

The Complete Sybase Replication Server Quick Reference Guide

***Versions 15.1, 15.2, 15.5, 15.6, 15.7 & 15.7.1
3rd edition***

These pages are taken from "*The Complete Sybase Replication Server Quick Reference Guide*", 3rd edition, as a sample of the contents. Note that the original page size is 11 by 22 cm (4.3 by 8.6 inch).

The complete book can be ordered from <http://www.sypron.nl/repqr> .

Copyright © 2004-2012 Sypron B.V

Rob Verschoor

Sypron Publications

ISBN 978-90-806117-3-3

The Complete Sybase Replication Server Quick Reference Guide

Replication Server versions 15.1, 15.2, 15.5, 15.6, 15.7 & 15.7.1
3rd edition

by Rob Verschoor

ISBN 978-90-806117-3-3

Published by:

Sypron B.V.
Het Wolfseind 24
3823 VS Amersfoort
The Netherlands
Internet www.sypron.nl
Email sypron@sypron.nl

Printed in The Netherlands

First edition	:	July 2004	(RS 12.0, 12.1, 12.5 & 12.6)
Second edition	:	June 2008	(RS 12.6, 15.0.1 & 15.1)
Third edition	:	July 2012	(RS 15.1, 15.2, 15.5, 15.6, 15.7 & 15.7.1)

Sypron is a registered trademark of Sypron B.V.

Sybase, Transact-SQL, Adaptive Server Enterprise, Replication Server, Replication Agent, Adaptive Server Anywhere, SQL Anywhere, OpenServer, OpenSwitch, RepConnector, OmniConnect, DirectConnect, EnterpriseConnect, Enterprise Connect Data Access and ExpressConnect (among others) are registered trademarks of Sybase, Inc. Other product or brand names may be (registered) trademarks of their respective owners.

Copyright © 2004-2012 Sypron B.V. All rights reserved. No part of this publication may be reproduced in any form, or by any means, without the prior written permission of the publisher.

While this book has been prepared with care, neither the author, nor the publisher, nor Sybase Inc. (nor its subsidiaries or parent company), assume any responsibility for errors or omissions, nor do they accept any liabilities for damages resulting from the use of the information herein.

Contents

Introduction	6
Replication Server versions covered in this edition	6
How complete is "Complete"? (topics not covered)	6
How to use this Quick Reference Guide	7
Syntax conventions	7
Terminology and Acronyms	7
Errata	8
Electronic Supplement	8
Undocumented commands: warning & disclaimer	8
Overview & General Concepts	9
1 Starting points for common RepServer DBA tasks	9
2 RepServer basics	9
2.1 Data model considerations for replication systems	10
2.2 Why replication systems can be difficult	11
3 Sybase data replication products	12
4 Products required for heterogeneous replication	12
5 License keys & editions	13
5.1 RepServer Editions	14
6 RepServer documentation	14
RepAgents & Primary Databases	15
7 Marking tables/procedures/databases for replication	15
7.1 Primary ASE databases	15
7.2 Marking data for warm standby or database replication	17
7.3 Be careful when marking big text/image tables in ASE!	18
7.4 Using SQL statement replication (RS 15.2, ASE 15.0.3)	18
7.5 ASE secondary truncation point vs. RSSD locator	19
7.6 Replicating to/from ASE In-Memory databases (15.6)	20
7.7 Primary non-ASE databases	20
8 Replication Agents	20
8.1 ASE RepAgent configuration	21
8.2 ASE RepAgent traceflags (undocumented)	23
8.3 Non-ASE RepAgents	24
Core Replication Functionality	26
9 RepServer syntax basics	26
10 Getting RepServer status information	26
11 Datatypes, datatype classes & -translations	27
11.1 Specifying datatypes in a repdef	28
11.2 Datatype classes and datatype translations	28
11.3 RSSD tables - datatypes	29
12 Connections to RepAgents	29
13 Connections to primary/replicate ASE databases	30
13.1 RSSD tables - database connections	32
13.2 Getting information about database connections	33
13.3 Configuration parameters for database connections	33
14 Connection Profiles for non-ASE databases (15.2)	38
14.1 RSSD tables - connection profiles	39
15 Bulk replication with HVAR/RTL (15.5)	39
15.2 Replicating into Sybase IQ with RTL (15.5)	40
15.3 Configuration parameters for HVAR/RTL (15.5)	41
16 Replication definitions	42
16.2 Table repdefs	42
16.3 Schema changes for replicated tables (15.5)	46
16.4 Schema changes for replicated tables (pre-15.5)	47
16.5 Function repdefs (stored procedure replication)	47
16.6 Applied functions & Request functions	49
16.7 Database repdefs (MSA)	51
16.8 Switching primary/replicate with database repdefs/subs	54
16.9 Warm standby vs. Database repdefs/subscriptions	55
16.10 Replicating data in the ASE 'master' database	55
16.11 'send standby' option in repdefs	56
16.12 Replicating minimal columns	56
16.13 Mixing table replication and procedure replication	57
16.14 Mixing database repdefs and table/function repdefs	58
16.15 Using table repdefs in warm standby, DB replication	58
16.16 RSSD tables - repdefs	59
16.17 Getting information about repdefs	59
17 Replicating text/image/rawobject/LOB columns	59

18	Replicating identity columns.....	61
18.1	Replicating identity column inserts.....	61
18.2	Replicating identity column updates.....	61
18.3	Common problems when replicating identity columns.....	62
19	Replicating ASE timestamp & computed columns.....	62
20	Publications & Articles.....	62
20.1	Getting information about publications and articles.....	63
20.2	RSSD tables - publications and articles.....	64
21	Subscriptions.....	64
21.1	Subscription materialization methods.....	64
21.2	Subscriptions: users and permissions.....	66
21.3	Queues for subscription (de)materialization.....	66
21.4	Automatic materialization: subscription commands.....	67
21.5	Bulk/manual materialization: subscription commands.....	69
21.6	RSSD tables - subscriptions.....	70
21.7	Getting information about subscriptions.....	70
21.8	Subscribing to ASE 'truncate table'.....	71
21.9	Subscribing to bitmaps.....	72
21.10	Common subscription creation problems.....	72
22	Warm Standby ('logical connection').....	73
22.2	ASE warm standby.....	73
22.3	Oracle warm standby (15.5).....	74
22.4	Warm standby commands.....	74
22.5	Tips for setting up an ASE warm standby.....	75
22.6	Warm standby - switching active and standby.....	77
22.7	Switching client applications to the new active database.....	77
22.8	Configuration parameters for logical connections.....	77
22.9	Warm standby - notes and remarks.....	78
22.10	RSSD tables - warm standby.....	79
22.11	Getting information about a warm standby.....	79
23	Functions, Function strings & Function string classes.....	79
23.1	Summary of function/function string concepts.....	79
23.2	Function string classes.....	80
23.3	Function string inheritance.....	81
23.4	Function string commands.....	82
23.5	RSSD tables - functions, function strings & -classes.....	84
23.6	Getting information about function strings.....	84
23.7	Function categories.....	85
23.8	Table repdef-specific functions (data access).....	85
23.9	Function repdef-specific functions ('user-defined').....	86
23.10	Class-specific functions (transaction control).....	86
23.11	SQL input/output templates for function strings.....	88
23.12	System variables in input/output templates.....	89
23.13	Tricks with function strings.....	90
23.14	User-defined functions: obsolete commands.....	91
24	Error classes.....	91
24.1	RSSD tables - error classes.....	93
25	RSSD exceptions log.....	94
26	Routes.....	94
26.2	Route commands.....	95
26.3	RSSD tables - routes.....	96
26.4	Getting information about routes.....	96
26.5	Configuration parameters for routes.....	97
26.6	Direct & indirect routes.....	97
26.7	Site-, System- and Route versions.....	98
27	Configuration parameters.....	99
27.1	Configuration parameters - commands.....	99
27.2	RSSD tables - configuration parameters.....	99
27.3	RepServer-wide configuration parameters.....	100
27.4	Configuration parameters for network security.....	103
28	Parallel DSI.....	104
28.1	Configuration parameters for parallel DSI.....	104
29	Multi-Path Replication (MPR) (15.7).....	106
	Replication System Administration.....	108
30	Installation/setup: the rs_init utility.....	108
30.1	Common rs_init problems.....	108
30.2	Character set & sort order.....	109
30.3	Installation tips on Windows.....	109
30.4	RepServer environment variables.....	109
31	Startup & shutdown.....	109
32	isql & 'interfaces' file.....	110
33	RepServer traceflags (undocumented).....	110

34	Disk space & RepServer devices/disk partitions.....	111
34.1	RSSD tables - disk partitions	112
34.2	Estimating disk space requirements	112
35	Queues	112
35.1	Diagnostic & maintenance commands for queues.....	112
35.2	Configuration parameters for queues	115
35.3	RSSD tables - queues.....	116
36	Special maintenance commands.....	116
37	Users, logins, passwords and permissions.....	116
37.1	RepServer user	116
37.2	ID server user.....	117
37.3	RepAgent user	117
37.4	Primary user (RSSD/ERSDD).....	117
37.5	Maintenance user.....	117
37.6	Route user.....	118
37.7	RepServer subscription user.....	118
37.8	Users, passwords, permissions - Commands.....	118
37.9	Passwords.....	119
38	Connectivity with RS Gateway (15.2).....	120
39	RSSD.....	121
39.1	Classic RSSD (in ASE)	121
39.2	Embedded RSSD/ERSDD (in SQLA)	121
39.3	Configuration parameters for embedded RSSD	122
39.4	Embedded RSSD: miscellaneous SQLA information	122
39.5	List of RSSD tables.....	123
40	Configuration (.cfg) file.....	127
41	Replication system monitoring features.....	127
42	Quiescing a replication system.....	127
43	Determining latency	128
44	Coordinated dumps.....	128
44.1	Restoring coordinated dumps.....	129
45	Database Resynchronization (15.5).....	129
46	Replication system recovery	130
46.1	Recovery-related commands.....	130
	Miscellaneous Topics	131
47	Performance tips.....	131
47.1	Caching RSSD tables.....	132
47.2	Monitor counters/statistics.....	133
48	Generation numbers	135
49	OQID layout.....	136
50	Troubleshooting tips.....	136
50.1	General troubleshooting steps.....	137
50.2	Connectivity problems.....	137
50.3	Problem: Everything looks OK but it doesn't replicate.....	137
50.4	Problem: Transaction log full in primary ASE database.....	138
50.5	Problem: Disk partitions full due to an open transaction.....	139
50.6	If nothing else works.....	139
50.7	If replication is slow.....	139
51	The rs_subcmp utility.....	139
52	Miscellaneous tips, tricks & topics.....	141
52.1	Reverse-engineering RepServer objects.....	141
52.2	Accessing a replicate database by multiple RepServers.....	141
52.3	When you've locked yourself out.....	141
52.4	The 'rs_ticket' feature.....	142
52.5	Commands for waiting.....	143
52.6	Command scheduling (15.5).....	143
53	Non-standard maintenance operations	144
53.1	Renaming a RepServer.....	144
53.2	Renaming a database connection	144
53.3	Changing the maintenance user for a database	145
53.4	Removing a database from the replication system	145
53.5	Moving a RepServer to another host.....	145
53.6	Backing up an entire RepServer.....	146
	Index.....	147

Introduction

Sybase Replication Server was designed as a versatile and powerful software product, and that goal has certainly been achieved. Unfortunately, it seems as if RepServer was never meant to be easy to use -- some of the RepServer syntax isn't easy to remember, and troubleshooting may require you to look for clues in a dozen different places. As a result, a DBA can quickly find him/herself at the wrong side of all that power and versatility.

To be fair, some of these issues can hardly be avoided: replication systems are never simple, and will quickly be an order of magnitude more complex than a single database. Nevertheless, many infrequently used commands and concepts of RepServer tend to slip your mind, especially when your replication system keeps running without any problems -- which, fortunately, is the case most of the time.

Invariably though, the moment comes when you need to take action, and that's when this book will be convenient: it contains all RepServer commands, RSSD procedures and ASE procedures you might need in practice. Apart from describing the basics of each command, I have also included some background information and practically useful tips, as well as a few handy RSSD queries for some of those things you sometimes need to figure out (well, at least I had to...).

When you're a RepServer DBA, there are those moments when you'll need all the help you can get -- this book will be your companion.

I'd like to thank everyone who bought previous editions of this quick reference guide and/or provided input and feedback.

Most of all, my gratitude goes to Simone for her invaluable support and understanding while I was busy writing this book.

Rob Verschoor
Amersfoort, July 2012, 3rd edition

About this Quick Reference Guide

Replication Server versions covered in this edition

The information in this Quick Reference Guide applies to RepServer version 15.1 or later, a Unix/Linux environment, except where indicated otherwise.

Most of this book assumes ASE-to-ASE replication for ASE version 15.0 or later, although most commands and features apply just as well to non-ASE replication or earlier ASE version, though this may not be indicated explicitly.

Briefly, the following RepServer versions are covered in this book:

- **15.1** - the base RepServer version used for this book.
- **15.2** - main new features: SQL statement replication, and connection profiles.
- **15.5** - with the introduction of the new features of HVAR and RTL, RepServer requires significantly more memory to cache replicated data. Therefore, RepServer is now primarily available in 64-bit (32-bit versions are being phased out). Use **admin version** to determine if your RepServer is 32- or 64-bit.
- **15.6** - added support for RTL from Oracle to IQ.
- **15.7** - main new features: multi-path replication (MPR) for fundamentally higher replication throughput.
- **15.7.1** - added support for MPR from/to Oracle and to IQ; also introduced new password security features.

