specified procedure.

Note This command is available only for Oracle and Microsoft SQL Server.

Syntax

To return stored procedure replication marking status:

```
pdb_setrepproc [procname]|mark|unmark|enable|disable
```

To unmark, enable, or disable all marked stored procedures:

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

To mark a specified stored procedure for replication with a replicated name:

```
pdb_setrepproc procname, [repname,] mark
```

To unmark, enable, or disable a specified stored procedure:

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

Note For Microsoft SQL Server, the `pdb_dflt_object_repl` configuration parameter has no impact on `pdb_setrepproc` command.

Parameters

procname
The name of a user stored procedure in the primary database.

The *procname* option can be delimited with quote characters to specify the character case.

If mixed character case (both uppercase and lowercase) is required, the name must be delimited. For example:

"Proc"

Note If you must use an object name case that does not match the setting of the `lfl_character_case` parameter, the object name must be delimited.

If an object name contains any non-alphanumeric characters, such as spaces, periods, and so forth, it must be delimited with quote characters. For example:

```
"proc name"  
"proc.name"
```

If an object name contains a period, it must be *both* owner-qualified and delimited with quote characters. For example:

```
owner."proc.name"  
"proc.owner"."proc.name"
```

rename

The name of the stored procedure specified in a function replication definition for the primary stored procedure.

The *rename* option can be delimited with quote characters to specify character case. See the previous description of the *procname* option for details.

By specifying a replicated name, stored procedure invocations can be replicated to a stored procedure invocation in the replicate database that has a different stored procedure name from the primary database.

Note The replicated name you specify with the `pdb_setrepproc` command must match the name specified by a `with primary function named` clause in a Replication Server function replication definition for the primary database connection. The Replication Agent cannot validate the function replication definition, but if it does not exist, function replication from the primary database will fail.

all

A keyword that refers to all user stored procedures in the primary database. By using the `all` keyword, you can mark all user stored procedures, or apply an unmark, enable, or disable operation to all *marked* stored procedures.

mark

Note You must specify an owner when using the mark keyword.

A keyword that refers to marking user stored procedures for replication.

unmark

A keyword that refers to unmarking marked stored procedures.

force

A keyword that refers to the unmark operation. When the force keyword follows the unmark keyword, the `pdb_setrepproc` command immediately unmarks the specified stored procedure in the primary database, without first checking the enable status of the stored procedure. When the force keyword follows the unmark keyword and the all keyword, the `pdb_setrepproc` command immediately removes replication marking from all marked stored procedures in the primary database, regardless of their enable status.

The force keyword also forces complete execution of the unmarking script, even if errors occur during the unmarking process. Normally, when errors occur during script execution, the script terminates immediately without completing.

Note Sybase recommends that you read the “Usage” section that follows to better understand how scripts are used in Oracle procedure marking and unmarking.

The force keyword can be useful when a previous script execution failed and left the unmarking operation incomplete. When errors occur during a forced script execution, the `pdb_setrepproc` command returns the following message:

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

enable

A keyword that refers to enabling replication for marked stored procedures.

disable

A keyword that refers to disabling replication for marked stored procedures.

Examples

Example 1

```
pdb_setrepproc
```

This command returns replication marking information for all marked stored procedures in the primary database.

Example 2

```
pdb_setrepproc authors
```

This command returns replication marking information for the user stored procedure named “authors” in the primary database.

Example 3

```
pdb_setrepproc authors, mark
```

This command marks the user stored procedure named “authors” in the primary database.

Example 4

```
pdb_setrepproc authors, enable
```

This command enables replication for the marked stored procedure named “authors” in the primary database.

Example 5

```
pdb_setrepproc all, unmark
```

This command unmarks all marked stored procedures in the primary database.

Usage

- If a marked procedure is renamed or dropped and a new procedure with the original name is created, you must explicitly mark the new procedure because the new procedure has no marking-related information from the original procedure. The marking information is maintained internally by object id, not object name.
- How you use the `pdb_setrepproc` command depends on the type of replication definition that you have created at Replication Server. If you have created a database replication definition with no function replication definition, then the replicate procedure in the `pdb_setrepproc` command refers to the procedure in the replicate database. However, if you have created a function replication definition, then the replicate procedure in the `pdb_setrepproc` command refers to the name of the function replication definition, and it is the function replication definition that must map to the procedure in the replicate database.

If *no* function replication definition exists and will not be added prior to replication, but only a database replication definition exists, use the following commands to mark a procedure for replication:

- When the procedure in the replicate database has the same name as the procedure in the primary database:

```
pdb_setrepproc pdb_proc, mark
```

where:

`pdb_proc` is the name of the procedure in the primary database that you want to mark for replication.

- When the procedure in the replicate database has the different name name than the procedure in the primary database:

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

where:

`pdb_proc` is the name of the procedure in the primary database that you want to mark for replication.

`rep_proc` is the name of the procedure in the replicate database.

If a function replication definition exists or will be added prior to replication, regardless of whether or not a database replication definition exists, use the following commands to mark a procedure for replication:

- When the function replication definition has the same name as the procedure in the primary database:

```
pdb_setrepproc pdb_proc, mark
```

where:

`pdb_proc` is the name of the procedure in the primary database that you want to mark for replication.

- If the procedure in the replicate database also has the *same* name as the function replication definition, then there is no need to use the “deliver as” clause in the replication definition in the primary Replication Server. For example:

```
create function replication definition
  pdb_proc with primary at data_server.
  database ...
```

- If the procedure in the replicate database has a *different* name than the name of function replication definition, then the function replication definition must map to the procedure in the replicate database. For example:

```
create function replication definition
  pdb_proc with primary at data_server.database
```

deliver as 'rep_proc' ...

- When the name of the function replication definition is different than the procedure in the primary database:

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

where:

pdb_proc is the name of the procedure in the primary database that you want to mark for replication.

rdpri_proc is the name of the function replication definition.

- If the procedure in the replicate database also has the *same* name as the function replication definition, then there is no need to use the “deliver as” clause in the replication definition in the primary Replication Server. For example:

```
create function replication definition
rdpri_proc with primary at
data_server.database ...
```

- If the procedure in the replicate database has a *different* name from the function replication definition, then the function replication definition must map to the procedure in the replicate database. For example:

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

- When multiple keywords and options are specified, each must be separated by a comma. Blank space before or after a comma is optional. For example:

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

- When you specify a stored procedure name in the `pdb_setrepproc` command, you must use the name of a valid user stored procedure.
- You cannot specify the following items as a stored procedure name in the `pdb_setrepproc` command:
 - System procedures
 - Replication Agent transaction log procedures
- If a stored procedure name in the primary database is the same as a keyword, it can be identified by adding the string `proc=` to the beginning of the stored procedure name. For example:

```
pdb_setrepproc proc=unmark, mark
```

- When you use the unmark keyword to remove replication marking from a stored procedure, the Replication Agent verifies that replication is disabled for that stored procedure and there are no pending (unprocessed) operations for that stored procedure in the transaction log. If replication is not disabled for that procedure, or if there is a pending operation for that procedure in the transaction log, `pdb_setrepproc` returns an error.
- When `pdb_setrepproc` is invoked with either no option or a single option, it returns marking information about the stored procedures in the primary database.
 - If `pdb_setrepproc` is invoked with no option, it returns a list of all marked procedures in the primary database.

Note Invoking the `pdb_setrepproc` command with no option produces the same result as invoking the `pdb_setrepproc` command with only the mark keyword.

- If `pdb_setrepproc` is invoked with a procedure name, it returns complete marking information about the specified procedure.
- If `pdb_setrepproc` is invoked with the mark keyword, it returns a list of all marked procedures in the primary database.
- If `pdb_setrepproc` is invoked with the unmark keyword, it returns a list of all unmarked procedures in the primary database.
- If `pdb_setrepproc` is invoked with the enable keyword, it returns a list of all marked procedures in the primary database, for which replication is currently enabled.
- If `pdb_setrepproc` is invoked with the disable keyword, it returns a list of all marked procedures in the primary database, for which replication is currently disabled.

Stored procedures marked for replication are recorded in the RASD. All other user procedures are considered unmarked.

Note The Replication Agent system procedures are not included in the list of unmarked procedures. Also not included are any synonyms or aliases for these procedures.

For procedures listed as unmarked or disabled, their invocations will not be captured for replication.

- When `pdb_setrepproc` is invoked with the `all` keyword and an action keyword (`unmark`, `enable`, or `disable`), the action specified is applied to either all user stored procedures in the primary database, or to all marked procedures in the primary database.
 - If `pdb_setrepproc` is invoked with the `all` and `unmark` keywords, it removes replication marking from all marked procedures in the primary database.

You can specify the `force` keyword after the `unmark` keyword to force immediate unmarking of all marked procedures, including procedures for which replication is still enabled.
 - If `pdb_setrepproc` is invoked with the `all` and `enable` keywords, it enables replication for all marked procedures in the primary database.
 - If `pdb_setrepproc` is invoked with the `all` and `disable` keywords, it disables replication for all marked procedures in the primary database.
- When `pdb_setrepproc` is invoked with a valid user stored procedure name and followed by an action keyword (`mark`, `unmark`, `enable`, or `disable`), the action specified is applied to the specified procedure.
 - If `pdb_setrepproc` is invoked with a procedure name and the `mark` keyword, it marks the specified procedure in the primary database for replication.
 - If `pdb_setrepproc` is invoked with a procedure name and the `unmark` keyword, it removes replication marking from the specified procedure in the primary database.
 - If `pdb_setrepproc` is invoked with a procedure name and the `enable` keyword, it enables replication for the specified marked procedure in the primary database.
 - If `pdb_setrepproc` is invoked with a procedure name and the `disable` keyword, it disables replication for the specified marked procedure in the primary database.

Note Sybase recommends the use of `[mark | unmark]` instead of `[enable | disable]` since the results are the same.

- If you specify a stored procedure name that does not exist in the primary database, the `pdb_setrepproc` command returns an error.

- When `pdb_setreproc` is invoked with a procedure name and a replicated name, followed by the `mark` keyword, the primary procedure is marked for replication with the specified replicated name.

If the primary procedure name you specify does not exist in the primary database, the `pdb_setreproc` command returns an error.

By specifying a replicated name, procedure invocations can be replicated to a procedure in the replicate database that has a different name from the primary procedure.

Note The replicated name you specify with the `pdb_setreproc` command must match the name of a Replication Server function replication definition for the primary database connection. The Replication Agent cannot validate the function replication definition, but if it does not exist, function replication from the primary database will fail.

- If RASD is not initialized (Oracle and Microsoft SQL Server), the `pdb_setreproc` command returns an error.

For Oracle:

To support stored procedure replication in Oracle, a stored procedure that is marked for replication must be modified. The modification is required to record the stored procedures execution in the Oracle transaction log. As a result of the modifications, the following behavior should be considered when marking and unmarking stored procedures in Oracle:

- You must disable DDL replication before marking or unmarking a procedure, and re-enable it after marking or unmarking to prevent modifications from replicating to standby.
- Marking and unmarking a stored procedure for replication requires that the Replication Agent drop, and then re-create the procedure. However, Replication Agent sets all the same privileges on the re-created procedure as those defined on the original procedure.

Note Do not remove or alter the Replication Agent comments in a marked stored procedure.

- When `pdb_setreproc` is invoked to mark a procedure for replication, Replication Agent does the following:
 - Modifies the user procedure to add code that captures input parameter values and generates Replication Agent transaction log records.

- Generates a SQL script that creates the procedures required for the Replication Agent transaction log in the primary database.
- Saves the generated script in a file called *partmark.sql* in the *RAX-15_5\inst_name\scripts\procname* directory, where *inst_name* is the name of the Replication Agent instance, and *procname* is the name of the stored procedure being marked. This script can *not* be manually executed—it is for informational purposes only.

Note If the value of the `pdb_auto_run_scripts` configuration parameter is false, the *partmark.sql* script will be saved but not executed automatically. You cannot manually run the script. To complete marking the procedure, you must first set `pdb_auto_run_scripts` to *true*, then re-run the `pdb_setrepproc` command.

- Executes the script to mark the stored procedure and create the transaction log objects in the primary database (if the value of the `pdb_auto_run_scripts` configuration parameter is true).
- After the script completes successfully, moves the *partmark.sql* file to the *RAX-15_5\inst_name\scripts\procname\installed* directory.
- If the mark script fails, it is stored in a file (*partmark.sql*) in the *RAX-15_5\inst_name\scripts\procname* directory, the stored procedure is not marked, and transaction log objects are not created. You can examine the script by viewing the *mark.sql* file.
- When `pdb_setrepproc` is invoked to unmark a marked stored procedure, Replication Agent does the following:
 - Modifies the user procedure to remove Replication Agent code that captures input parameter values and generates transaction log records.
 - Generates a SQL script that removes the tables and procedures required for the transaction log in the primary database.

- Saves the generated script in a file called *partunmark.sql* in the *RAX-15_5\inst_name\scripts\procname* directory, where *inst_name* is the name of the Replication Agent instance and *procname* is the name of the stored procedure being unmarked. For Oracle, this script named *partunmark.sql* because it can *not* be manually executed—it is for informational purposes only.

Note If the value of the `pdb_auto_run_scripts` configuration parameter is `false`, the *partunmark.sql* script will be saved but not executed automatically. You cannot manually run the script. To complete unmarking the procedure, you must first set `pdb_auto_run_scripts` to `true`, then re-run the `pdb_setrepproc` command.

- Executes the script to unmark the stored procedure and remove the transaction log objects in the primary database (if the value of the `pdb_auto_run_scripts` configuration parameter is `true`).
- After the script completes successfully, moves the *partunmark.sql* file to the *RAX-15_5\inst_name\scripts\procname\installed* directory.
- If the unmark script fails, it is stored in a file (*partunmark.sql*) in the *RAX-15_5\inst_name\procname\scripts* directory and the stored procedure is not unmarked and the transaction log objects are not removed. You can examine the script by viewing the *partunmark.sql* file.

When the unmark script execution encounters a fatal error on any database object, the `pdb_setrepproc` command returns the following message:

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

- The `pdb_setrepproc` command is used in replicating Oracle stored procedures that have an argument of type Boolean. For instructions on how to replicate Oracle stored procedures that have an argument of type Boolean, see the *Replication Agent Primary Database Guide*.

See also

`pdb_setrepcol`, `pdb_setreptable`, `ra_config`

pdb_setrepseq

Description Returns the sequence replication marking status; marks specified sequence for replication; unmarks all marked sequences or a specified sequence; enables or disables replication for all marked sequences or a specified sequence.

Note This command is available only for Oracle.

Syntax

To return sequence replication marking status:

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

To unmark, enable, or disable all marked sequences:

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

To mark, unmark, enable, or disable a specified sequence:

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

To mark a specified sequence for replication with a replicated name:

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

Parameters

sequence_name

The name of a user sequence in the primary database. The *sequence_name* option can be delimited with quote characters to specify the character case. If mixed character case (both uppercase and lowercase) is required, the name must be delimited. For example:

```
"Sequence"
```

The *sequence_name* parameter can be owner-qualified to include the primary sequence owner name, with each element separated by a period. For example:

```
owner.sequence
```

Note If you must use an object name case that does not match the setting of the `lfl_character_case` parameter, the object name must be delimited. If an object name contains any non-alphanumeric characters, such as spaces and periods, it must be delimited with quote characters. For example, “sequence name” or owner.”sequence name.”

rename

The replicated name of the sequence to be updated at the replicate site, if desired to be different than the sequence name at the primary site. The *rename* option can be delimited with quote characters to specify character case. See the previous description of the *sequence_name* parameter for details. By specifying a replicated name, sequence updates can be replicated to a sequence in the replicate database that has a different sequence name from the primary database.

The *rename* option can be owner-qualified to include the replicate sequence owner name, with each element separated by a period. For example:

```
repowner . rename
```

all

A keyword that refers to all user sequences in the primary database. By using the *all* keyword, you can unmark all user sequences, or apply an enable or disable operation to all marked sequences.

mark

A keyword that refers to marking user sequences for replication.

unmark

A keyword that refers to unmarking user sequences for replication.

force

A keyword that refers to the unmark operation. When the *force* keyword follows the *unmark* keyword, the `pdb_setrepseq` command immediately unmarks the specified sequence in the primary database, without first checking the enable status of the sequence. When the *force* keyword follows the *unmark* keyword and the *all* keyword, the `pdb_setrepseq` command immediately removes replication marking from all marked sequences in the primary database, regardless of their enable status

enable

A keyword that refers to enabling replication for marked sequences.

disable

A keyword that refers to disabling replication for marked sequences.

Usage

- When `pdb_setrepseq` is invoked, its function is determined by the keywords and options you specify.
- When multiple keywords and options are specified, each must be separated by a comma. Blank space before or after a comma is optional. For example:

`pdb_setrepseq all, unmark, force`

- When you specify a sequence in the `pdb_setrepseq` command, you must use the name of a valid user sequence.

pdb_setreptable

Description Returns replication marking status; marks all user tables or a specified table for replication; unmarks all marked tables or a specified table; enables or disables replication for all marked tables or a specified table.

Syntax To return replication marking status:

```
pdb_setreptable tablename
```

To mark all user tables (available only for Oracle and Microsoft SQL Server):

```
pdb_setreptable all, mark
```

To unmark, enable, or disable all marked tables:

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

To mark, unmark, enable, or disable a specified table:

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

Note Marking or unmarking all tables at once in the primary database using `pdb_setreptable all, mark` or `pdb_setreptable all, unmark` is not supported in Replication Agent for UDB. You must mark or unmark each table individually.

To mark a specified table for replication with a replicated name:

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

To mark a specified table for replication immediately for any occurrence, that may or may not be marked:

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

To return a list of all marked tables:

```
pdb_setreptable
```

Parameters

tablename

The name of a user table in the primary database.

The *tablename* parameter can be owner-qualified to include the primary table owner name, with each element separated by a period. For example:

```
owner.table
```

This parameter can be delimited with quote characters to specify the character case.

If mixed character case (both uppercase and lowercase) is required, the name must be delimited. For example:

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

Each mixed case element of the *tablename* option must be delimited separately, as shown in the previous example.

If an object name contains any non-alphanumeric characters, such as spaces or periods, it must be delimited with quote characters. For example:

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

If an object name contains a period, it must be *both* owner-qualified and delimited with quote characters. For example:

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

rename

The name of the table specified in the replication definition for a primary table.

Note The replicated name you specify with the `pdb_setreptable` command must match a table name specified by a `with primary table named` clause in a Replication Server replication definition for the primary database connection. The Replication Agent cannot validate the replication definition, but if it does not exist, or if the `with primary table named` clause does not match the replicated name specified with `pdb_setreptable`, replication from the primary table will fail.

The *rename* option can be owner-qualified to include the replicate table owner name, with each element separated by a period. For example:

`reowner.reptable`

The *reptable* option can also be delimited with quote characters to specify the character case. See the previous description of the *tablename* option for details.

Note If the replicate table name contains a period (for example, *table.name*), without owner qualification, you must set the value of the Replication Agent `use_rssd` parameter to `true`.

all

A keyword that refers to all tables in the primary database. By using the `all` keyword, you can mark all user tables, or apply an `unmark`, `enable`, or `disable` operation to all marked tables.

mark

A keyword that refers to marking a table.

owner

A keyword that refers to the mark operation.

The `owner` keyword turns on the `SEND OWNER` mode. When you specify the owner of a table in a replication definition, you must always use the `owner` keyword if you want to enable the `SEND OWNER` mode.

`owner mode` sets a flag in the LTL telling Replication Server that any table level Replication definition must be owner qualified to match this table.

If the `owner mode` is set, the replication definition must be owner qualified. If the `owner mode` is not set, the replication definition must not be owner qualified.

unmark

A keyword that refers to unmarking a marked table.

force

A keyword used with the unmark operation or mark operation:

Note The combination of mark and force keywords with the `pdb_setreptable` command is only valid for UDB and Oracle.

- When the force keyword follows the unmark keyword, the `pdb_setreptable` command immediately removes replication marking for the specified table in the primary database, without first checking the enable status of the table. When the force keyword follows the unmark keyword and the all keyword, `pdb_setreptable` immediately removes replication marking from all marked tables in the primary database, regardless of their enable status.

The force keyword also forces complete execution of the unmarking script, even if errors occur during the unmarking process. Normally, when errors occur during script execution, the script terminates immediately without completing. The force keyword can be useful when a previous script execution failed and left the unmarking operation incomplete.

When errors occur during a forced script execution, the `pdb_setreptable` command returns the following message:

```
Errors were encountered and ignored during FORCED
```

script execution. See error log for details.

- *For UDB and Oracle:*

When the force keyword follows the mark keyword, the `pdb_setreptable` command allows a table that contains one or more columns with unsupported datatypes to be marked for replication. No data for the unsupported columns is sent to Replication Server. As a result, any replicate table must have a suitable default value defined for the unsupported columns, since no data is received by the replicate database to be inserted into the unsupported columns.

The force keyword can not be used in combination with the all keyword. Tables with unsupported datatypes must be individually marked using the `pdb_setreptable` command and the force keyword (they will never be automatically marked, or marked by default if they have columns with unsupported datatypes).

In addition, tables with unsupported datatypes are not automatically marked when the `pdb_automark_tables` configuration parameter is true. Tables with unsupported datatypes must be individually marked using the `pdb_setreptable` command and the mark and force keywords. For a list of supported and unsupported datatypes, see the *Replication Agent Primary Database Guide*.

Note If a replication definition is created using the command `rs_create_repdef`, for a table that was marked using the force keyword, only columns with supported datatypes are listed in the replication definition. Any column with an unsupported datatype is excluded from the replication definition.

enable

A keyword that refers to enabling replication for marked tables.

disable

A keyword that refers to disabling replication for marked tables.

immediate

A keyword that allows a table to be immediately marked for any occurrence. This keyword is for Microsoft SQL Server and Oracle only.

Examples

Example 1

```
pdb_setreptable authors
```

This command returns replication marking information for the table named “authors” in the primary database.

Example 2

```
pdb_setreptable mark
```

This command returns replication marking information for all marked tables in the primary database.

Example 3

```
pdb_setreptable disable
```

This command returns replication marking information for all marked tables for which replication has been disabled in the primary database.

Example 4

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

This command forces unmarking for all marked tables in the primary database.

Example 5

```
pdb_setreptable all, enable
```

This command enables replication for all marked tables in the primary database.

Example 6

```
pdb_setreptable authors, mark
```

This command marks for replication the table named “authors” in the primary database. The primary table name in the replication definition must be authors.

Example 7

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

This command marks for replication the table named “authors” in the primary database so that the OWNER_MODE is enabled in the LTL. Therefore, any table replication definition created for this table must also be owner qualified.

Example 8

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

The primary table name in the replication definition must be:

```
powner.rtable
```

Example 9

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

The primary table name in the replication definition must be:

```
rowner.rtable
```

Example 10

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

The primary table name in the replication definition must be:

```
rowner.rtable
```

Example 11

```
pdb_setreptable authors, enable
```

This command enables replication for the marked table “authors” in the primary database.

Example 12

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

This command enables replication for the marked table named “mark” in the primary database.

Example 13

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

This command forces unmarking for the marked table “authors” in the primary database.

Example 14 (UDB and Oracle only)

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

This command forces table “authors” to be marked, even if it contains columns with unsupported datatypes. The columns with unsupported datatypes will not be replicated.

Usage

- When a marked table is renamed or dropped and a new table with the original name is created, you must explicitly mark the new table because the new table has no marking-related information from the original table. The marking information is maintained internally by object ID, not table name.
- When a table is marked for replication and the owner mode is set to on, the replication definition must contain the owner name in the “with primary table named” clause, or the “with all tables named” clause. If the owner mode setting and the existence of the owner name in the replication definition do not match, the replication definition is not be used.

For example:

- Issuing the following command with the owner mode set to on:

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

causes the `rs_create_repdef` command to generate the following replication definition for the primary and replicate database, which the Replication Server expects to receive:

```
create replication definition ra$0xda_"mytable"
with primary at oral02.dco
with primary table named "qa4user"."mytable"
with replicate table named "qa4user"."mytable"
.
.
.
```

- Issuing the following command with the owner mode set to off:

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

causes the `rs_create_repdef` command to generate the following replication definition for the primary and replicate database, which the Replication Server expects to receive:

```
create replication definition ra$0xda_"mytable"
with primary at oral02.dco
with primary table named "mytable"
with replicate table named "qa4user"."mytable"
.
.
.
```

- How you use the `pdb_setreptable` command depends on the type of replication definition that you have created at the Replication Server. If you have created a database replication definition with no table replication definition, then the replicate procedure in the `pdb_setreptable` command refers to the table in the replicate database. However, if you have created a table replication definition, then the replicate table in the `pdb_setreptable` command refers to the name of the table replication definition, and it is the table replication definition that must map to the table in the replicate database.

If *no* table replication definition exists and will not be added prior to replication, but only a database replication definition exists, use the following commands to mark a table for replication:

- When the table in the replicate database has the same name as the table in the primary database:

```
pdb_setreptable pdb_table, mark
```

where:

pdb_table is the name of the table in the primary database that you want to mark for replication.

- When the table in the replicate database has the different name than the table in the primary database:

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

where:

pdb_table is the name of the table in the primary database that you want to mark for replication.

rep_table is the name of the table in the replicate database.

If a table replication definition exists or will be added prior to replication, regardless of whether or not a database replication definition exists, use the following commands to mark a table for replication:

- When the primary table in the table definition has the *same* name as the table in the primary database:

```
pdb_setreptable pdb_table, mark
```

where:

pdb_table is the name of the table in the primary database that you want to mark for replication.

- If the table in the replicate database also has the *same* name as the table replication definition, then you can use the “with all tables named” clause in the replication definition in the primary Replication Server. For example:

```
create replication definition my_table_repdef  
with primary at data_server.database  
with all tables named pdb_table ...
```

- If the table in the replicate database has a *different* name than the primary table in the table replication definition, then the table replication definition must map to the table in the replicate database. For example:

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

- When the name of the table replication definition is *different* than the table in the primary database:

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

where:

`pdb_table` is the name of the table in the primary database that you want to mark for replication.

`rdpri_table` is the name of the primary table in the replication definition.

- If the table in the replicate database also has the *same* name as the primary table in the table replication definition, then you can use the “with all tables named” clause in the replication definition in the primary Replication Server. For example:

```
create replication definition my_table_repdef
with primary at data_server.database
with all tables named rdpri_table ...
```

- If the table in the replicate database has a *different* name from the primary table in the table replication definition, then the table replication definition must map to the table in the replicate database. For example:

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

- When multiple keywords and options are specified, each must be separated by a comma. Blank space before or after a comma is optional. For example:

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

- When you specify a primary table in the `pdb_setreptable` command, you must use the name of a valid user table.
- You cannot specify the following items as a primary table in the `pdb_setreptable` command:
 - System tables
 - Views
 - Replication Agent transaction log tables

- If you specify an alias or synonym as a primary table in the `pdb_setreptable` command, the actual table that the alias or synonym refers to is acted upon. The actual table name is the table name sent to the primary Replication Server.
- If a table name in the primary database is the same as a keyword, it can be identified by adding the `table=string` to the beginning of the name. For example:

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

This is true for both primary table names and replicated names.

- For UDB only, if the `mark` script fails, it is stored in a file (`mark.sql`) in the `RAX-15_5\inst_name\scripts\tablename` directory, the table is not marked, and transaction log objects are not updated. You can examine the script by viewing the `mark.sql` file.
- For UDB only, if the `unmark` script fails, it is stored in a file (`unmark.sql`) in the `RAX-15_5\inst_name\tablename\scripts` directory, the table is not unmarked, and the transaction log objects are not removed. You can examine the script by viewing the `unmark.sql` file. When the `unmark` script execution encounters a fatal error on any database object, the `pdb_setreptable` command returns the following message:

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

- If you create a new table using a table name that was previously marked you must mark the new table by executing the `pdb_setreptable` command with the `mark` option, even if you did not unmark the previous table.
- When you use the `unmark` keyword to remove replication marking from a primary table, the Replication Agent verifies that replication is disabled for that table and checks to make sure that there are no pending (unprocessed) operations for that table in the transaction log. If replication is not disabled, or there is a pending operation for that table in the transaction log, `pdb_setreptable` returns an error.
- When you use the `unmark` keyword to remove replication marking from primary tables, you can also specify the `force` keyword to immediately remove replication marking from primary tables, without regard to whether replication is disabled.

The `force` keyword also ignores script execution errors. If the `unmark` script execution encounters a fatal error on any database object, the `pdb_setreptable` command returns the following message:

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

- When `pdb_setreptable` is invoked with either no option or a single option, it returns marking information about the user tables in the primary database:

- If `pdb_setreptable` is invoked with no option, it returns a list of all marked tables in the primary database.

Note Invoking the `pdb_setreptable` command with no option produces the same result as invoking the `pdb_setreptable` with the `mark` keyword.

- If `pdb_setreptable` is invoked with a table name, it returns complete marking information about the specified primary table.
- If `pdb_setreptable` is invoked with the `mark` keyword, it returns a list of all marked tables in the primary database.
- If `pdb_setreptable` is invoked with the `unmark` keyword, it returns a list of all unmarked tables in the primary database.
- If `pdb_setreptable` is invoked with the `enable` keyword, it returns a list of all marked tables in the primary database for which replication is enabled.
- If `pdb_setreptable` is invoked with the `disable` keyword, it returns a list of all marked tables in the primary database for which replication is disabled.

Tables marked for replication are listed in the marked objects table. All other user tables are considered unmarked.

Note The Replication Agent transaction log tables and shadow tables are not included in the list of unmarked tables. Also not included are any synonyms, views, or aliases of these database objects.

For tables listed as unmarked or disabled, transactions will not be captured for replication.

- When `pdb_setreptable` is invoked with the `all` keyword and an action keyword (`mark`, `unmark`, `enable`, or `disable`), the action specified is applied to either all tables in the primary database, or all marked tables in the primary database.

- If `pdb_setreptable` is invoked with the `all` and `mark` keywords, all user tables in the primary database are marked for replication.

Note Tables owned by users contained in the owner filter list will not be marked. However, you will be able to mark any individual table.

- If `pdb_setreptable` is invoked with the `all` and `unmark` keywords, it removes replication marking from all marked tables in the primary database.

You can specify the `force` keyword after the `unmark` keyword to force immediate unmarking of all marked tables, or to unmark tables for which replication is still enabled, or to force the script execution to ignore errors and continue an unmarking operation that failed previously.

- If `pdb_setreptable` is invoked with the `all` and `enable` keywords, it enables replication for all marked tables in the primary database.
- If `pdb_setreptable` is invoked with the `all` and `disable` keywords, it disables replication for all marked tables in the primary database.
- When `pdb_setreptable` is invoked with a valid user table name, followed by an action keyword (`mark`, `unmark`, `enable`, or `disable`), the action specified is applied to the specified table.
- If `pdb_setreptable` is invoked with a table name and the `mark` keyword, it marks the specified table in the primary database for replication.

Note When an individual table is marked the owner filter list is not checked. This allows users to mark a table that has an owner in the owner filter list.

- If `pdb_setreptable` is invoked with a table name and the `unmark` keyword, it removes replication marking from the specified table in the primary database.

You can specify the `force` keyword after the `unmark` keyword to force immediate unmarking of the specified table, to unmark a table for which replication is still enabled, or to force the script execution to ignore errors and continue an unmarking operation that failed previously.

- If `pdb_setreptable` is invoked with a table name and the `enable` keyword, it enables replication for the specified marked table in the primary database.

If the enable script execution encounters a fatal error on any database object, the `pdb_setreptable` command returns the following message:

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

- If `pdb_setreptable` is invoked with a table name and the `disable` keyword, it disables replication for the specified marked table in the primary database.

If the disable script execution encounters a fatal error on any database object, the `pdb_setreptable` command returns the following message:

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

- If the table name you specify does not exist in the primary database, the `pdb_setreptable` command returns an error.
- When `pdb_setreptable` is invoked with a primary table name and a replicated name, followed by the `mark` keyword, the primary table is marked for replication with the specified replicated name.

If the primary table name you specify does not exist in the primary database, the `pdb_setreptable` command returns an error.

By specifying a replicated name, transactions can be replicated to a table in the replicate database that has a different name from the primary table.

Note The replicated name you specify with the `pdb_setreptable` command must match a table name specified by a `with all tables named` clause in a Replication Server replication definition for the primary database connection. The Replication Agent cannot validate the replication definition, but if it does not exist, or if the `with all tables named` clause does not match the replicated name specified with `pdb_setreptable`, replication from the primary table will fail.

You can also specify the owner keyword after the mark keyword so that when operations against the primary table are replicated, the primary table owner name will be attached to the replicate table name in the form `owner.tablename`.

Note If you want to use an owner-qualified replicate table name with the replicate owner's name, use the owner keyword with the `pdb_setreptable` command. If you specify an unqualified replicate table name, the primary table owner name is sent with the replicate table name in the LTL.

- If the Replication Agent transaction log does not exist in the primary database (UDB) or the RASD is not initialized (Oracle and Microsoft SQL Server), the `pdb_setreptable` command returns an error.
- To replicate a table that contains column names that have spaces, you must set `structured_tokens` to `true`.
- (For UDB and Oracle) If a table contains a column with a datatype that is not supported for replication, the `pdb_setreptable` command using the mark keyword may fail with an error similar to:

```
Command <pdb_setreptable> failed - Table <MYTABLE>
could not be marked because:
```

```
The table contains an unsupported data type.
```

To force the table to be marked, excluding the unsupported datatype columns from replication, add the `force` keyword to the `pdb_setreptable` command.

See also

`pdb_setrepcol`, `pdb_setrepproc`, `ra_config`

pdb_skip_op

Description Returns, adds to, or removes operations from a list of operations to skip during processing. The format of the record locator is database-specific.

Note This command is available only for Oracle and Microsoft SQL Server.

Syntax `pdb_skip_op [add | remove], [locator | scn, thread, rba] | [remove, all]`

Parameters

add

Adds a specified ID to the list of identifiers of records to skip.

remove

Removes a specified ID from the list of identifiers to skip.

locator

The locator keyword from the list of LTM locators to identify the operations to skip.

scn

(For Oracle) The system change number (SCN) keyword identifies a specified log record to skip.

thread

(For Oracle) The thread keyword of the redo log thread of the operation to skip.

rba

(For Oracle) The record byte address (RBA) keyword of the log record to skip.

all

Allows you to add or remove all IDs in the list of identifiers to skip.

Examples

Example 1

```
pdb_skip_op
```

This command with no parameters returns a list of the identifiers for the records you want to skip.

Example 2

```
pdb_skip_op add, id
```

This command adds an ID to the list of identifiers you want to skip.

Example 3

```
pdb_skip_op remove, id
```

To remove an ID to the list of identifiers you want to skip:

Example 4

```
pdb_skip_op remove, all
```

This command removes all the IDs on the list of identifiers you want to skip.

Example 5

```
pdb_skip_op add, locator
```

This command adds an operation, referred to by its location, to the list of identifiers that you want to skip.

Usage

- The `pdb_skip_op` command allows you to skip problem records, thereby avoiding having to reinitialize the Replication Agent.
- Skipped records are written to the system log as a warning message.
- The `pdb_skip_op` command is valid when the Replication Agent is in *Admin* or *Replication Down* state.

- The format of the identifier is database specific:

- For Microsoft SQL Server, the identifier is the RBA and it has the following form:

```
virtualfileseqno:pageid:operationsseqno
```

where:

- `virtualfileseqno` is the virtual log sequence number (LSN).
- `pageid` is the page number.
- `operationsseqno` is the sequence number of the operation.

You can get the LSN using the Microsoft SQL Server DBCC command, or you can parse it from the locator. For example:

- LSN in decimal format:

```
489:112:27
```

- LSN in hexadecimal format:

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

- For Oracle, the identifier contains the system change number (SCN), redo log thread, and record byte address (RBA). It has the following form:

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

where:

- wrap is the SCN wrap number.
- base is the SCN base number.
- sub is the SCN subindex.
- thread is the redo log thread number that the operation occurred on.
- lsn is the RBA log sequence number.
- blknum is the RBA block number.
- blkoffset is the RBA offset into the block where this record resides.

The values must be specified in the `pdb_skip_op` command together, as shown above, enclosed in quotes, with each item separated by a period.

For example:

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

All values can be described as hexadecimal by prefixing the identifier with an “0x” as follows:

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

or:

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

See also `ra_helplocator`, `ra_locator`

pdb_truncate_xlog

Description Truncates the Replication Agent primary database transaction log on demand.