Much of the information in this book is still valid for pre-15.1 RepServer versions, but this may not be indicated explicitly.

Throughout this Quick Reference Guide, notations like '(15.2)' indicate new or changed functionality in the corresponding RepServer version (here, 15.2); likewise, '(pre-15.2)' marks functionality that existed until version 15.1, but not in 15.2 or later.

Unless indicated otherwise, all version numbers refer to RepServer. When versions of other products are involved, this is indicated explicitly, for example 'ASE 15.5 and RS 15.7'. Here, 'RS' indicates RepServer to avoid any confusion.

How complete is 'Complete'? (topics not covered)

I have called this RepServer Quick Reference Guide 'Complete', because it covers almost all aspects of RepServer you may need in daily practice -- although any book about a software product as complex as RepServer must be incomplete by definition. Also, it is obvious that a 160-page booklet simply cannot contain the same amount of information as a bookshelf filled with full-size manuals -- no matter how small you print. For practical reasons, the following topics have therefore been omitted:

- System design aspects are not covered (except for some basic data modelling

aspects; >p.10). Although essential for any replication system, a quick reference guide is not the best place for discussing system design issues. See the *Replication Server Design Guide* (>p.14) for details.

- Only some aspects of heterogeneous replication are covered, such as a selection of the commands for non-ASE RepAgents (>p.24).
- Replication system recovery is not included, apart from mentioning it briefly (>p.130). There is simply too much to say about this rather complex topic, and including a summary would only create a false impression of completeness. Instead, see the *Replication Server Administration Guide* and *Replication Server Troubleshooting Guide* (>p.14).
- Some features are covered only partly, or not at all, for example LDAP, SSL, security mechanisms and High-Availability failover for the RSSD.
- Although most actions can also be performed via Sybase Control Center or Sybase Central, this book is primarily focused on the actual RepServer commands and procedures, as executed manually through `isql`.

Note that this Quick Reference Guide is not a tutorial: the reader is expected to be familiar with the basic concepts of RepServer. Nevertheless, if you're new to RepServer, you may find it useful to read section 2 (>p.9). Also, section 1 contains some starting points for common DBA tasks.

How to use this Quick Reference Guide

All information in this Quick Reference Guide is grouped by topic, such as 'Table repdefs' or 'Routes' (see the table of contents; >p.3). The index (>p.147) contains all commands, functions, stored procedures, etc. in alphabetical order. Many references are included in the index more than once; for example, `rs_helppub` is also included as `helppub`, `rs_`.

Each command description includes the things you need most: the syntax, a short description of the functionality and parameters, and often an example. In addition, there is often a bit of background information and some practical tips. This information will usually be sufficient; when you need more details or background information, see one of the RepServer product manuals, like the *Replication Server Reference Manual* or the *Replication Server Administration Guide* (>p.14).

Syntax conventions

The syntax notation follows commonly used conventions, and will usually be self-explanatory. Still, here are some guidelines:

- Literal text, such as command keywords, is printed in **bold**. Placeholders for variables or parameters are printed in *italic* - you should replace this with something. For example: **drop replication definition** *repdef_name*.
- Parentheses '(' and ')' and commas are part of the command itself. Square brackets '[' and ']', curly brackets '{' and '}', pipes '|', and 3-dots '...' are never part of a command, but indicate optional parts of the command.
- In some cases, the syntax has been simplified for practical reasons, omitting non-essential parts for better readability. For example [**NULL** | *table_name*] is used, although [{ **NULL** | *table_name* }] would have been more formally correct.

Terminology and Acronyms

The following terminology and acronyms are used in this Quick Reference Guide:

RS, RepServer	Replication Server
RepAgent	Replication Agent
LTM	Log Transfer Manager (an old name for RepAgent)
DSI	Data Server Interface, an RS thread connecting to a replicate database
RSI	Replication Server Interface, an RS thread connecting to another RepServer for a route
RCL	Replication Server Command Language; this acronym is used in the Sybase documentation, but not in this book
command	a RepServer language command (alter connection)
statement	an SQL language statement (select, create table)
repdef	replication definition
route	a one-way connection between two RepServers
dataserver	generic name for a database server, either ASE or non-ASE, containing or managing a database
primary dataserver/ database	a dataserver/database containing primary data
replicate dataserver/ database	a dataserver/database containing replicate data
primary table/ procedure	a table/stored procedure, located in a primary database, which is the source of replicated data

replicate table/ procedure	a table/stored procedure, located in a replicate database, which is the target of replicated data
primary RepServer	a RepServer where repdefs are created
replicate RepServer	a RepServer where subscriptions are created; when no routes are used, this is the same as the primary RepServer
RSSD	Replication Server System Database
ERSSD	Embedded RSSD, hosted by SQL Anywhere
primary RSSD	RSSD for a primary RepServer
replicate RSSD	RSSD for a replicate RepServer
.cfg file	the configuration file for a RepServer, typically named 'repserver_name.cfg' (>p.127)
warm standby pair	one active database and one standby database, synchronized by continuous replication, and related to a logical connection that can look like a single database
database replication	database repdefs and -subscriptions (>p.51)
quiescing	stopping the data flow through a replication system to allow maintenance to be performed (>p.127)
replication domain	one or more RepServers registered in the same ID server
heterogeneous replication	replication between different types of databases, for example from ASE to Oracle
bi-directional replication	a replication architecture where a database is both a primary and a replicate database at the same time
system [stored] procedures	database-independent ASE stored procedures, starting with sp_* (sp_setreptable), and residing in the sysystemprocs database
ASE	Adaptive Server Enterprise
SQLA	SQL Anywhere
ASA	Adaptive Server Anywhere, old name for SQL Anywhere
DDL	SQL Data Definition Language statements (create table , create database , etc.)
DML	SQL Data Manipulation Language statements (select , insert , update , delete , etc.)
LTL	Log Transfer Language; used for communication between RepAgents and RepServer
OQID	Origin Queue ID, a 36-byte identifier for each transaction
T-SQL	Transact-SQL, the SQL implementation by Sybase
bcp	an OS-utility to 'bulk copy' data from files to ASE tables and vice versa
OS	Operating System
a.k.a.	also known as
>p.49 / >19.5	see page 49 / see section 19.5

Errata

Errata will be available on-line at www.sypron.nl/repqr/errata.html. Please email any errors or comments to repqr@sypron.nl.

Electronic Supplement

All RSSD queries appearing in this book, as well as some additional RSSD queries and handy RSSD procedures, are available electronically for owners of this book. To download a copy, go to www.sypron.nl/repqr/supplement.

Undocumented commands, warning & disclaimer

Most information in this Quick Reference Guide concerns documented and supported functionality. However, some useful undocumented or unsupported commands are included as well. Please note that using undocumented or unsupported commands may constitute a risk: such commands may have unexpected side effects, may cause irreversible damage to your databases or replication system and/or lead to loss of data. Use such commands entirely at your own risk, and do not contact Sybase Technical Support for assistance. When a command is not documented (and therefore not supported), this is indicated in the description.

In any case, neither the author, nor the publisher, nor Sybase Inc. (nor its subsidiaries or parent company), assume any responsibility for errors or omissions, nor do they accept any liabilities for damages resulting from the use of the information in this book.

Overview & General Concepts

1 Starting points for common RepServer DBA tasks

Task	Recommended action
Create a new RepServer	Use rs_init (>p.108)
Start or stop a RepServer	See section 31 (p.108)
Determine the current status of a RepServer	Use RepServer commands like admin who (>p.26)
Determine the RSSD for a RepServer	Run admin rssd_name and/or sysadmin erssd . See section 39 (>p.121)
Run an RSSD stored procedure (like rs_helprep)	First, connect to the RSSD database server with isql (>p.110); then run the RSSD procedure
Create a connection to a primary or replicate database	Use rs_init (>p.108)
Set up publish-subscribe replication for tables or stored procedures	<ol style="list-style-type: none"> 1. Create connections to a primary and a replicate database (see above) 2. Create a repdef (for a primary table or stored procedure; >p.42,47). Multiple repdefs can also be grouped in a publication (>p.62) 3. Mark the primary table or stored procedure for replication (>p.15) 4. Create a subscription to the repdef or publication (>p.64)
Set up a warm standby	See section 22.5 (>p.75)
Set up database replication	<ol style="list-style-type: none"> 1. Create connections to a primary and replicate database (see above) 2. Create a database repdef (>p.51) 3. Mark the database for replication (>p.15) 4. Create one or more subscriptions to the database repdef (>p.64)
Enable password encryption for a RepServer	Use rs_init (>p.108)
Reverse-engineering of RepServer objects	Use RMP, RSP or PowerDesigner (>p.141)
Change configuration parameters	See section 27 (>p.99)
Find out why replication isn't working	See section 50 (>p.136)
Improve performance of your replication system	See section 47 (>p.131)
Set up an ASE-to-ASE or Oracle-to-Oracle replication-reference environment	(15.5) Follow the instructions in the <i>Replication Server Administration Guide, Vol. 2, appendix Implement a Reference Replication Environment</i>

If you need to create RepServers, database connections, warm standbys (etc.) regularly, it is recommended to create an **rs_init** resource file that can be used as a template. This allows you to run these common operations from the command line with **rs_init -r** instead of having to go through the interactive **rs_init** dialog every time. For more information on creating an **rs_init** resource file, see p.108.

2 RepServer basics

This section briefly describes some elementary aspects of RepServer:

- RepServer is Sybase's enterprise-level data replication product for log-based, publish-subscribe replication. RepServer provides best performance and functionality for ASE-to-ASE replication, but also supports heterogeneous replication (between different database types, like MS SQL Server, DB2 and Oracle).
- RepServer's unit of replication is a **transaction** (not just an individual row modification): either the entire transaction is successfully applied to the replicate database, or it isn't applied at all. Only committed transactions are replicated.
- By default, RepServer guarantees that replicated transactions will commit in the same order in the replicate database as they did in the primary database. When using HVAR or RTL in 15.5 (>p.39), or MPR in 15.7 (>p.106), this guarantee is relaxed as transactions can be merged or split up by RepServer.
- By default, RepServer replicates the **effect** of a DML statement (i.e. the changes to the data), not the DML statement itself. In 15.2, when enabling SQL state-

outside the database server (as for non-ASE databases).

8.1 ASE RepAgent configuration

The ASE RepAgent is a thread inside ASE. The ASE procedures in this section should be executed in the primary ASE database.

When RepServer has no connection for a particular database, but the ASE RepAgent tries to connect anyway, the ASE RepAgent will not start and ASE error 9261 will appear in the ASE errorlog ('aborting due to an unrecoverable communications or Replication Server error'). To resolve this, either create a database connection in RepServer (see section 13; >p.30), or disable the ASE RepAgent.

sp_configure 'enable rep agent threads' {, 1 | 0 }

(ASE procedure) Enables (1) or disables (0, =default) the ASE RepAgent (dynamic).

(ASE 15.7) sp_configure 'replication agent memory size' {, nr_of_2KB_pages }

(ASE proc/ASE config) The number of 2KB memory pages for the ASE RepAgent (dynamic). Default= 4096 (=8MB); this may need to be increased when replicating many tables and columns.

sp_help_rep_agent [database_name | NULL [, filter]]

(ASE procedure) Displays information about all RepAgent threads in this ASE server, or only for the specified *database_name*. *filter* ('recovery', 'config', 'process', 'scan', 'security' or 'all') limits the output to specific aspects.

To show which RepAgents are running, run **sp_help_rep_agent null, process**. To list the databases where a RepAgent is configured, run **sp_config_rep_agent**.

sp_start_rep_agent database_name [, 'recovery_foreground'] [, 'connect_dataserver', 'connect_database' [, 'repserver_name', 'repserver_username', 'repserver_password']]

(ASE procedure) Starts the RepAgent thread for *database_name*, where the secondary truncation point must be valid. For normal operations, only *database_name* is specified. To start the RepAgent in recovery mode, 'recovery' must be specified; with 'recovery_foreground', progress messages are sent to the client. For the optional other options, see **sp_config_rep_agent**.

Once started, the RepAgent is automatically started after a future ASE restart.

sp_stop_rep_agent database_name [, 'nowait']

(ASE procedure) Stops the RepAgent for this database; 'nowait' stops immediately. Once stopped, the RepAgent is not automatically started after an ASE restart.

sp_config_rep_agent [database_name [, option, value [, ...more values...]]

(ASE procedure) Without parameters, lists the databases where a RepAgent is configured. With only *database_name*, displays all configuration settings for the RepAgent in that database. With *option*, displays the current setting; with also *value*, modifies the setting. Most changes take effect once the RepAgent is restarted.

RepAgent config settings are stored in the **sysattributes** table in *database_name*.