- The behavior of this command changes based on the value of the configuration parameter `pdb_include_archives`:
 - When `pdb_include_archives` is false, triggers the archive process to archive any online redo logs that have already been processed by Replication Agent.

- When `pdb_include_archives` is true, removes old archive redo log files from the path specified by `pdb_archive_path`.

Note Truncation of the old archive log files from the `pdb_archive_path` directory is performed only if the `pdb_archive_remove` parameter is set to true.

- For UDB, if `pdb_archive_remove` is set to true, the archived transaction log files are removed from the location specified by the `pdb_archive_path` configuration parameter. The primary transaction log files will be deleted up to, but not including, the log file that contains the lsn referenced in the Replication Agent truncation point locator.
- For Microsoft SQL Server, the primary database log is truncated. Based on the LTM locator from the Replication Server, Replication Agent queries the primary database to obtain the newest transaction ID that can be truncated, and marks the transaction log space prior to the newest transaction as reusable.

For more information on how Replication Agent affects each type of database when `pdb_truncate_xlog` is executed, see the *Replication Agent Primary Database Guide*.

Syntax

`pdb_truncate_xlog`

Usage

- When `pdb_truncate_xlog` is invoked, Replication Agent immediately truncates the primary database transaction log based on the most recent truncation point received from the primary Replication Server. The truncation point is part of the information contained in the LTM Locator.
- To update the LTM Locator from the primary Replication Server, use the `ra_locator` command.
- The `pdb_truncate_xlog` command is asynchronous and it does not return success or failure (unless an immediate error occurs). You must examine the Replication Agent system log to determine success or failure of the `pdb_truncate_xlog` command.
- If the Replication Agent primary database log does not exist or if a connection failure occurs, the `pdb_truncate_xlog` command returns an error message.
- You can use the `ra_config` command to specify the type of automatic truncation you want. You can use the `pdb_truncate_xlog` command to truncate the transaction log if automatic truncation is not sufficient to manage the size of the transaction log.

- The `pdb_truncate_xlog` command is valid when the Replication Agent instance is in the *Admin*, *Replicating*, or *Replication Down* state.

See also `ra_config`, `ra_locator`

pdb_version

Description Returns the type and version of the primary data server.

Syntax `pdb_version`

Usage The actual results returned vary depending on the type of primary data server.

See also `ra_version`, `ra_version_all`

pdb_xlog

Description Returns the names of transaction log base objects; creates transaction log base objects in the primary database; or removes transaction log base objects from the primary database.

For Oracle and Microsoft SQL Server, this command verifies permissions are valid for the Replication Agent to obtain system data from the primary database. It also checks the condition of the primary database to determine if archiving is turned on or off, and then loads the RASD with system data from the primary database.

Note Because the `pdb_xlog create` command is deprecated, Sybase recommends that you use `pdb_xlog init`.

Syntax `pdb_xlog [[[init | create | remove] [, force]] | move_trunct]`

Parameters

`init`
The keyword for moving the truncation point to the end of the transaction log.

`create`
The keyword for creating a transaction log. Deprecated; use the `init` keyword instead.

remove

The keyword for removing a transaction log.

force

A keyword that refers to the remove or the init operation.

move_truncpt

A keyword that moves the truncation point; for Oracle and Microsoft SQL Server only.

Examples

```
pdb_xlog init
```

This command initializes the Replication Agent, creating any required transaction log base components.

```
pdb_xlog init, force
```

This command re-initializes the Replication Agent, creating or re-loading any required transaction log base components.

```
pdb_xlog remove
```

This command removes any Replication Agent transaction log base components.

```
pdb_xlog remove, force
```

This command removes any Replication Agent transaction log base components and ignores any individual errors that occur during removal.

```
pdb_xlog init, move_truncpt
```

This command moves the transaction log truncation point to the end of the current transaction log (Oracle and Microsoft SQL Server).

Usage

- When `pdb_xlog` is invoked with no option, it returns the actual names (not synonyms or aliases) of all Replication Agent transaction log base objects in the primary database. For Oracle and Microsoft SQL Server, if you have initialized the Replication Agent, it returns the name of the component and the primary database instance name.

Note See the chapter for your specific primary data server in the *Replication Agent Primary Database Guide* for more information on Replication Agent object names.

- If `pdb_xlog` is invoked with no option, and the Replication Agent transaction log base objects do not exist in the primary database, or the RASD has not been initialized (For Oracle and Microsoft SQL Server), the command returns no information.

- If `pdb_xlog` is invoked with the `init` keyword, the truncation point is established at the end of the primary database transaction log.

Note For Microsoft SQL Server, during the `pdb_xlog` `init` process, Replication Agent may connect to the Microsoft SQL Server using `pds_dac_port_number`. For more information, see the *Replication Agent Primary Database Guide*.

- If `pdb_xlog` is invoked with the `init, force` keywords, the truncation point is moved to the end of the log if the Replication Agent is not already initialized. However, if the Replication Agent is already initialized, the truncation point is not moved.

Note `pdb_xlog` `init` with the `force` keyword should only be used when advised by Sybase Technical support.

- If `pdb_xlog` is invoked with the `move_truncpt` keyword, the truncation point is moved to the end of the log without change or modification to any Replication Agent components. (For Oracle, this is the end of the current online redo log.) The `move_truncpt` option has no effect if the Replication Agent has not been initialized.

Note To prevent Replication Server from requesting a log starting point that occurs earlier in the log than the location established by the `move_truncpt` option, the Replication Server's LTM locator value for the primary connection must be zeroed. Execute Replication Server System Database (RSSD) command `rs_zerohtm` against the primary database connection to zero the LTM locator.

(For Oracle and Microsoft SQL Server) If you move the secondary truncation point to the end of the primary database transaction log using `pdb_xlog` `init, move_truncpt`, you risk skipping over any DDL commands record in the log. The DDL commands might have been used by the Replication Agent to update information stored within the Replication Agent System Database (RASD). If the RASD contents are incorrect due to skipping processing of some log records, you may force all of the schema information in the RASD to be refreshed using command `pdb_xlog` `init, force`. If only the schema for a single object stored in the RASD is of concern, you can unmark and remark just that single object, which will force the schema of the object to be reread into the RASD.

- When `pdb_xlog` is invoked with the `init` keyword, Replication Agent does the following:
 - Generates a SQL script that creates the Replication Agent tables and procedures in the primary database.
 - Saves the generated script in a file called *partinit.sql* in the `RAX-15_5\inst_name\scripts\xlog` directory, where *inst_name* is the name of the Replication Agent instance.

Note If the value of the `pdb_auto_run_scripts` configuration parameter is `false`, the *partinit.sql* script will be saved but not executed. However, you cannot manually run the script. To complete initializing Replication Agent, you must first set `pdb_auto_run_scripts` to `true`, then re-run the `pdb_xlog init` command.

- Executes the script to create the Replication Agent transaction log base objects in the primary database (if the value of the `pdb_auto_run_scripts` configuration parameter is `true`).
- After the script completes successfully, moves the *partinit.sql* file to the `RAX-15_5\inst_name\scripts\xlog\installed` directory.
- If the create script fails, it is stored in a file (*partinit.sql*) in the `RAX-15_5\inst_name\scripts\xlog` directory and the transaction log is not created. You can examine the script by viewing the *partinit.sql* file.
- If `pdb_xlog` is invoked with the `init` keyword and the Replication Agent objects already exist in the primary database or the RASD has been initialized (For Oracle and Microsoft SQL Server), then `pdb_xlog` returns an error message.
- When `pdb_xlog` is invoked with the `remove` keyword, Replication Agent does the following:
 - For UDB, `pdb_xlog remove` command will uninstall the jar files from the primary database (the jars are installed by the `pdb_xlog init` command).

It is necessary to use `pdb_xlog remove` command to de-initialize Replication Agent for UDB, and remove the truncation stored procedures and jars from the database.
 - Generates a SQL script that deletes the tables and procedures required for the transaction log base objects in the primary database.

- Saves the generated script in a file called *partdeinit.sql* in the *RAX-15_5\inst_name\scripts\xlog* directory, where *inst_name* is the name of the Replication Agent instance.

Note If the value of the `pdb_auto_run_scripts` configuration parameter is `false`, the *partdeinit.sql* script will be saved but not executed automatically. You cannot manually run the script. To complete de-initializing Replication Agent, you must first set `pdb_auto_run_scripts` to `true`, then re-run the `pdb_xlog remove` command.

- Executes the script to delete the Replication Agent objects from the primary database (if the value of the `pdb_auto_run_scripts` configuration parameter is `true`).
- After the script completes successfully, moves the *partdeinit.sql* file to the *RAX-15_5\inst_name\scripts\xlog\installed* directory.
- If the script fails, it is stored in a file (*partdeinit.sql*) in the *RAX-15_5\inst_name\scripts\xlog* directory and the Replication Agent objects are not deleted from the primary database. You can examine the script by viewing the *partdeinit.sql* file.
- When `pdb_xlog` is invoked with the `remove` keyword followed by the `force` keyword, the *partdeinit.sql* script continues executing, even if errors occur. The `force` keyword may be useful when a previous `remove` operation failed and the *partdeinit.sql* script terminated with an error.
- If `pdb_xlog` is invoked with the `remove` keyword, and Replication Agent objects do not exist in the primary database or the RASD has not been initialized (For Oracle or Microsoft SQL Server), then `pdb_xlog` returns an error message.
- If `pdb_xlog` is invoked with the `remove` keyword and any objects in the primary database are still marked for replication, then `pdb_xlog` returns an error message.

You can use the `pdb_setrepproc` and `pdb_setreptable` commands to determine which stored procedures and tables in the primary database are still marked. You also can use the `pdb_setreppddl` command to determine if DDL is enabled.

Even if objects are marked in the primary database, you can use the `pdb_xlog` command with the `remove` keyword followed by the `force` keyword to unmark any marked objects, and then remove the transaction log objects.

- If `pdb_xlog` is invoked with no option, the command is valid when the Replication Agent instance is in the *Admin*, *Replicating*, or *Replication Down* states.
- If `pdb_xlog` is invoked with either the `init` or `remove` keyword, the command is valid only when the Replication Agent instance is in the *Admin* or *Replication Down* state.
- The `pdb_xlog init` command will verify that the following privileges have been granted to `pds_username`:
 - EXECUTE_CATALOG_ROLE
 - SELECT ON V_\$LOGMNR_CONTENTS
 - SELECT ON V_\$LOGMNR_LOGS

These privileges are necessary for the `ra_dumptran` and `ra_helpop` commands to function properly. These privileges are not required for replication, only for using the `ra_dumptran` and `ra_helpop` commands, which are used in debugging and troubleshooting. If these privileges have not been granted at the time `pdb_xlog init` is invoked, a warning message is returned and logged in the Replication Agent log file.

- For more information about the Replication Agent transaction log, see the chapter for your specific primary data server in the *Replication Agent Primary Database Guide*.

See also `pdb_setrepcol`, `pdb_setrepproc`, `pdb_setreptable`, `ra_config`

quiesce

Description Stops all Replication Agent processing in *Replicating* state, and puts the Replication Agent instance in *Admin* state.

Syntax `quiesce`

Usage

- When the `quiesce` command is invoked, it stops all current replication processing in the Replication Agent instance:
 - The Log Reader component stops reading operations from the transaction log when the scan reaches the end of the log. It continues to send change-set data to the Log Transfer Interface component until it finishes processing the last operation scanned.

- The Log Transfer Interface component stops sending LTL commands to the Replication Server as soon as it finishes processing the last change set it receives from the Log Reader.
- When the Log Transfer Interface component is finished processing its input queue and sending the resulting LTL, the Replication Agent instance releases all of its connections to the primary database, and drops its connection to the primary Replication Server (and RSSD, if connected).
- The Replication Agent instance goes from *Replicating* state to *Admin* state.
- If the Replication Agent internal queues are full when the quiesce command is invoked, the quiesce processing may take a while to complete, and there may be a delay before the Replication Agent instance completes its transition to *Admin* state.
- Before moving the Replication Agent to the *Admin* state, the quiesce command waits until all data in the primary log has been read and sent to the Replication server.
- If the Replication Agent instance is in *Admin* state, the quiesce command returns an error.
- The quiesce command is valid only when the Replication Agent instance is in *Replicating* state.

Note The action of the suspend command is similar to that of the quiesce command, except that the suspend command stops Replication Agent processing immediately and flushes all data in the internal queues.

See also `ra_status`, `resume`, `shutdown`, `suspend`

ra_config

Description Returns help information for Replication Agent configuration parameters, or sets the value of a specified configuration parameter.

Syntax `ra_config [param [, value]]`

Parameters *param*
The name of a Replication Agent configuration parameter.

value

The value to be assigned to the configuration parameter specified in the *param* option. You can use the keyword *default* to set the specified parameter to its default value.

Examples

Example 1

```
ra_config
```

When this command is issued with *no* parameters, it returns a list of all Replication Agent configuration parameters.

Example 2

```
ra_config use_rssd
```

This command returns the current value of the *use_rssd* configuration parameter.

Example 3

```
ra_config scan_sleep_max, 60
```

This command changes the value of the *scan_sleep_max* parameter to 60.

Usage

- If *ra_config* is invoked with no option, it returns a list of all Replication Agent configuration parameters.
- If *ra_config* is invoked with the *param* option, it returns information only for the specified configuration parameter.
- If *ra_config* is invoked with the *param* and *value* options, it changes the setting of the specified configuration parameter to the value specified in the *value* option.
- You can use the keyword *default* in place of the *value* option to reset a configuration parameter to its default value. For example:

```
ra_config use_rssd, default
```

- The following information is returned for each configuration parameter:
 - Parameter name – the name of the parameter.
 - Parameter type – the datatype of the parameter's value (for example, string, numeric, or Boolean).
 - Current value – the value of the parameter in effect at the time *ra_config* is invoked.

- Pending value – if different from the current value, the value to which the parameter was set by a previous invocation of the `ra_config` command, but which has not yet taken effect.
- Default value – the value of the parameter when the Replication Agent instance configuration file is created.
- Legal values – the values that are allowed for the parameter, for example, a range of numbers or a list of specific strings.
- Category – refers to the Replication Agent component affected by the value of the parameter.
- Restart – refers to parameters that require the Replication Agent instance to be shut down and restarted before a change in value takes effect.
- When `ra_config` is invoked with either no option, or only the `param` option, the command is valid when the Replication Agent instance is in the *Admin*, *Replicating*, or *Replication Down* state.
- If `ra_config` is invoked when the Replication Agent instance is in *Replicating* state, with the `param` and `value` options for a parameter that can be changed only in *Admin* state, it returns an error.
- When `ra_config` is invoked with the `param` and `value` options, the command is always valid when the Replication Agent instance is in the *Admin* or *Replication Down* state.
- See Chapter 2, “Configuration Parameters” for more information.

See also

`ra_help`, `ra_set_login`

ra_date

Description

Returns the current date and time from the Replication Agent instance.

Syntax

`ra_date`

Usage

- When `ra_date` is invoked, it returns the current date and time from the Replication Agent instance in the form of a Sybase datetime datatype, as follows:

```
Current RA Date
-----
Jan 11 2010 12:09:47.310
```

(1 row affected)

- The `ra_date` command is valid when the Replication Agent instance is in the *Admin*, *Replicating*, or *Replication Down* state.

See also `pdb_date`, `ra_config`

ra_deviceoffset

Description Changes the raw disk device read offset for a log device recorded in the RASD.

Note This command is available only for Oracle and Microsoft SQL Server.

Syntax `ra_deviceoffset device_id, device_offset`

Parameters

device_id

For Oracle, the device ID is the Oracle redo log “Group number.” For Microsoft SQL Server, this device ID can be obtained using the `ra_helpdevice` command.

device_offset

The offset of the raw device from which Replication Agent starts a log scan.

Examples

```
ra_deviceoffset 1,20
```

This command specifies the raw device read offset to the log device ID “1” as:

```
20
```

Usage

- When `ra_devicepath` is invoked, Replication Agent records the specified raw device read offset for the specified log device in the RASD.
- To get information about log devices stored in the RASD, use the `ra_helpdevice` command.
- When `ra_updatedevices` is invoked, device offset is not reset to the default value. The default device read offset is zero.
- If `ra_deviceoffset` is invoked when the Replication Agent instance is in *Replicating* state, it returns an error.
- The `ra_deviceoffset` command is valid only when the Replication Agent instance is in the *Admin*, *Replicating*, or *Replication Down* state.

See also `ra_helpdevice`, `pdb_xlog`, `ra_updatedevices`

ra_devicepath

Description	Changes the disk device path for a log device recorded in the RASD. For Oracle and the ASM log devices, see the <code>pdb_asmdiskmap</code> command.
	<hr/> Note This command is available only for Oracle and Microsoft SQL Server. <hr/>
Syntax	<code>ra_devicepath device, dev_path</code>
Parameters	<p><i>device</i> The device ID is the Oracle redo log “Group number.”</p> <p><i>dev_path</i> The path that points to the disk log device for the device specified in the <i>device</i> option.</p>
Examples	<pre>ra_devicepath 3, d:\software\oracle\devices\redo001.log</pre> <p>This command specifies the disk device path to the log device ID “3” as:</p> <pre>d:\software\oracle\devices\redo001.log</pre>
Usage	<ul style="list-style-type: none"> • When <code>ra_devicepath</code> is invoked, Replication Agent records the specified disk device path for the specified log device in the RASD. • To get information about log devices stored in the RASD, use the <code>ra_helpdevice</code> command. • If you invoke <code>ra_updatedevices</code> after you set a device path using <code>ra_devicepath</code>, you must use <code>ra_devicepath</code> again to re-set the path if you need to alter the default path for a log device. The default device path is the device path returned by the primary data server. • If you invoke <code>ra_devicepath</code> when the Replication Agent instance is in <i>Replicating</i> state, it returns an error. • The <code>ra_devicepath</code> command is valid only when the Replication Agent instance is in the <i>Admin</i> or <i>Replication Down</i> state.
See also	<code>ra_helpdevice</code> , <code>pdb_xlog</code> , <code>ra_updatedevices</code>

ra_downgrade_accept

Description The `ra_downgrade_accept` command is executed by the instance to which Replication Agent is being downgraded (the earlier version). This command completes the downgrade process started by the `ra_downgrade_prepare` command, which is executed by the instance from which Replication Agent is being downgraded (the later version).

Syntax `ra_downgrade_accept export_file`

Parameters *export_file*
The file name to which the RASD was extracted during execution of the `ra_downgrade_prepare` command. This can be the file name alone or the file name with its absolute path. The file is named *timestamp.export*, where *timestamp* is a timestamp taken at the moment `ra_downgrade_prepare` was invoked.

Note The *export_file* parameter is only used for Microsoft SQL Server and Oracle. When the `ra_downgrade_accept` command is used for UDB, the command takes no parameters.

Usage

- (Microsoft SQL Server and Oracle only) The `ra_downgrade_accept` command restores the RASD from the specified file. By default, this file is located in the *import* subdirectory under the directory specified by the `rasd_backup_dir` configuration parameter of the Replication Agent instance to which you are downgrading (the earlier version). The path to this file was returned if `ra_downgrade_prepare` executed successfully.
- To use the `ra_downgrade_accept` command, Replication Agent must be able to connect to the primary database.
- (Microsoft SQL Server and Oracle only) If the `ra_downgrade_accept` command executes successfully, Replication Agent shuts down.
- The `ra_downgrade_accept` command is valid when the Replication Agent instance is in the *Admin* or *Replication Down* state.

See also `ra_downgrade_prepare`

ra_downgrade_prepare

Description	The <code>ra_downgrade_prepare</code> command prepares Replication Agent to downgrade to an earlier version. When executed in Replication Agent for Microsoft SQL Server or Replication Agent for Oracle, the <code>ra_downgrade_prepare</code> command copies the RASD contents to an export file. This file is then applied by the instance to which Replication Agent is being downgraded (the earlier version) to complete the downgrade.
Syntax	<code>ra_downgrade_prepare inst_path</code>
Parameters	<i>inst_path</i> The absolute path of the Replication Agent instance to which you are downgrading (the earlier version).
Usage	<ul style="list-style-type: none"> To prepare for downgrading to an earlier version of Replication Agent, the <code>ra_downgrade_prepare</code> command first changes Replication Agent system objects in the primary database to match those of the instance to which Replication Agent is being downgraded. (Microsoft SQL Server and Oracle only) The <code>ra_downgrade_prepare</code> command also extracts the contents of the Replication Agent System Database (RASD) to a file named <i>timestamp.export</i>, where <i>timestamp</i> is a timestamp taken at the moment <code>ra_downgrade_prepare</code> was invoked. By default, this file is located in the <i>import</i> subdirectory under the directory specified by the <code>rasd_backup_dir</code> configuration parameter of the Replication Agent instance to which you are downgrading (the earlier version). The path to this file is returned if <code>ra_downgrade_prepare</code> executes successfully. The <code>ra_downgrade_prepare</code> command is valid when the Replication Agent instance is in the <i>Admin</i> or <i>Replication Down</i> state.
See also	<code>ra_downgrade_accept</code>

ra_dump

Description	Emulates the Replication Server <code>rs_dumpdb</code> and <code>rs_dumptran</code> system functions.
Syntax	<code>ra_dump [database transaction,] dbname, dump_label</code>
Parameters	<i>database</i> A keyword that causes the primary Replication Server to apply the function string associated with the <code>rs_dumpdb</code> system function.

transaction

A keyword that causes the primary Replication Server to apply the function string associated with the `rs_dumptran` system function.

dbname

The name of the database to be dumped.

dump_label

A `varchar(30)` value that contains information to identify the database dump.

Examples

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

Here, `MSSQL_source` is the database name, and `rssddmp` is the dump label.

Usage

- When `ra_dump` is invoked, Replication Agent places a dump marker in the Replication Agent transaction log to facilitate coordinated dumps.
- The `ra_dump` command returns an error message if the transaction log does not exist.
- The `ra_dump` command is valid when the Replication Agent instance is in the *Admin*, *Replicating*, or *Replication Down* state.
- For more information about the Replication Server `rs_dumpdb` and `rs_dumptran` system functions, see the *Replication Agent Administration Guide* and *Replication Agent Primary Database Guide*.

See also

`ra_config`, `ra_migrate`

ra_dumptran

Description

This command returns information for use in troubleshooting a specified database transaction.

Note This command is available only for Oracle.

Syntax

```
ra_dumptran "[opid | locator | tranid]"
```

Parameters

opid

The Replication Agent operation ID for a database operation.

locator

The Replication Agent locator for a database operation.

transid

The Oracle transaction ID for the database transaction.

Examples

Example 1

```
ra_dumptran
0x0000.01783d95.0000:0001.000003fe.00000031.0010
go
```

This command returns information about the operation specified by the Replication Agent operation ID:

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, 24657301, 1, 1022.49.16
DUMP FILE NAME	C:\somepath\XID0004.0016.00000016.log
(8 rows affected)	

Example 2

```
ra_dumptran
0000000001783d9600020001000003fe0000003400e8000001783d
9500000000
go
```

This command returns information about the operation specified by the Replication Agent locator:

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)	

Example 3

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

This command returns information about the transaction specified by the Oracle transaction ID:

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)

Example 4 The following is an example of log-file output from the ra_dumptran command. The log file contains both transaction information and information about all operations in the transaction:

```
File name: C:\somepath\XID0004.0016.00000016.log
File contents:
BEGIN OPERATION ID 0x0000.01783d95.0000:0001.000003fe.00000031.0010
BEGIN SCN          24657302
TRANSACTION ID     0004.0016.00000016
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      0x0000.01783d95.0000:0001.000003fe.00000031.0010 0
NULL      NO          START          set transaction read write;
24657301 1      0x0000.01783d95.0000:0001.000003fe.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;

```

Usage

- The `ra_dumptran` command dumps all operations for a specified transaction to an exclusive log file used in troubleshooting a failed operation or transaction. The log-file header consists of the result set returned by `ra_dumptran` and includes the following rows:
 - **BEGIN OPERATION ID** – the Replication Agent operation ID for the transaction begin operation. This field is in the format *wrap.scn.subscn.thread.lsn.block.offset*, where:
 - *wrap.scn.subscn* is the system change number (SCN) for the begin operation.
 - *thread* is the database thread number.
 - *lsn* is the log sequence number for the begin operation.
 - *block* is the block where the begin operation resides.
 - *offset* is the offset into the operation where the begin operation resides.
 - **BEGIN SCN** – the SCN for the transaction operation as logged in a redo log file.
 - **TRANSACTION ID** – the ID of the transaction that the operation is a part of.
 - **USERNAME** – the name of the user that executed the transaction.
 - **EXECUTION TIME** – the date and time at which the transaction was executed.
 - **THREAD NUMBER** – the Oracle thread that executed the transaction.
 - **TRANSACTION SKIP COMMAND** – the Replication Agent command that causes the transaction to be skipped by Replication Agent during replication.
 - **DUMP FILE NAME** – The name of the log file to which the transaction is written by `ra_dumptran`.

- The log file specified by the DUMP FILE row also contains the operation results for the specified transaction:
 - SCN – the SCN for the operation as logged in a redo log file.
 - THREAD – the thread that executed the operation.
 - OPERATION ID – the Replication Agent operation ID for the transaction begin operation. This field is in the format *wrap.scn.subscn.thread.lsn.block.offset*.
 - OBJECT ID – the object ID of the affected object.
 - OBJECT NAME – the name of the affected object.
 - REPLICATE – whether or not (YES or NO) the object affected by the operation is marked for replication by Replication Agent.
 - OPERATION – the operation type.
 - SQL – the SQL statement for the operation.
- The ra_dumptran command cannot operate properly unless the Oracle LogMiner script, *\$ORACLE_HOME/rdbms/admin/dbmslm.sql*, has been installed at the primary database. If this script has not been installed, ra_dumptran will return an error.
- After LogMiner is installed, create a public synonym so that you do not have to log in as the owner to execute LogMiner functions:

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

Note This is required if you are using Oracle 10g.

- The following privileges must be granted to pds_username for the ra_dumptran command to function properly:
 - EXECUTE_CATALOG_ROLE
 - SELECT ON V_\$LOGMNR_CONTENTS
 - SELECT ON V_\$LOGMNR_LOGS
 - SELECT ANY TRANSACTION

- If the `ra_dumptran` command returns no result for a specified *opid* or *locator* value, the corresponding database operation may be one of many operations in a database transaction. In this case, you should instead specify the transaction ID of the transaction to which the database operation belongs.

ra_help

Description	Returns help information for Replication Agent commands.
Syntax	<code>ra_help [command]</code>
Parameters	<i>command</i> The name of a Replication Agent command for which you want to view help information.
Examples	<p>Example 1</p> <pre>ra_help</pre> <p>This command returns help for all Replication Agent commands.</p> <p>Example 2</p> <pre>ra_help pdb_gen_id</pre> <p>This command returns help for the <code>pdb_gen_id</code> command.</p>
Usage	<ul style="list-style-type: none"> • If <code>ra_help</code> is invoked with no option, it returns help information for all Replication Agent commands. • If <code>ra_help</code> is invoked with the <i>command</i> option, it returns help information only for the specified command. • The <code>ra_help</code> command is valid when the Replication Agent instance is in the <i>Admin</i>, <i>Replicating</i>, or <i>Replication Down</i> state.
See also	<code>ra_config</code>

ra_helparticle

Description Returns information about primary database articles from the RASD.

Note This command is available only for Oracle and Microsoft SQL Server.

Syntax ra_helparticle [*article*, [*version*]]

Parameters

article

The name or object ID of an article (table or procedure) in the primary database. Article names can be qualified with an owner name in the following form:

```
owner.article
```

Owner qualification of article names is optional.

version

A hexadecimal locator value that identifies the version of the article specified in the *article* option.

Examples

Example 1

```
ra_helparticle
```

This command returns information about all versions of all articles in the RASD.

Example 2

```
ra_helparticle table1
```

This command returns information about the current version of the article named “table1” in the RASD.

Example 3

```
ra_helparticle table1,  
000000000000210a400003334000700003334000699940000d413c50000000000
```

This command returns information about version 000000000000210a400003334000700003334000699940000d413c5000000000 of the article named “table1” in the RASD.

Usage

- The ra_helparticle command returns the following information for articles (tables and procedures):
 - Article object ID
 - Primary database name

- Article owner name or alias
- Article name
- Article type (table or procedure)
- Article status (Current, Archived, or Dropped)
- Article version number

All information except the article type, article status, and article version number are the values returned by the primary database when the Replication Agent is initialized with the `ra_init` command.

- If `ra_helparticle` is invoked with no option, it returns information for all versions of all articles (tables and procedures) in the RASD.
- If `ra_helparticle` is invoked with the *article* option, it returns information only for the current version of the specified article in the RASD.
- If `ra_helparticle` is invoked with the *article* and *version* options, it returns information only for the specified version of the specified article in the RASD.
- The `ra_helparticle` command is valid when the Replication Agent instance is in the *Admin*, *Replicating*, or *Replication Down* state.

See also

`ra_helpdb`, `ra_helpfield`, `ra_helplocator`, `ra_helpuser`

ra_helpdb

Description

Returns information about the primary database from the RASD.

Note This command is available only for Oracle and Microsoft SQL Server.

Syntax

`ra_helpdb`

Usage

- When `ra_helpdb` is invoked, it returns the following information about the primary database:
 - Database object ID
 - Database name

The database ID and database name are the values returned by the primary database when the Replication Agent is initialized with the `pdb_xlog` init command.

- The `ra_helppdb` command is valid when the Replication Agent instance is in the *Admin*, *Replicating*, or *Replication Down* state.
- The `ra_helppdb` command is valid only after the RASD has been initialized, that is, only after you have executed `pdb_xlog` init.

See also

`ra_devicepath`, `ra_helpparticle`, `ra_helpdevice`, `ra_helpfield`, `ra_helplocator`, `ra_helpuser`, `ra_updatedevices`

ra_helpdevice

Description

This command returns information about the primary database log devices from the RASD log device repository.

Note This command is available only for Oracle and Microsoft SQL Server.

Syntax

`ra_helpdevice [device]`

Parameters

device

The device ID of the primary database log device.

Examples

Example 1

```
ra_helpdevice
```

This command returns information about all primary database log devices recorded in the log device repository.

Example 2

```
ra_helpdevice 1
```

This command returns information about the primary database log device ID “1” in the log device repository.

Usage

- The `ra_helpdevice` command returns the following information for each primary database log device recorded in the RASD:

- Device ID – the log device ID defined by the primary data server.

Note For Oracle, the ID is the value of the Oracle Redo Log Group that this file belongs to.

- Database name – the name of the primary database associated with the log device.
 - Device name – the logical name of the log device defined by the primary data server.
 - Server device path – the path to a multiplexed version of the log device.
 - Disk mirror path – the path to the log device (at the standby site).
 - Disk device status – the current status of the server device path (ACCESSIBLE, NOT_VALID, or OPEN).
- The log device ID, primary database name, log device name, and server log device path are values returned by the primary data server when the Replication Agent is initialized with the `pdb_xlog init` command, or when the log device repository is updated with the `ra_updatedevices` command.
 - The disk mirror path is the current value recorded in the RASD. To find each log device, Replication Agent uses the disk mirror path recorded in its RASD.

For each log device recorded in the RASD, you can set or change the disk device path with the `ra_devicepath` command.

If you do not specify a disk device path using `ra_devicepath`, the value recorded for the disk mirror path is `DEFAULT`, and Replication Agent uses the value recorded for the server device path to find the log device.

(For Oracle) The disk mirror path for ASM is specified using the file created with the `pdb_asmdiskmap` command.

- The disk device status is updated by the Log Reader component each time you invoke the `ra_helpdevice` command.
- If `ra_helpdevice` is invoked with no option, it returns information for all log devices recorded in the RASD log device repository.
- If `ra_helpdevice` is invoked with the `device` option, it returns information only for the specified log device.
- The `ra_helpdevice` command is valid when the Replication Agent instance is in the *Admin*, *Replicating*, or *Replication Down* state.

See also `ra_devicepath`, `ra_helpdb`, `ra_updatedevices`

ra_helpdeviceoffset

Description Returns device offset information about primary database log devices from the RASD log device repository.

Note This command is available only for Oracle and Microsoft SQL Server.

Syntax `ra_helpdeviceoffset [device]`

Parameters *device*
The device ID of the primary database log device.

Examples **Example 1**

```
ra_helpdeviceoffset
```

This command returns the device offset information about all primary database log devices recorded in the log device repository.

Example 2

```
ra_helpdeviceoffset 1
```

This command returns information about the primary database log device ID 1 in the log device repository.

Usage

- The `ra_helpdeviceoffset` command returns the following information for each primary database log device recorded in the RASD:

- Device ID – the log device ID defined by the primary data server.

Note For Oracle, the device ID is the value of the Oracle Redo Log Group that contains this file.

- Database name – the name of the primary database associated with the log device.
- Device name – the logical name of the log device defined by the primary data server.
- Server device path – the path to a multiplexed version of the log device.

- Disk device path – the path to the log device (at the standby site).
- Disk device offset – the offset from which Replication Agent starts to scan the Oracle redo log in the log device. The value of the device offset is the value of the `raw_device_offset` parameter in the Replication Agent configuration file added to the offset value set by the command `ra_deviceoffset`.
- Disk device status – the current status of the server device path (ACCESSIBLE, NOT_VALID, or OPEN).
- The disk device path is the current value recorded in the RASD. Replication Agent uses the disk device path recorded in the RASD to locate each log device.
- For each log device recorded in the RASD, you can set or change the disk device path using `ra_devicepath`, and set or change the device offset using `ra_deviceoffset`.
- If you do not specify a disk device path using `ra_devicepath`, the value recorded for the disk device path is DEFAULT, and Replication Agent uses the value recorded for the server device path to find the log device.
- The disk device status is updated by the Log Reader component each time you invoke `ra_helpdeviceoffset`.
- If `ra_helpdeviceoffset` is invoked with no option, it returns information for all log devices recorded in the RASD log device repository.
- If `ra_helpdeviceoffset` is invoked with the device option, it returns information only for the specified log device.
- The `ra_helpdeviceoffset` command is valid when the Replication Agent instance is in *Admin*, *Replicating*, or *Replication Down* state.

See also

`ra_helpdevice`, `ra_deviceoffset`, `ra_devicepath`, `ra_updatedevices`

ra_helpfield

Description

Returns information about primary database fields (columns in tables, or input parameters in stored procedures) from the RASD.

Note This command is available only for Oracle and Microsoft SQL Server.

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

Parameters

article

The name or object ID of an article (table or procedure) in the primary database. Article names can be qualified with an owner name in the following form:

`owner.article`

Owner qualification of article names is optional.

version

A hexadecimal locator value that identifies the version of the specified article.

field

The name or object ID of a field (column or input parameter) in the specified article.

Examples

Example 1

`ra_helpfield authors`

This command returns information about all fields in the current version of the article named authors in the RASD.

Example 2

`ra_helpfield authors,
00000000000210a400003334000700003334000699940000d413c50000000000`

This command returns information about all fields in version 00000000000210a400003334000700003334000699940000d413c50000000000 of the article named “authors” in the RASD.

Example 3

`ra_helpfield authors,
00000000000210a400003334000700003334000699940000d413c50000000000, au_fname`

This command returns information about the field named au_fname in version 00000000000210a400003334000700003334000699940000d413c50000000000 of the article named “authors” in the RASD.

Usage

- The ra_helpfield command returns the following information for fields:
 - Field (column or input parameter) object ID
 - Field name
 - Field type ID
 - Field datatype (with precision, length, and scale)

- Field NULL mode
- Field IDENTITY status
- Field primary key status

All field information items are the values returned by the primary database when the Replication Agent is initialized with the `pdb_xlog init` command.

- If `ra_helpfield` is invoked with the *article* option, it returns information for all fields in the current version of the specified article in the RASD.
- If `ra_helpfield` is invoked with the *article* and *version* options, it returns information for all fields in the specified version of the specified article in the RASD.
- If `ra_helpfield` is invoked with the *article*, *version*, and *field* options, it returns information for the specified field in the specified version of the specified article in the RASD.
- The `ra_helpfield` command is valid when the Replication Agent is in either *Admin*, *Replicating*, or *Replication Down* state.
- No results are returned by this command if the RASD has not yet been initialized with the `pdb_xlog init` command.