Examples: **sp_config_rep_agent prod_db, 'retry timeout', '20'**

sp_config_rep_agent prod_db, 'send warm standby xacts', true

option and *value* can be one of the following (note that is sufficient to specify a unique substring for *option*; also note that numeric values must be quoted):

- 'enable', 'repserver_name', 'repserver_username', 'repserver_password' - configures the RepAgent for connecting to the specified RepServer with the specified username and password; also sets the secondary truncation point to valid in the database. *repserver_name* must be defined in the **interfaces** file used by this ASE server.
- 'disable' [, 'preserve secondary truncpt'] - removes the RepAgent configuration for this database. With 'preserve secondary truncpt', leaves the secondary truncation point unchanged; otherwise (=default), it is disabled.
- 'rs servername', 'repserver_name' - the name of the RepServer to connect to.
- 'rs username', 'repserver_username' - the RepServer username to be used for connecting to the RepServer.
- 'rs password', 'repserver_password' - the password to be used for connecting to the RepServer.
- 'connect dataserver', 'dataserver_name' - when RepAgent is started in recovery mode to scan the log of a database dump (and/or log dumps) that were restored into a different database, specifies the name of the original dataserver.

This datatype conversion can be done on two levels:

- For an individual primary column (or parameter) in a repdef. For each column (or parameter), a datatype must be specified. By adding the **map to datatype** clause to the column declaration in the repdef (see above), all incoming values for that column will be translated to the specified datatype before being replicated further through the replication system.
The official RepServer documentation calls this 'column-level translations'.
- For all data replicated to a replicate database. By creating datatype classes and datatype translations through the predefined SQL scripts mentioned below, a set of datatype translations is attached to a function string class. For a replicate database using that function string class, all data values replicated to that database are converted by the DSI according to those translations (only for those datatypes for which a translation has been defined).
The official RepServer documentation calls this 'class-level translations'.

The two types of translations can be mixed. Column-level translations (**map to** in the repdef) are applied before class-level translations (performed by the DSI).

Setting up datatype classes/translations is simpler in 15.2 than in pre-15.2:

- 15.2 provides predefined connection profiles (➤ p.38). When specifying a connection profile name with **create connection .using profile**, the required class-level translations are created automatically.
- In pre-15.2, to initialize datatype classes, datatype definitions and datatype translations for heterogeneous replication, certain predefined SQL scripts, located in the **\$\$SYBASE/\$SYBASE_REP/scripts** directory, must be executed in the RSSD by the DBA. See p.38 for more information.

11.3 RSSD tables - datatypes

Datatype classes are stored in the RSSD table **rs_classes**, with **rs_classes.classstype = 'D'**. For each datatype class, the RSSD table **rs_datatype** contains datatype definitions for that class.

The RSSD table **rs_translation** contains datatype translations for a particular function string class (the so-called 'class-level translations'). The RSSD table **rs_columns** contains the datatype conversions specified in the repdef with **map to** (the so-called 'column-level translations'), in columns **declared_dtid** (the declared datatype) and **publ_dtid** (the published datatype).

The RSSD **rs_*** procedures do not display any information about datatype translations. Instead, query the RSSD tables directly.

(Note: the exact relationship between datatype classes and function string classes remains unclear; in fact, there may not even be a direct relationship - the RepServer documentation just doesn't contain much information about this topic.)

In pre-15.2, 'empty' datatype classes, like **rs_oracle_dt_class** (in 15.5 also **rs_iq_dt_class**), are defined in the RSSD table **rs_classes**, with no corresponding rows in **rs_datatype**. Only the standard class **rs_rs_dt_class** is fully defined. The SQL scripts in **\$\$SYBASE/\$SYBASE_REP/scripts** insert rows into these RSSD tables, thus creating the datatype definitions and translations for these classes.

In 15.2, some (not all) additional datatype classes are defined by default. When a connection is created, additional definitions are added depending on the connection profile specified.

In principle, it should be possible to create your own datatype classes and corresponding translations. Because no RepServer commands exist for this purpose, it would require manually inserting rows into the RSSD tables **rs_classes**, **rs_datatype** and **rs_translation**. Unfortunately, exactly how to define such rows is mostly undocumented, but the SQL scripts mentioned above, and the RSSD table descriptions in the *Replication Server Reference Manual* (➤ p.14), may be a source of inspiration. Creating your own datatype classes and -translations is probably not officially supported by Sybase.

12 Connections to RepAgents

resume distributor *dataserver.database* [**skip transaction**]

Resumes a suspended distributor (DIST thread) for a connection to a primary database. The distributor matches incoming primary transactions against repdefs and subscriptions, writing replicated transactions from the inbound queue into outbound queues. There is one DIST thread for each inbound queue.

skip transaction deletes the first transaction from the inbound queue; this may be needed when the distributor shuts down due to invalid incoming data (for example, when incoming primary data is incompatible with the corresponding repdef datatype).

The skipped transaction is written into the RSSD exceptions log tables (➤ p.94), and

with `rs_init` as for any other database.

- An ASE RepAgent will be running in the primary **master** database, so its transaction log may need to be enlarged to ensure it does not fill up.
- The primary **master** database must be marked for replication with `sp_reptostandby 'master', 'all'`.
- The ASE RepAgent in the primary **master** database must be configured with `'send warm standby xacts' = true` (>p.22)
- The **master** database can be replicated either as a warm standby, or as a database replication definition and corresponding subscription.
- When using a database repdef for the ASE **master** database, the repdef properties **replicate dll** and **replicate system procedures** must both be enabled (or the DDL commands and `sp_*` calls above, respectively, will not be replicated).
- Materialization of the replicate **master** database should be done manually.
- Replication of the **master** database does not replicate the actual `uid/suid` values for new logins and roles. Rather, it relies on identical contents for **syslogins** and **sysrroles** on both sides so that the algorithms will pick the same values for new logins or roles.
- Since passwords for logins and roles are replicated as ciphertext, cross-platform synchronization is not supported (though it may actually work in some cases).
- Ensure that the `sp_*` procedures listed above are executed from within the **master** database if they are to be replicated.

16.11 'send standby' option in repdefs

send standby can be specified as an option in table repdefs (**send standby all columns**; **send standby replication definition columns**) and function repdefs (**send standby all parameters**; **send standby replication definition parameters**). This option applies only to a warm standby, and to database replication (MSA) when combined with table or function repdefs.

The absence or presence of **send standby** affects how primary data is replicated:

- **Replicated columns:** for a table repdef without a **send standby** option, or with only **send standby** (which is identical to **send standby all columns**), all columns of the primary table are replicated. With **send standby replication definition columns**, only the columns declared in the repdef are replicated.
- **Primary key columns:** for a table repdef with a **send standby** option (no matter which one), only the **primary key** columns of this repdef are used in the **where**-clause of replicate SQL **update delete** statements. Without a **send standby** option, the union of all **primary key** columns for all repdefs for this table is used in the **where**-clause (Completeness note 1: without a repdef, all table columns are used in the **where**-clause of replicate SQL statements. Completeness note 2: when multiple repdefs exist for a table, only one repdef can have a **send standby** option enabled at a time).
- **Replicated parameters:** for a function repdef without a **send standby** option, or with only **send standby** (which is identical to **send standby all parameters**), all parameters of the primary stored procedure are replicated. With **send standby replication definition parameters**, only the parameters declared in the repdef are replicated.

For a warm standby only, the **send standby** options in the repdefs can be overridden by the configuration parameter `send_standby_repdef_cols` for the logical connection (>p.78). NB: the current **send standby** setting for a repdef is not displayed by any RSSD procedure (the column **Used by Standby** of `rs_helprep` reflects if the repdef is used by a standby, but not the **send standby** clause).

16.12 Replicating minimal columns

'Minimal column replication' is a feature to improve performance for replicating **update** and **delete** statements on primary tables. Without minimal column replication, all columns for a modified row are replicated, including columns that have not changed. In contrast, minimal column replication only replicates those columns whose values have really been changed, in addition to the primary key columns. Especially for tables with many and/or long columns, minimal column replication can improve performance in the DSI (due to more efficient replicate SQL statements), over the network and in the stable queues (due to the replicated messages being shorter). Minimal column replication is specified in a table repdef either as **replicate all columns** (disabled; all columns are replicated; =default) or **replicate minimal columns**

electronic supplement to this book (➤p.8) or **admin config** (➤p.99).

21.9 Subscribing to bitmaps

RepServer supports the special datatype **rs_address**, which can be used for subscribing to bitmaps (i.e. specific bits in a column or parameter). This can be useful when individual bits have a functionally relevant meaning, for example in status fields where a group of on/off indicators is packed together in one 4-byte integer.

Warning: before using bitmap subscriptions, please pay attention to the following. With bitmap subscriptions, a primary update will not be replicated if only **rs_address** columns are updated (unless a change in the bitmap would cause a row to migrate out of its current subscription or into a different subscription); this means that the bitmap values in replicate data rows could potentially get out of sync with the bitmap values in the primary rows. Therefore, you should verify that the actual update statements on your primary data are suitable for bitmap subscriptions.

To subscribe to specific bits, use the **&** operator and a bitmask value in the **where**-clause of **create/define subscription** or **create article**. This will perform a logical 'AND' on the actual value of the column/parameter and the bitmask; when the result is non-zero, the row qualifies for the subscription.

In the example below, a bitmap indicates auxiliary employee skills. Let's assume bit 0 indicates a 'First Aid' qualification, bit 1 indicates Dutch language ability and bit 2 indicates VMS experience (indeed, not a well-normalized relational data model...):

```
create replication definition EmployeeAuxSkills with primary at PROD.staff_db
(EmpNo int, EmpName varchar(50), Skills rs_address) primary key (EmpNo)
searchable columns (Skills)
```

```
create subscription FirstAid_sub for EmployeeAuxSkills
with replicate at ADMIN.office_mgmt_db
where Skills & 1 -- folks with a First Aid qualification... (bit 0: 20 = binary 001)
and Skills & 2 -- ... who also speak Dutch (bit 1: 21 = binary 010)
```

```
create subscription VMS_sub for EmployeeAuxSkills
with replicate at ADMIN.diehards_db
where Skills & 6 -- folks who speak Dutch or know VMS
-- (bits 1+2: 21 + 22 = binary 110)
```

Note the difference between the two **where**-clauses: the first subscription requires both bits 0 and 1 to be set; the second subscription requires either bit 1 or 2, or both (this way, bitmap subscriptions can sometimes be used to simulate a logical 'OR' condition in the **where**-clause).

The bitmask can in principle also be specified as an 8-digit hexadecimal value: on Linux for example, **Skills & 0x00000006** is equivalent to **Skills & 6** above. However, this may not work cross-platform or on other platforms, since platform-specific byte ordering may produce unexpected results when using such hexadecimal values; with integer values, this problem does not apply.

rs_address represents a 4-byte (32-bit) datatype. Columns declared as **rs_address** in the repdef should normally be 4-byte integers in the primary table. **rs_address** columns cannot be part of the **primary key** clause in a repdef.

Although undocumented (and probably unsupported by Sybase), some other column datatypes -such as ASE's **smallint**, **tinyint** and **binary** (if not longer than 4 bytes)- can also be used for **rs_address** bitmaps, however, when using this, please note that you are fully responsible yourself for issues such as alignment and byte swapping!

21.10 Common subscription creation problems

When creating a subscription, the most common problems are:

- **Permission problems:** during materialization, the maintenance user does not have sufficient permissions for accessing tables in the replicate or primary database. Such permission problems invariably cause the DSI to be shut down; the RepServer errorlog will contain error messages with respect to this problem. When such permission problems occur, they can usually be solved by assigning the required permissions and resuming the DSI.
- **Disk partitions get full:** when a primary table is large, atomic materialization may cause the disk partitions to fill up (or at least, take a long time to complete). If this happens, one approach is to add as many disk partitions as needed, and be patient. However, it may be better to abort subscription creation (see below) and use a different method instead, like 'incremental atomic' materialization (➤p.64).
- **Replicate transaction log gets full:** when a primary table is large, atomic materialization may also cause the transaction log of the replicate database to fill up,

- 'derived class' - a function string class which has a parent class;
- 'base class' - a function string class which has no parent class.

A derived class inherits all function strings from its parent class; hence the name 'function string inheritance':

- When altering a function string in a parent class, this modification is immediately inherited by all derived classes for this parent class where this function string was not overridden (see next bullet).
- When creating a function string in a derived class, this overrides the inherited function string from the parent class (but does not affect the function string in the parent class itself).

When creating or modifying a function string, this takes effect immediately on the local RepServer; when there is an outgoing route, the modification takes effect in the destination RepServer once it has arrived in the destination RSSD.

Function string inheritance can have multiple levels: a parent class can also be a derived class at the same time. Parent classes can be established or changed with **create function string class** or **alter function string class**.

When creating a base class (i.e. a function string class without a parent class), all function strings must be explicitly created for the base class (with **create function string**); otherwise, any attempt to access a database with this function string class will cause errors in the DSI.

Notes:

- **rs_sqlserver_function_class** (the default class used for ASE connections) cannot be used as a parent class; use **rs_default_function_class** or another function class instead.
- Function strings cannot be created or modified for **rs_default_function_class** or **rs_db2_function_class** instead, create a derived class and create customized function strings in that class.
- Function string inheritance is dynamic (unlike the static class initialization for error classes with the RSSD procedure **rs_init_erroractions**: the latter is a one-time operation that is not maintained further).

23.4 Function string commands

```
{ create function string repdef_name.function_name[,function_string_name] |
  create function string function_name }
for [(15.7.1) function class ] function_class_name
[ with overwrite ]
[ scan 'input_whereclause_template' ]
[ output
  { language 'output_sql_template' |
    rpc 'execute procedure_name [ @param= ] { constant | ?variable!modifier? } [,...]' |
    writetext [ use primary log | with log | no log ] |
    none } ]
```

Creates or modifies a function string for the specified function and function string class (a.k.a. a 'custom' function string), either for a table repdef or function repdef. This takes effect immediately in the primary RepServer for the function string class; for a replicate RepServer, it takes effect once the modification arrives at the replicate RSSD. **create function string...with overwrite** is identical to **alter function string**.

(15.7.1)

```
create function string [owner]object_name.function_name[,function_string_name]
for database dataserver.database
...other clauses same as classic syntax above...
```

Creates or modifies a 'target-specific' function string for a replicate table or stored procedure. Unlike the classic syntax above, these function strings are unrelated to a repdef or function string class, and can therefore be used in combination with table redefs, database redefs, or a warm standby. This can also be used together with automatic primary key detection (➤p.58). Target-specific function strings can be created for the same functions as for table repdefs, except for **rs_select[with_lock]** (➤p.85); for stored procedures, the function name is identical to the procedure name. To display these function strings, use **rs_helpobjfstring** (➤p.85).

If the object is not owned by the maintenance user, the object owner should be specified or the function string won't work. No error is raised when the object doesn't exist.

Examples: **create function string Accounts_rdf.rs_delete**
for rs_sqlserver_function_class with overwrite output language
'delete Accounts where AccountNo = ?AccountNo!old? ;
insert LogTable values (getdate(), ?rs_origin_user!sys?,
"Deleted Account# ?AccountNo!old?")'

40 Configuration (.cfg) file

The RepServer configuration file (referred to as **.cfg** file in this book) is an ASCII file read by RepServer during startup to determine the server name, login and password for the ID server and RSSD server. Because the **.cfg** file contains passwords, it should be protected properly on file-system level.

By default, the **.cfg** file is located in **\$\$SYBASE/\$SYBASE_REP/install** and named **repserver_name.cfg**. The actual location of the **.cfg** file is passed to RepServer with the **-C** command line parameter in the **RUN_servername** file (>p.110).

The **.cfg** file contains some rather self-explanatory keywords. It is not recommended to edit the **.cfg** file manually, unless you understand what you're doing. When the **.cfg** file does not exist or contains invalid keywords, RepServer will not start. The **.cfg** file can also contain traceflags (>p.110). When the **.cfg** file contains encrypted passwords, these passwords can only be modified through **rs_init** (>p.108). When a line starts with **#**, the entire line is considered a comment.

For RepServers with an embedded RSSD that have outgoing routes, the ERSSD RepAgent also has a file ending in **.cfg**, by default located in **\$\$SYBASE** (>p.95).

41 Replication system monitoring features

Over the years, RepServer has included different tools and features for monitoring the status of the various RepServers and dataservers in the replication system (not to be confused with the monitoring counters in section 47.2, which are about the internal workings of a single RepServer). A quick overview of the various abbreviations:

- RMP - Replication Manager (Plug-in for Sybase Central)
- RMS - Replication Monitoring Services
- RSM - Replication Server Manager (pre-15.0)
- RSP - RepServer Plug-in for Sybase Central (pre-15.0)

RMP and RMS are no longer part of the default RepServer installation, but can be downloaded from <http://downloads.sybase.com> (under 'Replication Server') and installed in the RepServer environment.

Most of the functionality of these tools has been moved into Sybase Control Center (SCC). Some aspects, such as reverse-engineering of RepServer constructs, is not supported by SCC, but still requires RMP (or Sybase PowerDesigner).

Sybase Control Center (SCC) is a web-based administration and monitoring tool for RepServer (it can also monitor ASE and IQ), supporting RS 15.0 and later. SCC must be downloaded from <http://downloads.sybase.com> (under 'Sybase Control Center') and installed separately (optionally, in the same RepServer environment or elsewhere; also optionally, on a dedicated host). Note that recent versions of RS, ASE and IQ install a directory **\$\$SYBASE/SCC-X_Y** which in some cases do not contain the full SCC tool, but only the SCC agent that must be started (with **\$\$SYBASE/SCC-X_Y/bin/scc.sh**) in order to monitor/manage a product through SCC. It may be best to always download and install the latest SCC version.

Once SCC is installed and started, go to <https://scc-host:8283/scc> (default login/password=**sccadmin**/blank); **scc-host** is the host on which SCC is installed. See the SCC manuals for further details.

42 Quiescing a replication system

A replication system is said to be 'quiesced' (or 'quiescent') when all RepServers have processed all incoming messages and delivered all outgoing messages, when there are no open transactions, and when no subscription (de)materialization is in progress. Quiescing a replication system or an individual RepServer is sometimes required, for example to modify the schema of a replicated table, or for troubleshooting.

To determine whether a RepServer is currently quiesced, run **admin quiesce_check**. Unless all log transfer and incoming routes have been suspended, there is no guarantee that a quiesced state will be maintained.

To quiesce a replication system, perform these steps:

1. In all RepServers, run **suspend log transfer from all**.
2. In all RepServers with outgoing routes, run **admin quiesce_force_rsi** (>p.97), forcing RepServer to send all messages in the queues for its routes. Wait briefly before proceeding with the next step, to allow the messages to be processed.
3. In all RepServers, run **admin quiesce_check** to determine whether the RepServer is quiesced. For those RepServers that are not quiesced, repeat step 2.
4. Finally, run steps 2 and 3 one last time.

Once a quiesced RepServer can return to its normal state, first resume log transfer for the RSSD (if the RSSD is replicated) before resuming log transfer for the other primary databases.

Index

- #temporary tables (ASE) 10
 - & (rs_address bitmask operator)..... 72
 - .cfg file (configuration file)..... 127
 - 32/64-bit 6, 98
- ## A
- abort switch 75
 - activate subscription 69
 - active database (in warm standby) 73
 - active-standby pair..... 73
 - add partition..... 111
 - address, rs_ (datatype)..... 27, 72
 - ad-hoc SQL statements, replicating 91
 - admin config 99
 - admin cwd 27
 - admin disk_space 26, 111
 - admin echo 27
 - admin get_generation 20, 136
 - admin health 26
 - admin log_name 27
 - admin logical_status 27, 79
 - admin pid 27
 - admin quiesce_check 128
 - admin quiesce_force_rsi 97
 - admin rssid_name 27, 121
 - admin 'schedule' 143
 - admin security_property 104
 - admin security_setting 104
 - admin set_log_name 27
 - admin show_connection_profiles 39
 - admin show_connections 33, 97
 - admin show_function_classes 81
 - admin show_route_versions 98
 - admin show_site_version 98
 - admin sqm_readers 113
 - admin stats 133, 134
 - admin stats tps / cps / bps 134
 - admin stats, backlog 134
 - admin stats, cancel 134
 - admin stats, reset 134
 - admin stats, status 134
 - admin time 27
 - admin translate 28
 - admin verify_repserver_cmd 48
 - admin version 26, 98
 - connection 99
 - admin who 26
 - admin who, dist 30
 - admin who, dsi 33, 135
 - admin who, rs 97
 - admin who, sqm 112, 135
 - admin who, sqt 113
 - admin who is down 26
 - admin who is up 26
 - Advanced Services Option 39
 - Advanced Services Option (ASO) 13
 - agent (SOCC) 127
 - alias to dbo, maintenance user 108
 - allow connections 30, 130
 - alt_writetext, dsi_ (config) 36
 - alter applied function replication def..... 49
 - alter connection 31
 - for replicate table named 41
 - set command retry 93
 - set error class 93
 - set function string class 81
 - alter connector 32
 - alter database replication definition 53
 - alter encryption key 120
 - alter error class 93
 - alter function 91
 - alter function replication definition 49
 - alter function string 84
 - alter function string class 81
 - alter logical connection 71, 74
 - alter partition 111
 - alter queue 115
 - set sqm_cache_enable 115
 - set sqm_cache_size 115
 - set sqm_cmd_cache_size 115
 - set sqm_max_cmd_in_block 115
 - set sqm_page_size 115
 - set sqm_xact_with_large_msg 116
 - alter replication definition 44
 - alter request function replication def..... 49
 - alter route 96
 - alter schedule 143
 - alter subscription 70
 - alter user 118
 - alternate connection 31, 106
 - logical 74
 - always_replicate (text/image data) 59
 - applied functions 47, 49
 - apply_truncate_table, sysadmin 71
 - articles, for publications 62
 - articles, rs_ (RSSD table) 64, 123
 - ASA See SQLA (SQL Anywhere)
 - asa_function_class, rs_ (class) 80
 - ascii_pack_ibq (config) 38
 - ASE environment variables 109
 - ASE errorlog, location 137
 - ASE RepAgent 21
 - ASE Replicator 12
 - ASO (Advanced Services Option) . 13, 39
 - assign action 92
 - async_parser (config) 38
 - asyncfuncs, rs_ (RSSD table) 84, 123
 - atomic materialization 64
 - atomic materialization, incremental 65
 - audit_dest (config) 100
 - audit_enable (config) 100
 - auditing 100
 - autocorrection 45
 - and database repdefs 46
 - and function strings 46
 - and warm standby 46
 - automatic materialization 65, 67
- ## B
- backlog, in queues 134
 - backup, of a RepServer 146
 - base class 82
 - Basics, RepServer 9
 - batch (config) 33
 - batch_begin (config) 34
 - batch_end, rs_ (function) 87
 - batch_start, rs_ (function) 87
 - begin, rs_ (function) 86
 - bitmap subscriptions 72
 - block_size (config) 101
 - bulk materialization 65, 69
 - bulk_copy, dsi_ (config) 36
 - bulk_threshold, dsi_ (config) 36

byte_order (config) 101

C

caching RSSD tables 132
 canonical interface 12, 37
 capacity, rs_ (RSSD proc) 112
 captable, rs_ (RSSD table) **112**, 123
 CDB 39, **40**, 42
 cdb, sysadmin 40
 cdb_max_size, dsi_ (config) 42
 change database (CDB) 39, **40**, 42
 change password, RepServer user 118
 character set in replication system 109
 charset_convert, dsi_ (config) 34
 check publication 63
 check subscription **68**, 70
 check_repl, rs_ (function) 87
 classes, rs_ (RSSD table) 84, 93, **123**
 classes,rs_ (RSSD table) 29
 class-level translations (datatype) 29
 clsfunctions, rs_ (RSSD table) **84**, 124
 cm_fadeout_time (config) 101
 cm_max_connections (config) 101
 cmd_batch_size, dsi_ (config) 34
 cmd_direct_replicate (config) 37
 cmd_direct_replicate, dist_ (config) 37
 cmd_prefetch, dsi_ (config) 34
 cmd_separator, dsi_ (config) 34
 column-level translations (datatype) 29
 columns, rs_ (RSSD table) 124
 command scheduling 143
 command_convert, dsi_ (config) 41
 command_retry (config) 34
 comments, syntax 26
 commit, rs_ (function) 86
 commit_check_locks_intrvl, dsi_ (conf) 106
 commit_check_locks_log, dsi_ (conf) 106
 commit_check_locks_max, dsi_ (conf) 106
 commit_control, dsi_ (config) 106
 common DBA tasks, starting points 9
 compilation (HVAR/RTL) 39
 compile_enable, dsi_ (config) 41
 compile_max_cmds, dsi_ (config) 41
 compile_retry_threshold, dsi_ (config) 42
 complexity of replication systems 11
 compressed data, replicating 28
 computed columns 62, 87
 config, admin 99
 config, rs_ (RSSD table) **99**, 124
 config_rep_agent, sp_ (ASE proc) 21
 configuration file (.cfg file) 127
 Configuration parameters 99
 and admin config 99
 ascii_pack_ibq 38
 async_parser 38
 audit_dest 100
 audit_enable 100
 batch 33
 batch_begin 34
 block_size 101
 byte_order 101
 cm_fadeout_time 101
 cm_max_connections 101
 cmd_direct_replicate 37
 command_retry 34
 current_rssd_version 101
 db_packet_size 34
 deferred_name_resolution 37, 78
 deferred_queue_size 101

disk_affinity 33, 97, 111, 112
 dist_cmd_direct_replicate 37
 dist_direct_cache_read 101
 dist_sqt_max_cache_size 35
 dist_stop_unsupported_cmd **36**, 78
 dsi_alt_writetext 36
 dsi_bulk_copy 36
 dsi_bulk_threshold 36
 dsi_cdb_max_size 42
 dsi_charset_convert 34
 dsi_cmd_batch_size 34
 dsi_cmd_prefetch 34
 dsi_cmd_separator 34
 dsi_command_convert 41
 dsi_commit_check_locks_intrvl 106
 dsi_commit_check_locks_log 106
 dsi_commit_check_locks_max 106
 dsi_commit_control 106
 dsi_compile_enable 41
 dsi_compile_max_cmds 41
 dsi_compile_retry_threshold 42
 dsi_connector_type 37
 dsi_dataserver_make 37
 dsi_exec_request_sproc 34
 dsi_fadeout_time 34
 dsi_ignore_underscore_name 105
 dsi_isolation_level 36
 dsi_keep_triggers 34
 dsi_large_xact_size 105
 dsi_max_cmds_in_batch 34
 dsi_max_cmds_to_log 34
 dsi_max_text_to_log 34
 dsi_max_xacts_in_group 34
 dsi_non_blocking_commit **37**
 dsi_num_large_xact_threads 105
 dsi_num_threads 105
 dsi_partitioning_rule 105
 dsi_quoted_identifier 34
 dsi_replication 35
 dsi_replication_ddl 35
 dsi_row_count_validation 37
 dsi_rs_ticket_report 35, **142**
 dsi_serialization_method 105
 dsi_sql_data_style 35
 dsi_sqt_max_cache_size 35
 dsi_stage_all_ops 42
 dsi_text_convert_multiplier 35
 dsi_timer 37
 dsi_xact_group_size 34
 dump_load 35
 dynamic_sql, for DSI 36
 dynamic_sql_cache_management 36
 dynamic_sql_cache_size 36
 enable_rep_agent_threads (ASE) 21
 erssd_backup_dir 122
 erssd_backup_interval 122
 erssd_backup_start_date 122
 erssd_backup_start_time 122
 erssd_ra 122
 exec_cmds_per_timeslice 35
 exec_max_cache_size 37
 exec_nrm_request_limit 35
 exec_prs_num_threads 38
 exec_sqm_write_request_limit 35
 ha_failover 101
 id_msg_confidentiality 104
 id_msg_integrity 104
 id_msg_origin_check 104
 id_msg_replay_detection 104
 id_msg_sequence_check 104
 id_mutual_auth 104