See also

`ra_config`, `ra_help`, `ra_helparticle`, `ra_helppdb`, `ra_helpdevice`, `ra_helplocator`, `ra_helpuser`

ra_helplocator

Description Returns information about fields in the LTM Locator value.

Note This command is available only for Oracle and Microsoft SQL Server.

Syntax `ra_helplocator [locator_value]`

Parameters *locator_value*
The hexadecimal string value of an LTM Locator.

Examples **Example 1**

```
ra_helplocator
```

This command returns information about fields in the current LTM Locator value.

Example 2

```
ra_helplocator locator_value
```

This command returns information about fields in the specified LTM Locator value.

Usage

- The `ra_helplocator` command returns the following information about the LTM Locator value:
 - Locator field names
 - Locator field hexadecimal values
 - Locator field decimal values
- If `ra_helplocator` is invoked with no option, it returns information about fields in the current LTM Locator value.
- If `ra_helplocator` is invoked with the `locator_value` option, it returns information about fields in the specified LTM Locator value.
- The `ra_helplocator` command is valid when the Replication Agent instance is in the *Admin*, *Replicating*, or *Replication Down* state.

See also

`ra_config`, `ra_help`, `ra_locator`

See the *Replication Agent Primary Database Guide* for more information about locator fields and contents.

ra_helpop

Description

This command returns information for use in troubleshooting a specified database transaction log operation.

Note This command is available only for Oracle.

Syntax

```
ra_helpop “[opid | locator]”
```

Parameters

opid

The Replication Agent operation ID for a database operation.

locator

The Replication Agent locator for a database operation.

Examples

Example 1

```

ra_helpop
0x0000.01783d96.0000:0001.000003fe.00000034.00e8
go

```

This command returns information about the operation specified by the Replication Agent operation ID:

Name	Value
OPERATION ID	0x0000.01783d96.0000:0001.000003fe.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)

Example 2

```

ra_helpop
0000000001783d9600020001000003fe0000003400e8000001783d
9500000000
go

```

This command returns information about the operation specified by the Replication Agent locator:

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)
```

Usage

- The `ra_helpop` command displays database and Replication Agent information for a specified operation for use in troubleshooting a failed operation or transaction. The result set returned by `ra_helpop` includes the following rows:
 - OPERATION ID – the Replication Agent operation ID in the format `wrap.scn.subscn.thread.lsn.block.offset`, where:
 - `wrap.scn.subscn` is the system change number (SCN) for the specified operation.
 - `thread` is the database thread number.
 - `lsn` is the log sequence number for the specified operation.
 - `block` is the block where the specified operation resides.
 - `offset` is the offset into the operation where the specified operation resides.
 - SCN – the SCN for the operation as logged in a redo log file.
 - THREAD – the thread that executed the operation.
 - USERNAME – the name of the user that executed the operation.
 - EXECUTION TIME – the date and time at which the operation was executed.
 - OBJECT ID – the database ID of the affected object.
 - OBJECT NAME – the name of the affected object.
 - OPERATION – the operation type.
 - REPLICATE – whether or not (YES or NO) the object affected by the operation is marked for replication by Replication Agent.
 - TRANSACTION ID – the ID of the transaction that the operation is a part of.
 - REDO SQL – the SQL that can be used to replay the operation.
 - UNDO SQL – the SQL that can be used to undo the operation.

- **SKIP COMMAND** – the Replication Agent command that causes the operation to be skipped by Replication Agent during replication.
- The `ra_helpop` command can display information for more than one operation. Operation information is returned for each operation that has the SCN specified by *locator* or *opid*.
- The `ra_helpop` command cannot operate properly unless the Oracle LogMiner script, `$ORACLE_HOME/rdbms/admin/dbmslm.sql`, has been installed at the primary database. If this script has not been installed, `ra_helpop` will return an error.
- After LogMiner is installed, create a public synonym so that you do not have to log in as the owner to execute LogMiner functions:

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

Note This is required if you are using Oracle 10g.

- The following privileges must be granted to `pds_username` for the `ra_helpop` command to function properly:
 - EXECUTE_CATALOG_ROLE
 - SELECT ON V_\$LOGMNR_CONTENTS
 - SELECT ON V_\$LOGMNR_LOGS
 - SELECT ANY TRANSACTION
- A single DML command may be represented in the Oracle redo log as a succession of two or more operations. Oracle LogMiner, however, will display only the SCN of the first operation. Consequently, `ra_helpop` may return no result for an operation that occurs in the middle of a succession of operations in the Oracle redo log. If `ra_helpop` returns no result for a specified *opid* or *locator* value, use the `ra_dumptran` command, specifying the transaction ID of the transaction to which the database operation belongs, and dump the entire transaction from the Oracle redo log. Then, search the dump for the operation that Oracle LogMiner did not find.

ra_helpuser

Description Returns information about primary database users from the RASD.

Note This command is available only for Oracle and Microsoft SQL Server.

Syntax ra_helpuser [*user* [, *version*]]

Parameters

user
The name or user ID of a user in the primary database.

version
The version number of the database user in the RASD.

Examples **Example 1**

```
ra_helpuser
```

This command returns information about all versions of all users in the RASD.

Example 2

```
ra_helpuser bob
```

This command returns information about the current version of the database name “bob” in the RASD.

Example 3

```
ra_helpuser bob,  
00000000000210a400003334000700003334000699940000d413c50000000000
```

This command returns information about version
00000000000210a400003334000700003334000699940000d413c50000000000
of the database user named “bob” in the RASD.

Usage

- The ra_helpuser command returns the following information about primary database users:

- User ID
- User name
- User status (Current, Archived, or Dropped)
- Primary database version (locator value)

The user ID and user name are the values returned by the primary database when the Replication Agent is initialized with the ra_init command.

- If ra_helpuser is invoked with no option, it returns information about all users in all versions of the primary database in the RASD.

- If `ra_helpuser` is invoked with the `user` option, it returns information about the current version of the specified user in the primary database in the RASD.
- If `ra_helpuser` is invoked with the `user` and `version` options, it returns information about the specified user in the specified version of the primary database in the RASD.
- The `ra_helpuser` command is valid when the Replication Agent instance is in the *Admin*, *Replicating*, or *Replication Down* state.
- No results are returned by this command if the RASD has not been initialized by the `ra_init` command.

See also

`ra_config`, `ra_help`, `ra_helpparticle`, `ra_helppdb`, `ra_helpdevice`, `ra_helpfield`, `ra_helplocator`

ra_locator

Description

Returns the current value of the LTM Locator maintained by the Replication Agent, requests an LTM Locator value from the primary Replication Server, or sets the value of the LTM Locator maintained by the Replication Agent to zero.

Syntax

`ra_locator [update|zero]`

Parameters

`update`

The optional keyword to request a new LTM Locator value from the primary Replication Server.

`zero`

The optional keyword to set the value of the LTM Locator stored in the Replication Agent transaction log to zero.

Examples

Example 1

```
ra_locator
```

This command returns the current value of the LTM Locator maintained by the Replication Agent, as shown:

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

Example 2

```
ra_locator update
```

This command requests a new LTM Locator value from the primary Replication Server.

Example 3

```
ra_locator zero
```

This command sets the value of the LTM Locator maintained by the Replication Agent to all zeros.

Usage

- When `ra_locator` is invoked with no option, it returns the current value of the LTM Locator maintained by the Replication Agent instance. For UDB, the Replication Agent stores the value of the LTM Locator in a table in the primary database; for Oracle and Microsoft SQL Server this value is stored in the RASD.

Note The value of the LTM Locator that is maintained by the Replication Agent is also known as the *origin queue ID*.

- When `ra_locator` is invoked with the `update` keyword, it requests a new LTM Locator value from the primary Replication Server, and the Replication Agent saves the value.

Note When the `ra_locator` command is invoked with the `update` keyword, the change takes effect only if the Replication Agent instance is in *Replicating* state.

- When `ra_locator` is invoked with the `zero` keyword, it sets the value of the LTM Locator maintained by the Replication Agent to zero.
- The LTM Locator contains information that the Replication Agent uses to determine where to start reading the transaction log.

Upon start-up or recovery from a connection failure, the Replication Agent automatically requests an LTM Locator value from the primary Replication Server.

- If the value of the LTM Locator returned from the primary Replication Server is zero, then Replication Agent uses the LTM Locator value stored in the transaction log system table.

- If the value of the LTM Locator in the transaction log system table is zero, then Replication Agent starts reading the transaction log from either the current beginning of the log, or from the end of the log for UDB.
- For more information about the format of the origin queue ID, see the chapter for your specific primary data server in the *Replication Agent Primary Database Guide*.
- If the Replication Agent transaction log does not exist, the `ra_locator` command returns an error message.
- The `ra_locator` command with the zero keyword is valid only when the Replication Agent instance is in the *Admin* or *Replication Down* state.
- Without the zero keyword, the `ra_locator` command is valid when the Replication Agent instance is in the *Admin*, *Replicating*, or *Replication Down* state.

See also `pdb_gen_id`, `pdb_truncate_xlog`

ra_maintid

Description Returns the login name of the primary database maintenance user.

Syntax `ra_maintid`

Usage

- Replication Server requires a maintenance user login name for each database connection. The maintenance user login name for a database connection is specified with the Replication Server create connection or alter connection command.

When the primary database maintenance user login name is changed in the Replication Server (using the alter connection command), Replication Server automatically sends the new maintenance user login name to the Replication Agent, if the Replication Agent is in *Replicating* state.

Each time the Replication Agent goes into *Replicating* state, it automatically retrieves the primary database maintenance user login name from the primary Replication Server, and caches it.

- When `ra_maintid` is invoked, it returns the login name of the primary database maintenance user that is cached, as follows:

```

maintenance user
-----

```

```
SYS
(1 row affected)
```

- If `ra_maintid` is invoked when the Replication Agent is in *Replicating* state, it always returns the correct maintenance user login name.

If `ra_maintid` is invoked when the Replication Agent is in *Admin* or *Replication Down* state, it may not return the correct maintenance user login name, because the maintenance user login name could have changed in the Replication Server after the last time the Replication Agent retrieved the value and stored it.

- The `filter_maint_userid` configuration parameter is provided to support bidirectional replication, wherein the primary database also acts as a replicate database that has transactions applied to it by a Replication Server.

If the value of the `filter_maint_userid` parameter is true, database operations applied by the maintenance user are *not* replicated from the primary database. When it reads the transaction log, the Replication Agent Log Reader component filters out data-changing operations applied by the maintenance user.

- The `ra_maintid` command is valid when the Replication Agent instance is in the *Admin*, *Replicating*, or *Replication Down* state.

See also

`ra_config`, `ra_statistics`

ra_marker

Description

Places a marker in the primary database transaction log.

Syntax

`ra_marker command_tag`

Parameters

`command_tag`

A varchar value that contains information used for subscription materialization. When used for Microsoft SQL Server data servers, this value is `varchar(8000)`. When used for Oracle data servers, this value is `varchar(4000)`.

Examples

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

This command places a marker object in the Primary Database transaction log that invokes the Replication Server `activate subscription` command.

Usage	<ul style="list-style-type: none"> When <code>ra_marker</code> is invoked, the Replication Agent executes a transaction in the Primary Database that is captured in the Primary Database transaction log. The replicated transaction is sent as a marker object to the primary Replication Server. The <code>ra_marker</code> command returns an error message if the Replication Agent transaction log does not exist. The <code>ra_marker</code> command is valid when the Replication Agent instance is in the <i>Admin</i>, <i>Replicating</i>, or <i>Replication Down</i> state. For more information about the Replication Server <code>rs_marker</code> system function, refer to the <i>Replication Server Administration Guide</i> and <i>Replication Server Reference Manual</i>.
See also	<code>ra_dump</code>

ra_migrate

Description	Performs any necessary migration task between releases of Replication Agent.
Syntax	<code>ra_migrate</code>
Parameters	<p><i>None</i></p> <p>There are no parameters.</p>
Usage	<ul style="list-style-type: none"> After upgrading to a new release of Replication Agent, you must first run this command to update to the latest version of the Replication Agent. The <code>ra_migrate</code> command is valid when the Replication Agent instance is in the <i>Admin</i> or <i>Replication Down</i> state. (Oracle only) The <code>ra_migrate</code> command will verify that the following privileges have been granted to <code>pds_username</code>: <ul style="list-style-type: none"> <code>EXECUTE_CATALOG_ROLE</code> <code>select on V_\$LOGMNR_CONTENTS</code> <code>select on V_\$LOGMNR_LOGS</code>

These privileges are necessary for the `ra_dumptran` and `ra_helpop` commands to function properly. These privileges are not required for replication, only for using the `ra_dumptran` and `ra_helpop` commands, which are used in debugging and troubleshooting. If these privileges have not been granted at the time `ra_migrate` is invoked, a warning message is returned and logged in the Replication Agent log file.

ra_set_autocorrection

Description	Enables or disables autocorrection for marked tables.
Syntax	<code>ra_set_autocorrection [all <i>tablename</i>] {, [enable disable]}</code>
Parameters	<p>all</p> <p>To enable autocorrection for all marked tables, follow the <code>ra_set_autocorrection</code> command with the <code>all</code> and <code>enable</code> keywords:</p> <pre>ra_set_autocorrection all, enable</pre> <p>To disable autocorrection for all marked tables, follow the <code>ra_set_autocorrection</code> command with the <code>all</code> and <code>disable</code> keywords:</p> <pre>ra_set_autocorrection all, disable</pre> <p><i>tablename</i></p> <p>To enable autocorrection for one marked table, follow the <code>ra_set_autocorrection</code> command with the <i>tablename</i> parameter and the <code>enable</code> keyword:</p> <pre>ra_set_autocorrection <i>tablename</i>, enable</pre> <p>To disable autocorrection for one marked table, follow the <code>ra_set_autocorrection</code> command with the <i>tablename</i> parameter and the <code>disable</code> keyword:</p> <pre>ra_set_autocorrection <i>tablename</i>, disable</pre> <p>To display autocorrection status for one marked table, follow the <code>ra_set_autocorrection</code> command with the <i>tablename</i> parameter alone:</p> <pre>ra_set_autocorrection <i>tablename</i></pre> <p>enable</p> <p>Use the <code>enable</code> keyword to enable autocorrection for one marked table or all marked tables.</p>

disable

Use the disable keyword to disable autocorrection for one marked table or all marked tables.

Examples

Example 1

```
ra_set_autocorrection mytable, enable
```

This command enables autocorrection for the marked table named mytable.

Example 2

```
ra_set_autocorrection all, enable
```

This command enables autocorrection for all marked tables.

Example 3

```
ra_set_autocorrection mytable, disable
```

This command disables autocorrection for the marked table named mytable.

Example 4

```
ra_set_autocorrection all, disable
```

This command disables autocorrection for all marked tables.

Example 5

```
ra_set_autocorrection mytable
```

This command displays autocorrection status for the marked table named mytable.

Usage

- This command is used to support Replication Server autocorrection functionality.
- You cannot set autocorrection for tables that have not been marked for replication. If a marked table for which autocorrection is enabled is subsequently unmarked, autocorrection is automatically disabled for the table.
- The `column_compression` and `l1_send_only_primary_keys` configuration parameters are disregarded when Replication Agent is replicating a marked table for which autocorrection has been enabled.
- When autocorrection is enabled, Replication Server converts each update or insert operation into a pair of operations: one delete operation followed by an insert.

- If your primary database is Oracle and table-level supplemental logging has not already been enabled, enabling autocorrection will enable supplemental logging for all columns of the specified table.

ra_set_login

Description Sets the Replication Agent administrator login and password.

Syntax `ra_set_login username, password`

Parameters *username*

The login name of the Replication Agent administrator.

password

The password of the Replication Agent administrator.

Examples

```
ra_set_login bob3, bug3wag
```

This command sets the Replication Agent administrator login to “bob3” and the password to “bug3wag.”

Usage

- The Replication Agent administrator login has permission to log in to the Replication Agent instance through the administration port.
- Only one Replication Agent administrator login name is valid at any time.
- Any change in the Replication Agent administrator login or password takes place immediately, and you must use the new login and password the next time you log in to the Replication Agent instance.
- The password specified for the administrator login is encrypted in the Replication Agent configuration file.
- The `ra_set_login` command is valid when the Replication Agent instance is in the *Admin*, *Replicating*, or *Replication Down* state.

See also

`ra_config`

ra_statistics

Description Returns performance-related statistics for Replication Agent components and the Java Virtual Machine (Java VM), or resets the statistics counters.

Note The statistics counters may vary by primary database.

Syntax `ra_statistics [component|reset]`

Parameters *component*
The optional keyword that identifies a Replication Agent component or the Java VM. Valid *component* keywords are:

LR – Log Reader
LTI – Log Transfer Interface
LTM – Log Transfer Manager
VM – Java Virtual Machine

reset
The optional keyword that resets the statistics counters.

Examples **Example 1**

```
ra_statistics
```

This command returns performance statistics for the Replication Agent instance and the Java VM.

Example 2

```
ra_statistics reset
```

This command resets the statistics counters for the Replication Agent instance.

Example 3

```
ra_statistics VM
```

This command returns statistics for the Java VM as described in Table 1-4.

Usage

- If you invoke `ra_statistics` with no option, it returns statistics for all Replication Agent components and the Java VM.
- If you invoke `ra_statistics` with a *component* option, the `ra_statistics` command returns statistics for the specified Replication Agent component or the Java VM.
- Table 1-4 lists the statistics returned for the Java VM.

Table 1-4: Java VM statistics

Component	Statistic	Description
VM	VM maximum memory	Maximum memory (in bytes) available to the Java VM
VM	VM total memory allocated	Total memory (in bytes) allocated to the Java VM at start-up
VM	VM free memory	Memory (in bytes) allocated but not used by the Java VM
VM	VM memory usage	Memory (in bytes) allocated and in use by the Java VM
VM	VM % max memory used	Percentage of the maximum memory available to the Java VM, currently in use by the Java VM

- Table 1-5 lists the statistics returned for the Log Transfer Manager component.

Table 1-5: Log Transfer Manager statistics

Component	Statistic	Description
LTM	Time statistics obtained	Day, date, and time ra_statistics was invoked and information returned
LTM	Time replication last started	Day, date, and time that <i>Replicating</i> state was entered
LTM	Time statistics last reset	Day, date, and time that statistics counters were reset
LTM	Items held in Global LRUcache	Number of object references in the internal Least Recently Used cache

- Table 1-6 lists the Log Reader statistics for UDB.

Table 1-6: Log Reader statistics for UDB

Component	Statistic	Description
LR	Number of transaction logs scanned	Number of operations read from log devices
LR	Average unprocessed operations per transaction log scan	Average number of unprocessed operations for each transaction log scan
LR	Average transaction log scan time	Average transaction log scan time for operations read from log devices
LR	Number of operations replicated	Number of operations that were successfully replicated
LR	Number of transactions replicated	Number of transactions that were successfully replicated
LR	Number of transaction log operations skipped (maint_user, unmarked tables)	Number of transaction log operations that were skipped
LR	Average wait time on empty transaction log	Average time that the transaction log was not in use
LR	Average PDB Service Time/Operations	Average service and operations time for each database
LR	Operation Queue Size	The queue size used for the operations
LR	Operation Data Hash Size	The data hash size for the operations
LR	Number of transactions truncated	Number of transactions that were truncated

- Table 1-7 lists the Log Reader statistics for Microsoft SQL Server.

Table 1-7: Log Reader statistics for Microsoft SQL Server

Component	Statistic	Description
LR	Total operations scanned	Number of operations read from log devices since last reset
LR	Total operations processed	Number of operations read from log devices and passed to LTI since last reset
LR	Total operations skipped	Number of operations read from log devices and not processed for any reason since last reset
LR	Total maintenance user operations filtered	Number of maintenance user operations read from log devices and skipped since last reset
LR	Avg operation processing time	Average Log Reader operation processing time (in milliseconds) since last reset
LR	Total transactions processed	Number of transactions read from log devices since last reset
LR	Total transactions skipped	Number of transactions read from log devices and not processed for any reason since last reset
LR	Total transactions opened	Number of begin transaction commands read from log devices since last reset
LR	Total transactions closed	Number of commit and rollback commands read from log devices since last reset
LR	Total transactions committed	Number of commit commands read from log devices since last reset
LR	Total transactions aborted (rolled back)	Number of rollback commands read from log devices since last reset
LR	Total system transactions skipped	Number of system transactions read from log devices and skipped since last reset
LR	Avg operations per transaction	Average number of operations in each transaction read from log devices since last reset
LR	Current scan buffer size	Current size (in bytes) of the Log Reader scan buffer
LR	Current operation queue size	Current size (in bytes) of the Log Reader input queue
LR	Current session cache size	Current size (in bytes) of the session cache
LR	Log reposition point locator	Locator value of reposition point in log device

Component	Statistic	Description
LR	Last processed operation locator	Locator value of most recently processed operation read from log devices
LR	Average transaction log operation wait time (ms)	Average time (in milliseconds) that Log Reader had to wait for each new operation to appear in the log since last reset
LR	Average sender operation processing time (ms)	Average time (in milliseconds) that Log Reader sender took to process each operation since last reset
LR	Average sender operation wait time (ms)	Average time (in milliseconds) that Log Reader sender had to wait to send each processed operation to the LTI input queue since last reset
LR	Average ChangeSet send time (ms)	Average time (in milliseconds) that Log Reader sender took to send each processed operation to the LTI input queue since last reset
LR	Total sender operations processed	Number of operations that Log Reader sender processed since last reset
LR	Marked objects cache size	Current marked objects cache size

- Table 1-8 lists the statistics returned for the Log Reader component for Oracle.

Table 1-8: Log Reader statistics for Oracle

Component	Statistic	Description
LR	Average RBA search time (ms)	The average record byte address (RBA) search time during log scanner positioning
LR	Total bytes read	The total number of bytes read from the primary database transaction log
LR	Total log records read	The total number of log records read from the primary database transaction log
LR	Average number of bytes read per second	The average number of bytes read from the primary database transaction log per second
LR	Average number of bytes per record	The average number of bytes per log record read
LR	Average time (ms) per log read	The average time per primary database transaction log read
LR	Total online log read time (ms)	The total time spent reading the primary database online transaction redo log
LR	Total archive log read time (ms)	The total time spent reading primary database transaction redo log archives
LR	Average time (ms) per online log device read	The average time per online log device read
LR	Average time (ms) per archive log device read	The average time per archive log device read
LR	Total log records queued	The total number of log records queued for processing
LR	Total log records filtered	The total number of log records filtered
LR	Log scan checkpoint set size	The current number of log records in the checkpoint set
LR	Average number of log records per checkpoint	The average number of log records for each checkpoint log record read
LR	Average number of seconds between log record checkpoints	The average number of seconds between reading log record checkpoints
LR	Total operations scanned	Number of operations read from log devices since last reset
LR	Total operations processed	Number of operations read from log devices and passed to LTI since last reset
LR	Total operations skipped	Number of operations read from log devices and not processed for any reason since last reset

Component	Statistic	Description
LR	Total maintenance user operations filtered	Number of maintenance user operations read from log devices and skipped since last reset
LR	Avg operation processing time	Average Log Reader operation processing time (in milliseconds) since last reset
LR	Total transactions processed	Number of transactions read from log devices since last reset
LR	Total transactions skipped	Number of transactions read from log devices and not processed for any reason since last reset
LR	Total transactions opened	Number of begin transaction commands read from log devices since last reset
LR	Total transactions closed	Number of commit and rollback commands read from log devices since last reset
LR	Total transactions committed	Number of commit commands read from log devices since last reset
LR	Total transactions aborted (rolled back)	Number of rollback commands read from log devices since last reset
LR	Total system transactions skipped	Number of system transactions read from log devices and skipped since last reset
LR	Avg ops per transaction	Average number of operations in each transaction read from log devices since last reset
LR	Current scan buffer size	Current size of the Log Reader scan buffer
LR	Current operation queue size	Current size of the Log Reader operation queue
LR	Current session cache size	Current size of the session cache
LR	Current Op Proc RASD marked object cache size	Current size of the operation processor marked object repository cache
LR	Total number of Op Proc RASD marked object cache hits	Total number of operation processor marked object repository cache hits
LR	Total number of Op Proc RASD marked object cache misses	Total number of operation processor marked object repository cache misses
LR	Log reposition point locator	Locator value of reposition point in log device

Component	Statistic	Description
LR	Last processed operation locator	Locator value of most recently processed operation read from log devices
LR	Average xlog operation wait time (ms)	Average time (in milliseconds) that Log Reader had to wait for each new operation to appear in the log since last reset
LR	Average sender operation processing time (ms)	Average time (in milliseconds) that Log Reader sender took to process each operation since last reset
LR	Average sender operation wait time (ms)	Average time (in milliseconds) that Log Reader sender had to wait to send each processed operation to the LTI input queue since last reset
LR	Average change set send time (ms)	Average time (in milliseconds) that Log Reader sender took to send each processed operation to the LTI input queue since last reset
LR	Number of sender operations processed	Number of operations that Log Reader sender processed since last reset
LR	Current marked objects cache size	Current marked objects cache size

- Table 1-9 lists the statistics returned when the primary database is Oracle RAC. These statistics exist in addition to the normal Log Reader statistics listed in Table 1-8.

Table 1-9: Additional statistics for Oracle RAC

Component	Statistic	Description
LR	Log scan reader current LSN	The current log sequence number of the log being read for each cluster instance
LR	Log scan reader end-of-log status	The current end of log status for each cluster log scanner
LR	Log scan reader last read time	The number of seconds since the last read for each cluster scanner
LR	Log scan record set distribution	Distribution of the log scan checkpoint set across all log scan threads
LR	Log scan reader last record SCN	The SCN of the last log record read by each cluster scanner
LR	Log scan reader checkpoints	The checkpoint SCN of the last checkpoint log record read by each cluster scanner
LR	Log scan checkpoint SCN	The current checkpoint SCN, based on all cluster scanners
LR	Log scan active checkpoint SCN	The active checkpoint SCN, based on all cluster scanner
LR	Total log records read per redo log thread	The distribution of total log records read across all log scan threads
LR	Log scan record set sizes	The current scan record set size for each cluster scanner
LR	Log scan checkpoint queue sizes	The current checkpoint queue size for each cluster scanner

- Table 1-10 lists the statistics returned for the Log Transfer Interface component.

Table 1-10: Log Transfer Interface statistics

Component	Statistic	Description
LTI	Number of LTL commands sent	Total number of LTL commands sent to Replication Server since last reset
LTI	Average LTL command size	Average size (in bytes) of each LTL command sent to Replication Server since last reset
LTI	Average LTL commands/sec	Average number of LTL commands sent per second to Replication Server since last reset
LTI	Total bytes sent	Number of bytes sent to Replication Server since last reset
LTI	Average Bytes/second during transmission	Average bytes per second sent over connection to Replication Server since last reset
LTI	Average LTL buffer cache time	Average time (in milliseconds) it takes between placing the LTL commands into the LTL buffer to the time it is actually sent to the Replication Server
LTI	Average Rep Server turnaround time	Average time (in milliseconds) it takes Replication Server to acknowledge each LTL command buffer sent since last reset
LTI	Average data arrival time	Average time (in milliseconds) LTI waits between receiving change sets from Log Reader since last reset
LTI	Average time to create distributes	Average time (in milliseconds) LTI takes to convert a change-set into LTL since last reset
LTI	Average LTL buffer size	Average size (in bytes) of each LTL buffer sent to Replication Server since last reset
LTI	Average LTM buffer utilization (%)	Average utilization (in percentage of LTL buffer size) of each LTL buffer sent to Replication Server since last reset
LTI	Average LTL commands/buffer	Average number of LTL commands per buffer sent to Replication Server since last reset
LTI	Encoded column name cache size	Current encoded column name cache size
LTI	Input queue size	Current number of change sets in the LTI input queue
LTI	Output queue size	Current number of distributes in the LTI output queue

Component	Statistic	Description
LTI	Last QID sent	Hex value of most recent origin queue ID sent to Replication Server
LTI	Last transaction id sent	Hex value of most recent transaction ID sent to Replication Server

- Statistics counters are reset automatically each time the Replication Agent instance goes into *Replicating* state.
- If you invoke `ra_statistics` with the `reset` keyword, Replication Agent immediately resets all of the statistics, except the following:
 - Time statistics obtained (LTM)
 - Time replication last started (LTM)
 - Time statistics last reset (LTM)
 - Last QID sent (LTI)
 - Last transaction ID sent (LTI)
 - All Java VM statistics

Note All Java VM statistics are refreshed each time you invoke `ra_statistics`. All queue, buffer, and cache size statistics are also refreshed and are not reset in the way averages or sums are reset.

- The `ra_statistics` command is valid when the Replication Agent instance is in the *Admin*, *Replicating*, or *Replication Down* state.

See also

`ra_status`

ra_status

Description

Returns the current state of the Replication Agent instance.

Syntax

`ra_status`

Usage

- When `ra_status` is invoked, it returns the current state of the Replication Agent instance, and a brief description of the current state, as follows:

```

State  Action
-----
ADMIN  Waiting for operator command

```

(1 row affected)

Note If the first word in the description is “Transitioning,” the Replication Agent instance is in transition between states. Some commands are not valid when the Replication Agent instance is in state transition.

- Replication Agent states are:
 - *Admin* – in this state, the Replication Agent instance is running, but no connections are up. You can change any configuration parameter when the Replication Agent instance is in *Admin* state.
 - *Replicating* – in this state, the Log Reader component is scanning the transaction log for operations to replicate from the primary database. If there are operations to be replicated, the Log Transfer Interface component is sending LTL commands to the Replication Server.
 - *Replicating (Resynchronization)* – in this state, Replication Agent has been restarted and is resynchronizing the primary and replicate databases.
 - *Replication Down* – in this state, replication has stopped due to an error. After the error has been resolved, Replication Agent may return to the *Replicating* state.

Note Replication Agent behavior in the *Replication Down* state is the same as behavior in the *Admin* state, the only difference between the two states being that the *Replication Down* state is reached through a Replication Agent error.

See the *Replication Agent Administration Guide* for more information about Replication Agent states.

- The `ra_status` command is valid when the Replication Agent instance is in the *Admin*, *Replicating*, or *Replication Down* state.

See also

`quiesce`, `ra_statistics`, `resume`, `shutdown`, `suspend`

ra_truncatearticles

Description	Truncates unused articles in the RASD.
	<hr/> <p>Note This command is available only for Oracle and Microsoft SQL Server.</p> <hr/>
Syntax	<code>ra_truncatearticles locator</code>
Parameters	<p><i>locator</i></p> <p>The log locator value (LTM Locator) that identifies the cutoff point for truncating older versions of articles from the system data repository.</p>
Usage	<ul style="list-style-type: none"> When <code>ra_truncatearticles</code> is invoked, it truncates all non-current versions of all primary database articles in the system data repository older than the version identified by the <i>locator</i> value. <p>If the current (most recent) version of an article is older than the version identified by the <i>locator</i> value, it is not truncated.</p> <ul style="list-style-type: none"> Most common DDL commands and stored procedures executed in the primary database (such as <code>alter table</code>) are recorded in the transaction log, and replicated to the standby database. When it processes those DDL transactions for replication, Replication Agent updates its RASD automatically, creating a new version of the affected primary database articles. <p>Use <code>ra_truncatearticles</code> as part of a periodic maintenance procedure to prevent the RASD from growing indefinitely. See the <i>Replication Agent Administration Guide</i> for more information.</p> <hr/> <p>Note Be sure to back up the RASD using <code>rasd_backup</code> before you truncate it.</p> <hr/> <ul style="list-style-type: none"> The <code>ra_truncatearticles</code> command is valid when the Replication Agent instance is in the <i>Admin</i>, <i>Replicating</i>, or <i>Replication Down</i> state.
See also	<code>ra_truncateusers</code>

ra_truncateddlfilters

Description Truncates old lists of DDL commands that are filtered in the RASD.

Note This command is available only for Oracle.

Syntax ra_truncateddlfilters *locator*

Parameters *locator*

The log locator value (LTM Locator) that identifies the cutoff point for truncating older lists of DDL commands that are filtered from the system data repository.

Usage

- When ra_truncateddlfilters is invoked, it truncates all lists of filtered DDL commands in the system data repository that are older than the list version identified by the *locator* value.

ra_truncateusers

Description Truncates older versions of primary database users in the system data repository in the RASD.

Note This command is available only for Oracle and Microsoft SQL Server.

Syntax ra_truncateusers *locator*

Parameters *locator*

The log locator value (LTM Locator) that identifies the cutoff point for truncating older versions of database users from the system data repository.

Usage

- When ra_truncateusers is invoked, it truncates all non-current versions of all primary database users in the system data repository older than the version identified by the *locator* value.

If the current (most recent) version of a user is older than the version identified by the *locator* value, it is not truncated.

- The ra_truncateusers command is valid when the Replication Agent instance is in the *Admin*, *Replicating*, or *Replication Down* state.

See also ra_truncatearticles

ra_updatedevices

Description Updates information about primary database log devices in the RASD.

Note This command is available only for Oracle and Microsoft SQL Server.

Syntax ra_updatedevices

Usage

- For Oracle, when ASM manages the redo log files and the disk group is changed by either adding or dropping disks, you must invoke the `ra_updatedevices` command to be sure the log device repository is updated with correct ASM storage information.
- When `ra_updatedevices` is invoked, Replication Agent does the following:
 - Refreshes the archive log information
 - Deletes all of the data in its log device repository

Note If the device location is set, it is not overwritten.

- Queries the primary database for information about all of its log devices
- Re-populates the log device repository in the RASD with current information about primary database log devices returned by the primary database
- If any log device associated with the primary database is added, dropped, extended, or moved at the primary data server, you must:
 - Stop replication (using `quiesce` or `suspend`) to put the Replication Agent instance in *Admin* state
 - Invoke `ra_updatedevices` to update the log device repository in the RASD

See the *Replication Agent Administration Guide* for more information.

Note The primary database need not be quiesced when you update the log device repository.

- If the primary data server writes to a new (or altered) log device before you update the log device repository, the Replication Agent instance will stop replication processing and go to *Replication Down* state.

Sybase recommends that you coordinate all log device changes at the primary database with updating the Replication Agent log device repository.

- Because Replication Agent re-creates the entire log device repository when you invoke `ra_updateddevices`, any log device path that you modified previously (using `ra_devicepath`) is overwritten with the current log device information from the primary database.

For example:

```
ID=1 serverpath=/dev1 mirror=/dev1a
```

becomes the following when you change the `server` path to “dev44”:

```
ID=1 serverpath=/dev44 mirror=/dev1a
```

Note If you need to alter the “default” path for a log device (that is, the log device path returned by the primary database), you must use the `ra_devicepath` command *after* you invoke `ra_updateddevices`.

- For each log device recorded in the RASD, you can set or change the disk device path with the `ra_devicepath` command.

If you do not specify a disk device path (using `ra_devicepath`), the value recorded for the disk device path is `DEFAULT`, and Replication Agent uses the value recorded for the server device path to find the log device.

- The `ra_updateddevices` command is valid only when the Replication Agent instance is in the *Admin* or *Replication Down* state.
- (*For Oracle*) The Replication Agent uses the disk map file, to create mirror log devices, when log devices are created during transaction log initialization and when devices are updated using the `ra_updateddevices` command. When Replication Agent is in the *Replicating* state, it reads data from the mirrored disks specified in the map file.
- The `ra_updateddevices` command updates the ASM disk map file. When executed the ASM disk map file is updated as follows:
 - ASM is queried to see if the disk groups required to read any redo logs have changed. If new disks have been added to any of the ASM disk groups, a default mirror entry is added in the ASM disk map file for the new disk.
 - The ASM disk group specified by the archive log path parameter is checked for new disks as well as the ASM disk group for each online redo log.

- The ASM disk map file is updated before the log devices are updated to ensure any ASM disk path changes are included in updated log devices.
- If new disk entries are added to the ASM disk map file, the log devices will not be updated in the repository. A message is returned to the user, indicating that new entries are in the file that may need to be changed before devices are updated. The next time the `ra_updatedevices` command is executed, the log devices will be updated.
- The Replication Agent command `ra_helpdevice` provides device information for the log device status where the device is physically located and if it is being mirrored to another device. The physical information is a simple path to a file or raw device. ASM uses disk groups with potentially many disks, the physical information for ASM devices is provided to show all disks required for the device. There will be one row output for each disk in the group where the device is stored.

See also `ra_devicepath`, `ra_helpdevice`

ra_version

Description Returns the version of the Replication Agent instance, the host operating system version, and the JRE version.

Syntax `ra_version`

Usage When `ra_version` is invoked, it returns the Replication Agent version string in a row, as follows:

```
Sybase Replication Agent for Unix &
Windows/15.5.0.5602/P/generic/JDK
5.0/main/5602/VM: Sun Microsystems Inc.
1.5.0_13/OPT/Fri Jul 09 03:42:03 MST 2010
```

See also `pdb_version`, `ra_status`, `ra_version_all`

ra_version_all

Description Returns the name, type, and version of the Replication Agent instance, and version information for the primary data server, primary Replication Server, and communications drivers.

Syntax ra_version_all

Note (For Oracle only) When the Replication Agent is configured to connect to ASM (an `asm_tns_connection` is configured with a non-null value that is not the default value), `ra_version_all` includes an additional line of output that describes the version of ASM being connected to. When `asm_tns_connection` is not configured, no ASM entry is listed in `ra_version_all` output.

Usage

- When `ra_version_all` is invoked, it returns the following information:

Component	Version

Instance:	mro2 - Oracle
RepAgent:	Sybase Replication Agent for Unix & Windows/15.5.0.5700/P/generic/JDK 5.0/main/5700/VM: Sun Microsystems Inc. 1.5.0_13/OPT/Fri Jul 09 03:03:03 MST 2010
JRE:	Sun Microsystems Inc. Java(TM) 2 Runtime Environment, Standard Edition/1.5.0_13-b05/SunOS 5.8/sparc/32
RASD:	Sybase SQL Anywhere/11.0.1/WindowsXP
Primary Data Server:	Oracle Oracle9i Enterprise Edition Release 10.2.0.1.0 - Production With the Partitioning, OLAP and ORACLE Data Mining options JServer Release 10.2.0.1.0 - Production
PDS JDBC Driver:	Oracle JDBC driver 10.2.0.1.0
RepServer:	Replication Server/15.5 ESD#5/NT (IX86)/Windows 2000 /1/OPT/Fri Jul 09 14:01:20 2010
RSSD:	Sybase SQL Anywhere/11.0.1/WindowsXP
Sybase JDBC Driver:	jConnect(TM) for JDBC(TM)/6.05(Build 26169)/P/EBF15181/JDK14/Wed Oct 24 1:06:46 2007

See also pdb_version, ra_status, ra_version

rasd_backup

Description	Backs up the Replication Agent System Database (RASD).
	<hr/> Note This command is available only for Oracle and Microsoft SQL Server. <hr/>
Syntax	rasd_backup
Usage	<ul style="list-style-type: none"> When rasd_backup is invoked, it starts the database backup process for the RASD. <hr/> <p>Note Sybase recommends that you always back up the RASD before you truncate using ra_truncatearticles or ra_truncateusers.</p> <hr/> <ul style="list-style-type: none"> Replication Agent places RASD backup files in the directory identified by the rasd_backup_dir configuration parameter. <p>When you create a Replication Agent instance, a RASD backup directory is created automatically as part of the instance directory structure. The default value of the rasd_backup_dir parameter points to that directory.</p> <ul style="list-style-type: none"> The rasd_backup command is valid when the Replication Agent instance is in the <i>Admin</i>, <i>Replicating</i>, or <i>Replication Down</i> state.
See also	rasd_restore, ra_truncatearticles, ra_truncateusers

rasd_helpbackup

Description	Displays a list of RASD backups.
	<hr/> Note This command is available only for Oracle and Microsoft SQL Server. <hr/>
Syntax	rasd_helpbackup
Usage	<ul style="list-style-type: none"> When rasd_helpbackup is invoked, it displays a list of RASD backups stored in the directory, identified by the rasd_backup_dir configuration parameter. The backups are named using the date and time the backup was created.
See also	rasd_restore, rasd_backup, rasd_removebackup

rasd_removebackup

Description Removes RASD backups.

Note This command is available only for Oracle and Microsoft SQL Server.

Syntax `rasd_removebackup [backup_name | all]`

Parameters `backup_name`

The name of the backup that you are removing.

`all`

A keyword that allows you to remove all RASD backups.

Examples

Example 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)
```

Example 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)
```

Usage	<ul style="list-style-type: none"> • When <code>rasd_removebackup</code> is invoked, it removes a RASD backup that is stored in the directory identified by the <code>rasd_backup_dir</code> configuration parameter. • When <code>rasd_removebackup</code> is invoked with the <code>all</code> keyword, all RASD backups that are stored in the directory identified by the <code>rasd_backup_dir</code> configuration parameter are removed. • When <code>rasd_backup</code> is invoked, the names of all backups removed appear. • The backups are named using the date and time the backup was created.
See also	<code>rasd_restore</code> , <code>rasd_backup</code> , <code>rasd_helpbackup</code>

rasd_restore

Description	Allows you to restore the RASD.
	<hr/> <p>Note This command is available only for Oracle and Microsoft SQL Server.</p> <hr/>
Syntax	<code>rasd_restore [backup_name]</code>
	<hr/> <p>Note After executing <code>rasd_restore</code>, Replication Agent automatically shuts down if <code>rasd_restore</code> is successful.</p> <hr/>
Parameters	<p><code>backup_name</code></p> <p>The name of the backup that you are restoring from. If you omit the backup name, the most recent backup is restored.</p>
Examples	<p>Example 1</p> <pre>rasd_restore</pre> <p>This command with no parameters restores the RASD from the most recent backup.</p> <p>Example 2</p> <pre>rasd_restore 2008-07-24_15.41.10</pre> <p>This command restores the RASD from the 2008-07-24_15.41.10 backup.</p>
Usage	<ul style="list-style-type: none"> • When <code>rasd_restore</code> is invoked, it starts the restore process for the RASD.

- When no parameters are used, Replication Agent looks for the most recent RASD backup in the directory identified by the `rasd_backup_dir` configuration parameter.
- If a backup name is provided as a parameter, Replication Agent restores from the specified backup in the directory identified by the `rasd_backup_dir` configuration parameter.
- When you create a Replication Agent instance, an RASD backup directory is automatically created as part of the instance directory structure. The default value of the `rasd_backup_dir` parameter points to that directory.
- If you invoke `rasd_restore` when the Replication Agent instance is in *Replicating* state, it returns an error.
- The `rasd_restore` command is valid only when the Replication Agent instance is in the *Admin* or *Replication Down* state.

See also `rasd_backup`, `rasd_helpbackup`, `rasd_removebackup`

rasd_trunc_schedule

Description Manages a truncation schedule. `rasd_trunc_schedule` returns a list of the repository truncation schedule, and can also add and remove a specific schedule.

Note This command is available only for Oracle and Microsoft SQL Server.

Syntax `rasd_trunc_schedule [add | remove | clear | force] schedule`

Parameters `schedule`
The day and time string in the form of restricted UNIX cron style that indicates the time automatic repository truncation is to be performed.

The following is a valid schedule string in UNIX cron style format:

[mm] [HH] [DOM] [MON] [DOW]

where:

- mm – is the minutes past the hour.
- HH – is the hour in 24-hour notation.
- DOM – represents the days of the month, 2-digit number between 1 and 31, which represents the day of the month.
- MON – represents the month of the year, abbreviated in 3-character format, such as “Jan”, “Feb”, and so on, or a 2-digit number between 1 and 12, which represents the month in a year from January to December.
- DOW – represents the day of the week, abbreviated in 3-character format, such as “Sun”, “Sat”, and so on, or a 2-digit number between 1 and 7, which represents the day in a week from Sunday to Saturday.
- Use an asterisk to match any valid value in a specific schedule field, ([mm],[HH],[DOM],[MON],[DOW]):

- For example, “1720***” represents a daily schedule at 8:17 p.m.

- When both the DOW and DOM are specified, the schedule represents two days that match either DOW or DOM.

For example, “*1216*Mon” represents 12:00 a.m. every Monday or 12:00 a.m. the 16th of every month.

- Multiple entries can be provided using a semicolon.

For example, “*1216*Mon” or “1720***;*1216*Mon”.

- Do not leave spaces between fields; otherwise, the schedule is rejected as an invalid schedule format.

For example, “* 12 16 *Mon” is as an invalid schedule.

- Use the dash “-” operator to specify a range of values. For example, “1-6” represents the sequence “1,2,3,4,5,6.”
- Use the slash “/” operator to skip a given number of values. For example, “*/3” in the hour time field represents the sequence “0,3,6,9,12,15,18,21.”

clear

To remove all repository truncation schedules, enter:

```
rasd_trunc_schedule clear
```

When the repository truncation schedule list is not set or empty, repository truncation by schedule is disabled.

force

To perform an immediate repository truncation manually, regardless of the automatic truncation time schedule, enter:

```
rasd_trunc_schedule force
```

Examples

Example 1

```
rasd_trunc_schedule
```

This command returns a list of all repository truncation schedule times when repository truncation occurs.

Example 2

```
rasd_trunc_schedule add, 1720***
```

This command adds daily repository truncation schedule at 8:17 PM to the schedule list.

Example 3

```
rasd_trunc_schedule remove, 1720***
```

This command removes the daily repository truncation schedule at 8:17 PM from the schedule list.

Example 4

```
rasd_trunc_schedule clear
```

This command clears all repository truncation schedules that have been set.

Example 5

```
rasd_trunc_schedule force
```

This command truncates the repository immediately, regardless of the existence of any truncation schedule.

Usage

- When `rasd_trunc_schedule` is invoked, its function is determined by the keywords and options you specify.
- When you specify multiple keywords and options, separate each must using a comma. A blank space before or after a comma is optional.

For example:

```
rasd_trunc_schedule add, 1720***
```


See also `ra_truncatearticles`, `ra_truncateusers`

resume

Description	Starts replication processing in the Replication Agent instance.
Syntax	<code>resume [[resync[, init]] [purge]]</code>
Parameters	<p><code>resync</code> (Oracle only) Replication Agent sends a resync database marker to Replication Server.</p> <p><code>init</code> (Oracle only) This keyword is used only with the <code>resync</code> keyword. Replication Agent sends both a resync database marker and an initialization command to Replication Server.</p> <p><code>purge</code> (Oracle and Microsoft SQL Server only) Replication Agent sends a purge command to Replication Server when replication resumes.</p>
Usage	<ul style="list-style-type: none"> When <code>resume</code> is invoked, the Replication Agent instance attempts to go to <i>Replicating</i> state and start replication operations, as follows: <ul style="list-style-type: none"> Replication Agent attempts to open network connections to the primary database, primary Replication Server, and RSSD. If it fails to establish a connection, the Replication Agent logs a warning message in its system log, and it attempts to retry the connection, based on its configuration parameters for the connection. If the Replication Agent cannot establish a connection to the primary database after exhausting its configured retry attempts, it aborts all subsequent resume processing, returns to <i>Replication Down</i> state, and logs the error. Replication Agent requests the current LTM Locator value from the primary Replication Server, and it stores the value in the Replication Agent transaction log. The Log Reader component begins scanning the transaction log, looking for operations to be replicated. Log Reader begins scanning the log at the point identified by the LTM Locator value.

- When it finds transactions to replicate, Log Reader passes them (as change-set data) to the input queue of the Log Transfer Interface component.
- The Log Transfer Interface component reads the change-set data from its input queue, generates LTL commands, and places the LTL commands in its output queue for transmission to the Replication Server.
- When `resume resync` is invoked, the Replication Agent instance attempts to start in the *Replicating (Resynchronization)* state.
 - Replication Agent sends a resync database marker to Replication Server, which processes this resync database marker and awaits a dump database marker from Replication Agent.
 - When `ra_init, force` is invoked before `resume resync`, Replication Agent sends instructions for Replication Server to purge all open transactions from the inbound queue and reset duplicate detection before receiving any new inbound transactions. Replication Server then awaits a dump database marker from Replication Agent.
 - When `resume resync, init` is invoked, Replication Agent sends a resync database marker and an initialization command instructing Replication Server to purge all open transactions from the inbound queue, reset duplicate detection, and suspend the outbound DSI. Use this option when you want to reload the primary database from the same dump as the replicate database.

For more information about configuring database resynchronization, see the *Replication Server Administration Guide*.

- When `resume purge` is invoked, the Replication Agent instance sends a purge command to Replication Server when replication resumes. Replication Server will then purge data from the inbound queue for the connection to which this Replication Agent instance is connected. The purge keyword should be used only after downgrading to an earlier version of Replication Agent or when otherwise recommended by Sybase Technical Support.
- If any start-up operation fails, the Replication Agent instance returns to *Replication Down* state, and it logs the error.
- If the `resume` command is successful, the Replication Agent instance goes to *Replicating* state. To determine the current state of the Replication Agent, use the `ra_status` command.

- The resume command returns an error under any of the following conditions:
 - The Replication Agent instance is already in *Replicating* state.
 - The system data repository in the RASD does not exist or is not initialized (Oracle and Microsoft SQL Server).
 - The Replication Agent connection configuration parameters are not set correctly, or it fails otherwise to connect with the primary database or the primary Replication Server.
 - The database connection for the primary database is not defined correctly in the primary Replication Server.
- If the resume command is successful, the Replication Agent instance goes into *Replicating* state.
- The resume command is valid only when the Replication Agent instance is in the *Admin* or *Replication Down* state.

See also `quiesce`, `ra_status`, `shutdown`, `suspend`

rs_create_repdef

Description Creates a replication definition at Replication Server for a marked table and procedure, or for all marked tables and procedures.

Note This command is available only for Oracle and Microsoft SQL Server.

Replication Agent is pre-configured to match replication definition datatypes available in Replication Server 15.0 and later. If replication definitions are to be generated against an earlier version of Replication Server, this configuration needs to be changed. Contact Sybase Technical Support for assistance in making this adjustment.

Syntax `rs_create_repdef {all | name}`

Parameters *all*
 A replication definition is created for all tables and procedures that are marked for replication.

name

A replication definition is created for the table or procedure specified by *name*.

Note rs_create_repdef always assumes that a database replication definition exists for the primary database.

Usage

- When a table is marked for replication and the owner mode is set to on, the replication definition created by rs_create_repdef includes the owner name as part of the table name for a table replication definition in the “with primary table named” clause.
- This command always assumes that a database replication definition exists for the primary database. All replication definitions created by this command include the send standby clause, which means the replication definition will only be used by Replication Server if there is already a database level replication definition. The replication definition created by rs_create_repdef can not be individually subscribed to. If you do not wish to have a database level replication definition, you must use a different tool, or create replication definitions manually, and not use rs_create_repdef.
- Replication definitions created by rs_create_repdef always define the datatypes using available user defined datatypes that are installed in Replication Server. This means that customers using rs_create_repdef should not set Replication Agent configuration parameter pdb_convert_datetime to true, as doing so converts date and timestamp datatypes to Sybase format, instead of UDD format.
- Using the Replication Agent configuration parameter pdb_auto_create_repdefs has the same result as executing rs_create_repdef.
- When rs_create_repdef is invoked and the parameter “all” or “ALL” is entered, a replication definition is created for all tables or procedures that are marked for replication.
- When rs_create_repdef is invoked and the name of a table or procedure that is marked for replication is entered, a replication definition is created only for that table or procedure.

- For each table or procedure for which a replication definition create is attempted, a result set is returned. The result set contains the replication definition name and status of the create. If the replication definition was created, the status will be “created.” If an error occurred, an error message from Replication Server will be returned.
- The character case of the object names in the replication definition will be set according to the `ttl_character_case` setting.
- The following applies to replication definition table and procedure names:
 - All non-alphanumeric characters and spaces are removed and are not part of the table or procedure name.
 - Underscores are kept as part of the name even though they are non-alphanumeric characters.
 - Periods are replaced with underscores.
- Replication definition names for tables always begin with the prefix “*ra\$*,” followed by a unique alphanumeric identifier (maximum of 8 characters), and ending with a table or object name. For example, for a replicate name of “My Table,” the resulting replication definition name is “*ra\$0x7952_mytable*.” For an especially long replicate name of “mytable89012345678901234567890” (30 characters), the resulting replication definition name is “*ra\$0x7952_mytable8901234567890*” (30 or 255 characters maximum, depending on whether or not the `pdb_support_large_identifier` configuration parameter is set).
- For date columns, the `rs_create_repdef` command creates a replication definition with a column datatype defined that assumes the Replication Agent `pdb_convert_datetime` configuration parameter is set to false. If `pdb_convert_datetime` is set to true, the format of the date value does not match the format expected by Replication Server. To avoid this problem, change the `pdb_convert_datetime` configuration parameter to false, or manually create the replication definitions (without using the `rs_create_repdef` command).

See also

`rs_drop_repdef`

rs_drop_repdef

Description	A replication definition at the configured Replication Server for a table and procedure is dropped.
	<hr/> Note This command is available only for Oracle and Microsoft SQL Server. <hr/>
Syntax	rs_drop_repdef <i>name</i>
Parameters	<i>name</i> A replication definition is dropped for that table or procedure.
Usage	<ul style="list-style-type: none">• When rs_drop_repdef is invoked, a replication definition for that table is dropped at the Replication Server.• When rs_drop_repdef is invoked and the name of a table or procedure that is marked for replication is entered, a replication definition is created for that table or procedure.• For each table or procedure for which a replication definition is dropped, a result set is returned. The result set contains the table name and status of the create. If the replication definition was created, the status will be “dropped.” If an error occurred, an error message from Replication Server will be returned.• The character case of the object names in the replication definition will be set according to the <code>l1_character_case</code> setting.• The following applies to replication definition table and procedure names:<ul style="list-style-type: none">• All non-alphanumeric characters and spaces are removed and are not part of the table or procedure name.• Underscores are kept as part of the name even though they are non-alphanumeric characters.• Periods are replaced with underscores.• Replication definition names for tables always begin with the prefix “<i>ra\$</i>,” followed by a unique alphanumeric identifier (maximum of 8 characters), and ending with a table or object name. For example, for a replicate name of “My Table,” the resulting replication definition name is “<i>ra\$0x7952_mytable</i>.” For an especially long replicate name of “mytable89012345678901234567890” (30 characters), the resulting replication definition name is “<i>ra\$0x7952_mytable8901234567890</i>” (30 or 255 characters maximum, depending on whether or not the <code>pdb_support_large_identifier</code> configuration parameter is set).

See also `rs_create_repdef`

rs_ticket

Description Supports Replication Server `rs_ticket` processing by placing an `rs_ticket` marker in the primary database transaction log. This command was created in support of the Replication Server `rs_ticket` feature.

Syntax `rs_ticket H1 [, H2[, H3 [, H4]]]`

Parameters *H1, H2, H3*
Each parameter contains from 1-10 characters. It is free form and is to be used as an identifier.

H4

It contains from 1-50 characters. It is free form and is also to be used as an identifier.

Examples **Example 1**

The following executes `rs_ticket` and monitors the processing time for the record identified by the four parameters (only one parameter is required):

```
rs_ticket test1, 1221, appxyz.monitoring_system
```

Note The parameters are optional, and can be used to identify or differentiate executions of `rs_ticket`.

It can be used independently or grouped with additional executions to allow processing times to be compared.

In this example, the following information will be sent to Replication Server.

```
rs_ticket 'V=1;H1=test1;H2=1221;H3=appxyz;
H4=monitoring_system;PDB(name)=hh:mm:ss.ddd'
```

where “name” is the name of the primary database.

When this command reaches the replicate database, Replication Server will add additional time values for the EXEC, DIST and DSI components of Replication Server. The final result seen by the replicate database will look similar to:

```
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'
```

You can use the information provided to monitor replication latency and performance. By using different or descriptive H1-H4 parameters, users can more easily identify which `rs_ticket` data matches the activity or timing of the command when entered at the primary database.

Example 2

To measure performance of a batch of work, you can surround the work with `rs_ticket` executions, similar to the following sequence:

(Execute in Replication Agent)

```
rs_ticket start
```

(Execute in primary data server)

```
execute replication benchmarks
```

(Execute in Replication Agent)

```
rs_ticket stop
```

Usage

- The Replication Server EXEC, DIST, and DSI modules parse and process `rs_ticket` subcommands.
- There are no subscriptions for `rs_ticket`. DIST does not send `rs_ticket` to DSI unless there is at least one subscription from the replicate site.

For more information regarding the `rs_ticket` usage in Replication Server, refer to the Replication Server documentation.

server_xlog

Description

Creates or removes transaction log base objects in the Microsoft SQL Server system resource database. This command is used for first-time initialization of Replication Agent for Microsoft SQL Server and should be run before `pdb_xlog` init. This command is used for first-time initialization of Replication Agent for Microsoft SQL Server and should be run before `pdb_xlog` init.

Note This command is only for use with Microsoft SQL Server.

Syntax

```
server_xlog [init | remove] [, force]
```


Parameters	<p>init The keyword for creating transaction log base objects in the Microsoft SQL Server system resource database.</p> <p>remove The keyword for removing transaction log base objects from the Microsoft SQL Server system resource database.</p> <p>force A keyword that refers to the init or remove operation.</p>
Examples	<p>Example 1</p> <pre>server_xlog init</pre> <p>This command creates transaction log base objects in the Microsoft SQL Server system resource database.</p> <p>Example 2</p> <pre>server_xlog remove, force</pre> <p>This command removes transaction log base objects from the Microsoft SQL Server system resource database.</p>
Usage	<ul style="list-style-type: none"> If you attempt to execute <code>pdb_xlog init</code> for Microsoft SQL Server before the transaction log has been initialized, <code>pdb_xlog init</code> stops and prompts you to go back and first execute <code>server_xlog init</code>.
See also	<code>pdb_xlog</code>

shutdown

Description	Shuts down the Replication Agent instance, terminating its process.
Syntax	<code>shutdown [immediate]</code>
Parameters	<p>immediate The optional keyword that shuts down the Replication Agent instance immediately.</p>
Usage	<ul style="list-style-type: none"> When <code>shutdown</code> is invoked with no option, the Replication Agent starts a normal (graceful) shutdown. <p>In a normal shutdown, the Replication Agent first quiesces, and then the process terminates. See <code>quiesce</code> on page 80 for more information about quiescing the Replication Agent.</p>

- When shutdown is invoked with the immediate keyword, the Replication Agent starts an immediate shutdown.

In an immediate shutdown, the Replication Agent:

- Stops all of its replication processing, without regard to transactions in process or in transit
- Drops all of its connections
- Terminates the application process
- The shutdown command with the immediate keyword is valid at any time, when the Replication Agent instance is in any state, including transition between states.
- The shutdown command with no keyword (normal shutdown) is valid when the Replication Agent instance is in the *Admin*, *Replicating*, or *Replication Down* state, but not in state transition.

See also `quiesce`, `ra_status`, `resume`, `suspend`

suspend

Description Stops all current replication processing and puts the Replication Agent instance into *Admin* state.

Syntax `suspend`

Usage

- When `suspend` is invoked, it stops all current replication processing in the Replication Agent instance.
 - The Log Reader component stops scanning the transaction log immediately, and the Log Transfer Interface component stops sending LTL to the Replication Server immediately.
 - Any data in the Replication Agent internal queues (input and output queues of the Log Reader and Log Transfer Interface components) is removed without further processing.
 - The Replication Agent instance immediately releases all of its connections to the primary database, and drops its connection to the primary Replication Server (and RSSD, if connected).

- The Replication Agent instance goes from *Replicating* state to *Admin* state.

Note The action of the `quiesce` command is similar to that of the `suspend` command, except that `quiesce` allows pending transactions in the Replication Agent internal queues to be processed first, before putting the Replication Agent instance in *Admin* state.

- If the Replication Agent instance is in *Admin* state, the `suspend` command returns an error.
- The `suspend` command is valid only when the Replication Agent instance is in *Replicating* state.

See also `quiesce`, `ra_status`, `resume`, `shutdown`

test_connection

Description Tests Replication Agent connection configurations and network connectivity.

Syntax `test_connection [conn_name]`

Note (For Oracle only) When the Replication Agent is configured to connect to ASM (an `asm_tns_connection` is configured with a non-null value that is not the default value), `test_connection` includes an additional line of output that describes the version of ASM being connected to. When `asm_tns_connection` is not configured, no ASM entry is listed in `test_connection` output.

Parameters

conn_name

The keyword for a Replication Agent connection to be tested. Valid keywords are:

- PDS – primary data server
- RS – primary Replication Server (and RSSD, if so configured)

Note If the value of the `use_rssd` configuration parameter is true, the `test_connection` command tests Replication Agent connectivity to the RSSD when it tests connectivity to the Replication Server. If the value of the `use_rssd` configuration parameter is false, the `test_connection` command does *not* test Replication Agent connectivity to the RSSD.

Examples

Example 1

```
test_connection
```

This command tests all Replication Agent connections, including the primary data server connection, the primary Replication Server connection, and the RSSD connection (if so configured).

Example 2

```
test_connection PDS
```

This command tests only the Replication Agent connection for the primary data server.

Usage

- When `test_connection` is invoked with no option, Replication Agent tests all of its connections by attempting to log in to the corresponding server for each connection, using the connection parameters stored in its configuration file.
- When `test_connection` is invoked with either the RS or PDS keyword, Replication Agent tests the specified connection.
- The `test_connection` command verifies both network connectivity and the following Replication Agent connection configuration parameters for the primary database:
 - connection type (connectivity driver and protocol) – `pds_connection_type`
 - database name – `pds_database_name`
 - data server name – `pds_server_name`
 - Data source name (ODBC drivers only) – `pds_datasource_name`

- host machine name – pds_host_name
- port number – pds_port_number
- user login access – pds_password and pds_username

Note The test_connection command does *not* validate Replication Agent user login permissions in the primary database. It verifies only that the user login and password specified in the pds_username and pds_password parameters can log in to the primary data server.

- The test_connection command verifies both network connectivity and the following Replication Agent connection configuration parameters for the primary Replication Server (and RSSD, if so configured):
 - Database name – rssid_database_name (RSSD only)
 - Replication Server data source (as specified in the Replication Server primary database connection) – rs_source_db and rs_source_ds (Replication Server only)
 - Host machine name – rs_host_name (and rssid_host_name)
 - Network packet size – rs_packet_size (Replication Server only)
 - Port number – rs_port_number (and rssid_port_number)
 - User login access – rs_password, rs_username (and rssid_password and rssid_username)

Note The test_connection command verifies that the Replication Agent user login (specified in the rs_username and rs_password parameters) has connect source permission in the primary Replication Server.

- The test_connection command returns the connection type and its status, as follows:

```
Type Connection
----
PDS succeeded
RS    succeeded

(2 rows affected)
```

If the connection status is failed, it indicates one of the following:

- The Replication Agent connection configuration parameters are not set correctly.

- A network failure or communication error prevents the connection.
- The server associated with the connection is down.
- If the connection status is failed, check the Replication Agent system log to determine the cause of the failure.

Note You may also need to check the system log of the server associated with the connection to determine the cause of the failure.

- See the *Replication Agent Administration Guide* for information about setting up Replication Agent connection configuration parameters.
- See Chapter 2, “Configuration Parameters” for information about specific connection configuration parameters.
- The `test_connection` command is valid when the Replication Agent instance is in the *Admin*, *Replicating*, or *Replication Down* states.

See also

`ra_config`, `ra_statistics`, `ra_status`

trace

Description

Returns current trace flag settings, or changes trace flag settings for the Replication Agent instance.

Syntax

`trace [flag]all, switch`

Parameters

flag

The name of the trace flag to change the setting for.

all

A keyword that allows you to apply a switch value to all of the trace flags at once.

switch

A Boolean (true or false) value that enables or disables tracing for the trace point identified in the *flag* option.

Usage

- The trace command is intended for use by Sybase Technical Support engineers when troubleshooting Replication Agent.
- When trace is invoked with no option, it returns the current settings for all Replication Agent trace flags.

- When trace is invoked with the *flag* and *switch* options, it changes the setting of the trace flag identified, and it returns the current (new) setting for the trace flag.
- When trace is invoked with the *all* keyword and a *switch* option, it sets all Replication Agent trace flags to the value specified in the *switch* option, and it returns the current (new) setting for all of the trace flags.
- Changes made with the trace command take effect immediately.
- When a trace flag is set to true, tracing is enabled for the trace points identified by the flag. When set to false, tracing is disabled for the trace points.
- Output from all trace points (except *LTITRACELTL*) is sent to the Replication Agent system log file. Use the *log_system_name* command to find the name and path of the Replication Agent system log file.
- Output from the *LTITRACELTL* trace point is sent to a separate trace output file named *LTITRACELTL.log*. To view the contents of the *LTITRACELTL.log* file, your file viewer must be capable of handling very long lines.

Note The *LTITRACELTL.log* file contains a human-readable representation of the LTL, not the actual LTL commands sent to the primary Replication Server.

- Table 1-11 lists Replication Agent trace flags:

Table 1-11: Replication Agent trace flags

Trace flag	Description
<i>BMGRTRACE</i>	When set to “true,” this flag enables Bean Management event tracing.
<i>CACHETRC</i>	When set to “true,” this flag enables tracing of internal cache events.
<i>DBCCONTEXT</i>	When set to “true,” this flag turns on tracing of database context events.
<i>LATRC</i>	When set to “true,” this flag traces general Log Administrator operations.
<i>LATRCSQL</i>	When set to “true,” this flag traces SQL conversations between Log Administrator and the primary database.
<i>LICTRACE</i>	When set to “true,” this flag traces feature license check-in/checkout events.
<i>LOGREADTRC</i>	When set to “true”, turns on trace of database log reading.
<i>LRTRACE</i>	When set to “true,” this flag traces general execution of the Log Reader component.
<i>LTITRACE</i>	When set to “true,” this trace flag enables tracing operations of the Log Transfer Interface component.
<i>LTITRACELTL</i>	When set to “true,” this trace flag enables LTL statement tracing in the <i>LTITRACELTL.log</i> file.
<i>LTMCI</i>	When set to “true,” causes tracing of LTM component interface invocations and LTM invocations of other components' interfaces.
<i>LTLFMTTRC</i>	When set to “true,” this trace flag enables tracing of the LTL formatter.
<i>LTMHL</i>	When set to “true,” causes highlights in the LTM execution path to be noted.
<i>LTMSC</i>	When set to “true,” causes tracing of all Replication Agent state changes.
<i>RACONTRC</i>	When set to “true,” causes tracing of connection and query execution.
<i>RACONTRCSQL</i>	When set to “true,” causes tracing of SQL statements to be executed.
<i>RASDTRC</i>	(For Oracle and Microsoft SQL Server) When set to “true,” turns on tracing of Replication Agent System Data Repository events.
<i>RATRACE</i>	When set to “true,” causes tracing of Replication Agent events.
<i>RSTICKETTRC</i>	When set to “true,” causes Replication Agent to log trace message including the <i>rs_ticket</i> value to the Replication Agent system log during LTL formatting.

Trace flag	Description
<i>STMTRACE</i>	When set to “true,” causes tracing of LTM state monitor events.
<i>THREADTRC</i>	When set to “true,” logs ThreadPool trace events.

- You *cannot* change the settings of SYSTEM trace flags.

Table 1-12 lists Replication Agent SYSTEM trace flags:

Table 1-12: Replication Agent SYSTEM trace flags

Trace flag	Description
<i>CONFIG</i>	Configuration change event logged.
<i>ERROR</i>	Serious error; manual intervention may be needed to recover.
<i>FATAL</i>	Critical error; application shut down; manual intervention required to recover.
<i>INFORMATION</i>	Information only; no action required.
<i>WARNING</i>	Minor error; operation not affected, or problem is recoverable.

- The trace command is valid when the Replication Agent instance is in the *Admin*, *Replicating*, or *Replication Down* state.

See also

log_system_name

Configuration Parameters

This chapter describes the Replication Agent configuration file and configuration parameters.

Topic	Page
Configuration parameter overview	157
Configuration parameter reference	159

Configuration parameter overview

Configuration parameters record the user-configurable settings that control how a Replication Agent instance operates. The current values of all configuration parameters are stored in the *configuration* file of each Replication Agent instance.

Replication Agent configuration file

The configuration file is created automatically when you create a Replication Agent instance. It resides in the instance subdirectory, under the Replication Agent base directory.

The configuration file is named after the Replication Agent instance, with the extension *.cfg* (for example, if the instance is named “my_ra,” the configuration file is *my_ra.cfg*).

Each time a Replication Agent instance starts up, it reads the configuration file to get the configuration information it needs to run. After start-up, the only time the Replication Agent accesses the configuration file is when the *ra_config* or *ra_set_login* command is invoked to change the value of a configuration parameter.

When the value of a configuration parameter is changed, Replication Agent saves the new value, overwriting the entire configuration file.

Configuration file format

The configuration file is a flat ASCII file that contains configuration information for a single Replication Agent instance.

The first two lines in the configuration file identify the file as a Replication Agent configuration file and record the time that the file was last modified. For example:

```
#RAX Property File
#Fri Jan 12 07:33:18 MST 2008
```

Each configuration parameter name appears on a separate line, followed by the equal symbol (=) and the current value of the parameter. For example:

```
compress_ltl_syntax=true
```

If the Replication Agent instance is not running, you can view the configuration file to examine the current Replication Agent configuration.

Note Sybase recommends that you do *not* edit the configuration file, because Replication Agent overwrites the entire configuration file every time the `ra_config` or `ra_set_login` command is invoked to change a parameter value.

If the Replication Agent instance is running, use the `ra_config` command to view the current Replication Agent configuration.

Changing configuration parameters

To view, set, or change the current value of a Replication Agent configuration parameter, use the `ra_config` command.

To change the current Replication Agent administrator login (`ltm_admin_user`) or administrator password (`ltm_admin_pw`), you must use the `ra_set_login` command.

Note You cannot use the `ltm_admin_user` and `ltm_admin_pw` parameters; additionally, they do not appear in the parameter list returned by `ra_config`.

See Chapter 1, “Command Reference” for more information about using the `ra_config` and `ra_set_login` commands.

Copying a Replication Agent configuration

When you create a new Replication Agent instance with the `ra_admin` utility, you can specify the new instance to use the same configuration parameter values as an existing Replication Agent instance.

Note When you copy an existing configuration instance when creating a new Replication Agent instance, certain configuration parameter values are not copied to the new configuration. See the *Replication Agent Administration Guide* for more information.

If you do not copy an existing configuration when you create a new Replication Agent instance, the `ra_admin` utility creates a default configuration file, with default values for all configuration parameters.

Configuration parameter reference

Table 2-1 lists all of the Replication Agent configuration parameters and a brief description of each parameter. The target column indicates when the parameter is used for “All” targets or specific targets; Oracle, Microsoft SQL Server (MSSQL), or DB2 UDB (UDB).

Table 2-1: Replication Agent configuration parameters

Parameter name	Target	Description
<code>admin_port</code>	All	Port number that the Replication Agent will use to listen for administrative connections.
<code>archive_read_block_count</code>	Oracle	Determines the number of blocks or pages that are read at a time from the primary database archive log.
<code>asm_password</code>	Oracle	Password for Oracle Automatic Storage Management (ASM) access for the user specified in the <code>asm_username</code> .
<code>asm_tns_connection</code>	Oracle	Identifies the Oracle ASM connection name found in the <code>tnsnames.ora</code> file.
<code>asm_tns_filename</code>	Oracle	Identifies the Oracle <code>tnsnames.ora</code> file name where the ASM connection information is located.

Parameter name	Target	Description
asm_username	Oracle	Identifies the Oracle user name to be used when connecting to the ASM server.
auto_adjust_block_count	Oracle MSSQL	Determines whether or not the Replication Agent should automatically adjust the log_read_block_count parameter if any part of a read is discarded
column_compression	All	Use minimal column information in LTL.
compress_ltl_syntax	All	Use abbreviated LTL syntax.
connect_to_rs	All	Enable/disable connection from LTI to Replication Server.
ddl_password	Oracle MSSQL	Password for ddl_username.
ddl_username	Oracle MSSQL	The database user name included in LTL for replicating DDL commands to the replicate database.
dump_batch_timeout	All	Number of seconds to wait before sending an incomplete LTL buffer to Replication Server.
filter_maint_userid	All	Log Reader filters operations with maintenance user ID.
function_password	Oracle MSSQL	Password for user ID passed in LTL with replicated stored procedure invocations.
function_username	Oracle MSSQL	User ID passed in LTL with replicated stored procedure invocations.
log_backup_files	All	Determines the number of log backup files kept in the log directory.
log_directory	All	Directory where Replication Agent system log file is located.
log_read_block_count	Oracle MSSQL	Determines the number of blocks or pages that are read at a time from the primary database transaction log.
log_trace_verbose	All	Switch on/off verbose mode in trace log file.
log_wrap	All	Number of 1KB blocks written to log file before wrapping.

Parameter name	Target	Description
lr_max_scan_queue_size	Oracle MSSQL	Maximum number of log records permitted in the log reader log scan queue during replication.
lr_max_op_queue_size	Oracle MSSQL	Maximum number of operations permitted in the log reader operation queue during replication.
lr_ntext_byte_order	Oracle MSSQL	Specifies which byte order to use when replicating NCLOB for Oracle, and NTEXT for Microsoft SQL Server.
lr_parallel_scan	Oracle	Determines whether redo log threads for all nodes in an RAC database are scanned in parallel or with a serial scan.
lr_read_buffer_size	UDB	Size in bytes of the log read buffer.
lr_send_trunc_partition_ddl	Oracle	Determines whether truncate partition commands are sent as DDL or DML to the replicate database.
lti_batch_mode	All	Switches on/off LTI batch mode.
lti_max_buffer_size	All	Maximum number of change sets stored in the LTI input buffer.
lti_update_trunc_point	All	Number of LTL commands sent before LTI requests new LTM Locator.
ltl_batch_size	All	Size of the LTL batch buffer.
ltl_big_endian_unitext	All	Specifies whether unicode LOB data should be converted from little endian to big endian before sending LTL to Replication Server.
ltl_character_case	All	Case of database object names sent to Replication Server.
ltl_origin_time_required	All	Specifies whether to send origin_time command tag in LTL.
ltl_send_only_primary_keys	All	Controls whether the Replication Agent sends only primary key columns data for the <i>before</i> image for update and delete operations.
ltm_admin_pw	All	Password for Replication Agent administrative port.
ltm_admin_user	All	User ID for Replication Agent administrative port.
max_ops_per_scan	All	Maximum number of operations Log Reader will read in a single log scan.

Parameter name	Target	Description
pdb_archive_path	Oracle UDB	Identifies the directory path where the Replication Agent expects to find archived Oracle redo log files or the archived UDB transaction log files.
pdb_archive_remove	Oracle UDB	Enables or disables the removal of archived transaction log files from the path specified by <code>pdb_archive_path</code> .
pdb_auto_create_repdefs	Oracle MSSQL	If set to true, when tables and procedures are marked for replication, a replication definition is automatically created at Replication Server for that table or procedure.
pdb_automark_tables	Oracle MSSQL	Determines if the Replication Agent automatically marks tables for replication during initialization or DDL replication.
pdb_auto_run_scripts	All	Automatic execution of SQL scripts used to create/remove transaction log objects and mark/unmark primary database objects.
pdb_commit_generation	Oracle	Determines whether or not Replication Agent generates commit log records at the primary database.
pdb_commit_interval	Oracle	Determines the interval in seconds at which Replication Agent generates commit log records.
pdb_convert_datetime	All	Converts native date/time formats to Sybase datetime format.
pdb_dflt_column_repl	All	Enables replication for LOB columns by default when table is marked.
pdb_dflt_object_repl	Oracle UDB	Enables replication by default when object is marked.
pdb_ignore_unsupported_anydata	Oracle	Determines whether or not Replication Agent ignores data of unsupported datatypes stored in columns of type ANYDATA.
pdb_include_archives	Oracle	Enables or disables the use of Oracle archive log files.
pdb_skip_missing_user	All	Determines whether or not Replication Agent skips the processing of any command for which there is no matching session or user information.

Parameter name	Target	Description
pdb_support_large_identifier	All	To support replication of large identifiers up to 255 characters in length with Replication Server 12.6 or later.
pdb_timezone_file	Oracle	Specifies the file to read at Replication Agent initialization to obtain Oracle time zone information.
pdb_xlog_device	All	Name of the primary database device.
pdb_xlog_prefix	All	Character string prefix used to identify transaction log objects.
pdb_xlog_prefix_chars	All	Non-alphabetic characters allowed in <code>pdb_xlog_prefix</code> .
pds_connection_type	All	Type of connection to primary data server.
pds_dac_port_number	MSSQL	The primary database dedicated administrative connection port number used by Replication Agent during the server level initialization of the primary database.
pds_database_name	All	Name of database replicated from the primary data server.
pds_datasource_name	UDB	Data source name or database alias of the database replicated from the primary data server.
pds_host_name	All	Name of primary data server host machine.
pds_integrated_security	MSSQL	Determines if the Replication Agent should use Windows authentication when connecting to the primary SQL Server.
pds_password	All	Password for user ID that the Replication Agent uses to access the primary data server.
pds_port_number	All	Port number for the primary data server.
pds_retry_count	All	Number of times to retry connection to primary data server.
pds_retry_timeout	All	Number of seconds to wait between connection retry attempts.
pds_server_name	All	Server name of primary data server.

Parameter name	Target	Description
pds_tns_connection	Oracle	Oracle connection name found in the <i>tnsname.ora</i> file which identifies the connection information for the primary database.
pds_tns_filename	Oracle	Oracle file name identifying the Oracle <i>tnsname.ora</i> file to be used to identify the connection information for the primary database.
pds_username	All	User ID that Replication Agent uses to access primary data server.
ra_retry_count	All	Number of times LTM attempts to get back to <i>Replicating</i> state after a failure.
ra_retry_timeout	All	Number of seconds to wait between LTM attempts to get back to <i>Replicating</i> state.
rasd_backup_dir	Oracle MSSQL	The directory path for Replication Agent System Database (RASD) backup files.
rasd_database	Oracle MSSQL	The directory path for the Replication Agent System Database (RASD) database file.
rasd_mirror_tran_log	Oracle MSSQL	Enables or disables Replication Agent System Database (RASD) transaction log mirroring.
rasd_trace_log_dir	Oracle MSSQL	The directory path for the Replication Agent System Database (RASD) trace log file.
rasd_tran_log	Oracle MSSQL	The directory path for the Replication Agent System Database (RASD) transaction log file.
rasd_tran_log_mirror	Oracle MSSQL	The directory path for the Replication Agent System Database (RASD) transaction log file mirror.
ra_standby	Oracle	Determines whether or not Replication Agent functions in standby mode.
rman_enabled	Oracle	Indicates whether or not Replication Agent uses the Oracle RMAN utility to truncate old archive log files.
rman_password	Oracle	Password used with <i>rman_username</i> to connect to the Oracle RMAN utility.
rman_username	Oracle	Login name used with <i>rman_password</i> to connect to the Oracle RMAN utility.

Parameter name	Target	Description
rs_charset	All	Character set used to communicate with Replication Server.
rs_host_name	All	Name of primary Replication Server host machine.
rs_packet_size	All	Network I/O packet size sent to Replication Server.
rs_password	All	Password for user ID Replication Agent uses to access Replication Server.
rs_port_number	All	Port number for primary Replication Server.
rs_replicate_owner_required	All	Indicates if the owner is always included with the replicate table clause when generating replication definitions.
rs_retry_count	All	Number of times to retry connection to primary Replication Server.
rs_retry_timeout	All	Number of seconds to wait between connection retry attempts.
rs_source_db	All	Name of primary database identified to Replication Server.
rs_source_ds	All	Name of primary data server identified to Replication Server.
rs_ticket_version	All	Determines whether Replication Agent records the primary database time or the primary database date and time into the rs_ticket marker.
rs_username	All	User ID that Replication Agent uses to access primary Replication Server.
rssd_charset	All	Character set used to communicate with RSSD.
rssd_database_name	All	Name of RSSD database.
rssd_host_name	All	Name of RSSD host machine.
rssd_password	All	Password for user ID Replication Agent uses to access RSSD.
rssd_port_number	All	Port number for RSSD.
rssd_username	All	User ID that Replication Agent uses to access RSSD.
scan_eol_threshold	Oracle	Number of seconds that Replication Agent waits on an RAC node before the node is deemed inactive.

Parameter name	Target	Description
scan_sleep_increment	All	Number of seconds to increase Log Reader wait before next scan after finding no operations to replicate.
scan_sleep_max	All	Maximum number of seconds for Log Reader to wait before next scan after finding no operations to replicate.
skip_ltl_errors	All	LTI ignores error messages returned by Replication Server.
structured_tokens	All	LTI uses structured tokens when generating LTL output.
truncation_interval	All	Number of minutes to wait between automatic log truncations.
truncation_type	All	Methods of log truncation allowed.
use_rssd	All	Switches on/off access to RSSD for replication definitions.

admin_port

The client socket port number of the Replication Agent.

Default

10000

Value

A valid port number on the Replication Agent host machine.

Comments

- When you create a Replication Agent instance, you must specify a client socket port number for the instance administration port. Client applications use this port number to connect to the Replication Agent instance.
- You must specify a port number that does not conflict with any port numbers already in use on the Replication Agent host machine.
- If you change the value of the admin_port parameter with the ra_config command, the new value is recorded in the configuration file immediately, but you must shut down and restart the Replication Agent instance to make the new port number take effect.
- After you change the value of the admin_port parameter with the ra_config command, the next time you log in to the Replication Agent administration port, you must use the new port number.

archive_read_block_count

Determines the number of blocks or pages that are read at a time from the primary database archive log.

Note This parameter is available only for Oracle.

Default	64
Value	An integer between 1 - 8192
Comments	<ul style="list-style-type: none"> • Redo block sizes are 512 bytes for all platforms except HP, for which the block size is 1024 bytes. • The block size is not configurable. • The archive_read_block_count affects only the read of Oracle archive redo logs. • The log_read_block_count configuration determines the block count for Oracle online redo logs.

asm_password

The password associated with the configuration parameter asm_username user to access the Oracle Automatic Storage Management (ASM) server instance.

Note This parameter is available only for Oracle.

Default	"" (empty string)
Value	A valid password.
Comments	<ul style="list-style-type: none"> • Setting of the configuration parameter is only required when the Oracle ASM product is being used to manage Oracle online redo or archive redo storage. • The value of the asm_password parameter is encrypted in the Replication Agent instance configuration file.
See also	asm_tns_connection, asm_tns_filename, and asm_username configuration parameters.

asm_tns_connection

The Oracle connection name that identifies the connection parameters for the Oracle Automatic Storage Management (ASM) connection in the Oracle *tnsnames.ora* file. If configuration parameter `asm_tns_filename` is not configured, the *tnsnames.ora* file identified by `pds_tns_filename` will be used.

Note This parameter is available only for Oracle.

Default	<not configured>
Value	A valid ASM connection name from the Oracle <i>tnsnames.ora</i> file specified by either the <code>asm_tns_filename</code> or <code>pds_tns_filename</code> configuration parameter.
Comments	Setting of the configuration parameter is only required when the Oracle ASM product is being used to manage Oracle online redo or archive redo storage
See also	<code>pds_tns_filename</code> , <code>asm_tns_filename</code> , <code>asm_username</code> , and <code>asm_tns_password</code> configuration parameters.

asm_tns_filename

The fully-qualified file name identifying the Oracle *tnsnames.ora* file that contains the Oracle ASM connection parameters. This configuration parameter is required only when the connection parameter information required for ASM does not exist in the *tnsnames.ora* file pointed to by the `pds_tns_filename` configuration parameter.

Note This parameter is available only for Oracle.

Default	<not configured>
Value	A valid Oracle <i>tnsnames.ora</i> file that contains the connection parameters to the Oracle ASM server. This file normally resides in the ASM instance's <code>ORACLE_HOME\network\admin</code> directory.
Comments	<ul style="list-style-type: none">Setting of the configuration parameter is only required when the Oracle ASM product is being used to manage Oracle online redo or archive redo storage.

- This parameter is required only when the `pds_tns_filename` configuration parameter is not populated, or the `tnsnames.ora` file pointed to by configuration parameter `pds_tns_filename` does not contain the connection definition for the ASM connection to be used. This may occur if the primary Oracle instance and ASM instance was installed in different `ORACLE_HOME` locations, resulting in different `tnsnames.ora` files for each server.

Warning! The Replication Agent process must have *read* permission to this file. Access failures prevent the Replication Agent from connecting to the Oracle ASM server.

See also `pds_tns_filename` and `asm_tns_filename` configuration parameters.

asm_username

The login name that the Replication Agent uses to access the Oracle ASM server.

Note This parameter is available only for Oracle.

Default	<not configured>
Value	A valid Oracle ASM user name.
Comments	<ul style="list-style-type: none"> • Setting of the configuration parameter is only required when the Oracle ASM product is being used to manage Oracle online redo or archive redo storage • The value of the <code>asm_username</code> parameter is the login name that the Replication Agent uses to log in to the Oracle ASM server that is managing storage for Oracle online or archived redo logs. • This login name must be defined in the ASM server, with appropriate privileges to query ASM system views. • The Replication Agent uses this login to query ASM in order to determine Oracle online or archived redo log locations. • To remove archive log files from ASM that are no longer required for Replication, the user must have update privileges in ASM.

See also `asm_tns_connection`, `asm_tns_filename`, `asm_password`, and `pds_archive_remove` configuration parameters.

auto_adjust_block_count

Determines whether or not the Replication Agent should automatically adjust the `log_read_block_count` parameter if any part of a read is discarded.

Note This parameter is available only for Oracle and Microsoft SQL Server.

Default	true
Value	<p>true – enables automatic adjusting of the log read block count.</p> <p>false – disables automatic adjusting of the log read block count.</p>
Comments	<ul style="list-style-type: none"> • The <code>auto_adjust_block_count</code> is relevant only if the <code>log_read_block_count</code> is configured to a value greater than 1. • The Replication Agent never automatically adjusts to a block count greater than the block count specified by the <code>log_read_block_count</code> configuration parameter. • For Oracle, the <code>auto_adjust_block_count</code> is relevant only for online redo log reading, since archive logs have a defined end.
See also	<code>log_read_block_count</code> configuration parameter.

column_compression

Determines whether the Log Transfer Interface component sends all columns in row *after* images, or only the columns that changed in an update operation.

Default	true
Values	<p>true – enables minimal column information (only changed columns in row <i>after</i> images) in Log Transfer Language (LTL) for update operations.</p> <p>false – disables minimal column information in LTL for update operations.</p>
Comments	<ul style="list-style-type: none"> • When the <code>column_compression</code> parameter is set to false, the LTI component sends complete row <i>after</i> images in LTL, including columns in which no data changed as a result of an update operation. • When the <code>column_compression</code> parameter is set to true, the LTI component sends minimal column information in the row <i>after</i> images in LTL, with only the columns that changed as a result of an update operation. Columns in which no data changed as a result of the update are not sent in LTL.

- In general, setting the value of the `column_compression` parameter to `true` provides better Replication Agent throughput.

See also `ltl_send_only_primary_keys` configuration parameter.

compress_ltl_syntax

Determines whether the Log Transfer Interface component compresses Log Transfer Language (LTL) commands using abbreviated syntax.

Default `true`

Values `true` – enables LTL compression, using abbreviated LTL syntax.
`false` – disables LTL compression.

Comments

- Setting the value of the `compress_ltl_syntax` parameter to `true` will provide better Replication Agent throughput.
- See the *Replication Agent Administration Guide* for more information about LTL commands and abbreviated LTL syntax.

connect_to_rs

Enables or disables the network connection to the primary Replication Server.

Default `true`

Values `true` – enables the network connection to the Replication Server.
`false` – disables the network connection to the Replication Server.

Comments

- When the value of the `connect_to_rs` parameter is `false`, the network connection from the Replication Agent to the Replication Server is disabled, and no replication can occur.
- When the network connection to the Replication Server is disabled by the `connect_to_rs` parameter, the Replication Agent instance can still go to *Replicating* state, with the following limitations:
 - A “dummy” connection in the Replication Agent emulates a real connection to the Replication Server.
 - The value of the LTM Locator stored in the Replication Agent transaction log is set to zero.

- The maintenance user name is set to an invalid user ID.

Note maintenance user operations cannot be filtered when the value of the `connect_to_rs` parameter is false.

- You can use the `connect_to_rs` parameter to temporarily disable the network connection to the Replication Server for testing.
- When the value of the `connect_to_rs` parameter is false, you can put the Replication Agent instance in *Replicating* state, set the value of the `LTITRACELTL` trace flag to true, and view a readable representation of the LTL that would have been sent to the Replication Server if the connection had not been disabled.
- During normal Replication Agent operation, the value of the `connect_to_rs` parameter must be true.

ddl_password

Updates the log device repository in the RASD. Identifies the password for `ddl_username`.

Note This parameter is available only for Oracle and Microsoft SQL Server.

Default

"" (empty string)

Value

A valid password.

Comments

- The value of the `ddl_password` parameter can be up to 30 characters.
- The value of the `ddl_password` parameter is the password for the database user name specified in the `ddl_username` parameter.
- The value of the `ddl_password` parameter is encrypted in the Replication Agent configuration file.

ddl_username

The database user name included in LTL for replicating DDL commands to the standby database. This user must have permission to execute all replicated DDL commands at the standby database.

Note This parameter is available only for Oracle and Microsoft SQL Server.

Default	<not_configured>
Value	A valid user name in the standby database.
Comments	<ul style="list-style-type: none"> • The value for the <code>ddl_username</code> must not be the same as the value of the maintenance user defined in Replication Server for the standby connection. Failure to provide different names results in a Replication Server error. • The value of the <code>ddl_username</code> parameter is sent in the LTL for all replicated DDL statements. • The value of the <code>ddl_password</code> parameter is the password for the database user name specified in the <code>ddl_username</code> parameter. • When DDL is replicated, Replication Server will connect to the replicate database using the <code>ddl_username</code> and <code>ddl_password</code>. • For Oracle, Replication Server issues the following message: <pre>ALTER SESSION SET CURRENT_SCHEMA=<i>user</i></pre> <p>where <i>user</i> is the user ID that generated the DDL operation at the primary database. The actual DDL command is then executed against the replicate database. If the <code>ddl_username</code> does not have permission to issue <code>ALTER SESSION SET CURRENT_SCHEMA</code> or to execute the DDL command against the <i>user</i> schema, the command fails.</p> • For Microsoft SQL Server, Replication Agent will send: <pre>execute as login = <i>user</i></pre> <p>where <i>user</i> is the user ID that generated the DDL operation at the primary database. The actual DDL command is then executed against the replicate database. If the <code>ddl_username</code> does not have permission to issue <code>execute as login</code> or to execute the DDL command against the <i>user</i> schema, the command fails.</p>

dump_batch_timeout

Specifies the number of seconds to wait before sending the contents of the Log Transfer Interface (LTI) buffer to the Replication Server, even though the buffer is not full.

Default	5
Value	An integer from 1 to 60.
Comments	<ul style="list-style-type: none"> • The value of the <code>dump_batch_timeout</code> parameter is the number of seconds from the time the previous LTI buffer was sent to the Replication Server until the next buffer will be sent. • The <code>dump_batch_timeout</code> parameter has no effect if the value of the <code>lti_batch_mode</code> parameter is false.

filter_maint_userid

Determines whether operations applied by the maintenance user are ignored.

Default	true
Values	<p>true – enables the Log Reader to ignore maintenance user operations.</p> <p>false – disables the Log Reader filter to allow replicating maintenance user operations.</p>
Comments	<ul style="list-style-type: none"> • The <code>filter_maint_userid</code> configuration parameter is provided to support bidirectional replication, in which the primary database also serves as a replicate database that has transactions applied to it by a Replication Server maintenance user. • If the value of the <code>filter_maint_userid</code> parameter is true, database operations applied by the maintenance user are <i>not</i> replicated. The Log Reader component filters out (ignores) operations applied by the maintenance user when it reads the transaction log. • If the value of the <code>filter_maint_userid</code> parameter is false, database operations applied by the maintenance user are replicated. The Log Reader component replicates all operations on marked objects, regardless of the user that applied the operation. • The maintenance user login is specified when the database connection for the primary database is created in the Replication Server.

function_password

The password included in Log Transfer Language for replication of “request” stored procedures.

Note This parameter is available only for Oracle and Microsoft SQL Server.

Default	"" (empty string)
Values	A valid password.
Comments	<ul style="list-style-type: none">• The value of the function_password parameter can be up to 30 characters.• The value of the function_password parameter is the password for the database user name specified in the function_username parameter.• The value of the function_password parameter is encrypted in the Replication Agent configuration file.• For more information about “request” stored procedures, see Replication Server documentation.

function_username

The database user name included in Log Transfer Language (LTL) for replication of “request” stored procedures.

Note This parameter is available only for Oracle and Microsoft SQL Server.

Default	sa
Values	A valid user name in the primary database.
Comments	<ul style="list-style-type: none">• The value of the function_username parameter is sent in the LTL for all replicated stored procedures in the primary database.• The value of the function_password parameter is the password for the database user name specified in the function_username parameter.• For more information about “request” stored procedures, see Replication Server documentation.

log_backup_files

	The number of backup log files kept in the Replication Agent instance <i>log</i> directory.
Default	3
Values	An integer greater than or equal to 1.
Comments	<ul style="list-style-type: none"> When the system log wraps, Replication Agent copies the current log file to a backup file, with a generated number appended to the file's name. For example, if the system log file is named <i>my_ra.log</i>, the first backup file created when the system log wraps would be named <i>my_ra1.log</i>. The second backup file created would be named <i>my_ra2.log</i>, and so on. When the number of backup files exceeds the value of the <code>log_backup_files</code> parameter, the oldest backup file (that is, the one with the lowest generated number) is deleted from the <i>log</i> directory before the next backup file is created.

log_directory

	The directory for Replication Agent system log files.
Default	<p>The path to the <i>log</i> directory created when the Replication Agent instance was created. For example:</p> <ul style="list-style-type: none"> On Microsoft Windows platforms: <pre style="margin-left: 40px;">%SYBASE%\RA-15_5\inst_name\log</pre> <p>where:</p> <ul style="list-style-type: none"> <code>%SYBASE%</code> is the path to the Replication Agent installation directory. <code>inst_name</code> is the name of the Replication Agent instance. On UNIX platforms: <pre style="margin-left: 40px;">\$SYBASE/RA-15_5/inst_name/log</pre> <p>where:</p> <ul style="list-style-type: none"> <code>\$SYBASE</code> is the path to the Replication Agent installation directory. <code>inst_name</code> is the name of the Replication Agent instance.
Value	A valid path on the Replication Agent host machine.

- Comments
- When a Replication Agent instance is created, the *log* directory is created as part of the instance directories. The default value of the `log_directory` parameter points to that directory.
 - If you specify any valid path as the value of the `log_directory` parameter, the Replication Agent instance places its system log files in the directory you specify the next time it is started.
 - If you specify the default value of the `log_directory` parameter by using the default keyword in the `ra_config` command, then the next time it is started, Replication Agent will place its system log files in the *log* directory that was created when the Replication Agent instance was created.
 - If you change the value of the `log_directory` parameter with the `ra_config` command, the new value is recorded in the configuration file immediately, however, you must shut down and restart the Replication Agent instance to make the new value take effect.

log_read_block_count

Determines the number of blocks or pages that are read at a time from the primary database transaction log.

Note This parameter is available for Oracle and Microsoft SQL Server.

- Default 64
- Values An integer between 1 - 8192
- Comment
- In Microsoft SQL Server, transaction log page sizes can be 2K, 4K, 8K, or 16K.
 - In Oracle, redo block sizes are 512 bytes for all platforms except HP, for which the block size is 1024 bytes. The block size is not configurable.
 - In Oracle, the `log_read_block_count` affects only the read for online redo logs. The `archive_log_read_block_count` controls the read for archive redo logs.
- See also `auto_adjust_block_count` configuration parameter.

log_trace_verbose

	Enables or disables additional diagnostic information in Replication Agent system log files.
Default	false
Values	true – enables detailed diagnostic information in log files. false – disables detailed diagnostic information in log files.
Comment	Detailed diagnostic information is intended for troubleshooting only, with assistance from Sybase Technical Support.

log_wrap

	The maximum size, in 1K blocks, of the Replication Agent system log file before wrapping.
Default	1000
Value	An integer greater than or equal to 1000.
Comments	<ul style="list-style-type: none">• The value of the log_wrap parameter is the number of 1KB blocks written by Replication Agent, before it wraps the system log file.• Larger values for the log_wrap parameter allow more log history in each file. Smaller values produce smaller log files.• When the log file wraps, Replication Agent copies the current log file to a backup file, with a generated number appended to the file's name. For example, if the system log file is named <i>my_ra.log</i>, the first backup file created when the system log wraps would be named <i>my_ra1.log</i>. The second backup file created would be named <i>my_ra2.log</i>, and so on.• When the number of backup files exceeds the value of the log_backup_files parameter, the oldest backup file (that is, the one with the lowest generated number) is deleted from the <i>log</i> directory before the next backup file is created.

lr_max_scan_queue_size

The maximum number of log records permitted in the log reader log scan queue during replication.

Note This parameter is available only for Oracle and Microsoft SQL Server.

Default	1000
Values	An integer from 25 to $2^{31} - 1$.
Comments	<ul style="list-style-type: none">• The <code>lr_max_scan_queue_size</code> parameter can be used to tune Replication Agent performance.

lr_max_op_queue_size

The maximum number of operations permitted in the log reader operation queue during replication.

Note This parameter is available only for Oracle and Microsoft SQL Server.

Default	1000
Values	An integer from 25 to $2^{31} - 1$.
Comments	<ul style="list-style-type: none">• The <code>lr_max_op_queue_size</code> parameter can be used to tune Replication Agent performance.

lr_ntext_byte_order

Ensures the byte order of NLOB data (for Oracle) or NTEXT data (for Microsoft SQL Server) is sent correctly to the replicate database. The byte order being correct is necessary when you are replicating with different primary and replicate database types (Oracle to Microsoft SQL Server), or primary and replicate databases that are on different platforms (Windows to big endian UNIX).

Note This parameter is available only for Oracle and Microsoft SQL Server.

Default	big
---------	-----

Values	big – big endian little – little endian
Examples	<ul style="list-style-type: none">• For Microsoft SQL Server: When replicating Microsoft SQL Server to Microsoft SQL Server, both the primary and replicate database must be little endian because Microsoft SQL Server runs only on Windows. The parameter should be set to little endian. If the replicate database is not Microsoft SQL Server, you need to know whether it is little endian or big endian, and set the parameter to match.• For Oracle: When replicating Oracle to Oracle, you need to know the endian for the replicate database and set the parameter so the correct endian is sent to the replicate. The same applies when replicating to databases other than Oracle.
Comments	<ul style="list-style-type: none">• If <code>lr_nnext_byte_order</code> is set to little, <code>ltl_big_endian_unitext</code> must be set to false.• Big endian indicates a left-to-right byte order architecture; little endian indicates a right-to-left byte order architecture.• Microsoft SQL Server stores double byte ntext datatype values in little endian byte order. By default, the byte order of ntext data will be converted during replication to big endian, to allow the data to be transmitted over networks using the common “network” byte order, or big endian.

If the target database is also Microsoft SQL Server, the SQL Server will *not* automatically convert the replicated data from the sent big endian order to the SQL Server desired little endian order. To support replicating ntext data to a Microsoft SQL Server (or other replicate server that fails to provide the necessary conversion), you may 'force' the byte order to be sent using the `lr_ntext_byte_order` parameter by specifying big or little, as desired to meet the expectations of your replicate database.

Note The default behavior of the Replication Agent is to force any unicode data to big endian order, as defined by configuration parameter `ltl_big_endian_unitext`. In order to allow configuration parameter `lr_ntext_byte_order` to successfully override the SQL Server byte order, you must also set `ltl_big_endian_unitext` configuration parameter to false whenever the `lr_next_byte_order` parameter is used.

- The following describes the relationship between `ltl_big_endian_unitext` and `lr_ntext_byte_order` configuration parameters:
 - When `ltl_big_endian_unitext` is set true by default, the Replication Agent ensures that all unicode data is sent in big endian order.
 - When set to false, `ltl_big_endian_unitext` allows unicode data to be sent in the byte order that is used when the data is stored in the transaction log file
 - In contrast, `lr_ntext_byte_order`, forces the result of Unicode data read from the transaction log to be in the requested byte order, regardless of how it normally exists in the transaction log file.

See also `ltl_big_endian_unitext` configuration parameter.

lr_parallel_scan

Determines whether redo log threads for all nodes in an RAC database are scanned in parallel or with a serial scan.

Note This parameter is available only for Oracle and is used only for Real Application Cluster (RAC) environments.

Default false

Values true – Replication Agent starts a redo log scan thread for each redo log thread in the RAC database. All redo log threads are scanned in parallel.

false – The redo log threads are scanned serially.

- Comments
- Parallel scan threads can improve performance and reduce read and process latency because all the redo log threads are scanned simultaneously.
 - The actual degree of parallelization that can be achieved depends on the hardware resources available and the number of nodes in the Oracle RAC database. Some environments may not have the resources to benefit from this configuration parameter.

lr_read_buffer_size

Size in bytes of the log read buffer.

Note This parameter is available only for UDB.

Default 16384

Values An integer from 10000 to $2^{31} - 1$.

- Comments
- The Replication Agent for UDB LogReader component uses the value of the lr_read_buffer_size parameter to determine the maximum number of bytes to be read from the transaction log during each scan. If the read buffer size is too small to read one operation, the LogReader component shuts down the Replication Agent instance and reports a -2650 or -30081 error. Unfortunately, these error messages cover general communication errors, not just an insufficient buffer size.
 - A read buffer size that is too large may result in the Replication Agent instance running out of memory.

lr_send_trunc_partition_ddl

Determines whether truncate partition commands are sent as DDL or DML to the replicate database.

Note This parameter is available only for Oracle.

Default true

Values	<p>true – the truncate partition command is sent as a DDL command (alter table).</p> <p>false – the truncate partition is sent as a DML operation.</p>
Comments	<ul style="list-style-type: none"> • If set to true, the truncate partition command is sent as a DDL command. Normally, it is set to replicate to Oracle. • Set to false, when replicating to databases that treat truncate partition commands as DML.

lti_batch_mode

	Enables or disables the Log Transfer Interface component LTL batch mode.
Default	true
Values	<p>true – enables LTL batch mode.</p> <p>false – disables LTL batch mode.</p>
Comments	<ul style="list-style-type: none"> • If the value of the <code>lti_batch_mode</code> parameter is true, the LTI component sends LTL commands to the Replication Server in batches, instead of one command at a time: <ul style="list-style-type: none"> • The LTI component fits as many LTL commands as it can into its LTL batch mode buffer, before it sends any commands to the Replication Server. • When the time interval specified in the <code>dump_batch_timeout</code> parameter expires, the LTI component sends the current LTL batch mode buffer contents to the Replication Server, even if the buffer is not full. • If the value of the <code>lti_batch_mode</code> parameter is false, the LTI component sends individual LTL commands to the Replication Server for each change set in its input queue. • When Replication Agent connects to the Replication Server, it determines the version of the Replication Server: <ul style="list-style-type: none"> • If the Replication Server version is earlier than 12.5, the size of the LTL batch mode buffer is set to 16KB automatically. • If the Replication Server version is 12.5 or later, Replication Agent sets the size of the LTL batch mode buffer to the size specified by the <code>ltl_batch_size</code> parameter.

- If the Replication Server version is 12.5 or later, you can use the Replication Agent `ltl_batch_size` parameter to set the size of the LTI component's LTL batch mode buffer.

Note Adjusting the size of the LTL batch mode buffer can help you optimize the performance of the replication system.

- If the Replication Server version is earlier than 12.5 and the value of the `lti_batch_mode` parameter is true, if any single LTL distribute command exceeds the 16K size of the LTL batch mode buffer, Replication Server returns an error and Replication Agent goes into *Replication Down* state.
- In general, setting the value of the `lti_batch_mode` parameter to true provides better Replication Agent throughput.

lti_max_buffer_size

The maximum size of the Log Transfer Interface (LTI) component's queues.

Default

5000

Value

An integer in the range of 1000 to 100000.

Comments

- The value of the `lti_max_buffer_size` parameter is the maximum number of operations that can be stored in the LTI component's inbound and outbound queues:
 - Operations in the inbound queue represent change sets received from the Log Reader component.
 - Operations in the outbound queue are the Log Transfer Language commands to be sent to the Replication Server.
- The LTI component's inbound queue is a bounded buffer that blocks the processing of the Log Reader component when it gets full.

lti_update_trunc_point

The number of Log Transfer Language (LTL) commands sent before requesting a new LTM Locator from the Replication Server.

Default

10000

Value

An integer from 1 to 100000.

- Comments
- The value of the `lti_update_trunc_point` parameter is the number of LTL commands that Replication Agent sends to the Replication Server, before it requests a new LTM Locator (secondary truncation point) from the Replication Server.
 - Lower numbers cause Replication Agent to request a new LTM Locator from the Replication Server more often.
 - If the value of the `truncation_type` parameter is `locator_update`, setting the value of the `lti_update_trunc_point` parameter to a lower number causes automatic log truncation to occur more frequently.
 - The value of the `lti_update_trunc_point` parameter is a trade-off between better system performance and longer recovery time:
 - Lower values reduce the time it takes to recover from a replication failure, but they may have an adverse affect on overall system throughput.
 - Higher values improve overall system throughput, but they may increase the time it takes to recover from a replication failure.
 - If the Replication Agent is operating in an unreliable network environment, it may be prudent to set the `lti_update_trunc_point` parameter to a lower value to ensure faster recovery.

ltl_batch_size

The size of the Log Transfer Interface component's Log Transfer Language (LTL) batch mode buffer.

Default

40000

Value

An integer from 16384 to 10485760.

Comments

- The value of the `ltl_batch_size` parameter is the size (in bytes) of the LTI component's LTL batch mode buffer.
- When Replication Agent connects to the Replication Server, it determines the version of the Replication Server:
 - If the Replication Server version is earlier than 12.5, the size of the LTL batch mode buffer is set to 16K automatically, and the value of the `ltl_batch_size` parameter is ignored.

- If the Replication Server version is 12.5 or later, Replication Agent sets the size of the LTL batch mode buffer to the size specified by the `ltl_batch_size` parameter.
- The Log Transfer Interface component uses the LTL batch mode buffer only if the value of the `lti_batch_mode` parameter is true. If the value of the `lti_batch_mode` parameter is false, the LTL batch mode buffer is not used.

ltl_big_endian_unitext

Specifies whether “unitext” data is converted from little endian to big endian before sending LTL to Replication Server.

Default

true

Values

true – Unitext data that is in little endian byte order will be changed to big endian byte order.

false – Unitext data byte order is *not* changed.

Comments

When setting this parameter, you must know how the `lr_nxtxt_byte_order` is set. If parameter `lr_nxtxt_byte_order` is set to send the correct byte order for the replicate database, then `ltl_big_endian_unitext` must be set to false so the byte order will not be changed.

See also

`lr_nxtxt_byte_order` configuration parameter.

ltl_character_case

The character case used for database object names in Log Transfer Language (LTL) sent to the Replication Server.

Default

asis

Values

asis – database object names are sent in the same character case as they are returned from the primary database, or (if the value of the `use_rssd` parameter is true) in the same character case as they are specified in replication definitions.

lower – database object names are sent in *all lowercase*, regardless of how they are returned from the primary database, or specified in replication definitions.

upper – database object names in LTL are sent in *all uppercase*, regardless of how they are returned from the primary database, or are specified in replication definitions.

Comments	<ul style="list-style-type: none"> • The <code>ltl_character_case</code> configuration parameter allows you to customize the handling of database object names in LTL to work with replication definitions that specify the object names differently than the way the primary database returns them. • If the value of the <code>ltl_character_case</code> parameter is <code>asis</code>, and the value of the <code>use_rssd</code> parameter is <code>true</code>, database object names are sent in the same character case as they are specified in replication definitions. • If the value of the <code>ltl_character_case</code> parameter is <code>asis</code>, and the value of the <code>use_rssd</code> parameter is <code>false</code>, database object names are sent in the same character case as they are returned from the primary database. • If replication definitions specify database object names in all lowercase, set the value of the <code>ltl_character_case</code> parameter to <code>lower</code>. • If replication definitions specify database object names in all uppercase, set the value of the <code>ltl_character_case</code> parameter to <code>upper</code>. • If you want to send database object names with “mixed” character case (for example, <code>MyTable</code>), set the value of the <code>ltl_character_case</code> parameter to <code>asis</code>.
----------	---

ltl_origin_time_required

	Enables or disables the Log Transfer Language (LTL) <code>origin_time</code> command tag.
Default	<code>false</code>
Values	<p><code>true</code> – enables the <code>origin_time</code> command tag in LTL.</p> <p><code>false</code> – disables the <code>origin_time</code> command tag in LTL.</p>
Comments	<ul style="list-style-type: none"> • If the value of the <code>ltl_origin_time_required</code> parameter is <code>true</code>, the Log Transfer Interface component includes the <code>origin_time</code> command tag in the LTL it generates. • If a Replication Server function string checks for the <code>origin_time</code> command tag, set the value of the <code>ltl_origin_time_required</code> parameter to <code>true</code>. • The <code>datetime</code> value placed in the LTL <code>origin_time</code> command tag is the time that the original primary database operation was recorded in the transaction log, not the time it was scanned and processed by the Log Reader component.

- Setting the value of the `ltl_origin_time_required` parameter to `false` provides better Replication Agent throughput.
- If you use Replication Manager to report latency, you must set the value of the `ltl_origin_time_required` parameter to `true`.

ltl_send_only_primary_keys

Determines whether Replication Agent sends only *before* image primary key columns, or sends all *before* image columns to Replication Server for update and delete operations to the replicate database.

Default

true

Value

true – sends only the *before* image primary key columns to Replication Server.
false – sends all *before* image columns to Replication Server.

Comments

- When set to `true` and a replication definition exists that identifies the primary key column(s) for a table, only the primary key column value(s) are sent for the *before* image in update and delete operations. Sending only primary key column data reduces the amount of data sent to Replication Server, since only primary keys are used to construct the “where” clauses for update and delete operations.
- When set to `false`, *before* image values are sent for all columns available, regardless of primary key definition.
- Setting of `ltl_send_only_primary_keys` to `false`, is only recommended when additional *before* image values provide benefit, such as for supporting customer function strings at the Replicate database, or for resolving other issues where additional column data provides benefit.
- Primary keys are defined within a table level Replication Definition. If configuration `use_rssd` is set to `false`, the setting of `ltl_send_only_primary_keys` has no impact, since Replication Definition information will not be gathered from the Replication Server System Database (RSSD).
- If you set the value to `false`, the performance will be slower.

ltm_admin_pw

The Replication Agent administrator login password.

Default	"" (empty string)
Value	A valid password.
Comments	<ul style="list-style-type: none">• The value of the <code>ltm_admin_pw</code> parameter is the password for the user name authorized to log in to the Replication Agent.• The value of the <code>ltm_admin_pw</code> parameter is encrypted in the Replication Agent configuration file.• To change the value of the <code>ltm_admin_pw</code> parameter, use <code>ra_set_login</code>.• When you change the value of the <code>ltm_admin_pw</code> parameter with <code>ra_set_login</code>, the new value is recorded in the configuration file immediately. However, you must shut down and restart the Replication Agent instance to make the new password take effect. <p>After you change the value of the <code>ltm_admin_pw</code> parameter with <code>ra_set_login</code>, you must use the new password the next time you log in to the Replication Agent.</p>

ltm_admin_user

The Replication Agent administrator login name.

Default	sa
Value	A valid user name on the Replication Agent host machine.
Comments	<ul style="list-style-type: none">• The value of the <code>ltm_admin_user</code> parameter is the user name authorized to log in to the Replication Agent.• To change the value of the <code>ltm_admin_user</code> parameter, use the <code>ra_set_login</code> command.• If you change the value of the <code>ltm_admin_user</code> parameter with the <code>ra_set_login</code> command, the new value is recorded in the configuration file immediately. However, you must shut down and restart the Replication Agent instance to make the new administrator name take effect.• After you change the value of the <code>ltm_admin_user</code> parameter with <code>ra_set_login</code>, you must use the new administrator name the next time you log in to the Replication Agent.

max_ops_per_scan

The max_ops_per_scan parameter is deprecated and only provided for backward compatibility. Changes made to the max_ops_per_scan parameter will not affect Replication Agent behavior.

Default	1000
Values	An integer from 25 to $2^{31} - 1$.
See also	lr_read_buffer_size configuration parameter.

pdb_archive_path

Identifies the directory path where the Replication Agent expects to find archived redo log files (For Oracle) or archived transaction log files (for UDB). When archived redo log files or archived transaction log files are stored in the file system, the configuration parameter is set to a file system path.

Note This parameter is available only for Oracle and UDB.

Default	<not configured>
Values	For Oracle, a valid directory path on the machine hosting the Replication Agent that points to a location where Oracle places the archived redo log files. For example,