id_security_mechanism	104	sql_max_prs_size	38
id_server	101	sql_max_read_delay	103
id_unified_login	104	sre_reserve	102
init_sqm_write_delay	101	stage_operations	42
init_sqm_write_max_delay	101	standby_func_class	79
materialization_save_interval	78	stats_reset_rssd	133
md_sqm_write_request_limit	35	stats_sampling	133
mem_reduce_malloc	102	stats_show_zero_counters	133
mem_thr_dsi	101	sts_cachesize	103
mem_thr_exec	101	sts_full_cache_*	103, 132
mem_thr_sqt	101	sub_daemon_sleep_time	103
mem_warning_thr1	101	sub_sqm_write_request_limit	35
mem_warning_thr2	101	unicode_format	37
memory_control	101	unified_login	104
memory_limit	101	use_batch_markers	36
memory_max (obsolete)	101	use_security_services	103
minimum_rssd_version	101	use_ssl	103
msg_confidentiality	103	varbinary_strip_trailing_zeros	37
msg_integrity	103	varchar_truncation	103
msg_origin_check	103	ws_sqldml_replication	78
msg_replay_detection	103	configure connection	32
msg_sequence_check	103	configure logical connection	74
mutual_auth	104	configure replication server	99
nrm_thread	102	configure route	96
num_client_connections	102	configure, rs (RSSD proc)	99
num_concurrent_subs	102	conflict resolution, with func strings	90
num_msgqueues	102	connect	120
num_msgs	102	connection profiles	29, 38, 39
num_mutexes	102	connection, show	121
num_stable_queues	102	connections to databases	30
num_threads	102	by multiple RepServers	141
oserver	101	non-ASE databases	38
parallel_dsi	105	renaming	32, 144
password_encryption	119	connectivity problems	137
prev_min_rssd_version	101	connector_type, dsi (config)	37
prev_rssd_version	101	coordinated dumps	128
queue_dump_buffer_size	102	create alternate connection	30
rec_daemon_sleep_time	102	create alternate logical connection	74
related RSSD tables	99	create applied function replication def.	47
rep_as_standby	35, 78	create article	63
replicate_minimal_columns	37, 78	create connection	30
replication agent mem.size (ASE)	21	for warm standby	75
rsi_batch_size	97	using profile	38
rsi_fadeout_time	97	create database replication definition	51
rsi_packet_size	97	create error class	92
rsi_sync_interval	97	create function	91
rsi_xact_with_large_msg	97	create function replication definition	48, 50
rssd_error_class	102	create function string	82
save_interval	35, 78, 97	create function string class	81
security_mechanism	104	create logical connection	74
security_services	103	create partition	111
send_enc_password	119	create publication	62
send_standby_repdef_cols	78	create replication definition	42
send_timestamp_to_standby	36	create request function replication def.	49
send_truncate_table	78	create route	95
smp_enable	102	create schedule	143
sqm_async_seg_delete	102	create server	41
sqm_cache_enable	115	create subscription	67
sqm_cache_size	115	create user	118
sqm_cmd_cache_size	115	creating a new RepServer	108
sqm_max_cmd_in_block	115	current_rssd_version (config)	101
sqm_page_size	115	custom function string	82, 84
sqm_recover_segs	102	cwd, admin	27
sqm_seg_preater	102		
sqm_warning_thr_ind	102		
sqm_warning_thr1	102		
sqm_warning_thr2	102		
sqm_write_flush	102, 111		
sqm_xact_with_large_msg	116		
sqt_init_read_delay	103		
sqt_max_cache_size	103		

D

Data Assurance	12
Data model, issues for replication	10
database connections	30

non-ASE databases	38	dematerialization queues	66
database generation number	20, 135	derived class	82
database repdefs	51	destination RepServer (routes)	94
and autocorrection	46	destination_db, rs_ (variable)	89
function strings	53	destination_ds, rs_ (variable)	89
marking data for replication	17	destination_ldb, rs_ (variable)	89
mixing with table/function repdefs	58	destination_lds, rs_ (variable)	89
switching primary and replicate	54	destination_ptype, rs_ (variable)	89
vs. warm standby	55	destination_user, rs_ (variable)	89
database replication		diagnostic RepServer	110
..... See database repdefs		dictionary, rs_ (RSSD table)	124
database resynchronization	129	direct I/O (O_DIRECT)	102
databases, rs_ (RSSD table)	32 , 79, 124	direct routes	96, 98
datarow_for_writetext, rs_ (function)	86	direct_cache_read, dist_ (config)	101
dataserver error class	92	DirectConnect	12
dataserver_make, dsi_ (config)	37	disconnect	120
datatype class	28	disk partitions	111
datatype translations	28	full, due to open transaction	139
datatype, rs_ (RSSD table)	29 , 124	disk space requirements, estimate	112
datatypes	27	disk_affinity (config)	83, 97, 111, 112
class-level translations	29	disk_space, admin	26, 111
column-level translations	29	diskaffinity, rs_ (RSSD table)	112 , 124
declared datatype	28	diskpartitions, rs_ (RSSD table)	112 , 124
mapping (map to clause)	28 , 29	DIST thread (distributor)	29 , 36
published datatype	28	dist_cmd_direct_replicate (config)	37
RepServer native datatypes	27	dist_direct_cache_read (config)	101
db_packet_size (config)	34	dist_sqt_max_cache_size (config)	35
DB2 vs UDB	80	dist_stop_unstopp_cmd (conf)	36 , 78
db2_error_class, rs_ (class)	92	distributor (DIST thread)	29 , 36
db2_function_class, rs_ (class)	80	resume	29
dbcc dbrepair (ASE cmd)	20	SQT optimization	101
dbcc gettrunc (ASE cmd)	20	suspend	30
dbcc settrunc (ASE cmd)	20 , 136	do_not_replicate (text/image data)	59
dbltn (ERSSD RepAgent)	95	Documentation about RepServer	14
dbrepair, dbcc(ASE cmd)	20	dropspvr.exe (executable)	110
dbreps, rs_ (RSSD table)	59 , 124	drop article	63
dbsrv8 (ERSSD)	123	drop connection	32
dbsrv9 (ERSSD)	123	drop database replication definition	54
dbstop (ERSSD)	123	drop error class	93
dbsubsets, rs_ (RSSD table)	59 , 124	drop function	91
DDL replication		drop function replication definition	49
ASE error 208	37	drop function string	84
ASE error 2762	17, 22 , 52	drop function string class	81
ASE warning 2007	37	drop logical connection	74
deferred name resolution	37, 78	drop partition	111
dsi_replication_ddl (config)	35	drop publication	62
list of replicated statements/procs	16	drop replication definition	45
not working	137	drop route	95
required ASE RepAgent config	22	drop schedule	143
set replication (ASE cmd)	17	drop subscription	68 , 70
sp_reptostandby (ASE proc)	16	drop user	118
with database repdef	52	drop_queue, sysadmin	113
with warm standby	17	dropdb, sysadmin	32
deactivate subscription, for ASE IMDB	20	dropldb, sysadmin	74
declared datatype	28	droprs, sysadmin	116
dedicated route	94 , 106	DSI EXEC thread	104
default function string, restoring	84	DSI generation number	135
default RepServer username/password	116	DSI thread	104
default_fs, rs_ (variable)	90	multiple DSIs	106
default_function_class, rs_ (class)	80	dsi_alt_writetext (config)	36
deferred_name_resolution (config)	37, 78	dsi_bulk_copy (config)	36
deferred_queue_size (config)	101	dsi_bulk_threshold (config)	36
define subscription	69	dsi_cdb_max_size (config)	42
delayed replication	37	dsi_charset_convert (config)	34
delete, rs_ (function)	85	dsi_check_thread_lock, rs_ (function)	87
delexception, rs_ (RSSD proc)	94	dsi_cmd_batch_size (config)	34
delexception_date, rs_ (RSSD proc)	94	dsi_cmd_prefetch (config)	34
delexception_id, rs_ (RSSD proc)	94	dsi_cmd_separator (config)	34
delexception_range, rs_ (RSSD proc)	94	dsi_command_convert (config)	41
deliver_as_name, rs_ (variable)	89	dsi_commit_check_locks_intrvl (conf)	106
dematerialization	66	dsi_commit_check_locks_log (conf)	106

dsi_commit_check_locks_max (conf)	106	encrypted data, replicating	87
dsi_commit_control (config)	106	encrypted passwords	
dsi_compile_enable (config)	41	disable	119
dsi_compile_max_cmds (config)	41	enable in rs_init	108
dsi_compile_retry_threshold (config)	42	encryption key, for passwords	120
dsi_connector_type (config)	37	encryptionkeys, rs_ (RSSD table)	124
dsi_dataserver_make (config)	37	Enterprise Connect Data Access	12
dsi_exec_request_sproc (config)	34	Enterprise Edition	14
dsi_fadeout_time (config)	34	Environment variables	109
dsi_ignore_underscore_name (config)	105	Errata	8
dsi_isolation_level (config)	36	error actions	92
dsi_keep_triggers (config)	34	error classes	92
dsi_large_xact_size (config)	105	error message number	
dsi_max_cmds_in_batch (config)	34	10354 (ASE error)	62
dsi_max_cmds_to_log (config)	34	11065 (ASE error)	62
dsi_max_text_to_log (config)	34	15162	83
dsi_max_xacts_in_group (config)	34	15184	92
dsi_non_blocking_commit (config)	37	15230	76
dsi_num_large_xact_threads (config)	105	15287	83
dsi_num_threads (config)	105	15537	92
dsi_partitioning_rule (config)	105	16022	26
dsi_quoted_identifier (config)	34	2007 (ASE warning)	37
dsi_replication (config)	35	2056	26, 67
dsi_replication_ddl (config)	35	208 (ASE error)	37
dsi_row_count_validation (config)	37	2762 (ASE error)	17, 22, 52
dsi_rs_ticket_report (config)	35, 142	28049	73
dsi_serialization_method (config)	105	32020 (warning)	15
dsi_sql_data_style (config)	35	5152	32
dsi_sqt_max_cache_size (config)	35	5158	51
dsi_stage_all_ops (config)	42	5185	37
dsi_text_convert_multiplier (config)	35	5186	19, 37
dsi_timer (config)	37	5187	37
dsi_xact_group_size (config)	34	5193	19
dump/load materialization	66	5203	37
dump_dbname, rs_ (variable)	90	584 (ASE error)	62
dump_file, sysadmin	113	9261 (ASE error)	21, 47
dump_label, rs_ (variable)	90	9289 (ASE error)	24
dump_load (config)	36	9290 (ASE warning)	24
dump_queue, sysadmin	113	errors/actions, rs_ (RSSD table)	94, 124
dump_status, rs_ (RSSD proc)	135	errorlog location	
dump_status, rs_ (variable)	90	ASE	137
dump_thread_stacks, sysadmin	116	non-ASE RepAgent	25
dump_timestamp, rs_ (variable)	90	RepServer	27
dump_tran, sysadmin	114	errorlog, writing strings	27
dumpdb, rs_ (function)	87, 128	ERSSD (embedded RSSD)	121
dumps, coordinated	128	default login/password	121
dumpran, rs_ (function)	87, 128	determine if used	121
duplicate transactions	135	miscellaneous SQLA commands	122
dynamic SQL		start/stop manually	123
for a repdef and all DSIs	44	erssd, sysadmin	122
for DSI	36	erssd_backup_dir (config)	122
for repdef and DSI	45	erssd_backup_interval (config)	122
dynamic_sql (config), for DSI	36	erssd_backup_start_date (config)	122
dynamic_sql, set	45	erssd_backup_start_time (config)	122
dynamic_sql_cache_management (config)	36	erssd_ra (config)	122
dynamic_sql_cache_size (config)	36	exceptions log, RSSD	94
		exceptscmd, rs_ (RSSD table)	94, 124
		exceptshdr, rs_ (RSSD table)	94, 124
		exceptslast, rs_ (RSSD table)	124, 136
		exec_cmds_per_timeslice (config)	35
		exec_max_cache_size (config)	37
		exec_nrm_request_limit (config)	35
		exec_prs_num_threads (config)	38
		exec_request_sproc, dsi_ (config)	34
		exec_sqm_write_request_limit (config)	35
		Executor	See REP AGENT thread
		ExpressConnect	12, 13
E			
EBF installation override	109		
ECDA	12		
echo, admin	27		
editions			
Enterprise Edition	14		
Heterogenous Edition	14		
Messaging Edition	12, 14		
Real-Time Loading (RTL) Edition	14		
Electronic supplement to this book	8		
embedded RSSD	See ERSSD		
enable rep agent threads (ASE config)	21		

F

fadeout_time, dsi_ (config) 34
 fast_route_upgrade, sysadmin 99
 fillcapttable, rs_ (RSSD proc) 112
 filters, database repdef 52
 First Seg.Block (queues) 113
 float (datatype) 10, 27
 forgotten, RS password 141
 funcstrings, rs_ (RSSD table) **84**, 124
 function repdefs 47
 mixing with database repdefs 58
 mixing with table repdefs 57
 function string class inheritance 81
 function string classes 80
 definition 80
 list of system-supplied classes 80
 function strings 82
 conflict resolution 90
 custom 82, 84
 db repdef 53
 definition 79
 input/output templates 88
 modifiers (for variables) 88
 restoring default 84
 system variables 89
 target-specific **82**, 84
 various tricks 90
 warm standby 78
 functions 79
 definition 79
 function class-specific 86
 function repdef-specific (user-def'd) 86
 table repdef-specific 85
 target-specific **82**, 84
 functions, rs_ (RSSD table) **84**, 124

G

gateway 120
 generation numbers 135
 DSI 135
 primary database 135
 get_charset, rs_ (function) 87
 get_generation, admin **136**
 get_lastcommit, rs_ (function) 87
 get_lastcommit, rs_ (proc) 33
 get_sortorder, rs_ (function) 87
 get_textptr, rs_ (function) 86
 get_thread_seq, rs_ (function) 87
 get_thread_seq_noholdlock, rs_ (func) 87
 gettrunc, dbcc (ASE cmd) 20
 grant 119

H

ha_failover (config) 101
 heartbeat 142
 Help: replication doesn't work! 137
 help_rep_agent/sp_ (ASE proc) 21
 helpcfg, rs_ (RSSD proc) 100, 133
 helpcfg_db, rs_ (RSSD proc) 71, 78, 100
 helpcfg_queue, rs_ (RSSD proc) 115
 helpcfg_route, rs_ (RSSD proc) 100
 helpcheckrepdef, rs_ (RSSD proc) 58
 helpclass, rs_ (RSSD proc) 81, 93
 helpclassfstring, rs_ (RSSD proc) 85
 helpcounter, rs_ (RSSD proc) **135**
 helpdb, rs_ (RSSD proc) 33

helpdbrep, rs_ (RSSD proc) 59
 helpdbsub, rs_ (RSSD proc) 71
 helperror, rs_ (RSSD proc) 93
 helpexception, rs_ (RSSD proc) 94
 helpfstring, rs_ (RSSD proc) 84
 helpfunc, rs_ (RSSD proc) 85
 helpobjfstring, rs_ (RSSD proc) 85
 helppartition, rs_ (RSSD proc) 111
 helppub, rs_ (RSSD proc) 63
 helppubsub, rs_ (RSSD proc) **64**, **71**
 helprep, rs_ (RSSD proc) 59
 helprepdb, rs_ (RSSD proc) 33, **59**, 71
 helpreptable, rs_ (RSSD proc) 59
 helprepversion, rs_ (RSSD proc) 47
 helproute, rs_ (RSSD proc) 96
 helpsub, rs_ (RSSD proc) 70
 helpuser, rs_ (RSSD proc) 119
 heterogeneous RepAgents 24
 heterogeneous replication 12
 Heterogenous Edition 14
 hibernate_off, sysadmin 116
 hibernate_on, sysadmin 116
 hibernation mode 116
 HVAR
 PK-FK constraint **43**
 PK-FK constraints in repdef 45
 retry failed transactions 40
 ID server 117
 ID server user 117
 name of ID server 117
 id_msg_confidentiality (config) 104
 id_msg_integrity (config) 104
 id_msg_origin_check (config) 104
 id_msg_replay_detection (config) 104
 id_msg_sequence_check (config) 104
 id_mutual_auth (config) 104
 id_security_mechanism (config) 104
 id_server (config) 101
 id_unified_login (config) 104
 identifiers 26
 identifiers, in primary database 11
 identity columns, replicating 61, 62
 idnames, rs_ (RSSD table) 124
 ids, rs_ (RSSD table) 125
 ignore loss **130**, 138
 ignore_underscore_name (config) 105
 image/text data, mark for replication 18
 image/text/rawobject data 59
 IMDB, in ASE 20
 inbound queue 112
 incremental atomic materialization 65
 indexes in replicate database 11
 indirect routes 96, **98**
 info, rs_ (table) 33
 InfoPrimer 41
 Information Liquidity Model 12
 inheritance, function string classes 81
 init, rs_ (utility) 108
 init_erroractions, rs_ (RSSD proc) 93
 init_sqm_write_delay (config) 101
 init_sqm_write_max_delay (config) 101
 initial RepServer username/password 116
 initialize_threads, rs_ (function) 87
 initializing a new RepServer 108
 in-memory ASE database 20
 input templates (function strings) 88
 insert, rs_ (function) 85

interfaces file	110	marker, rs_ (proc).....	87
intermediate RepServer (routes).....	94	marking primary data for replication ...	15
Internet, RepServer resources on	14	for database repdefs	17
IQ, replicating to	40	in a warm standby	17
iq_error_class, rs_ (class).....	92	master database (ASE), replicating	55
iq_function_class, rs_ (class).....	80	materialization	
isolation_level, dsi_ (config).....	36	(de)materialization queues	66
isql (utility)	110	drop when materialization failed	73
		identifying.....	66
		disk space requirements	66
		materialization methods	64
		atomic materialization	64
		automatic materialization	65
		bulk materialization	65
		dump/load materialization	66
		incremental atomic mat	65
		manual materialization	65
		no materialization	64
		non-atomic materialization	65
		simulated (non)atomic bulk mat	66
		users and permissions	66
		materialization queues	66
		materialization_save_interval (config)	78
		max_cmds_in_batch, dsi_ (config)	34
		max_cmds_to_log, dsi_ (config).....	34
		max_text_to_log, dsi_ (config).....	34
		max_xacts_in_group, dsi_ (config)	34
		md_sqm_write_request_limit (config) ..	35
		mem_reduce_malloc (config)	102
		mem_thr_dsi (config).....	101
		mem_thr_exec (config).....	101
		mem_thr_sqt (config).....	101
		mem_warning_thr1 (config).....	101
		mem_warning_thr2 (config).....	101
		memory_control (config).....	101
		memory_limit (config).....	101
		memory_max (config, obsolete)	101
		message bus integration	12, 14
		message loss	130
		causing rejected messages	138
		Messaging Edition	12, 14
		minimal column replication	56
		and rs_default_fs	90
		for normal connections	37
		for warm standby	78
		limitations	57
		minimum_rssd_version (config).....	101
		Mirror Activator	12
		missing replicate rows, unnoticed.....	11
		missing updates, and parallel DSI	104
		Mobilink	12
		modifiers, for function string variables	88
		monitor counters	133
		move a subscription	70
		move primary	
		of error class	93
		of function string class	81
		MPR	See multi-path replication
		MSA (=database repdef/subscription) .	51
		msg_confidentiality (config).....	103
		msg_integrity (config).....	103
		msg_origin_check (config)	103
		msg_replay_detection (config)	103
		msg_sequence_check (config).....	103
		msgs, rs_ (RSSD table).....	125
		msss_error_class, rs_ (class).....	92
		msss_function_class, rs_ (class).....	80
		multi-path replication (MPR).....	106
		multiple DSIs	106, 141
		Multiple Site Availability (MSA).....	51
		Multiple Warm Standby (=MSA).....	51
J			
job scheduling.....	143		
K			
keep_triggers, dsi_ (config).....	34		
L			
l_origin, rs_ (variable)	89		
LAN RepAgents.....	24		
language templates (function strings) .	88		
large transactions (DSI).....	104, 105		
large_xact_size, dsi_ (config)	105		
Last Seg.Block (queues).....	113		
last_text_chunk, rs_ (variable).....	90		
lastcommit, rs_ (table)	32, 128		
latency	128		
license keys	13		
served/unserved	13		
license manager (SySAM)	13		
line continuation character (\).....	26		
lmconfig, sysadmin	13		
lmutil (license manager).....	13		
LOB data	59		
local queue ID (LQID)	114		
locator, in RSSD table rs_locator	19		
locator, rs_ (RSSD table).....	20, 125		
locked user	120		
locking yourself out	141		
log transfer			
resume	30		
suspend	30		
Log Transfer Language (LTL).....	24		
log_first_tran, sysadmin	115		
log_name, admin	27		
log_system_name(non-ASE RepAgent) ..	25		
logical_status, admin	79		
loss detection.....	130		
lost RS password.....	141		
LQID	114		
LTM	20		
LTM locator (non-ASE RepAgent) .	19, 25		
LTM truncation point	19		
M			
maintenance user	117		
alias to dbo	108		
changing	145		
ERSSD	121		
replicating primary transactions by	22		
maintusers, rs_ (RSSD table).....	125		
manual materialization	65, 69		
manuals for RepServer	14		
map to (datatype mapping)	28, 29		
mapping (map to clause)	43		
marker, rs_ (function).....	87		

mutual_auth (config)..... 104

N

name, of ID server 117
 name, of RepServer 26
 native datatypes 27
 net-change database (CDB).... 39, 40, 42
 network security 103
 new, new_raw (variable modifier) 88
 newsgroups 14
 Next.Read (queues) 113
 no materialization (mat.method) 64
 non_blocking_commit, dsi_ (config).... 37
 non_blocking_commit, rs_ (function).... 87
 non_blocking_commit_flush, rs_ (func) 87
 non-ASE RepAgent 24
 non-atomic materialization 65
 NRM thread 35, 102
 nrm_thread (config) 102
 num_client_connections (config) 102
 num_concurrent_subs (config) 102
 num_large_xact_threads, dsi_ (conf). 105
 num_msgqueues (config) 102
 num_msgs (config) 102
 num_mutexes (config) 102
 num_stable_queues (config) 102
 num_threads (config) 102
 num_threads, dsi_ (config) 105

O

objects, rs_ (RSSD table) 59, 125
 objfunctions, rs_ (RSSD table) 84, 125
 old, old_raw (variable modifier) 88
 opaque (datatype) 27
 open transactions 139
 OpenSwitch 12
 OQID, layout 136
 oqid, rs_ (RSSD table) 125
 Oracle
 ExpressConnect interface 37
 option for replication 12
 replicating to IQ 40
 replication agent 24
 warm standby 74
 oracle_error_class, rs_ (class) 92
 oracle_function_class, rs_ (class) 80
 Origin queue ID See OQID
 origin, rs_ (variable) 89
 origin_begin_time, rs_ (variable) 89
 origin_commit_time, rs_ (variable) 89
 origin_db, rs_ (variable) 89
 origin_ds, rs_ (variable) 89
 origin_ldb, rs_ (variable) 89
 origin_lds, rs_ (variable) 89
 origin_qid, rs_ (variable) 90
 origin_user, rs_ (variable) 89
 origin_xact_id, rs_ (variable) 89
 origin_xact_name, rs_ (variable) 90
 oserver (config) 101
 outbound queue 112
 output templates (function strings) 88

R

parallel DSI 104
 parallel_dsi (config) 105
 param, param_raw (variable modifier) . 88

parent class 81
 partitioning rules, for parallel DSI 105
 partitioning_rule, dsi_ (config) 105
 partitions, disk 111
 password 119
 change, for maintenance user 31
 change, RepServer user 118
 for RepServer 'sa'
 default password 116
 reset 'sa' password 141
 lost/forgotten 141
 password encryption 119
 disable 119
 enable in rs_init 108
 password expiration 118, 120
 password length (min, max) 120
 password policy settings 120
 password_encryption (config) 19
 password_encryption (config) 119
 passwords, rs_ (RSSD table) 125
 path See replication path
 pdb_gen_id (non-ASE RepAgent) 25
 pdb_setrepcol (non-ASE RepAgent) 25
 pdb_setrepproc (non-ASE RepAgent) . 25
 pdb_setreptable (non-ASE RepAgent) . 25
 pdb_truncate_xlog (Non-ASE RepAg.) . 25
 perf_listruns, rs_ (RSSD proc) 135
 performance tips 131
 permissions 119
 poster of RSSD tables 123
 PowerDesigner 12
 prepared statements
 for a redef and all DSIs 44
 for DSI 36
 for redef and DSI 45
 prev_min_rssd_version (config) 101
 prev_rssd_version (config) 101
 primary key
 automatically detect 58
 unique, why required 10
 updating, issues 10
 primary transaction log, full (ASE) 138
 primary user
 ERSSD 121
 RSSD/ERSSD 117
 print_rs_ticket (traceflag) 111
 profdetail, rs_ (RSSD table) 39, 125
 profile, rs_ (RSSD table) 39, 125
 publications 62
 publications, rs_ (RSSD table) 64, 125
 published datatype 28
 publish-subscribe replication 42
 purge_all_open, sysadmin 115
 purge_first_open, sysadmin 115
 purge_route_at_replicate, sysadmin ... 96