```
ORACLE_HOME\oradata\orcl\archive
```

Archive logs stored in and managed by ASM are owned by the corresponding unique Oracle database name. If the Oracle database name differs from the global unique database name, you must set pdb_archive_path to both the name of the ASM disk group and the globally unique name of the database in which the archive logs are stored:

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

You can also set pdb_truncate_xlog to manually remove archive logs. Set the pdb_archive_path to the ASM disk group name, and precede the archive logs to be manually removed with a plus "+" sign.

For UDB, a valid directory path on disk, as defined by the UDB LOGARCHMETH1 or LOGARCHMETH2 database configuration parameter, when it is configured to DISK. For example,

```
<path>
```

Comments	<ul style="list-style-type: none"> For Oracle, you must set <code>pdb_archive_path</code> when configuration parameter <code>pdb_include_archives</code> is set to <code>true</code>, and you must set it to a valid location before the Replication Agent can be placed in a <i>Replicating</i> state. If the Replication Agent cannot find an expected log record in the Oracle online redo logs, the Replication Agent will search this directory for the archived log file containing the required record. For UDB, <code>pdb_archive_path</code> must be configured before <code>pdb_archive_remove</code> is enabled (set to <code>true</code>). When <code>pdb_archive_remove</code> parameter is set to <code>true</code>, Replication Agent for UDB deletes the archived log files from the location specified by <code>pdb_archive_path</code>.
See also	<code>pdb_archive_remove</code> , and <code>pdb_include_archives</code> configuration parameters.

pdb_archive_remove

Enables or disables the removal of Oracle archived redo log files or UDB archived transaction logs from the path specified by `pdb_archive_path`.

Note This parameter is available only for Oracle and UDB.

Default	<code>false</code>
Values	<p><code>true</code> – Allows the removal of archived Oracle redo log and UDB archived transaction log files from the path specified by <code>pdb_archive_path</code>. Removal occurs based on the execution of command <code>pdb_truncate_xlog</code>, or the timing of automatic truncation based on parameters <code>truncation_type</code> and <code>truncation_interval</code>.</p> <p><code>false</code> – Disables the removal of archived Oracle redo log files or UDB archived transaction log files.</p>
Comments	<ul style="list-style-type: none"> For Oracle, set this configuration to <code>true</code> when the path specified by <code>pdb_archive_path</code> is established solely for Replication support, and automatic removal of unneeded archived log files is desired. For UDB, set this configuration to <code>true</code> when UDB is configured to archive logs by setting <code>LOGARCHMETH1</code> configuration parameter to <code>DISK:<path></code>, and the automatic removal of archived log files is desired.

- If the path specified by `pdb_archive_path` is shared by other processes, or the removal of archived log files is expected to be performed by processes other than the Replication Agent, this parameter should be `false`.
- Configuration parameters `truncation_type` and `truncation_interval`, and command `pdb_truncate_xlog` have no impact when this configuration parameter is set to `false`.

See also

`pdb_archive_path`, `truncation_type`, and `truncation_interval` configuration parameters, and the `pdb_truncate_xlog` command.

`pdb_auto_create_repdefs`

Configures Replication Agent to automatically create replication definitions at Replication Server when a table or procedure is marked for replication, after initialization. To improve performance, `pdb_auto_create_repdefs` is ignored during initialization.

Note This parameter is available only for Oracle and Microsoft SQL Server.

Note The `pdb_xlog create` command no longer checks the setting of `pdb_auto_create_repdefs` during initialization. To create replication definitions for all marked tables after executing `pdb_xlog create`, execute command `rs_create_repdef all`.

Default

`false`

Values

`true` – Replication Agent automatically creates replication definitions at the Replication Server when tables or procedures are marked after initialization.

`false` – no replication definitions are created when tables or procedures are marked.

Comments

Note Replication Agent always assumes that a database replication definition exists for the primary database.

- The table and procedure replication definitions that Replication Agent creates assume that a database level replication definition for the primary database already exists at Replication Server.

All replication definitions created by this command include the `send standby` clause, which means the replication definition will only be used by Replication Server if there is a database level replication definition or the primary Replication Server connection is for a warm standby configuration. The replication definition created by `rs_create_repdef` can not be individually subscribed to. If you do not wish to have a database level replication definition, or warm standby configuration, you must use a different tool, or create replication definitions manually, and not use `rs_create_repdef`.

- Replication definitions created by `rs_create_repdef` will always define the datatypes using available user defined datatypes that are installed in Replication Server. This means that customers using `rs_create_repdef` should not set Replication Agent configuration parameter `pdb_convert_datetime` to true, as doing so converts date and timestamp datatypes to Sybase format, instead of UDD format.
- If this parameter is set to true and when `pdb_setrepproc` is invoked to mark a procedure or procedures, a replication definition is created at Replication Server for each procedure that gets marked for replication.
- If this parameter is set to true and when `pdb_setreptable` is invoked to mark a table or tables, a replication definition is created at Replication Server for each table that gets marked for replication.
- If this parameter is set to true and when `pdb_setreptable` is invoked to unmark a table or tables, the replication definition is dropped at Replication Server for each table that gets unmarked for replication.
- If this parameter is set to true and when `pdb_setrepproc` is invoked to unmark a procedure or procedures, a replication definition is dropped at Replication Server for each procedure that gets unmarked for replication if this parameter is set to true.
- The following applies to replication definition table and procedure names:
 - All non-alphanumeric characters and spaces are removed and are not part of the table or procedure name.
 - Underscores are kept as part of the name even though they are non-alphanumeric characters.
 - Periods are replaced with underscores.

- Replication definition names for tables always begin with the prefix “*ra\$*,” followed by a unique alphanumeric identifier (maximum of 8 characters), and ending with a table or object name. For example, for a replicate name of “My Table,” the resulting replication definition name is “*ra\$0x7952_mytable*.” For an especially long replicate name of “mytable89012345678901234567890” (30 characters), the resulting replication definition name is “*ra\$0x7952_mytable8901234567890*” (30 or 255 characters maximum, depending on whether or not the `pdb_support_large_identifier` configuration parameter is set).
- Replication definition names for procedures are the same name as the procedure.

pdb_automark_tables

Determines if the Replication Agent automatically marks tables for replication during DDL replication.

Default

true

Values

true – user tables are automatically marked during DDL replication.

false – user tables are not automatically marked during DDL replication. They must always be marked using the `pdb_setreptable` command (default).

Comments

- The default value for `pdb_automark_tables` is set to true when a Replication Agent instance is created. In this default setting, all user tables (those whose owners are not contained in the `owner_filter_list`) are marked for replication when the `pdb_xlog` command is executed with the `init` keyword. In addition, when replication of DDL commands is enabled (`pdb_setrepddl` setting is enabled by default), any create table command for a user table (those whose owners are not contained in the `owner_filter_list`) is automatically marked for replication. If automatic marking of tables is not desired, this configuration parameter value should be changed to false.
- Automatic marking of new tables (those created in the primary database with the create table command) will only occur when replication of DDL commands is enabled (`pdb_setrepddl` is set to enable) and the table is a user table (those whose owners are not contained in the `owner_filter_list`) and `pdb_automark_tables` is set to true. Modifying the `owner_filter_list` after the initialization may cause inconsistencies.

- Tables are automatically unmarked for replication when a drop table command issued at the primary and is recorded in the transaction log, regardless of the settings of `pdb_setrepddl` or `pdb_automark_tables`. This is due to the fact that a dropped table cannot be replicated from.
- Automatic marking of user tables is independent of manual marking of tables using the `pdb_setreptable` command. In other words, you can always mark or unmark individual or all tables for replication using the `pdb_setreptable` command, regardless of the setting of `pdb_automark_tables`.

pdb_auto_run_scripts

Determines whether Replication Agent automatically runs scripts (for transaction log creation and removal, and object marking and unmarking) at the primary database.

Default

true

Values

true – Replication Agent automatically runs scripts.

false – Replication Agent generates and saves the scripts, but it does not automatically run them at the primary database.

Comments

- When the `pdb_xlog` command is invoked to create or remove the transaction log, Replication Agent generates a script to create or remove the transaction log base objects.
- When the `pdb_setrepproc` command is invoked to mark or unmark an object in the primary database, Replication Agent generates a script to create or remove the transaction log objects necessary for object marking.
- Replication Agent always saves the scripts in a file. Log creation and removal scripts are saved in files named *partinit.sql* and *partdeinit.sql*. Object marking and unmarking scripts are saved in files named *partmark.sql* and *unmark.sql*.
- When the `pdb_auto_run_scripts` parameter is set to false, the scripts are created but no action is taken. This allows you to review the scripts to see what action will be taken before execution. You cannot execute the scripts. You must set `pdb_auto_run_scripts` parameter back to true and re-execute the command to have the desired action take place.

- As described above for the `pdb_xlog` and `pdb_setreproc` commands, Oracle and Microsoft SQL Server creates the *partinit*, *partdeinit*, *partmark*, and *partunmark* scripts. For Oracle and Microsoft SQL Server, these scripts can *not* be executed (since they do not update the RASD) and are for informational purposes only.
- This parameter must be set to true for initialization to occur.

pdb_commit_generation

Determines whether or not Replication Agent generates commit log records at the primary database.

Note This parameter is available only for Oracle and only used for Oracle RAC.

Default	false
Values	true – Replication Agent generates commit log records at the interval specified by <code>pdb_commit_interval</code> . false – Replication Agent does not generate commit log records.
Comments	<ul style="list-style-type: none">• If <code>pdb_commit_generation</code> is set to true, Replication Agent generates commit log records only if it is in the <i>Replicating</i> state.• Generating commit log records lowers replication latency in Oracle RAC replication environments when cluster nodes are inactive but has no effect on replication latency for non-clustered replication environments. The <code>pds_username</code> must have the <code>execute on DBMS_FLASHBACK</code> permission granted for Replication Agent to generate commit log records.• The Replication Agent “Average number of seconds between log record checkpoints” statistic can be used to confirm that commit frequency results are in the desired range.
See also	<code>pdb_commit_interval</code> configuration parameter.

pdb_commit_interval

Determines the interval in seconds at which Replication Agent generates commit log records.

Note This parameter is available only for Oracle and only used for Oracle RAC.

Default	1
Value	An integer from 1 to 5.
Comments	<ul style="list-style-type: none"> • The <code>pdb_commit_interval</code> parameter is relevant only if <code>pdb_commit_generation</code> is set to true. • The Replication Agent “Average number of seconds between log record checkpoints” statistic should show a value less than or equal to the value of <code>pdb_commit_interval</code> when <code>pdb_commit_generation</code> is set to true.
See also	<code>pdb_commit_generation</code> configuration parameter.

pdb_convert_datetime

Determines whether Replication Agent converts non-Sybase temporal datatypes to the Sybase datetime format.

Default	true
Values	<p>true – Replication Agent converts all data in the primary database native date/time datatypes to the Sybase datetime format.</p> <p>false – Replication Agent replicates data in the primary database native datetime datatypes as character strings.</p>
Comments	<ul style="list-style-type: none"> • Replication definitions created by the <code>rs_create_repdef</code> command always define the datatypes using available user defined datatypes (UDDs) that are installed in Replication Server. If you use the <code>rs_create_repdef</code> command, do not set the Replication Agent configuration parameter <code>pdb_convert_datetime</code> to true, as doing so converts date and timestamp datatypes to Sybase format, instead of UDD format.

- The `pdb_convert_datetime` parameter is provided for backward compatibility with previous versions of Replication Agents and Replication Server. If you use Replication Server version 12.0 or later, Sybase recommends that you use the Replication Server heterogeneous datatype support (HDS) feature for all datatype conversion and translation.
- Replication Agent checks the value of the `pdb_convert_datetime` parameter at the time an object is marked for replication. Transaction log objects that support replication of the marked object are constructed to provide the desired date format.

If you change the value of the `pdb_convert_datetime` parameter after an object is marked, it has no effect on the marked object. To change the datetime datatype conversion for a marked object, you must unmark the object, change the value of the `pdb_convert_datetime` parameter, then remark the object.

- For log-based Replication Agents, the conversion takes place after the log records have been read and before LTL is generated to send to the Replication Server.
- Any missing component in the non-Sybase date/time datatype format is treated as an implied 0 (zero) when it is converted to the Sybase datetime format.
- When the value of the `pdb_convert_datetime` parameter is true, the replication definition for each table should specify that the declared datatype for all date/time columns is datetime.
- If the value of the `pdb_convert_datetime` parameter is false, the Replication Agent sends date/time data to the primary Replication Server as default-sized character strings. The default character string size varies by database and datatype:
 - DB2 Universal Database: DATE = char(10), TIME = char(8), TIMESTAMP = char(26)
 - Microsoft SQL Server: datetime or smalldatetime = char(23), timestamp = binary(8)
 - Oracle: DATE = char(19)
- Set the value of the `pdb_convert_datetime` parameter to true if *all* date/time values replicated from the primary database will be replicated as the Sybase datetime datatype.
- `pdb_convert_datetime` must be false if a table containing replicated LOB columns has datetime datatype in the primary key.

- Replication Agent date/time datatype conversion does not work with LOB column replication, unless either of the following conditions exist (these conditions are *not* required for Oracle and Microsoft SQL Server):
 - There are no date/time columns in the tables that have LOB column replication enabled, or
 - The primary keys in tables that have LOB column replication enabled do not contain date/time datatypes.

Otherwise, if you use the `pdb_setrepcol` command to enable LOB column replication, you must set the value of the `pdb_convert_datetime` parameter to `false`.

The Replication Agent for UDB is the only Replication Agent that must query LOB data directly from the primary database. To successfully query a primary database table for a LOB column value, any date column value must retain the primary database format and structure for the date value to appear correctly in the query. The format and structure for the date value cannot be converted to the Sybase datetime format.

- Sybase recommends that you set the value of the `pdb_convert_datetime` parameter to `false` for better Replication Agent throughput performance and optimal datatype handling.

pdb_dflt_column_repl

	Determines whether LOB column replication is enabled by default when tables are marked.
Default	false
Values	<p>true – LOB column replication is enabled by default (automatically) when tables are marked.</p> <p>false – LOB column replication is disabled by default when tables are marked.</p>
Comments	<ul style="list-style-type: none"> • If the value of the <code>pdb_dflt_column_repl</code> parameter is <code>false</code> when a table is marked for replication, no transactions that affect LOB columns in the table can be replicated until replication is explicitly enabled with the <code>pdb_setrepcol</code> command. • You can use the <code>pdb_setrepcol</code> command to enable or disable replication for all LOB columns in all marked tables at once.

- When replication is disabled for a LOB column, any part of an operation that affects that column will not be recorded in the transaction log, even if the operation also affects other columns for which replication is enabled.

pdb_dflt_object_repl

Determines whether replication is enabled by default when objects (tables or stored procedures) are marked.

Note This parameter is available only for Oracle and UDB.

Default	true
Values	true – enables replication by default (automatically) when objects are marked. false – disables replication by default when objects are marked.
Comments	<ul style="list-style-type: none">• If the value of the <code>pdb_dflt_object_repl</code> parameter is false when a table is marked for replication, no transactions can be replicated from that table until replication is explicitly enabled with the <code>pdb_setreptable</code> command.• If the value of the <code>pdb_dflt_object_repl</code> parameter is false when a stored procedure is marked for replication, no invocations of that stored procedure can be replicated until replication is explicitly enabled with the <code>pdb_setrepproc</code> command.• You can use the <code>pdb_setrepproc</code> or <code>pdb_setreptable</code> command to enable or disable replication for all marked stored procedures or tables at once.• When replication is disabled for a table, no operations that affect that table will be recorded in the transaction log.• When replication is disabled for a stored procedure, no invocations of that stored procedure are recorded in the transaction log.• For Microsoft SQL Server, the <code>pdb_setrepproc</code> command ignores the <code>pdb_dflt_object_repl</code> parameter. As a result, this command will always enable replication by default when objects are marked.

pdb_ignore_unsupported_anydata

Determines whether or not Replication Agent ignores data of unsupported datatypes stored in columns of type ANYDATA.

Note This parameter is available only for Oracle.

Default	false
Values	<p>true – Replication Agent ignores data of unsupported datatypes stored in columns of type ANYDATA, sending no data for these columns to Replication Server.</p> <p>false – Replication Agent sends the string <code>type not supported</code> to Replication Server for data of unsupported datatypes stored in columns of type ANYDATA. This causes Replication Server failure, after which corrections must be made at Replication Server or the replicate database for each table row containing unsupported data.</p>
Comments	<ul style="list-style-type: none"> • Replication Agent does not replicate data of the following Oracle datatypes stored in a column of type ANYDATA: <ul style="list-style-type: none"> • BFILE • NESTED TABLE • REF • UROWID • VARRAY • Replication Agent checks the setting of <code>pdb_ignore_unsupported_anydata</code> only when an object is marked for replication. To reset <code>pdb_ignore_unsupported_anydata</code> and change Replication Agent behavior for a marked object, you must unmark the object before you reset <code>pdb_ignore_unsupported_anydata</code>. The change to <code>pdb_ignore_unsupported_anydata</code> will take effect once you remark the object. • If <code>pdb_ignore_unsupported_anydata</code> is set to true and the replicate table has a default column value for the corresponding ANYDATA columns, the primary and replicate tables will be inconsistent. If the replicate table has no default column value for the corresponding ANYDATA columns, Replication Server will fail, even though <code>pdb_ignore_unsupported_anydata</code> is set to true.

- To recover from a Replication Server failure caused by data of unsupported datatypes found in an ANYDATA column, do one of the following:
 - Alter the corresponding replicate table so that the table has a default value for columns of type ANYDATA. This can only be done if `pdb_ignore_unsupported_anydata` is set to true.
 - Create a trigger in the corresponding replicate table to provide a default value for columns of type ANYDATA. This can only be done if `pdb_ignore_unsupported_anydata` is set to true.
 - Customize a Replication Server function string to provide a default value for columns of type ANYDATA in the replicate table.

pdb_include_archives

Enables or disables the use of Oracle archive log files.

Note This parameter is available only for Oracle.

Default

true

Values

true – allows reading of the archived Oracle redo log files from the path specified by `pdb_archive_path`. The configuration of Oracle automatic archiving is supported under this mode. Use `pdb_archive_remove` to remove old archives logs that are no longer needed to support replication.

false – only online redo logs files are read. Oracle automatic archiving must be disabled. The Replication Agent executes Oracle archive commands to archive the redo logs once they are no longer needed for replication.

Comments

- Set the configuration to true when use of archive logs is preferred or when Oracle must be configured to perform automatic archiving. Set this value to false if accessing only the on-line redo logs is preferred.
- Set this value to false if using only the online redo logs is preferred.

See also

`pdb_archive_path`, `truncation_type`, and `truncation_interval` configuration parameters.

pdb_skip_missing_user

Determines whether or not Replication Agent skips the processing of any command for which there is no matching session or user information.

Note This parameter is available only for Oracle.

Default	false
Values	<p>true – Replication Agent skips the processing of any command for which there is no matching session or user information. A message is logged identifying the skipped operation.</p> <p>false – Replication Agent continues to process any command for which there is no matching session or user information. The default session user is sent as “missing.”</p>

pdb_support_large_identifier

To support replication of large identifiers up to 255 characters in length with Replication Server 12.6 and later.

Default	false
Value	<p>true – objects containing large identifiers may be marked for replication.</p> <p>false – objects containing large identifiers may <i>not</i> be marked for replication.</p>
Comments	<ul style="list-style-type: none"> • If <code>pdb_support_large_identifier</code> value is false, when an object (Table/Procedure/Function) is being marked for replication, the object is checked for any identifiers that are longer than 30 characters. An error is returned and the object is not marked for replication if the object has identifiers longer than 30 characters. • This parameter may be set to true if the Replication Server being used is at version 12.6 or later and the replicate database can support large identifiers. • When <code>pdb_support_large_identifier</code> is set to true, objects being marked for replication are not checked for identifiers longer than 30 characters.

pdb_timezone_file

Specifies the file to read at Replication Agent initialization to obtain Oracle time zone information.

Note This parameter is available only for Oracle.

Default	<not configured>
Value	A valid path to the Oracle time zone file including the <i>time zone</i> file name.
Comments	<ul style="list-style-type: none"> If the value is not specified, it will default to the Oracle installation's <i>oracore/zoneinfo/timezone.dat</i> file. For example, <pre> \$ORACLE_HOME/oracore/zoneinfo/timezone.dat </pre> The <i>timezone</i> file specified must be for the same release and platform as the primary Oracle database. For example, an Oracle 9i <i>timezone</i> file is not compatible with an Oracle 10g primary database, and a Windows <i>timezone</i> file is not compatible with UNIX.

pdb_xlog_device

The primary database device on which Replication Agent transaction log objects are created.

Default	NULL
Value	A valid primary database device name or NULL.
Comments	<ul style="list-style-type: none"> The value of the <i>pdb_xlog_device</i> parameter is the device specification of the primary database device to be used in SQL scripts generated by the Replication Agent to create transaction log objects. The <i>pdb_xlog_device</i> parameter allows you to specify a single device on which all Replication Agent transaction log objects will be created, even if the database uses multiple devices. If the value of the <i>pdb_xlog_device</i> parameter is NULL, no device is specified in the SQL create statements, and Replication Agent transaction log objects are placed on the primary data server's system-defined default device.

pdb_xlog_prefix

The prefix string used in database object names to identify Replication Agent transaction log objects.

Default	ra_
Value	A character string of 1 to 3 characters.
Comments	<ul style="list-style-type: none"> • When Replication Agent generates database object names for transaction log components in the primary database, it uses the value of the <code>pdb_xlog_prefix</code> parameter as an object name prefix. • Replication Agent uses the value of the <code>pdb_xlog_prefix</code> parameter to recognize its transaction log objects in the primary database. Therefore, if you change the value of the <code>pdb_xlog_prefix</code> parameter after the transaction log objects are created, Replication Agent will not be able to find its transaction log objects. • The value of the <code>pdb_xlog_prefix_chars</code> parameter specifies the non-alphabetic characters that can be used in the prefix string.

pdb_xlog_prefix_chars

Non-alphabetic characters that are allowed in the database object name prefix string that identifies Replication Agent transaction log objects.

Default	_ (underscore)
Value	A string of characters with no separators.
Comments	<ul style="list-style-type: none"> • The default value of the <code>pdb_xlog_prefix_chars</code> parameter depends on the type of primary database that the Replication Agent instance was created for. The default value is based on the standard, non-alphabetic characters allowed by each non-Sybase database. • When you set or change the value of the <code>pdb_xlog_prefix_chars</code> parameter, the new value replaces any existing value; it does not add or append the new value to a previous value. • When you use the <code>ra_config</code> command to set the value of the <code>pdb_xlog_prefix</code> parameter, any non-alphabetic characters specified on the command line are validated against the value of the <code>pdb_xlog_prefix_chars</code> parameter. • Alphabetic characters a-z are always valid in the <code>pdb_xlog_prefix</code> parameter, and they need not be specified.

- Replication Agent does not support delimited names for transaction log objects, so you cannot use a space character in the value of the `pdb_xlog_prefix` parameter.
- The value you specify for the `pdb_xlog_prefix_chars` parameter is not validated. There are no restrictions on the characters you can include.

Note The primary data server may restrict the characters used in certain positions in a database object name. Refer to the documentation for your primary data server for more information.

`pds_connection_type`

The type of connectivity driver used on the primary database connection.

Default

One of the following values is set automatically when the Replication Agent instance is created.

Values

MSSQLJDBC – Replication Agent uses the Microsoft SQL Server JDBC driver to connect to the primary Microsoft SQL Server database.

ORAJDBC – Replication Agent uses the Oracle JDBC driver to connect to the primary Oracle database.

UDBODBC – Replication Agent uses the DB2 Universal Database ODBC driver to connect to the primary database in DB2 Universal Database.

Comments

- The value of the `pds_connection_type` parameter is set automatically at the time a Replication Agent instance is created. The specific value depends on the type of Replication Agent instance created.

Note Sybase recommends that you do *not* change the default value of the `pds_connection_type` parameter.

- The value of the `pds_connection_type` parameter determines which of several other Replication Agent configuration parameters related to the primary database connection must also have values specified.
 - MSSQLJDBC requires corresponding values for the following parameters:
 - `pds_server_name`
 - `pds_port_number`

- pds_database_name
- UDBODBC requires corresponding values for the following parameters:
 - pds_database_name
 - pds_datasource_name
- ORAJDBC requires corresponding values for the following parameters:
 - pds_host_name
 - pds_port_number
 - pds_database_name
- If using the *tnsnames.ora* file the following parameters are required:
 - pds_tns_filename
 - pds_tns_connection
- The value of the pds_connection_type parameter is automatically when a Replication Agent instance is created.

pds_dac_port_number

The dedicated administrative connection (DAC) port number for Microsoft SQL Server error log.

Note This parameter is available only for Microsoft SQL Server.

Default	1111
Value	A valid port number for Microsoft SQL Server.
Comments	<ul style="list-style-type: none"> • This parameter is required when executing the <code>pdb_xlog</code> init command against a Microsoft SQL Server for the first time. It is required to support installation of changes to the <i>mssqlsystemresource</i> database. • For information regarding the use of the <code>pds_dac_port_number</code> parameter, see the “Microsoft SQL Server” chapter of the <i>Replication Agent Primary Database Guide</i>, under “Initialization of the primary data server and Replication Agent” and “First-time initialization.”
See also	<code>pdb_xlog</code> configuration parameter.

pds_database_name

The name of the primary database.

Default <not_configured>

Value A valid database name.

Note You cannot set pds_database_name if pds_tns_connection is set.

Comments

- The value of the pds_database_name parameter is the name of the primary database on the primary data server.

Note Some primary data servers may not support multiple databases in a single instance of the data server. In that case, the value of the pds_database_name parameter should be the name of the data server instance.

- See the *Replication Agent Administration Guide* for more information about setting up Replication Agent connection configuration parameters.

pds_datasource_name

The data source name (DSN) specified for the ODBC driver used on the primary database connection.

Note This parameter is available only for Microsoft SQL Server and UDB.

Default <not_configured>

Value A valid ODBC data source name.

Comments

- The value of the pds_datasource_name parameter is the data source name (DSN) of the ODBC driver on the primary database connection.
- If the value of the pds_connection_type parameter is UDBODBC, the value of the pds_datasource_name parameter must be the database alias of the primary database in the DB2 Universal Database server.
- See the *Replication Agent Administration Guide* for more information about setting up Replication Agent connection configuration parameters.

pds_host_name

The name of the primary data server host machine.

Default <not_configured>

Value A valid host name.

Note You cannot set pds_host_name if the pds_tns_connection is set.

Comments

- The value of the pds_host_name parameter is the network name of the host machine on which the primary data server resides.
- See the *Replication Agent Administration Guide* for more information about setting up Replication Agent connection configuration parameters.

pds_integrated_security

Determines if the Replication Agent should use Windows authentication when connecting to the primary SQL Server.

Note This parameter is available only for Microsoft SQL Server.

Default false

Value true – Specifies that Replication Agent should connect to the primary SQL Server using Windows authentication.

false – Specifies that Replication Agent should connect to the primary SQL Server using SQL Server authentication (default).

Comments

- The default value for pds_integrated_security is set to false when a Replication Agent instance is created. In this default setting, you must configure the pds_password parameter, and the primary SQL Server must be configured to allow SQL Server authentication.
- Set this value to true to have the Replication Agent connect to the primary SQL Server using Windows authentication. The Windows environment and the primary SQL Server must be configured to use Windows authentication. For more information, see the Microsoft SQL Server section in the *Replication Agent Primary Database Guide*.

pds_password

The password that Replication Agent uses for primary data server access.

Default "" (empty string)

Value A valid password.

- Comments
- The value of the pds_password parameter is the password for the user login name that the Replication Agent uses to access the primary data server.
 - The value of the pds_password parameter is encrypted in the Replication Agent instance configuration file.
 - See the *Replication Agent Administration Guide* for more information about setting up Replication Agent connection configuration parameters.

pds_port_number

The client port number for the primary data server.

Default 1111

Value A valid port number on the primary data server host machine.

Note You cannot set pds_port_number if pds_tns_connection is set.

- Comments
- The value of the pds_port_number parameter is the client port number for the primary data server.
 - See the *Replication Agent Administration Guide* for more information about setting up Replication Agent connection configuration parameters.

pds_retry_count

The number of times the Replication Agent tries to establish a connection to the primary database.

Default 5

Value An integer from 0 to 2,147,483,647.

- Comments
- The value of the `pds_retry_count` parameter is the number of times that Replication Agent will try to establish a network connection to the primary database after a connection failure.
 - Sybase recommends a setting of 5 for this parameter.
 - See the *Replication Agent Administration Guide* for more information about setting up Replication Agent connection configuration parameters.

pds_retry_timeout

The number of seconds the Replication Agent waits between retry attempts to connect to the primary database.

Default 10

Value An integer from 0 to 3600.

- Comments
- The value of the `pds_retry_timeout` parameter is the number of seconds that the Replication Agent will wait between retry attempts to establish a network connection to the primary database after a connection failure.
 - See the *Replication Agent Administration Guide* for more information about setting up Replication Agent connection configuration parameters.

pds_server_name

The server name of the primary data server.

Default <not_configured>

Value A valid server name.

- Comments
- The value of the `pds_server_name` parameter is the server name of the primary data server.
 - See the *Replication Agent Administration Guide* for more information about setting up Replication Agent connection configuration parameters.

pds_tns_connection

The Oracle connection name that identifies the primary database connection in the Oracle *tnsnames.ora* file.

Note This parameter is available only for Oracle.

Default	<not configured>
Value	A valid primary database connection name from the Oracle <i>tnsnames.ora</i> file specified by the <code>pds_tns_filename</code> configuration parameter.
Comments	<ul style="list-style-type: none">Setting of the configuration parameter overrides settings of the configuration parameters <code>pds_host_name</code>, <code>pds_database_name</code>, and <code>pds_port_number</code>.This configuration parameter is required when the Oracle data server instance to be replicated is part of a Real Application Cluster (RAC) configuration.
See also	<code>pds_tns_filename</code> configuration parameter.

pds_tns_filename

The fully-qualified file name identifying the Oracle *tnsnames.ora* file that contains connection parameters for the primary Oracle data server.

Note This parameter is available only for Oracle.

Default	<not configured>
Value	A valid Oracle <i>tnsnames.ora</i> file that contains the connection parameters to the primary Oracle data server. This file normally resides in the <code>ORACLE_HOME\network\admin</code> directory.
Comments	<ul style="list-style-type: none">When configured, Replication Agent will use the connection information specified in the <code>interfaces</code> file to connect to the primary database and the <code>pds_host_name</code> and the <code>pds_port_number</code> are ignored. The <code>pds_tns_connection</code> name should be configured to the entry name in the Sybase <code>interfaces</code> file when <code>pds_tns_filename</code> is configured.

- Setting of the configuration parameter is required when the Oracle data server instance to be replicated is part of a Real Application Clusters (RAC) configuration.

Warning! The Replication Agent process must have *read* permission to this file. Access failures will prevent the Replication Agent from connecting to the Oracle server.

See also `pds_tns_connection` configuration parameter.

pds_username

The user login name that the Replication Agent uses for primary data server access.

Default `<not_configured>`

Value A valid user name.

- Comments
- The value of the `pds_username` parameter is the login name that the Replication Agent uses to log in to the primary data server.
This login name must be defined in the primary data server, with appropriate privileges or authority in the primary database.
 - The Replication Agent uses this login to access primary database objects and to create, remove, and manage its transaction log objects in the primary database.
 - See the *Replication Agent Administration Guide* for more information about setting up Replication Agent connection configuration parameters.

ra_retry_count

The number of times the Replication Agent attempts to restart replication after a failure.

Default 2

Value An integer greater than 0.

- Comments
- The value of the `ra_retry_count` parameter is the number of times that the Log Transfer Manager component will try to get the Replication Agent instance back into *Replicating* state after a failure or error causes the instance to go to *Replication Down* state.
 - When a network connection fails, the Replication Agent attempts to re-establish the connection, using the values stored in its connection configuration parameters for that connection.
 - If the Replication Agent is unable to re-establish a connection after the number of retries specified in the `pds_retry_count` or `rs_retry_count` parameter, then the Replication Agent instance goes to *Replication Down* state and the Log Transfer Manager component attempts to return the Replication Agent instance to *Replicating* state, based on the settings of the `ra_retry_count` and `ra_retry_timeout` parameters.

ra_retry_timeout

The number of seconds the Replication Agent waits between attempts to restart replication after a failure.

Default 10

Value An integer greater than 0.

Comment The value of the `ra_retry_timeout` parameter is the number of seconds that the Log Transfer Manager component will wait between its attempts to get the Replication Agent instance back into *Replicating* state after a failure causes the instance to go to *Replication Down* state.

rasd_backup_dir

The directory path for Replication Agent System Database (RASD) backup files.

Note This parameter is available only for Oracle and Microsoft SQL Server.

Default The path to the RASD *backup* directory created automatically when the Replication Agent instance was created. For example:

- On Microsoft Windows platforms:

```
%SYBASE%\RAX-15_5\inst_name\repository\backup
```

where:

- `%SYBASE%` is the path to the Replication Agent installation directory.
- `inst_name` is the name of the Replication Agent instance.
- On UNIX platforms:

```
$(SYBASE)/RAX-15_5/inst_name/repository/backup
```

where:

- `$(SYBASE)` is the path to the Replication Agent installation directory.
- `inst_name` is the name of the Replication Agent instance.

Value

A valid path on the Replication Agent host machine.

Comments

- When you create a Replication Agent instance, an RASD *backup* directory is created automatically as part of the instance directory structure. The default value of the `rasd_backup_dir` parameter points to that directory.
- If you specify any valid path as the value of the `rasd_backup_dir` parameter, Replication Agent places its RASD backup files in that directory during RASD backup operations, and it looks in that directory for the RASD backup files during restore operations.

rasd_database

The directory path for the Replication Agent System Database (RASD) database file.

Note This parameter is available only for Oracle and Microsoft SQL Server.

Default

The path to the RASD database file created automatically when the Replication Agent instance was created. For example:

- On Microsoft Windows platforms:

```
%SYBASE%\RAX-15_5\inst_name\repository\inst_name.db
```

where:

- `%SYBASE%` is the path to the Replication Agent installation directory.
- `inst_name` is the name of the Replication Agent instance.

- On UNIX platforms:

`$(SYBASE)/RAX-15_5/inst_name/repository/inst_name.db`

where:

- `$(SYBASE)` is the path to the Replication Agent installation directory.
- `inst_name` is the name of the Replication Agent instance.

Value A valid path and RASD database file name on the Replication Agent host machine.

Comments

- When you create a Replication Agent instance, the *repository* directory and the RASD database file are created automatically. The default value of the `rasd_database` parameter points to the RASD database file in that directory.
- If you specify any valid path and RASD database file name as the value of the `rasd_database` parameter, the Replication Agent instance looks in that directory for its RASD database file, with the file name you specified.

rasd_mirror_tran_log

Enables or disables Replication Agent System Database (RASD) transaction log mirroring.

Note This parameter is available only for Oracle and Microsoft SQL Server.

Default false

Values

true – enables mirroring the RASD transaction log to another file.

false – disables mirroring of the RASD transaction log.

Comment Setting the value of the `rasd_mirror_tran_log` parameter to true provides additional recovery options in the event of a device failure on the Replication Agent host machine.

rasd_trace_log_dir

The directory path for the Replication Agent System Database (RASD) trace log file.

Note This parameter is available only for Oracle and Microsoft SQL Server.

Default

The path to the *repository* directory created automatically when the Replication Agent instance was created. For example:

- On Microsoft Windows platforms:

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

where:

- *%SYBASE%* is the path to the Replication Agent installation directory.
- *inst_name* is the name of the Replication Agent instance.
- On UNIX platforms:

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

where:

- *\$SYBASE* is the path to the Replication Agent installation directory.
- *inst_name* is the name of the Replication Agent instance.

Value

A valid path on the Replication Agent host machine.

Comments

- When you create a Replication Agent instance, the *repository* directory is created automatically as part of the instance directory structure. The default value of the *rasd_trace_log_dir* parameter points to that directory.
- If you specify any valid path as the value of the *rasd_trace_log_dir* parameter, the Replication Agent instance writes its RASD trace log file in that directory.

rasd_tran_log

The directory path for the Replication Agent System Database (RASD) transaction log file.

Note This parameter is available only for Oracle and Microsoft SQL Server.

Default

The path to the RASD transaction log file created automatically when the Replication Agent instance was created. For example:

- On Microsoft Windows platforms:

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

where:

- `%SYBASE%` is the path to the Replication Agent installation directory.
- `inst_name` is the name of the Replication Agent instance.
- On UNIX platforms:

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

where:

- `$SYBASE` is the path to the Replication Agent installation directory.
- `inst_name` is the name of the Replication Agent instance.

Value

A valid path on the Replication Agent host machine.

Comments

- When you create a Replication Agent instance, the *repository* directory and RASD transaction log file are created automatically. The default value of the `rasd_tran_log` parameter points to that transaction log file.
- If you specify any valid path and RASD transaction log file name as the value of the `rasd_tran_log` parameter, the Replication Agent instance looks in that directory for its RASD transaction log file, with the name you specified.

rasd_tran_log_mirror

The directory path for the Replication Agent System Database (RASD) transaction log file mirror.

Note This parameter is available only for Oracle and Microsoft SQL Server.

Default

The path to the RASD transaction log file mirror in the *tran_log_mirror* directory created automatically when the Replication Agent instance was created. For example:

- On Microsoft Windows platforms:

```
%SYBASE%\RAX-15_5\inst_name\repository\tran_log_mirror\inst_name.log
```

where:

- *%SYBASE%* is the path to the Replication Agent installation directory.
- *inst_name* is the name of the Replication Agent instance.
- On UNIX platforms:

```
$SYBASE/RAX-15_5/inst_name/repository/tran_log_mirror/inst_name.log
```

where:

- *\$SYBASE* is the path to the Replication Agent installation directory.
- *inst_name* is the name of the Replication Agent instance.

Value

A valid path on the Replication Agent host machine.

Comment

If you specify any valid path and transaction log file name as the value of the *rasd_tran_log_mirror* parameter, the Replication Agent instance looks in that directory for its RASD transaction log file mirror, with the name you specified.

ra_standby

Determines whether or not Replication Agent functions in standby mode.

Note This parameter is available only for Oracle.

Default

false

Values

true – Replication Agent functions in standby mode.

	false – Replication Agent functions in normal mode.
Comment	<ul style="list-style-type: none"> • In standby mode, Replication Agent: <ul style="list-style-type: none"> • scans the transaction log and keeps the Replication Agent System Database (RASD) current. • does not send any Log Transfer Language (LTL) to Replication Server. • continues to perform log truncation. • To function in standby mode, Replication Agent should: <ul style="list-style-type: none"> • have the <code>rs_source_ds</code> and <code>rs_source_db</code> parameters configured as physical connections to Replication Server. • enable or disable the replication of DDL statements as desired using the <code>pdb_setrep_ddl</code> command. • set the <code>pdb_auto_create_repdefs</code>, <code>pdb_dflt_column_repl</code>, <code>pdb_dflt_object_repl</code>, and <code>pdb_automark_tables</code> parameters to true.
See also	<code>rs_source_ds</code> , <code>rs_source_db</code> , <code>pdb_auto_create_repdefs</code> , <code>pdb_dflt_column_repl</code> , <code>pdb_dflt_object_repl</code> , and <code>pdb_automark_tables</code> configuration parameters.

rman_enabled

The `rman_enabled` configuration parameter determines whether or not Replication Agent truncates old archive log files using the Oracle RMAN utility. This parameter can be used only for Oracle data servers.

Default	false
Values	<p>true – enables truncation of archive log files using the Oracle RMAN utility.</p> <p>false – disables truncation of archive log files using the Oracle RMAN utility.</p>
Comments	<ul style="list-style-type: none"> • If both the <code>rman_enabled</code> and <code>pdb_archive_remove</code> parameters are set to true, Replication Agent uses the Oracle RMAN utility to remove old archive redo log files from the path specified by the <code>pdb_archive_path</code> parameter. If the <code>rman_enabled</code> parameter is set to false but the <code>pdb_archive_remove</code> parameter is set to true, Replication Agent uses file system operations to remove old archive redo log files from the path specified by the <code>pdb_archive_path</code> parameter.

- When the `rman_enabled` parameter is set to true, the `ORACLE_HOME` environment variable must be set in the runtime context of the Replication Agent process, and the `rman_username` and `rman_password` parameters must be properly configured.
- The Oracle RMAN utility must be installed on the same machine as Replication Agent and must be compatible with the Oracle database that contains the archive log files being truncated.
- A change to the `rman_enabled` parameter will take effect only after Replication Agent is suspended and resumed.

See also `pdb_archive_remove`, `rman_password`, `rman_username` configuration parameters.

rman_password

The `rman_password` configuration parameter contains the password used with `rman_username` to connect to the Oracle RMAN utility. This parameter can be used only for Oracle data servers.

Default "" (empty string)

Value A valid password.

Comments

- Setting the `rman_password` configuration parameter is required only if `rman_enabled` is set to true.
- The value of the `rman_password` configuration parameter is encrypted in the configuration file for the Replication Agent instance.

See also `rman_enabled` and `rman_username` configuration parameters.

rman_username

The `rman_username` configuration parameter contains the login name used with `rman_password` to connect to the Oracle RMAN utility. This parameter can be used only for Oracle data servers.

Default <not configured>

Value A valid Oracle user name with sysdba privileges.

Comments

- Setting the `rman_username` configuration parameter is required only if `rman_enabled` is set to true.

- The value of `rman_username` is the login name that Replication Agent uses to connect to the Oracle RMAN utility and manage archive log files. This login name must be defined in the Oracle primary data server and have `sysdba` privileges. If a login name lacking `sysdba` privileges is set by `ra_config`, Replication Agent returns an error.

See also `rman_enabled` and `rman_password` configuration parameters.

rs_charset

The character set used in communication with the primary Replication Server.

The Replication Agent default character set must be set to match the primary database's character set. The value of the `rs_charset` parameter must be set to match Replication Server's character set. If they differ, Replication Agent will do character set conversion before sending data to Replication Server.

Note If Replication Agent can connect to Replication Server 15.0.1 or later, the `rs_charset` in Replication Agent is ignored and the `RS_charset` in Replication Server is used.

If the character set on your Replication Agent is different from the one on your primary database, you need to set the `RA_JAVA_DFLT_CHARSET` environment variable. The Replication Agent character set must be the same as that of the primary database. For more information on setting the `RA_JAVA_DFLT_CHARSET` environment variable, see Chapter 2 of the *Replication Agent Administration Guide*.

Note Setting this parameter to anything other than the character set of the primary Replication Server causes Replication Agent to do character set conversion before sending data to Replication Server. This will degrade the Replication Agent performance.

Default	Defaults to empty string ("").
Value	Any valid Sybase character set supported by the Java VM on the Replication Agent host machine.
Comments	<ul style="list-style-type: none">• Sybase recommends that you use the exact same value as that of the <code>RS_charset</code> parameter in the Replication Server configuration (<code>.cfg</code>) file which is located at: <code>\$SYBASE/REP-15_0/install/<instance>.cfg</code>. For example, <code>iso_1</code>.

- Sybase recommends that you configure the primary data server and primary Replication Server to use the same character set.

Note If `rs_charset` is not set at the time you try to resume replication, Replication Agent returns an error.

When the Replication Agent instance is created, the `rs_charset` parameter is set to its default value "" (empty string).

- If you specify a valid character set for the value of the `rs_charset` parameter, the Replication Agent instance sends replicated transaction data from the primary database to the primary Replication Server in that character set.
- If you do *not* specify a valid character set name for the value of the `rs_charset` parameter (including the default `rs_charset` value ""), the Replication Agent instance will not allow you to resume replication.
- If the values of the `rs_charset` and the system default character set are valid but *not* the same value, Replication Agent converts the replicated transaction data from the system-defined database character set to the Replication Server character set before sending it to the primary Replication Server.
- See the *Replication Agent Administration Guide* for more information about setting up Replication Agent connection configuration parameters.

rs_host_name

The name of the primary Replication Server host machine.

Default

<not_configured>

Value

A valid host name.

Comments

- The value of the `rs_host_name` parameter is the name of the host machine for the primary Replication Server.
- See the *Replication Agent Administration Guide* for more information about setting up Replication Agent connection configuration parameters.

rs_packet_size

	The network packet size on the connection to the primary Replication Server.
Default	2048
Value	An integer from 2048 to 65536.
Comments	<ul style="list-style-type: none">• The value of the <code>rs_packet_size</code> parameter is the maximum size (in bytes) of the network packets handled by the TCP/IP network protocol.• The Replication Agent <code>rs_packet_size</code> parameter is equivalent to the Replication Server <code>rs_packet_size</code> parameter.• When the network packet size is smaller, more packets must be processed to transmit a given amount of data to the Replication Server. When the network packet size is larger, more system resources are consumed to process the packets.• The optimum value of the <code>rs_packet_size</code> parameter is based on the nature of the typical data replicated. If the typical operation is very large, a larger packet size is more efficient.• A larger value of the <code>rs_packet_size</code> parameter is more efficient when the value of the <code>lti_batch_mode</code> parameter is true.• See the <i>Replication Agent Administration Guide</i> for more information about setting up Replication Agent connection configuration parameters.

rs_password

	The password that Replication Agent uses for Replication Server access.
Default	"" (empty string)
Value	A valid password.
Comments	<ul style="list-style-type: none">• The value of the <code>rs_password</code> parameter is the password for the user login name that Replication Agent uses to log in to the primary Replication Server.• The value of the <code>rs_password</code> parameter is encrypted in the Replication Agent instance configuration file.• See the <i>Replication Agent Administration Guide</i> for more information about setting up Replication Agent connection configuration parameters.

rs_port_number

The client port number of the primary Replication Server.

Default

1111

Value

A valid port number on the Replication Server host machine.

Comments

- The value of the `rs_port_number` parameter is the client port number of the primary Replication Server.
- See the *Replication Agent Administration Guide* for more information about setting up Replication Agent connection configuration parameters.

rs_replicate_owner_required

Indicates if the owner is always included with the replicate table clause when generating replication definitions.

Default

true

Value

true – the owner is always included in the replicate table clause.

false – the owner is not included in the replicate table clause unless the table is marked with the owner mode value set to on.

Comments

For additional information, see the `rs_create_repdef` command.

rs_retry_count

The number of times the Replication Agent will retry establishing a connection to the primary Replication Server.

Default

5

Value

An integer greater than 0.

Comments

- The value of the `rs_retry_count` parameter is the number of times that Replication Agent will try to establish a network connection to the Replication Server after a connection failure.
- Sybase recommends a setting of 5 for this parameter.
- See the *Replication Agent Administration Guide* for more information about setting up Replication Agent connection configuration parameters.

rs_retry_timeout

The number of seconds the Replication Agent waits between attempts to connect to the primary Replication Server.

Default	10
Value	An integer greater than 0.
Comments	<ul style="list-style-type: none">• The value of the <code>rs_retry_timeout</code> parameter is the number of seconds that the Replication Agent will wait between its retry attempts to establish a network connection to the primary Replication Server after a connection failure.• See the <i>Replication Agent Administration Guide</i> for more information about setting up Replication Agent connection configuration parameters.

rs_source_db

The name of the database identified in the Replication Server primary database connection.

Default	<not_configured>
Value	A valid database name.
Comments	<ul style="list-style-type: none">• The value of the <code>rs_source_db</code> parameter is the name of the primary database by which the primary Replication Server recognizes the primary database transaction log.• The value of the <code>rs_source_db</code> parameter must match the name of the database specified in the Replication Server create connection command for the primary database.• See the <i>Replication Agent Administration Guide</i> for more information about setting up Replication Agent connection configuration parameters.

rs_source_ds

The name of the data server identified in the Replication Server primary database connection.

Default	<not_configured>
Value	A valid server name.

- Comments
- The value of the `rs_source_ds` parameter is the name of the primary data server by which the primary Replication Server recognizes the primary database transaction log.
 - The value of the `rs_source_ds` parameter must match the name of the data server specified in the Replication Server `create connection` command for the primary database.
 - The value of the `rs_source_ds` parameter should *not* be the same as the name of the Replication Agent instance.
 - See the *Replication Agent Administration Guide* for more information about setting up Replication Agent connection configuration parameters.

rs_ticket_version

Determines whether Replication Agent records the primary database time or the primary database date and time into the `rs_ticket` marker.

- Default 1
- Value
- 1 – Replication Agent records only the primary database time.
 - 2 – Replication Agent records both the primary database date and time.
- Comments
- If the value is set to 1, Replication Agent records only the primary database time into `rs_ticket` marker. For example, 13:20:19.368.
 - If the value is set to 2, Replication Agent records both the primary database date and time into `rs_ticket` marker. For example, 12/14/07 13:20:19.368.
- See also `rs_ticket` command.

rs_username

The user login name that the Replication Agent uses for Replication Server access.

- Default <not_configured>
- Value A valid user name.
- Comments
- The value of the `rs_username` parameter is the user login name that Replication Agent uses to log in to the primary Replication Server.

- The value of the `rs_password` parameter is the password for the login name specified by the `rs_username` parameter.
- The user login name that Replication Agent uses to log in to the Replication Server must have connect source permission in the Replication Server.
- See the *Replication Agent Administration Guide* for more information about setting up Replication Agent connection configuration parameters.

rssd_charset

The character set used in communication with the RSSD of the primary Replication Server.

Default

"" (empty string)

Value

Any valid Sybase character set supported by the Java VM on the Replication Agent host machine.

Comments

- The value of the `rssd_charset` parameter must match (or be compatible with) the RSSD character set. The RSSD character set is usually the same as the Replication Server default character set identified by the Replication Server `rs_charset` configuration parameter.
- If you specify a valid character set for the value of the `rssd_charset` parameter, the Replication Agent instance communicates with the RSSD using that character set.
- If you do *not* specify a valid character set name for the value of the `rssd_charset` parameter (including the default `rssd_charset` value ""), the Replication Agent communicates with the RSSD using the RSSD charset.
- The `rssd_charset` parameter does not need to be set if the Replication Agent `use_rssd` parameter is set to `false`.
- See the *Replication Agent Administration Guide* for more information about setting up Replication Agent connection configuration parameters.

rssd_database_name

The database name of the RSSD of the primary Replication Server.

Default

<not_configured>

Value	A valid database name.
Comments	<ul style="list-style-type: none"> • The value of the <code>rssd_database_name</code> parameter is the database name of the RSSD of the primary Replication Server. • The <code>rssd_database_name</code> parameter does not need to be set if the Replication Agent <code>use_rssd</code> parameter is set to false. • See the <i>Replication Agent Administration Guide</i> for more information about setting up Replication Agent connection configuration parameters.

rssd_host_name

The name of the machine on which the RSSD of the primary Replication Server resides.

Default	<not_configured>
Value	A valid host name.
Comments	<ul style="list-style-type: none"> • The value of the <code>rssd_host_name</code> parameter is the name of the host machine on which the RSSD of the primary Replication Server resides. • The <code>rssd_host_name</code> parameter does not need to be set if the Replication Agent <code>use_rssd</code> parameter is set to false. • See the <i>Replication Agent Administration Guide</i> for more information about setting up Replication Agent connection configuration parameters.

rssd_password

The password that the Replication Agent uses for access to the RSSD of the primary Replication Server.

Default	"" (empty string)
Value	A valid password.
Comments	<ul style="list-style-type: none"> • The value of the <code>rssd_password</code> parameter is the password for the user login name that the Replication Agent uses to access the RSSD of the primary Replication Server. • The value of the <code>rssd_password</code> parameter is encrypted in the Replication Agent instance configuration file.

- The `rssd_password` parameter need not be set if the Replication Agent `use_rssd` parameter is set to false.
- See the *Replication Agent Administration Guide* for more information about setting up Replication Agent connection configuration parameters.

rssd_port_number

The client port number of the Replication Server System Database (RSSD) of the primary Replication Server.

Default	1111
Value	A valid port number on the RSSD host machine.
Comments	<ul style="list-style-type: none">• The value of the <code>rssd_port_number</code> parameter is the client port number of the RSSD data server.• The <code>rssd_port_number</code> parameter need not be set if the Replication Agent <code>use_rssd</code> parameter is set to false.• See the <i>Replication Agent Administration Guide</i> for more information about setting up Replication Agent connection configuration parameters.

rssd_username

The user login name that the Replication Agent uses to access the RSSD of the primary Replication Server.

Default	<not_configured>
Value	A valid user login name in the RSSD data server.
Comments	<ul style="list-style-type: none">• The value of the <code>rssd_username</code> parameter is the user login name that the Replication Agent uses to access the RSSD.• The <code>rssd_username</code> parameter need not be set if the Replication Agent <code>use_rssd</code> parameter is set to false.• See the <i>Replication Agent Administration Guide</i> for more information about setting up Replication Agent connection configuration parameters.

scan_eol_threshold

The number of seconds Replication Agent waits before determining that an RAC database redo log thread is inactive.

Note This parameter is available only for Oracle and is only applicable to Oracle RAC databases.

Default	300
Value	An integer from 300 to 2147483647.
Comments	<ul style="list-style-type: none"> • If no updates occur for a redo log thread of an RAC database for the number of seconds specified by the <code>scan_eol_threshold</code> parameter, Replication Agent considers the redo log thread to be inactive. • Even when a redo log thread is inactive, Replication Agent continues to check for new activity. When an inactive thread becomes active again, Replication Agent processes the activity without manual intervention. • If the <code>scan_eol_threshold</code> parameter is set too low, Replication Agent may treat all RAC database redo log threads as inactive. If the Replication Agent error log contains errors indicating inactivity for RAC database redo log threads that you know to be active, you may need to increase the value of the <code>scan_eol_threshold</code> configuration parameter.

scan_sleep_increment

The number of seconds to add to each wait interval before scanning the transaction log, after a previous scan yields no transaction to be replicated.

Default	5
Value	An integer from 0 to 3600.
Comments	<ul style="list-style-type: none"> • The value of the <code>scan_sleep_increment</code> parameter is the number of seconds added to each wait interval before the Log Reader component scans the log device for a transaction to be replicated, after a previous scan yields no such transaction. • The number of seconds specified by the <code>scan_sleep_increment</code> parameter is added to each wait interval, until the wait interval reaches the value specified by the <code>scan_sleep_max</code> parameter.

- For optimum Replication Agent performance, the value of the `scan_sleep_increment` parameter should be balanced with the average number of operations in the primary database over a period of time. In general, better performance results from reading more operations from the transaction log during each Log Reader scan.
- With a primary database that is less frequently updated, increasing the value of the `scan_sleep_increment` parameter may improve overall performance.
- If the database is continuously updated, the value of the `scan_sleep_increment` parameter may not be significant to Replication Agent performance.

scan_sleep_max

The maximum wait interval between Log Reader transaction log scans.

Default

60

Value

An integer from 5 to 86400.

Comments

- The value of the `scan_sleep_max` parameter is the maximum number of seconds that can elapse before the Log Reader component scans the transaction log for a transaction to be replicated, after a previous scan yields no such transaction.
- For reduced replication latency in an infrequently updated database, Sybase recommends lower number settings for the `scan_sleep_max` parameter.
- If the primary database is continuously updated, the value of the `scan_sleep_max` parameter is not significant to Replication Agent performance.

skip_ltl_errors

Determines whether the Replication Agent ignores Log Transfer Language (LTL) error messages.

Warning! Using the `skip_ltl_errors` parameter incorrectly may cause data inconsistencies between the primary and replicate databases.

Default	false
Values	true – enables skipping LTL errors to continue replication. false – disables skipping LTL errors.
Comments	<ul style="list-style-type: none">• If the <code>skip_ltl_errors</code> configuration parameter is set to true, the Replication Agent instance logs any LTL error messages returned by the Replication Server, along with the offending LTL commands, and then it continues processing transaction log records.• If the <code>skip_ltl_errors</code> configuration parameter is set to false, the Replication Agent instance stops all of its replication processing and goes to <i>Replication Down</i> state when it receives an LTL error message and the error is unrecoverable.• The <code>skip_ltl_errors</code> parameter is intended for troubleshooting only, with assistance from Sybase Technical Support.

structured_tokens

	Determines whether the Replication Agent uses LTL structured tokens.
Default	true
Values	true – enables LTL structured tokens. false – disables LTL structured tokens.
Comments	<ul style="list-style-type: none">• If the <code>structured_tokens</code> configuration parameter is set to true, the Log Transfer Interface (LTI) component uses LTL structured tokens when it generates LTL commands.• Using structured tokens in the LTL can significantly improve overall replication system performance.• Using structured tokens in the LTL can improve Replication Server performance, especially when non-Sybase datatypes in the primary database must be translated by Replication Server.• To replicate columns that have one or more spaces in the column name, you must set the value of the <code>structured_tokens</code> parameter to true.

truncation_interval

Specifies a time interval between automatic truncations of the Replication Agent transaction log.

Warning! If you configure automatic truncation, For UDB, the Replication Agent silently deletes the primary database log files, or for Oracle the Replication Agent deletes the archive log files that it no longer needs. For more information, see the *Replication Agent Primary Database Guide*.

Default	0
Value	An integer from 0 to 720.
Comments	<ul style="list-style-type: none">• The value of the truncation_interval parameter is the number of minutes between automatic transaction log truncations.• Automatic transaction log truncation based on the value of the truncation_interval parameter takes place only when the value of the truncation_type parameter is interval.• The maximum truncation interval is 720 minutes, or 12 hours.• If the value of the truncation_interval parameter is 0 (zero) and the value of the truncation_type parameter is interval, automatic truncation is disabled.• To truncate the transaction log manually, use the pdb_truncate_xlog command.
See also	pdb_archive_path, pdb_archive_remove, and truncation_type configuration parameters.

truncation_type

Configures transaction log truncation behavior of the Replication Agent.

Warning! For UDB, the Replication Agent deletes the UDB primary database log files and for Oracle the Replication Agent deletes the archived log files that it no longer needs. For more information, see the *Replication Agent Primary Database Guide*.

Default	locator_update
---------	----------------

Values	<p><code>command</code> – Replication Agent truncates the transaction log only when the <code>pdb_truncate_xlog</code> command is invoked.</p> <p>When the value of the <code>truncation_type</code> parameter is <code>command</code>, the only way you can truncate the transaction log is by invoking the <code>pdb_truncate_xlog</code> command. No automatic truncation takes place when the value of the <code>truncation_type</code> parameter is <code>command</code>.</p> <p><code>interval</code> – Replication Agent automatically truncates the transaction log when determined by a configurable interval of time.</p> <p><code>locator_update</code> – Replication Agent automatically truncates the transaction log whenever it receives a new LTM Locator value from the primary Replication Server.</p> <p>When the value of the <code>truncation_type</code> parameter is <code>locator_update</code>, the transaction log is automatically truncated when Replication Agent receives a new LTM Locator from the primary Replication Server.</p>
Comments	<hr/> <p>Note For Oracle and UDB, truncation of the archive log files that Replication Agent no longer needs from the <code>pdb_archive_path</code> directory is performed only if the <code>pdb_archive_remove</code> parameter is true.</p> <hr/> <ul style="list-style-type: none"> • Regardless of the value of the <code>truncation_type</code> parameter, you can truncate the Replication Agent transaction log manually at any time by invoking the <code>pdb_truncate_xlog</code> command. • If the value of the <code>truncation_interval</code> parameter is 0 (zero) and the value of the <code>truncation_type</code> parameter is <code>interval</code> (the default values for both parameters), automatic truncation is disabled. • Replication Agent receives a new LTM Locator based on the values of the <code>lti_update_trunc_point</code> parameter.
See also	<code>pdb_archive_path</code> , <code>pdb_archive_remove</code> , and <code>truncation_interval</code> configuration parameters.

use_rssd

	Determines whether the Replication Agent uses replication definitions.
Default	true
Values	true – enables using replication definitions.

false – disables using replication definitions.

Comments

- If the value of the `use_rssd` parameter is true, the Replication Agent instance connects to the Replication Server System Database (RSSD) to retrieve replication definitions for the primary database automatically whenever it goes from *Replication Down* state to *Replicating* state (for example, when the `resume` command is invoked).
 - Each time it retrieves replication definitions, Replication Agent stores the information in a cache. Replication Agent uses replication definitions stored in its cache when it generates Log Transfer Language (LTL) commands.
 - If the Log Transfer Interface (LTI) component encounters an operation on a database object for which it does not have a cached replication definition, Replication Agent reconnects to the RSSD to update its replication definition cache.
 - If a replication definition still cannot be found for the operation, the Replication Agent instance suspends all of its replication operations and goes to *Replication Down* state.
- Replication Agent can use information in table and function replication definitions (that is, replication definitions for individual primary database objects) stored in the RSSD to generate more efficient LTL, and thus improve throughput in the LTI component and the Replication Server.

Accessing replication definitions in the RSSD enables the LTI component to improve performance by:

- Omitting column names in LTL. When columns are sent in the order specified in the replication definition, column images can be sent without column names (headings), which reduces LTL overhead.
 - Omitting unneeded columns in LTL. When columns are sent as specified in the replication definition, images for unchanged columns need not be sent, which reduces LTL overhead.
 - Sending data for each column in the datatype specified by the replication definition. This allows data to be handled more efficiently all the way through the replication system.
 - Sending database object names in the same character case as defined in the replication definition.
- If the value of the `use_rssd` parameter is false, none of the previously described performance improvements are possible. In that case, the Replication Agent sends all data as a char datatype in the LTL.

Glossary

This glossary describes Replication Agent terms.

Adaptive Server

The brand name for Sybase relational database management system (RDBMS) software products.

- Adaptive Server® Enterprise manages multiple, large relational databases for high-volume online transaction processing (OLTP) systems and client applications.
- Adaptive Server IQ manages multiple, large relational databases with special indexing algorithms to support high-speed, high-volume business intelligence, decision support, and reporting client applications.
- SQL Anywhere™ (formerly Adaptive Server Anywhere) manages relational databases with a small DBMS footprint, which is ideal for embedded applications and mobile device applications.

See also **DBMS** and **RDBMS**.

atomic materialization

A materialization method that copies subscription data from a primary database to a standby database in a single, atomic operation. No changes to primary data are allowed until the subscription data is captured at the primary database. See also **bulk materialization** and **nonatomic materialization**.

BCP utility

A bulk copy transfer utility that provides the ability to load multiple rows of data into a table in a target database. See also **bulk copy**.

bulk copy

An Open Client™ interface for the high-speed transfer of data between a database table and program variables. Bulk copying provides an alternative to using SQL insert and select commands to transfer data.

bulk materialization

A materialization method whereby subscription data in a standby database is initialized outside of the replication system. You can use bulk materialization for subscriptions to table replication definitions or function replication definitions. See also **atomic materialization** and **nonatomic materialization**.

client	In client/server systems, the part of the system that sends requests to servers and processes the results of those requests. See also client application .
client application	Software that is responsible for the user interface, including menus, data entry screens, and report formats. See also client .
commit	An instruction to the DBMS to make permanent the changes requested in a transaction. See also transaction . Contrast with rollback .
data client	A client application that provides access to data by connecting to a data server. See also client , client application , and data server .
data distribution	A method of locating (or placing) discrete parts of a single set of data in multiple systems or at multiple sites. Data distribution is distinct from data replication, although a data replication system can be used to implement or support data distribution. Contrast with data replication .
data replication	The process of copying data to remote locations, and then keeping the replicated data synchronized with the primary data. Data replication is different from data distribution. Replicated data is stored copies of data at one or more remote sites throughout a system, and it is not necessarily distributed data. Contrast with data distribution . See also disk replication and transaction replication .
data server	A server that provides the functionality necessary to maintain the physical representation of a table in a database. Data servers are usually database servers, but they can also be any data repository with the interface and functionality a data client requires. See also client , client application , and data client .
database	A collection of data with a specific structure (or schema) for accepting, storing, and providing data for users. See also data server , DBMS , and RDBMS .
database connection	A connection that allows Replication Server to manage the database and distribute transactions to the database. Each database in a replication system can have only one database connection in Replication Server. See also Replication Server and route .
datatype	A keyword that identifies the characteristics of stored information on a computer. Some common datatypes are: char, int, smallint, date, time, numeric, and float. Different data servers support different datatypes.
DBMS	An abbreviation for database management system, a computer-based system for defining, creating, manipulating, controlling, managing, and using databases. The DBMS can include the user interface for using the database, or it can be a standalone data server system. Compare with RDBMS .

disaster recovery	A method or process used to restore the critical business functions interrupted by a catastrophic event. A disaster recovery (or business continuity) plan defines the resources and procedures required for an organization to recover from a disaster, based on specified recovery objectives.
ERSSD	An abbreviation for Embedded Replication Server System Database, which manages replication system information for a Replication Server. See also Replication Server .
failback	A procedure that restores the normal user and client access to a primary database, after a failover procedure switched access from the primary database to a standby database. See also failover .
failover	A procedure that switches user and client access from a primary database to a standby database, particularly in the event of a failure that interrupts operations at the primary database, or access to the primary database. Failover is an important fault-tolerance feature for systems that require high availability. See also failback .
function	A Replication Server object that represents a data server operation such as insert, delete, or begin transaction. Replication Server distributes operations to standby databases as functions. See also function string .
function string	A string that Replication Server uses to map a function and its parameters to a data server API. Function strings allow Replication Server to support heterogeneous replication, in which the primary and standby databases are different types, with different SQL extensions and different command features. See also function .
gateway	Connectivity software that allows two or more computer systems with different network architectures to communicate.
inbound queue	A stable queue managed by Replication Server to spool messages received from a Replication Agent. See also outbound queue and stable queue .
interfaces file	A file containing information that Sybase Open Client and Open Server™ applications need to establish connections to other Open Client and Open Server applications. See also Open Client and Open Server .
isql	An Interactive SQL client application that can connect and communicate with any Sybase Open Server application, including Adaptive Server Enterprise, Replication Agent, and Replication Server. See also Open Client and Open Server .
Java	An object-oriented programming language developed by Sun Microsystems. A platform-independent, “write once, run anywhere” programming language.

Java VM	The Java Virtual Machine. The Java VM (or JVM) is the part of the Java Runtime Environment (JRE) that is responsible for interpreting Java byte codes. See also Java and JRE .
JDBC	An abbreviation for Java Database Connectivity. JDBC is the standard communication protocol for connectivity between Java clients and data servers. See also data server and Java .
JRE	An abbreviation for Java Runtime Environment. The JRE consists of the Java Virtual Machine (Java VM or JVM), the Java Core Classes, and supporting files. The JRE must be installed on a machine to run Java applications, such as the Replication Agent. See also Java VM .
LAN	An abbreviation for “local area network,” a computer network located on the user’s premises that covers a limited geographical area (usually a single site). Communication within a local area network is not subject to external regulations; however, communication across the LAN boundary can be subject to some form of regulation. Contrast with WAN .
latency	<p>In transaction replication, the time it takes to replicate a transaction from a primary database to a standby database. Specifically, latency is the time elapsed between committing an original transaction in the primary database and committing the replicated transaction in the standby database.</p> <p>In disk replication, latency is the time elapsed between a disk write operation that changes a block or page on a primary device and the disk write operation that changes the replicated block or page on a replicate device.</p> <p>See also disk replication and transaction replication.</p>
LOB	An abbreviation for large object, a type of data element that is associated with a column that contains extremely large quantities of data.
Log Reader	An internal component of the Replication Agent that interacts with the primary database to capture transactions for replication. See also Log Transfer Interface and Log Transfer Manager .
Log Transfer Interface	An internal component of the Replication Agent that interacts with Replication Server to forward transactions for distribution to a standby database. See also Log Reader and Log Transfer Manager .
Log Transfer Manager	An internal component of the Replication Agent that interacts with the other Replication Agent internal components to control and coordinate Replication Agent operations. See also Log Reader and Log Transfer Interface .

maintenance user	A special user login name in the standby database that Replication Server uses to apply replicated transactions to the database. See also Replication Server .
materialization	The process of copying the data from a primary database to a standby database, initializing the standby database so that the Replication Agent system can begin replicating transactions. See also atomic materialization , bulk materialization , and nonatomic materialization .
nonatomic materialization	A materialization method that copies subscription data without a lock on the primary database. Changes to primary data are allowed during data transfer, which may cause temporary inconsistencies between the primary and standby databases. Contrast with atomic materialization . See also bulk materialization .
ODBC	An abbreviation for Open Database Connectivity, an industry-standard communication protocol for clients connecting to data servers. See also JDBC .
Open Client	A Sybase product that provides customer applications, third-party products, and other Sybase products with the interfaces needed to communicate with Open Server applications. See also Open Server .
Open Client application	An application that uses Sybase Open Client libraries to implement Open Client communication protocols. See also Open Client and Open Server .
Open Server	A Sybase product that provides the tools and interfaces required to create a custom server. See also Open Client .
Open Server application	A server application that uses Sybase Open Server libraries to implement Open Server communication protocols. See also Open Client and Open Server .
outbound queue	A stable queue managed by Replication Server to spool messages to a standby database. See also inbound queue and stable queue .
primary data	The version of a set of data that is the source used for replication. Primary data is stored and managed by the primary database. See also Replication Agent , primary database , and Replication Server .
primary database	The database that contains the data to be replicated to another database (the standby database) through a replication system. The primary database is the database that is the source of replicated data in a replication system. Sometimes called the active database. Contrast with standby database . See also primary data .
primary key	The column or columns whose data uniquely identify each row in a table.

primary site	The location or facility at which primary data servers and primary databases are deployed to support normal business operations. Sometimes called the active site or main site. See also primary database and standby site .
primary table	A table used as a source for replication. Primary tables are defined in the primary database schema. See also primary data and primary database .
primary transaction	A transaction that is committed in the primary database and recorded in the primary database transaction log. See also primary database , replicated transaction , and transaction log .
quiesce	To cause a system to go into a state in which further data changes are not allowed. See also quiescent .
quiescent	<p>In a replication system, a state in which all updates have been propagated to their destinations. Some Replication Agent and Replication Server commands require that you first quiesce the replication system.</p> <p>In a database, a state in which all data updates are suspended so that transactions cannot change any data and the data and log devices are stable.</p> <p>This term is interchangeable with quiesced and in quiesce. See also quiesce.</p>
RASD	An abbreviation for Replication Agent System Database. Information in the RASD is used by the primary database to recognize database structure or schema objects in the transaction log.
RCL	An abbreviation for Replication Command Language, the command language used to manage Replication Server.
RDBMS	An abbreviation for relational database management system, an application that manages and controls relational databases. Compare with DBMS . See also relational database .
relational database	A collection of data in which data is viewed as being stored in tables, which consist of columns (data items) and rows (units of information). Relational databases can be accessed by SQL requests. See also SQL .
replicated data	A set of data that is replicated from a primary database to a standby database by a replication system. See also primary database , replication system , and standby database .
replicated transaction	A primary transaction that is replicated from a primary database to a standby database by a transaction replication system. See also primary database , primary transaction , standby database , and transaction replication .

Replication Agent	An application that reads a primary database transaction log to acquire information about data-changing transactions in the primary database, processes the log information, and then sends it to a Replication Server for distribution to a standby database. See also primary database and Replication Server .
replication definition	A description of a table or stored procedure in a primary database, for which subscriptions can be created. The replication definition, maintained by Replication Server, includes information about the columns to be replicated and the location of the primary table or stored procedure. See also Replication Server and subscription .
Replication Server	The Sybase software product that provides the infrastructure for a robust transaction replication system. See also Replication Agent .
RSSD	An abbreviation for Replication Server System Database, which manages replication system information for a Replication Server. See also Replication Server .
replication system	A data processing system that replicates data from one location to another. Data can be replicated between separate systems at a single site, or from one or more local systems to one or more remote systems. See also disk replication and transaction replication .
rollback	An instruction to a database to back out of the changes requested in a unit of work (called a transaction). Contrast with commit . See also transaction .
SQL	An abbreviation for Structured Query Language, a nonprocedural programming language used to process data in a relational database. ANSI SQL is an industry standard. See also transaction .
stable queue	A disk device-based, store-and-forward queue managed by Replication Server. Messages written into the stable queue remain there until they can be delivered to the appropriate process or standby database. Replication Server provides a stable queue for both incoming messages (the inbound queue) and outgoing messages (the outbound queue). See also database connection , Replication Server , and route .
standby data	The data managed by a standby database, which is the destination (or target) of a replication system. See also data replication and standby database .

standby database	A database that contains data replicated from another database (the primary database) through a replication system. The standby database is the database that receives replicated data in a replication system. Sometimes called the replicate database. Contrast with primary database . See also standby data .
standby site	The location or facility at which standby data servers and standby databases are deployed to support disaster recovery, and normal business operations during scheduled downtime at the primary site. Sometimes called the alternate site or replicate site. Contrast with primary site . See also standby database .
subscription	A request for Replication Server to maintain a replicated copy of a table, or a set of rows from a table, in a standby database at a specified location. See also replication definition and Replication Server .
table	In a relational DBMS, a two-dimensional array of data or a named data object that contains a specific number of unordered rows composed of a group of columns that are specific for the table. See also database .
transaction	A unit of work in a database that can include zero, one, or many operations (including insert, update, and delete operations), and that is either applied or rejected as a whole. Each SQL statement that modifies data can be treated as a separate transaction, if the database is so configured. See also SQL .
transaction log	Generally, the log of transactions that affect the data managed by a data server. Replication Agent reads the transaction log to identify and acquire the transactions to be replicated from the primary database. See also Replication Agent , primary database , and Replication Server .
transaction replication	A data replication method that copies data-changing operations from a primary database transaction log to a standby database. See also data replication and disk replication .
transactional consistency	A condition in which all transactions in the primary database are applied in the standby database, in the same order that they were applied in the primary database.
WAN	An abbreviation for “wide area network,” a system of local-area networks (LANs) connected together with data communication lines. Contrast with LAN .

Index

A

abbreviated form of LTL 171
Admin state 83, 125–126, 148–149
admin_port configuration parameter 166
administrator login 114, 188–189
alias, of database object 33, 49, 67
archive_read_block_count configuration parameter 167
articles in RASD 94, 95
truncating 127
asm_password configuration parameter 167
asm_tns_connection configuration parameter 168
asm_tns_filename configuration parameter 168
asm_username configuration parameter 169
auto_adjust_block_count configuration parameter 170
automatic running of scripts 195

B

backing up RASD 133, 214–215
base objects, transaction log 65
batch mode, LTL 183–184, 185–186
buffers, Log Transfer Interface 80–81, 174, 183, 185–186

C

changing
configuration parameters 81
primary database log device path 85
character case in LTL
See also Log Transfer Language (LTL)
column names 186–187
stored procedure names 43–44, 186–187
table names 57, 186–187
character set

primary data server 223
Replication Server 222–223
RSSD 228
client ports
primary data server 210
Replication Server 225
RSSD 230
column_compression configuration parameter 170–171
columns
date/time conversion with LOB columns 199
enabling and disabling replication 30–34
enabling replication 199–200
fields in RASD 99–101
name in LTL 186–187
name of LOB column 32
name returned by database 14–16
primary key 17–18
replication status 33
sent in LTL 170–171
commands
help information 93
log_system_name 6
lr_dump_marker 7
pdb_capabilities 8
pdb_commit_generation_filter 9
pdb_commit_generation_status 10
pdb_date 11–12
pdb_execute_sql 12–13
pdb_gen_id 13–14
pdb_get_columns 14–16
pdb_get_databases 16–17
pdb_get_primary_keys 17–18
pdb_get_procedure_parms 18, 20
pdb_get_procedures 20–21
pdb_get_sql_database 22
pdb_get_tables 22–24
pdb_send_osuser_list 27
pdb_set_sql_database 29
pdb_setrepcol 30–34

- pdb_setrepddl** 35
- pdb_setrepproc** 43–53
- pdb_setrepseq** 54
- pdb_setreptable** 56–70
- pdb_skip_op** 71
- pdb_truncate_xlog** 75
- pdb_version** 75
- pdb_xlog** 75–80
- quiesce** 80–81
- ra_config** 81–83, 157, 158
- ra_date** 83–84
- ra_deviceoffset** 84, 99
- ra_devicepath** 85
- ra_downgrade_accept** 86
- ra_downgrade_prepare** 87
- ra_dump** 87–88
- ra_dumptran** 88
- ra_help** 93
- ra_helparticle** 94–95
- ra_helppdb** 95–96
- ra_helpdevice** 96
- ra_helpfield** 99–101
- ra_helplocator** 101
- ra_helppop** 102
- ra_helpuser** 106–107
- ra_locator** 107
- ra_maint_id** 109–110
- ra_marker** 110–111
- ra_migrate** 111
- ra_set_autocorrection** 112
- ra_set_login** 114, 157, 158
- ra_statistics** 115, 125
- ra_status** 125, 126
- ra_truncatearticles** 127
- ra_truncateddlfilters** 128
- ra_truncateusers** 128
- ra_updatedevices** 129, 172
- ra_version** 131
- ra_version_all** 132
- rasd_backup** 133
- rasd_helpbackup** 133
- rasd_removebackup** 134–135
- rasd_restore** 136
- resume** 139–141
- rs_ticket** 145
- shutdown** 147–148
- suspend** 148–149
- test_connection** 149–152
- trace** 152–155
- communications
 - driver version 75, 132
 - JDBC driver 206, 207
 - network packet size 224
 - ODBC driver 206
 - primary data server parameters 213
 - Replication Server parameters 222, 228
 - RSSD parameters 230
 - testing connections 149–152
- compress_lti_syntax** configuration parameter 171
- configuration parameters 157
 - admin_port** 166
 - archive_read_block_count** 167
 - asm_password** 167
 - asm_tns_connection** 168
 - asm_tns_filename** 168
 - asm_username** 169
 - auto_adjust_block_count** 170
 - column_compression** 170–171
 - compress_lti_syntax** 171
 - connect_to_rs** 171–172
 - ddl_password** 172
 - ddl_username** 173
 - dump_batch_timeout** 174
 - filter_maint_userid** 174
 - function_password** 175
 - function_username** 175
 - getting current values 81
 - log_backup_files** 176
 - log_directory** 6, 176–177
 - log_read_block_count** 177
 - log_trace_verbose** 178
 - log_wrap** 178
 - lr_max_op_queue_size** 179
 - lr_max_scan_queue_size** 179
 - lr_parallel_scan** 181
 - lr_read_buffer_size** 182
 - lti_batch_mode** 183–184
 - lti_max_buffer_size** 184
 - lti_update_trunc_point** 184–185
 - ltl_batch_size** 185–186
 - ltl_big_endian_unitext** 186
 - ltl_character_case** 186–187

- ltl_origin_time_required 187–188
- ltl_send_only_primary_keys 188
- ltm_admin_pw 158, 188–189
- ltm_admin_user 158, 189
- max_ops_per_scan 190
- pdb_archive_path 190
- pdb_archive_remove 191
- pdb_auto_create_repdefs 192
- pdb_auto_run_scripts 195
- pdb_automark_tables 194
- pdb_commit_generation 196
- pdb_commit_interval 197
- pdb_convert_datetime 197–199
- pdb_dflt_column_repl 199–200
- pdb_dflt_object_repl 200
- pdb_ignore_unsupported_anydata 201
- pdb_include_archives 202
- pdb_ownerfilter 24–26
- pdb_skip_missing_user 203
- pdb_support_large_identifier 203
- pdb_timezone_file 204
- pdb_xlog_device 204
- pdb_xlog_prefix 78, 79, 205
- pdb_xlog_prefix_chars 205, 206
- pds_connection_type 206
- pds_dac_port_number 207
- pds_database_name 208
- pds_datasource_name 208
- pds_host_name 209
- pds_integrated_security 209
- pds_password 210
- pds_port_number 210
- pds_retry_count 210–211
- pds_retry_timeout 211
- pds_server_name 211
- pds_tns_connection 212
- pds_tns_filename 212
- pds_username 213
- ra_retry_count 213–214
- ra_retry_timeout 214
- ra_standby 219–220
- rasd_backup_dir 214–215
- rasd_database 215–216
- rasd_mirror_tran_log 216
- rasd_trace_log_dir 217
- rasd_tran_log 218
- rasd_tran_log_mirror 219
- rman_enabled 220
- rman_password 221
- rman_username 221
- rs_charset 222–223
- rs_host_name 223
- rs_packet_size 224
- rs_password 224
- rs_port_number 225
- rs_replicate_owner_required 225
- rs_retry_count 225
- rs_retry_timeout 226
- rs_source_db 226
- rs_source_ds 226–227
- rs_ticket_version 227
- rs_username 227–228
- rssd_charset 228
- rssd_database_name 228–229
- rssd_host_name 229
- rssd_password 229–230
- rssd_port_number 230
- rssd_username 230
- scan_eol_threshold 231
- scan_sleep_increment 231–232
- scan_sleep_max 232
- setting 81, 83
- skip_ltl_errors 232, 233
- structured_tokens 233
- truncation_interval 234
- truncation_type 234–235
- use_rssd 235
- connect_to_rs configuration parameter 171–172
- connections
 - character sets 222–223, 228
 - dummy connection 171–172
 - pds_connection_type** parameter 206, 207
 - pds_dac_port_number** parameter 207
 - pds_database_name** parameter 208
 - pds_datasource_name** parameter 208
 - pds_host_name** parameter 209
 - pds_port_number** parameter 210
 - pds_retry_count** parameter 210–211
 - pds_retry_timeout** parameter 211
 - pds_server_name** parameter 211
 - primary data server character set 223
 - Replication Server character set 222–223

rs_charset parameter 222–223
rs_host_name parameter 223
rs_packet_size parameter 224
rs_password parameter 224
rs_port_number parameter 225
rs_replicate_owner_required parameter 225
rs_retry_count parameter 225
rs_retry_timeout parameter 226
rs_source_db parameter 226
rs_source_ds parameter 226–227
rs_username parameter 227–228
 RSSD character set 228
rssd_charset parameter 228
rssd_database_name parameter 228–229
rssd_host_name parameter 229
rssd_port_number parameter 230
 testing 149–152, 171
 converting temporal datatypes 197–199
 creating
 transaction log 78, 195
 current database for executing SQL 12–13, 22, 29

D

data source name (DSN)
 See ODBC driver
 database connection to Replication Server 57
 database connections
 in Replication Server 109–110
 database connections in Replication Server 70
 database devices
 help command 96–98
 primary database log device 85, 96, 129
 primary database mirror log device 96, 172
 database generation ID 13–14
 See also LTM Locator; origin queue ID
 database objects
 See also columns; primary tables; stored procedures
 aliases, synonyms, and views 33, 49, 65–66, 67
 articles in RASD 94–95
 character case of names in LTL 186–187
 columns 14–16, 99–101
 fields in articles 99–101
 LOB columns 30–34
 primary keys 17–18

 stored procedures 18, 21, 43–53, 94–95
 tables 22–24, 94–95
 transaction log prefix 205
 users 107, 128
 database operations
 help command 102
 troubleshooting 88
 databases
 See UDB; Microsoft SQL Server; Oracle; primary
 databases; RASD; replicate databases
 datatypes
 char (Sybase) 197–199
 converting non-Sybase date/time 197–199
 datetime (Sybase) 197–199
 date and time returned
 primary data server 11–12
 Replication Agent 83–84
 date/time datatype conversion 197–199
datetime Sybase datatype 197–199
 DDL in transaction log 127
ddl_password configuration parameter 172
ddl_username configuration parameter 173
 deleting
 transaction log 78–79, 195
 device name of primary database 204
 diagnostic, verbose logging 177, 178
 disabling column replication 30–34
 for all LOB columns 34
 disabling stored procedure replication 43–53
 for all stored procedures 50
 disabling table replication 56, 70
 for all tables 68
 DSN (ODBC data source name)
 See ODBC driver
 dummy connections 171–172
 dump marker in transaction log 87–88
dump_batch_timeout configuration parameter 174

E

enabling column replication 30–34
 by default 199–200
 for all LOB columns 34
 enabling stored procedure replication 43–53
 for all stored procedures 50

- enabling table replication 70
 - by default 200
 - for all tables 68
- errors 232
- errors, Log Transfer Language (LTL) 233
- executing SQL commands 12–13, 22, 29

F

- files
 - LTL trace log 153
 - mirrored RASD transaction log 219
 - RASD backup 214–215
 - RASD database file 215–216
 - RASD trace log 217
 - Replication Agent scripts directory 78, 79
 - system log 6, 152–155
- filter_maint_userid** configuration parameter 174
- forcing unmarking
 - stored procedures 45, 50
 - tables 60, 66, 68
- format of configuration file 158
- function replication definitions 44, 51
- function_password** configuration parameter 175
- function_username** configuration parameter 175

G

- gateway to primary database 75
- generation ID of primary database 13–14
 - See also* LTM Locator; origin queue ID
- getting help with Replication Agent commands 93
- getting information
 - primary database date and time 11–12
 - primary database objects 14, 24
 - primary database version 75
 - Replication Agent commands 107
 - Replication Agent date and time 83–84
 - Replication Agent performance 115, 125
 - Replication Agent status 125–126
 - Replication Agent version 131

H

- help
 - for commands 93
 - for configuration parameters 81
- help commands
 - articles in RASD 94, 95
 - database operations 102
 - fields in articles 99–101
 - LTM Locator 101
 - primary database 95–96
 - primary database log devices 96–98
 - primary database users in RASD 106–107
- host machines
 - primary data server 209
 - Replication Agent 131, 166
 - Replication Server 223
 - RSSD 229

I

- IBM DB2 Universal Database
 - See* UDB
- immediate shutdown 147–148
- instance, Replication Agent
 - administrator login 114
 - configuration file 157, 158
 - quiescing 80–81
 - resuming 139–141
 - shutting down 147–148
 - status 125–126

J

- Java Runtime Environment (JRE)
 - character set 222–223, 228
 - version 131
- JDBC driver
 - Oracle database server 206, 207
 - UDB 206–207
 - version 75, 132

L

LOB columns

- date/time conversion with 199
- disabling replication for 30–34
- enabling replication 199–200
- enabling replication for 30–34
- name of 32
- replication status 33

log devices

- help command 96–98
- path to log device 85
- path to mirror log device 96
- updating log device repository 129–131

log devices, primary database

- path to location 85
- updating in RASD 129–131

log files

- RASD trace log 217
- RASD transaction log 216
- Replication Agent system log 6, 152–155, 176
- truncating transaction log 75, 184
- wrapping 178

Log Reader component

- filter_maint_userid** parameter 174
- max_ops_per_scan** parameter 190
- operation queue 190
- operations per scan 190
- quiesce processing 80
- scan_sleep_increment** parameter 231–232
- scan_sleep_max** parameter 232
- statistics 115, 125

Log Transfer Interface component 170

- batch mode 183–184, 185–186
- batch timeout 174
- buffer size 185–186
- column_compression** parameter 170–171
- compress_ltl_syntax** parameter 171
- connect_to_rs** parameter 171, 172
- dump_batch_timeout** parameter 174
- lti_update_trunc_point** parameter 184–185
- LTL batch mode buffer 184, 185–186
- ltl_batch_size** parameter 185–186
- ltl_character_case** parameter 186–187
- ltl_origin_time_required** parameter 187–188
- quiesce processing 81
- statistics 115, 125

Log Transfer Language (LTL)

- character case of object names 43–44, 57, 186–187
- columns sent in 170–171
- compressed syntax 171
- error messages 232, 233
- LTL batch mode buffer 184, 185–186
- LTL trace log 153
- object owner name 58
- origin_time** command tag 187–188
- structured tokens 233

Log Transfer Manager component

- statistics 115, 125
- log_backup_files** configuration parameter 176
- log_directory** configuration parameter 6, 176–177
- log_read_block_count** configuration parameter 177
- log_system_name** command 6
- log_trace_verbos** configuration parameter 178
- log_wrap** configuration parameter 178
- lr_dump_marker** command 7
- lr_max_op_queue_size** configuration parameter 179
- lr_max_scan_queue_size** configuration parameter 179
- lr_parallel_scan** configuration parameter 181
- lr_read_buffer_size** configuration parameter 182
- lti_batch_mode** configuration parameter 183–184
- lti_max_buffer_size** configuration parameter 184
- lti_update_trunc_point** configuration parameter 184–185
- ltl_batch_size** configuration parameter 185–186
- ltl_big_endian_unitext** configuration parameter 186
- ltl_character_case** configuration parameter 186–187
- ltl_origin_time_required** configuration parameter 187–188
- ltl_send_only_primary_keys** configuration parameter 188
- LTM Locator 107–109, 235
 - help command 101
 - origin queue ID 13–14
 - position in transaction log 107–109, 139
 - updating 184–185
- ltm_admin_pw** configuration parameter 158, 188–189
- ltm_admin_user** configuration parameter 158, 189

M

Maintenance User
 filtered by Log Reader 174

markers in transaction log
ra_marker object 110–111
rs_dumpdb marker 87–88
rs_dumptran marker 87–88

marking a primary table 56–70
 all user tables 68
 items not allowed 33, 65–66, 67
 marking status 67
 running scripts automatically 195

marking a stored procedure 43–53
 items not allowed 49
 marking status 49
 running scripts automatically 195

max_ops_per_scan configuration parameter 190

Microsoft SQL Server
 connection type 206
 ODBC driver 206

mirror log devices, primary database
 path to location 96
 updating in RASD 172

mirrored RASD transaction log 216, 219

N

names
 columns returned by database 14–16
 host machine 209, 223, 229
 primary data server 211
 primary database 16–17, 207, 208
 primary table owner 57, 58
 RASD database name 215–216
 RSSD database name 228–229
 stored procedure owner 43–44
 stored procedures 20–21

network packet size 224

O

objects, database
 columns 14–16
 primary keys 17–18

stored procedures 18, 21
 tables 22–24
 users 107, 128

ODBC driver
 data source name (DSN) 208

operating system
 version 131

Oracle archive log file truncation
 using RMAN utility 220, 221

Oracle database server
 connection type 206
 JDBC driver 207

Oracle RMAN utility
 archive log file truncation 220, 221

origin queue ID
See also LTM Locator
 database generation ID 13–14

origin_time LTL command tag 187–188

owner of objects
 primary tables 57, 58
 stored procedures 43–44

P

parameters
See also configuration parameters
 Replication Agent configuration 81–83
 stored procedure 18, 20

passwords
 primary database user login 210
 Replication Agent administrator 114, 188–189
 Replication Server user login 224
 RSSD user login 229–230

path
 log devices 85, 95–96, 98, 129, 131
 mirror log devices 96, 172
 RASD backup directory 133, 136, 214–215
 RASD database file 215–216
 RASD trace log 217
 RASD transaction log 218
 RASD transaction log mirror 219
 Replication Agent scripts directory 78, 79
 Replication Agent system log 153, 176

pdb_archive_path configuration parameter 190

pdb_archive_remove configuration parameter 191

- pdb_asmdiskmap** command 7
- pdb_auto_create_repdefs** configuration parameter 192
- pdb_auto_run_scripts** configuration parameter 195
- pdb_automark_tables** configuration parameter 194
- pdb_capabilities** command 8
- pdb_commit_generation** configuration parameter 196
- pdb_commit_generation_filter** command 9
- pdb_commit_generation_status** command 10
- pdb_commit_interval** configuration parameter 197
- pdb_convert_datetime** configuration parameter 197–199
- pdb_date** command 11–12
- pdb_dflt_column_repl** configuration parameter 199–200
- pdb_dflt_object_repl** configuration parameter 200
- pdb_execute_sql** command 12–13
- pdb_gen_id** command 13–14
- pdb_get_columns** command 14–16
- pdb_get_databases** command 16–17
- pdb_get_primary_keys** command 17–18
- pdb_get_procedure_parms** command 18, 20
- pdb_get_procedures** command 20–21
- pdb_get_sql_database** command 22
- pdb_get_tables** command 22–24
- pdb_ignore_unsupported_anydata** configuration parameter 201
- pdb_include_archives** configuration parameter 202
- pdb_ownerfilter** configuration parameter 24
- pdb_send_osuser_list** command 27
- pdb_set_sql_database** command 29
- pdb_setrepcol** command 30–34
- pdb_setreppddl** command 35
- pdb_setrepproc** command 43–53
- pdb_setrepseq** command 54
- pdb_setreptable** command 56–70
- pdb_skip_missing_user** configuration parameter 203
- pdb_skip_op** command 71
- pdb_support_large_identifier** configuration parameter 203
- pdb_timezone_file** configuration parameter 204
- pdb_truncate_xlog** command 74, 75
- pdb_version** command 75
- pdb_xlog** command 75–80
- pdb_xlog_device** configuration parameter 204
- pdb_xlog_prefix** configuration parameter 78, 79, 205
- pdb_xlog_prefix_chars** configuration parameter 205
- pds_connection_type** configuration parameter 206, 207
- pds_dac_port_number** configuration parameter 207
- pds_database_name** configuration parameter 208
- pds_datasource_name** configuration parameter 208
- pds_host_name** configuration parameter 209
- pds_integrated_security** configuration parameter 209
- pds_password** configuration parameter 210
- pds_port_number** configuration parameter 210
- pds_retry_count** configuration parameter 210–211
- pds_retry_timeout** configuration parameter 211
- pds_server_name** configuration parameter 211
- pds_tns_connection** configuration parameter 212
- pds_tns_filename** configuration parameter 212
- pds_username** configuration parameter 213
- performance statistics 115, 125
 - resetting 125
- port numbers
 - primary data server 210
 - Replication Agent 166
 - Replication Server 225
 - RSSD 230
- prefix, transaction log 78, 79
- primary database
 - server port number 210
- primary databases
 - articles in RASD 94, 95
 - character set 223
 - column names returned 14–16
 - communications drivers 132
 - connection from Replication Agent 149–152, 206, 213
 - database connection in Replication Server 44, 51, 70
 - database connections to Replication Server 57
 - database name 16–17, 207, 208
 - device name 204
 - gateway 75
 - generation ID 13–14
 - host machine name 209
 - log devices 85, 96–98, 129–131, 172
 - object names returned 14–16
 - path to location 84, 85, 98
 - primary keys 17–18
 - Replication Agent user login 213
 - Replication Server database connection 109–110
 - Replication Server source definition 226
 - server date and time 11–12

- server name 211
- server port number 210
- server version 75
- SQL commands 12–13, 22, 29
- stored procedures 18, 21
- testing connections 149–152
- updating log devices 129–131, 172
- user logins in RASD 106–107, 128
- version 132

primary key columns 17–18

primary tables

- articles in RASD 127
- character case of name 57
- disabling replication 56, 70
- enabling replication 56, 70, 200
- forcing unmarking 60, 66, 68
- getting list from database 22–24
- LOB columns 30–34
- marking 33, 56–70
- marking status 67
- object owner 57, 58
- primary keys 17–18
- table name 22–24, 56, 65–66
- unmarking 56–70

Q

queues

- Log Reader 190
- Log Transfer Interface 80–81, 174, 183, 185–186
- LTM Locator 101, 107–109
- origin queue ID 13–14
- quiesce** processing 80–81
- suspend** processing 148–149

quiesce command 80–81

quiescing Replication Agent 80–81

R

ra_config command 81–83, 157, 158

ra_date command 83–84

ra_deviceoffset command 84, 99

ra_devicepath command 85

ra_downgrade_accept command 86

ra_downgrade_prepare command 87

ra_dump command 87–88

ra_dumptran command 88

ra_help command 93

ra_helparticle command 94–95

ra_helppdb command 95–96

ra_helpdevice command 96

ra_helpfield command 99–101

ra_helplocator 101

ra_helpop command 102

ra_helpuser command 106–107

ra_locator command 107

ra_maint_id command 109–110

ra_marker command 110, 111

ra_marker system function 110–111

ra_migrate command 111

ra_migrate system function 111

ra_retry_count configuration parameter 213–214

ra_retry_timeout configuration parameter 214

ra_set_autocorrection command 112

ra_set_login command 114, 157, 158

ra_standby configuration parameter 219–220

ra_statistics command 115, 125

ra_status command 125, 126

ra_truncatearticles command 127

ra_truncateddfilters command 128

ra_truncateusers command 128

ra_updatedevices command 129, 172

ra_version command 131

ra_version_all command 132

RASD

- articles 94–95, 127
- backing up database 133
- database backup files 214–215
- database file 215–216
- fields 99–101
- log devices, primary database 85
- mirror log devices, primary database 96–98
- mirrored RASD log 216, 219
- primary database 95–96
- primary database objects 94–95, 127
- primary database users 107, 128
- rasd_backup_dir** parameter 214–215
- rasd_database** parameter 215–216
- rasd_mirror_tran_log** parameter 216
- rasd_trace_log_dir** parameter 217

- rasd_tran_log** parameter 218
- rasd_tran_log_mirror** parameter 219
- restoring from backup 136
- transaction log file 218
- truncating 128
- updating log devices 129–131
- updating mirror log devices 172
- rasd_backup** command 133
- rasd_backup_dir** configuration parameter 214–215
- rasd_database** configuration parameter 215–216
- rasd_helpbackup** command 133
- rasd_mirror_tran_log** configuration parameter 216
- rasd_removebackup** command 134–135
- rasd_restore** command 136
- rasd_trace_log_dir** configuration parameter 217
- rasd_tran_log** configuration parameter 218
- rasd_tran_log_mirror** configuration parameter 219
- rasd_trunc_schedule** command 136
- replicate tables
 - name specified in replication definition 69
- Replicating* state 125–126, 139–141
- Replication Agent
 - Admin* state 83, 125–126, 148–149
 - administration port 166
 - administrator login 114, 188–189
 - articles in RASD 94, 95
 - asm_password 167
 - asm_tns_connection 168
 - asm_tns_filename 168
 - asm_tns_username 169
 - backing up RASD 133
 - creating transaction log 78
 - database generation ID 13–14
 - date and time returned 83–84
 - fields in articles 99–101
 - help commands 93
 - immediate shutdown 147–148
 - Log Reader component 80, 231
 - Log Transfer Interface component 81
 - LTL batch size 183–184, 185–186
 - LTL structured tokens 233
 - LTL trace log 153
 - LTM Locator 101, 107–109, 184–185, 235
 - origin queue ID 107–109
 - pds_tns_connection 212
 - pds_tns_filename 212
 - performance statistics 115, 125
 - primary database user login 210, 213
 - quiescing an instance 80–81
 - RASD 136, 214
 - removing transaction log 78–79
 - Replicating* state 81, 125–126, 139–141
 - Replication Server user login 224, 227–228
 - restoring RASD 136
 - rs_create_repdef 141
 - rs_drop_repdef 144
 - RSSD connection 228, 230
 - RSSD user login 229–230, 230
 - scripts directory 78, 79
 - shutting down an instance 147–148
 - statistics, performance 115, 125
 - status 125–126
 - system log file 6, 152–155, 176
 - testing connections 149–152, 171, 172
 - transaction log prefix 78, 79, 205
 - troubleshooting 152–155, 177, 178, 232, 233
 - updating log device repository 129–131, 172
 - version 131
- Replication Agent commands 81
- Replication Agent System Database
 - See* RASD
- replication definitions
 - character case of object names 43–44, 57, 186–187
 - function (stored procedure) 44, 51
 - table 57, 69, 70
- Replication Server
 - batch mode 183
 - character set 222–223
 - connection from Replication Agent 149–152, 171, 172, 222, 228
 - database connection 44, 51, 57, 70, 109–110
 - database generation ID 13–14
 - function replication definition 51
 - host machine name 223
 - LTL batch size 185–186
 - LTL errors 232, 233
 - LTM Locator 107–109, 184–185, 235
 - network packet size 224
 - port number 225
 - ra_marker** system function 110–111
 - ra_migrate** system function 111
 - Replication Agent user login 224, 227–228

replication definitions 44, 51, 57, 70
rs_dumpdb marker 87–88
rs_dumptran marker 87–88
 source database 226
 table replication definition 69
 testing connections 149–152
 version and LTL batch size 185–186
 Replication Server System Database
 See RSSD
 repository
 primary database log devices 85, 96–98, 129–131, 172
 system data 136, 214
 restoring RASD from backup 136
resume command 139–141
 Returns 24
rman_enabled configuration parameter 220
rman_password configuration parameter 221
rman_username configuration parameter 221
rs_charset configuration parameter 222–223
rs_host_name configuration parameter 223
rs_packet_size configuration parameter 224
rs_password configuration parameter 224
rs_port_number configuration parameter 225
rs_replicate_owner_required configuration parameter 225
rs_retry_count configuration parameter 225
rs_retry_timeout configuration parameter 226
rs_source_db configuration parameter 226
rs_source_ds configuration parameter 226–227
rs_ticket command 145
rs_ticket_version configuration parameter 227
rs_username configuration parameter 227–228
 RSSD
 character set 228
 connection from Replication Agent 228, 230
 database name 228–229
 host machine name 229
 port number 230
 Replication Agent user login 229–230, 230
 replication definitions 235
rssd_charset configuration parameter 228
rssd_database_name configuration parameter 228–229
rssd_host_name configuration parameter 229
rssd_password configuration parameter 229–230

rssd_port_number configuration parameter 230
rssd_username configuration parameter 230

S

scan_eol_threshold configuration parameter 231
scan_sleep_increment configuration parameter 231–232
scan_sleep_max configuration parameter 232
 scripts
 automatic running 195
 directory 78, 79
 transaction log creation 78
 setting
 configuration parameters 81
 primary database log device path 85
shutdown command 147–148
 shutting down Replication Agent 147–148
 size of log files 178
skip_ltl_errors configuration parameter 232, 233
 socket port number 167
 primary data server 210
 Replication Agent 166, 167, 168, 169, 170, 212
 Replication Server 225
 RSSD 230
 SQL command execution 12–13, 22, 29
 starting
 replication 139–141
 states of Replication Agent 125–126
 Admin state 83, 125–126, 148–149
 changing 80–81, 139–141, 148–149
 Replicating state 83, 125–126, 139–141
 statistics, performance 115, 125
 resetting 125
 status of Replication Agent 125–126
 stopping Replication Agent 147–148
 stored procedures
 articles in RASD 127
 character case of name 43–44
 disabling replication 50
 enabling replication 43–53
 forcing unmarking 45, 50
 marking 43–53
 name 20–21
 object owner 43–44

Index

- parameters returned 18, 20
- replicate name 51
- unmarking 43–53
- structured_tokens** configuration parameter 233
- suspend** command 148–149
- synonyms of database objects 33, 49, 65–66, 67
- syntax, LTL compression 171
- system data repository
 - backing up 133
 - restoring 136
- system database
 - See* RASD; RSSD
- system log file 6, 152–155

T

- table replication definitions 57, 70
- tables, primary database
 - See* primary tables
- test_connection** command 149–152
- testing connections 149–152, 171
- trace** command 152–155
- trace log file
 - See also* system log file
 - LTL output 153
 - RASD 217
- transaction logs
 - base objects 65
 - creating 78
 - creation script 78
 - database generation ID 13–14
 - DDL operations 127, 128
 - LTM Locator 107–109, 139
 - origin time in LTL 187–188
 - prefix 78, 79, 204
 - primary database devices 85, 96–98, 129–131, 172
 - ra_marker** object 110–111
 - removing 78–79
 - Replication Agent 75–80
 - Replication Agent objects 33, 34
 - rs_dumpdb** marker 87–88
 - rs_dumptran** marker 87–88
 - scanning 231, 232
 - shadow tables 49, 67
 - truncating 74, 75, 185, 234, 235

- transactions
 - help command 102
 - troubleshooting 88
- troubleshooting
 - dummy connection 171–172
 - LTL errors 153–155, 232, 233
 - trace flags 152–155
 - verbose logging 177, 178
- troubleshooting commands
 - database operations 88
- truncating RASD 127, 128
- truncation_interval** configuration parameter 234
- truncation_type** configuration parameter 234–235

U

- UDB
 - connection type 206–207
 - database alias in DSN 208
- unmarking a primary table 56–70
 - all tables 68
 - force** option 60, 66, 68
 - running scripts automatically 195
- unmarking a stored procedure 43–53
 - all stored procedures 50
 - force** option 45, 50
 - running scripts automatically 195
- updating
 - log devices in RASD 129–131, 172
 - LTM Locator 184–185
- use_rssd** configuration parameter 235
- user IDs
 - primary database 210, 213
 - primary database users in RASD 106–107, 128
 - Replication Agent administrator 114, 188–189
 - Replication Server 224, 227–228
 - RSSD user logins 229–230, 230

V

- values
 - configuration parameters 81
 - LTM Locator 101, 107–109
- verbose log output 177, 178

- version
 - articles in RASD 94–95, 99–101
 - primary data server 75
 - primary database users in RASD 106–107
 - Replication Agent 131
 - Replication Agent commands 132
 - Replication Server 185–186
- views of database objects 33, 49, 65–66, 67

W

- wait interval, connection retry
 - primary database 211
- warm standby
 - ra_standby** parameter 219–220
- wrapping log files 178