Q

queue ID See OQID
 queue_dump_buffer_size (config) 102
 queuemsg, rs_ (RSSD table) 116, 125
 queuemsgtxt, rs_ (RSSD table) . 116, 125
 queues 112
 (de)materialization queues 66
 identifying 112
 inbound, outbound 112
 queue size and save_interval 112
 queue size and slow replication ... 138
 understanding queue growth 113
 queues, rs_ (RSSD table) 116, 125

- quiesce_check, admin 128
 quiesce_force_rsi, admin 97
 quiescing/quiesced, RepServer 127
 quoted identifiers 43, 44, 58
 quoted_identifier, dsi_ (config) 34
- ## R
- ra (non-ASE RepAgent) 24
 ra_admin (non-ASE RepAgent) 24
 ra_config (non-ASE RepAgent) 25
 ra_dump (non-ASE RepAgent) 25
 ra_help (non-ASE RepAgent) 24
 ra_locator (non-ASE RepAgent) 25
 ra_set_autocorr (non-ASE RepAgent) 25
 ra_set_login (non-ASE RepAgent) 25
 ra_status (non-ASE RepAgent) 24
 ra_version[_all] (non-ASE RepAgent) 24
 raw_object_serialization, rs_ (function) 87
 rawobject [in row] (datatype) 27
 rawobject/text/image data 59
 RAX (non-ASE RepAgent) 24
 real (datatype) 10, 27
 Real-Time Events 14
 Real-Time Loading Edition 14
 rebuild queues 115, **130**
 rec_daemon_sleep_time (config) 102
 recovery 130
 recovery, rs_ (RSSD table) 125
 references (PK-FK constraint) **43**, 45
 refreshing, publication subscription 63, **68**
 registry (Windows), startup params 110
 renaming
 - a database connection **32**, 144
 - a RepServer 144
 REP AGENT thread 31, 35, 111
 - remove 32
 rep_as_standby (config) **35**, 78
 RepAgent
 - configuring (ASE) 21
 - connections to RepServer 29
 - for ASE primary databases 21
 - for non-ASE primary databases 24
 - starting (ASE) 21
 - stopping (ASE) 21
 RepAgents 14
 RepConnector 14
 repdbs, rs_ (RSSD table) 125
 repdef
 - and autocorrection 45
 - database repdefs 51
 - function repdefs 47
 - table repdefs 42
 - uniqueness of names 42
 - versions of 46
 repdef filters (database repdef) 52
 repl_off, rs_ (function) 87
 repl_on, rs_ (function) 87
 replicate_if_changed (text/image data) 59
 replicate_minimal_columns (conf) 37, 78
 replicated RSSD tables for routes 96
 replicating minimal columns 56
 - and rs_default fs 90
 - limitations 57
 Replication Agent See RepAgent
 replication agent mem size (ASE conf) 21
 replication definition See repdef
 replication index 18
 replication mode 16
 replication path 106
 - default path 107
 - logical path 106
 - physical path 106
 replication, dsi_ (config) 35
 replication_ddl, dsi_ (config) 35
 replication_path, sp_ (ASE proc) 107
 Replicator, ASE 12
 repobjs, rs_ (RSSD table) 125
 RepServer
 - creating 108
 - dropping 116
 - name 26
 - renaming 144
 repserver (executable) 170
 RepServer error class 92
 repserver.diag (executable) 110
 repserver_error_class, rs_ 19, 37, **92**
 repssrvr.exe (executable) 110
 reptostandby, sp_ (ASE proc) 16
 request functions 49
 - setup example 50
 request stored procedures 49
 reserved words 26
 reset_lastcommit, rs_ (function) 87
 resetting monitor counters **134**
 resetting the SA password 141
 resource file (for rs_init) 108
 restore_dsi_saved_seg, sysadmin 115
 restoring the default function string 84
 resume connection 32
 resume distributor 29
 resume log transfer 30
 resume queue 116
 resume route 96
 resynchronization, of database 129
 retry failed HVAR transactions 40
 reverse-engineering, objects 141
 revoke 119
 rollback, rs_ (function) 86
 route version 98
 routes 94
 - dedicated **94**, 106
 - direct routes 96, **98**
 - indirect routes 96, **98**
 - shared **94**, 106
 - upgrading 98, 99
 routes, rs_ (RSSD table) **96**, 125
 routeversions, rs_ (RSSD table) 126
 row_count_validation, dsi_ (config) 37
 rowcount verification 11, **37**
 row-level locking and parallel DSI 104
 rs_address (datatype) 27, **72**
 rs_articles (RSSD table) **64**, 123
 rs_asa_function_class (class) 80
 rs_asyncfuncs (RSSD table) **84**, 123
 rs_batch_end (function) 87
 rs_batch_start (function) 87
 rs_begin (function) 86
 rs_capacity (RSSD proc) 112
 rs_captable (RSSD table) **112**, 123
 rs_check_repl (function) 87
 rs_classes (RSSD table) 29, 84, 93, **123**
 rs_clsfuncs (RSSD table) **84**, 124
 rs_columns (RSSD table) 124
 rs_commit (function) 86
 rs_config (RSSD table) **99**, 124
 rs_configure (RSSD proc) 99
 rs_databases (RSSD table) **32**, 79, 124
 rs_datarow_for_writetext (function) 86
 rs_datatype (RSSD table) **29**, 124
 rs_db2_error_class (class) 92

rs_db2_function_class (class)	80	rs_info (table).....	33
rs_dbreps (RSSD table)	59 , 124	rs_init (utility).....	108
rs_dbsubsets (RSSD table)	59 , 124	rs_init_erroractions (RSSD proc)	93
rs_default_fs (variable).....	90	rs_initialize_threads (function).....	87
rs_default_function_class (class).....	80	rs_insert (function)	85
rs_delete (function).....	85	rs_iq_error_class (class).....	92
rs_delexception (RSSD proc)	94	rs_iq_function_class (class).....	80
rs_delexception_date (RSSD proc)	94	rs_l_origin (variable)	89
rs_delexception_id (RSSD proc)	94	rs_last_text_chunk (variable).....	90
rs_delexception_range (RSSD proc).....	94	rs_lastcommit (table).....	32, 128
rs_deliver_as_name (variable)	89	rs_locator (RSSD table)	19, 20, 125
rs_destination_db (variable)	89	rs_maintusers (RSSD table).....	125
rs_destination_ds (variable).....	89	rs_marker (function).....	87
rs_destination_ldb (variable).....	89	rs_marker (proc)	87
rs_destination_lds (variable).....	89	rs_msgs (RSSD table).....	125
rs_destination_ptype (variable).....	89	rs_msss_error_class (class).....	92
rs_destination_user (variable)	89	rs_msss_function_class (class).....	80
rs_dictionary (RSSD table).....	124	rs_non_blocking_commit (function).....	87
rs_diskaffinity (RSSD table).....	112 , 124	rs_non_blocking_commit_flush (func).....	87
rs_diskpartitions (RSSD table).....	112 , 124	rs_objects (RSSD table).....	59 , 125
rs_dsi_check_thread_lock (function).....	87	rs_objfunctions (RSSD table).....	84 , 125
rs_dump_dbname (variable).....	90	rs_oqid (RSSD table).....	125
rs_dump_label (variable).....	90	rs_oracle_error_class (class).....	92
rs_dump_stats (RSSD proc).....	135	rs_oracle_function_class (class).....	80
rs_dump_status (variable).....	90	rs_origin (variable).....	89
rs_dump_timestamp (variable).....	90	rs_origin_begin_time (variable).....	89
rs_dumpdb (function).....	87, 128	rs_origin_commit_time (variable).....	89
rs_dumptran (function)	87, 128	rs_origin_db (variable).....	89
rs_encryptionkeys (RSSD table).....	124	rs_origin_ds (variable).....	89
rs_erroractions (RSSD table).....	94 , 124	rs_origin_ldb (variable).....	89
rs_exceptscmd (RSSD table)	94 , 124	rs_origin_lds (variable).....	89
rs_exceptshdr (RSSD table).....	94 , 124	rs_origin_qid (variable).....	90
rs_exceptslast (RSSD table).....	124 , 136	rs_origin_user (variable).....	89
rs_fillcapable (RSSD proc)	112	rs_origin_xact_id (variable).....	89
rs_funcstrings (RSSD table).....	84 , 124	rs_origin_xact_name (variable).....	90
rs_functions (RSSD table).....	84 , 124	rs_passwords (RSSD table).....	125
rs_get_charset (function).....	87	rs_perf_listtruns (RSSD proc).....	135
rs_get_lastcommit (function).....	87	rs_profdetail (RSSD table).....	39 , 125
rs_get_lastcommit (proc).....	33	rs_profile (RSSD table).....	39 , 125
rs_get_sortorder (function).....	87	rs_publications (RSSD table).....	64 , 125
rs_get_textptr (function).....	86	rs_queuemsg (RSSD table).....	116 , 125
rs_get_thread_seq (function).....	87	rs_queuemsgtxt (RSSD table).....	116 , 125
rs_get_thread_seq_noholdlock (func).....	87	rs_queues (RSSD table).....	116 , 125
rs_helpcfg (RSSD proc).....	100, 133	rs_raw_object_serialization (function).....	87
rs_helpcfg_db (RSSD proc).....	71, 78, 100	rs_recovery (RSSD table).....	125
rs_helpcfg_queue (RSSD proc).....	115	rs_repdb (RSSD table).....	125
rs_helpcfg_route (RSSD proc).....	100	rs_repl_off (function).....	87
rs_helpcheckrepdef (RSSD proc).....	58	rs_repl_on (function).....	87
rs_helpclass (RSSD proc).....	81, 93	rs_repubjs (RSSD table).....	125
rs_helpclassfstring (RSSD proc).....	85	rs_repserver_error_class	19, 37, 92
rs_helpcounter (RSSD proc).....	135	rs_reset_lastcommit (function).....	87
rs_helpdb (RSSD proc).....	33	rs_rollback (function).....	86
rs_helpdbrep (RSSD proc).....	59	rs_routes (RSSD table).....	96 , 125
rs_helpdbsub (RSSD proc).....	71	rs_routeversions (RSSD table).....	126
rs_helperror (RSSD proc).....	93	rs_rules (RSSD table).....	64, 70, 126
rs_helpexception (RSSD proc).....	94	rs_schedule (RSSD table).....	126 , 143
rs_helpfstring (RSSD proc).....	84	rs_scheduledtxt (RSSD table).....	126 , 143
rs_helpfunc (RSSD proc).....	85	rs_secondary_qid (variable).....	90
rs_helpobjfstring (RSSD proc).....	85	rs_segments (RSSD table).....	116 , 126
rs_helppartition (RSSD proc).....	111	rs_select (function).....	85
rs_helppub (RSSD proc).....	63	rs_select_with_lock (function).....	85
rs_helppubsub (RSSD proc).....	64, 71	rs_send_repserver_cmd (proc).....	46
rs_helprep (RSSD proc).....	59	rs_session_setting (func).....	87
rs_helprepdb (RSSD proc).....	33, 59 , 71	rs_set_ciphertext (function).....	87
rs_helpreptable (RSSD proc).....	59	rs_set_dml_on_computed (function).....	87
rs_helprepversion (RSSD proc).....	47	rs_set_isolation_level (function).....	87
rs_helproute (RSSD proc).....	96	rs_set_timestamp_insert (function).....	87
rs_helpsub (RSSD proc).....	70	rs_setproxy (function).....	87
rs_helpuser (RSSD proc).....	119	rs_sites (RSSD table).....	96, 126
rs_idnames (RSSD table).....	124	rs_sqlserver_error_class (class).....	92
rs_ids (RSSD table).....	125	rs_sqlserver_function_class (class).....	80, 82

rs_stat_genreport (RSSD proc)	135	scheduletxt, rs_ (RSSD table) ...	126 , 143
rs_stat_populate (RSSD proc)	134	scheduling commands/jobs	143
rs_statcounters (RSSD table)	126, 133	schema changes, primary tables ...	46, 47
rs_statdetail (RSSD table)	126, 133	schema transformation	41
rs_statreport (RSSD table)	126, 133	secondary truncation point	19
rs_statrun (RSSD table)	126, 133	disable/remove	20
rs_status (table)	33	secondary_qid, rs_ (variable)	90
rs_subcmp (utility)	139	security_mechanism (config)	104
rs_subscriptions (RSSD table)	70 , 126	security_property, admin	104
rs_systext (RSSD table)	84, 94 , 126	security_services (config)	103
rs_targetobjs (RSSD table)	84 , 126	security_setting, admin	104
rs_tbcnfig (RSSD table)	39, 126	segments, disk space	112
rs_textptr_init (function)	86	segments, rs_ (RSSD table)	116 , 126
rs_threads (table)	33	select, rs_ (function)	85
rs_ticket (proc)	142	select_with_lock, rs_ (function)	85
rs_ticket_print_ (traceflag)	111	send standby, option in repdef	56
rs_ticket_history (table)	33, 142	send_enc_password (config)	119
rs_ticket_report (function)	87	send_repserver_cmd, rs_ (proc)	46
rs_ticket_report, dsi_ (config)	35, 142	send_standby_repdef_cols (config)	78
rs_ticket_v1 (proc)	142	send_timestamp_to_standby (config)	36
rs_translation (RSSD table)	29 , 126	send_truncate_table (config)	78
rs_triggers_reset (function)	87	serialization_method, dsi_ (config)	105
rs_trunc_reset (function)	87	served license keys	13
rs_trunc_set (function)	87	server, show	121
rs_truncate (function)	85, 88	servername, of RepServer	26
rs_tvalues (RSSD table)	126	session_setting, rs_ (func)	87
rs_udb_error_class (class)	92	set autocorrection	45
rs_udb_function_class (class)	80	set dynamic sql	45
rs_update (function)	85	set log recovery	130
rs_update_lastcommit (proc)	33	set proxy	119
rs_update_threads (function)	87	set replication (ASE cmd)	17
rs_usedb (function)	87	set repmode (ASE cmd)	19
rs_users (RSSD table)	126	set repthreshold (ASE cmd)	19
rs_version (RSSD table)	126	set ciphertext, rs_ (function)	87
rs_whereclauses (RSSD table)	64, 126	set_dml_on_computed, rs_ (function)	87
rs_writetext (function)	86	set_dsi_generation, sysadmin	136
rs_writetext_log (variable)	90	set_isolation_level, rs_ (function)	87
rs_zeroltm (RSSD proc)	20 , 136	set_log_name, admin	27
RSI thread	95	set_timestamp_insert, rs_ (function)	87
rsi_batch_size (config)	97	setproxy, rs_ (function)	87
rsi_fadeout_time (config)	97	setrepcol, sp_ (ASE proc)	15
rsi_packet_size (config)	97	setrepdbmode, sp_ (ASE proc)	18, 19
rsi_sync_interval (config)	97	setrepdefmode, sp_ (ASE proc) ...	15, 19
rsi_xact_with_large_msg (config)	97	setreplicate, sp_ (ASE proc)	15
RSSD	121	setrepproc, sp_ (ASE proc)	16
creation, scripts	121	setreptable, sp_ (ASE proc)	15
default login/password	117	settrunc, dbcc (ASE cmd)	20 , 136
determine RSSD type	121	shared route	94 , 106
exceptions log	94	show connection	121
location	27	show server	121
primary user	117	show_connection_profiles, admin	39
RSSD stored procedures	121	show_connections, admin	33 , 97
RSSD tables	123	show_function_classes, admin	81
replicated for routes	96	show_route_versions, admin	98
rssd_error_class (config)	102	show_site_version, admin	98
rssd_name, admin	27, 121	shutdown	110
RTL Edition	14	simulated (non)atomic bulk materializ. ...	66
rules, rs_ (RSSD table)	64, 70 , 126	single-user mode	110
		site version	98
		site_version, sysadmin	98
		sites, rs_ (RSSD table)	96, 126
		skip_bad_repserver_cmd, sysadmin ...	47
SA password reset	141	smtp_enable (config)	102
sa password, default	116	sort order in replication system	109
Save_Int:Seg (queues)	113	source RepServer (routes)	94
save_interval (config)	35, 78, 97	sp_config_rep_agent (ASE proc)	21
and queue size	112	sp_help_rep_agent (ASE proc)	21
scan templates (function strings)	83	sp_replication_path (ASE proc)	107
SCC	See Sybase Control Center	sp_repostandby (ASE proc)	16
SCC agent	127	with database repdef	51
schedule, rs_ (RSSD table)	126 , 143	with warm standby	17

sp_setrepcol (ASE proc).....	15	stats tps / cps / bps, admin.....	134
sp_setrepdbmode (ASE proc)	18, 19	stats, admin	133, 134
sp_setrepdefmode (ASE proc)	15, 19	stats, backlog, admin	134
sp_setreplicate (ASE proc).....	15	stats, cancel, admin	134
sp_setrepproc (ASE proc).....	16	stats, reset, admin.....	134
sp_setreptable (ASE proc)	15	stats, status, admin	134
sp_start_rep_agent (ASE proc).....	21	stats_reset_rssd (config).....	133
sp_stop_rep_agent (ASE proc)	21	stats_sampling (config).....	133
sp_sysmon (ASE proc).....	23	stats_show_zero_counters (config).....	133
SQL Anywhere ERSSD		status information, getting	26
login/password	121	status, rs_ (table)	38
SQL Remote.....	12	stop_rep_agent, sp_ (ASE proc).....	21
SQL statement replication ..	18, 44, 53, 78	stop_unsupp_cmd_dist (conf).....	36, 78
SQL.INI file.....	110	stopping RepServer	110
sql_data_style, dsi_ (config).....	35	stored procedure replication.....	47
SQLA (SQL Anywhere), for ERSSD ..	121	strict, save_interval setting.....	78
miscellaneous commands	122	sts_cachesize (config).....	103
start/stop manually	123	sts_full_cache_* (config).....	103, 132
sqlserver_error_class, rs_ (class).....	92	sub_daemon_sleep_time (config).....	103
sqlserver_function_class,rs_ (class)	80,82	sub_sqm_write_request_limit (config).....	35
SQM thread.....	112	subcmp.exe (utility).....	139
sqm_async_seg_delete (config).....	102	subscription materialization	
sqm_cache_enable (config).....	115	See Materialization	
sqm_cache_size (config).....	115	subscriptions.....	64
sqm_cmd_cache_size (config).....	115	aborting creation	73
sqm_max_cmd_in_block (config)	115	move between DSI's	70
sqm_page_size (config)	115	subscriptions, rs_ (RSSD table) ..	70, 126
sqm_purge_queue, sysadmin.....	114	suspend connection.....	32
sqm_recover_segs (config).....	102	suspend distributor.....	30
sqm_seg_prealloc (config)	102	suspend log transfer.....	30
sqm_unzap_command, sysadmin	115	suspend route.....	96
sqm_unzap_tran, sysadmin.....	115	switch active	75
sqm_warning_thr_ind (config).....	102	detailed steps.....	77
sqm_warning_thr1 (config).....	102	switching primary and replicate	
sqm_warning_thr2 (config).....	102	with database repdefs.....	54
sqm_write_flush (config).....	102, 111	Sybase Control Center (SCC).....	127
sqm_xact_with_large_msg (config) ..	116	Sybase IQ, replicating to	40
sqm_zap_command, sysadmin	115	Syntax conventions.....	7
sqm_zap_tran, sysadmin.....	114	syntax_of RepServer commands	26
SQT.....	112	sys, sys_raw (variable modifier) ..	88
info from admin who, sqt	113	sysadmin apply_truncate_table.....	71
sizing.....	35, 132	sysadmin cdb.....	40
sqt_dump_queue, sysadmin	114	sysadmin drop_queue.....	113
sqt_init_read_delay (config).....	103	sysadmin dropdb.....	32
sqt_max_cache_size (config).....	103	sysadmin dropldb	74
sqt_max_cache_size, dist_ (config) ..	35	sysadmin droprs.....	116
sqt_max_cache_size, dsi_ (config).....	35	sysadmin dump_file	113
sqt_max_prs_size (config).....	38	sysadmin dump_queue.....	113
sqt_max_read_delay (config)	103	sysadmin dump_thread_stacks.....	116
sre_reserve (config)	102	sysadmin dump_tran.....	114
SSL.....	103	sysadmin erssd	122
stable devices.....	111	sysadmin fast_route_upgrade.....	99
stable queues.....	112	sysadmin hibernate_off.....	116
stage_all_ops, dsi_ (config).....	42	sysadmin hibernate_on.....	116
stage_operations (config).....	42	sysadmin issue_ticket.....	142
standalone mode.....	110	sysadmin lmconfig	13
standby database (in warm standby) ..	73	sysadmin log_first_tran	115
standby_func_class (config).....	79	sysadmin purge_all_open	115
start_rep_agent, sp_ (ASE proc).....	21	sysadmin purge_first_open	115
starting points for common DBA tasks ..	9	sysadmin purge_route_at_replicate ..	96
starting RepServer	109	sysadmin restore_dsi_saved_seg.....	115
startserver (utility).....	109	sysadmin set_dsi_generation.....	136
startup params (Windows registry) ..	110	sysadmin site_version	98
stat_genreport, rs_ (RSSD proc).....	135	sysadmin skip_bad_repserver_cmd.....	47
stat_populate, rs_ (RSSD proc).....	134	sysadmin sqm_purge_queue.....	114
statcounters, rs_ (RSSD table) ..	126, 133	sysadmin sqm_unzap_command.....	115
statdetail, rs_ (RSSD table).....	126, 133	sysadmin sqm_unzap_tran	115
statement replication		sysadmin sqm_zap_command.....	115
.....See SQL Statement replication		sysadmin sqm_zap_tran	114
statreport, rs_ (RSSD table)	126, 133	sysadmin sqt_dump_queue	114
statrun, rs_ (RSSD table)	126, 133	sysadmin system_version	98

- sysadmin upgrade
 database..... 98
 route 98
- sysam (license manager)..... 13
 SySAM (license manager)..... 13
 sysmon, sp_ (ASE proc)..... 23
 system procedures, replicated 16
 system variables..... 89
 system version..... 98
 system_version, sysadmin..... 98
 systext, rs_ (RSSD table) 84, 94, 126
- ## T
- table repdefs..... 42
 mixing with database repdefs 58
 mixing with function repdefs 57
 using with warm standby/MSA..... 58
 no repdef needed in 15.7..... 58
 without subscriptions 58
- targetobjs, rs_ (RSSD table)..... 84, 126
 target-specific function string 82, 84
 tbconfig, rs_ (RSSD table)..... 39, 126
 templates, system variables..... 89
- test_connection (non-ASE RepAgent) 25
 text/image data, marking for replication 18
 text/image/rawobject data 59
 text_convert_multiplier, dsi_ (config)..... 35
 text_status (variable modifier)..... 88
 textptr_init, rs_ (function) 86
 threads, rs_ (table)..... 33
 throughput, by admin stats..... 134
 ticket feature 142
 ticket, print_rs_ (traceflag) 111
 ticket, rs_ (proc)..... 142
 ticket_history, rs_ (table)..... 33, 142
 ticket_report, rs_ (function) 87
 ticket_v1, rs_ (proc) 142
 time, admin 27
 time-delayed replication 37
 timer, dsi_ (config) 37
 timestamp
 ASE datatype..... 62
 replicating 36, 62
 replicating, function..... 87
 RS datatype..... 27
- trace (RepServer cmd) 110
 traceflags
 ASE RepAgent 23
 RepServer 110
- transaction log full (ASE) 138
 transformation stored procedures 41
 translate, admin 28
 translation, rs_ (RSSD table) 29, 126
 triggers, and replication 64
 triggers_reset, rs_ (function)..... 87
- ## troubleshooting
- connectivity problems 137
 disk partitions full 139
 general steps 137
 transaction log full (ASE) 138
- trunc_reset, rs_ (function) 87
 trunc_set, rs_ (function) 87
- truncate table replication
 enable/disable 71
 subscribing 71
 truncate table partition 71, 88
 with database repdefs 53
- truncate, rs_ (function)..... 85, 88
 truncation point, secondary..... 19
- tvalues, rs_ (RSSD table)..... 126
- ## U
- UDB vs DB2..... 80
 udb_error_class, rs_ (class) 92
 udb_function_class, rs_ (class) 80
 Undocumented commands, disclaimer.. 8
 Unicode..... 37
 unicode_format (config)..... 87
 unified_login (config) 104
 unique primary key, why required 10
 unit of replication 9
 unlocking a locked user 120
 unreserved license keys 13
- update, rs_ (function) 85
 update_lastcommit, rs_ (proc) 33
 update_threads, rs_ (function)..... 87
- ## upgrading
- routes..... 98, 99
 sysadmin upgrade, database 99
 sysadmin upgrade, route..... 98
 with 'repserver_upgrade' 110
 with rs_init 108
- use_batch_markers (config) 36
 use_security_services (config) 103
 use_ssl (config)..... 103
 usedb, rs_ (function) 87
 user, user_raw (variable modifier)..... 88
 username for RepServer, default 116
 username/password, forgotten 141
- ## users
- ID server user 117
 locked 120
 maintenance user..... 117
 primary user (RSSD/ERSSD)..... 117
 RepAgent user 117
 RepServer user 116
 route user..... 118
 subscription user 66, 118
 unlocking 120
- users, rs_ (RSSD table) 126
- ## V
- validate publication..... 63
 validate subscription..... 70
 varbinary_strip_trailing_zeros (config) 37
 varchar_truncation (config)..... 103
 verify_repserver_cmd, admin 46
 version 142
 version, admin..... 98
 version, rs_ (RSSD table)..... 126
- ## W
- wait for create standby 75, 143
 wait for delay 143
 wait for switch..... 75, 143
 wait for time 143
- ## warm standby
- and autocorrection 46
 DDL replication..... 17
 function strings 78
 getting status information 79
 list of replicated DDL stmts/procs... 16
 marking data for replication 17
 'multiple warm standby' (=MSA) 51
 Oracle 74

setup steps	75	writetext, rs_ (function).....	86
switch active, detailed steps	77	writetext_log, rs_ (variable).....	90
switching applications to standby ...	77	writing strings to RS errorlog.....	27
vs. database repdef.....	55	ws_sqldml_replication (config)	78
when	78		
where-clause			
display with rs_helppub	63		
display with rs_helpsub	70		
for a subscription	68		
for an article	63		
whereclauses, rs_ (RSSD table)..	64, 126		
who, admin.....	26		
Windows			
registry key, startup parameters ...	110		
service, for RS.....	109		
		X	
		xact_group_size, dsi_ (config).....	34
		Z	
		zeroltm, rs_ (RSSD prod).....	20, 136

Sample page