



New Features Guide

## **Replication Server®**

15.2

DOCUMENT ID: DC00783-01-1520-01

LAST REVISED: February 2009

Copyright © 2009 by Sybase, Inc. All rights reserved.

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

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

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

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

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

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

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

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

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

# Contents

<b>About This Book .....</b>	<b>ix</b>	
<b>CHAPTER 1</b>	<b>New Features in Replication Server Version 15.2.....</b>	<b>1</b>
	Support for DSI bulk copy-in .....	1
	Setting up bulk copy-in .....	2
	Changes to subscription materialization.....	3
	New counters for bulk copy-in .....	3
	Limitations .....	4
	Non-blocking commit.....	6
	Adaptive Server delayed commit .....	6
	Configuring non-blocking commit .....	6
	System functions that support non-blocking commit .....	7
	Support for non-Adaptive Server databases .....	8
	Product version requirements .....	8
	Quoted identifiers .....	9
	Enabling the quoted identifier feature.....	9
	Marking identifiers as quoted .....	10
	Forwarding quoted identifiers to data servers .....	11
	Changes to rs_helprep .....	12
	Product version requirements .....	15
	Replication Server gateway .....	15
	Cascading connection .....	15
	Enabling the Replication Server gateway.....	16
	Tracking connections .....	17
	Dropping connections.....	17
	Limitations .....	17
	Product version requirements .....	17
	SQL statement replication .....	18
	Modifications to system configuration .....	18
	SQL statement replication configuration .....	20
	Row count verification .....	23
	Configuring warm standby database for SQL replication .....	24
	Replication Server System Database (RSSD) modifications ..	25
	Product and mixed-version requirements.....	25

- Non-Adaptive Server replication support enhancements ..... 26
  - Simplified installation and configuration ..... 26
  - Connection profiles..... 27
- Non-Adaptive Server error class support ..... 30
  - Default non-ASE error classes ..... 30
  - Creating error classes ..... 31
  - Altering error classes..... 31
  - Creating and altering connections ..... 32
  - Native error codes ..... 32

**CHAPTER 2**

- New Features in Replication Server Version 15.1 ..... 33**
  - Dynamic SQL enhancements ..... 33
  - Function replication enhancements ..... 35
  - Adaptive Server shared-disk cluster support ..... 36
  - Monitor and counter enhancements..... 36
    - Active object identification ..... 37
    - New procedures interface ..... 37
  - Improved stable queue management..... 38
    - Changes to sysadmin dump\_queue..... 38
    - Changes to sysadmin sqt\_dump\_queue ..... 39
    - Changes to resume connection..... 39
    - Changes to sysadmin log\_first\_tran ..... 40
    - New sysadmin sqm\_zap\_tran command ..... 40
    - New sysadmin sqm\_unzap\_tran command ..... 40
    - New sysadmin dump\_tran command ..... 41
  - Changed locales directory..... 41
  - Extended password encryption support ..... 42
  - rs\_ticket stored procedure version 2 ..... 42
  - New Replication Server counters..... 43
  - Extended support for large-object (LOB) datatypes ..... 44
  - Partial update of LOB datatypes ..... 45
  - Extended timestamp support ..... 46
  - New opaque datatype ..... 46
  - Dump transaction enhancement ..... 47
    - LTL dump subcommand enhancement..... 47
    - rs\_dumptran enhancement ..... 48
  - Distributor status recording ..... 49
  - Enhanced text update ..... 49
  - Adaptive Server integer identity support ..... 50
  - Stable Queue Manager performance enhancements ..... 50
    - Stable queue caching..... 50
    - Segment preallocation..... 52
    - Support for direct I/O file access ..... 52

<b>CHAPTER 3</b>	<b>New Features in Replication Manager 15.1 .....</b>	<b>53</b>
	Enhanced support for dynamic SQL .....	53
	Enhanced support for function replication definitions.....	54
	Support for large-object datatypes .....	54
	Sybase Central 6.0.....	55
	Support for opaque datatypes.....	55
	Support for timestamp datatypes .....	56
<b>CHAPTER 4</b>	<b>New Features in Replication Server Version 15.0.1.....</b>	<b>57</b>
	Configuration and tuning enhancements .....	57
	Monitor and counter enhancements.....	58
	Dynamic SQL for enhanced Replication Server performance .....	60
	Master database replication .....	62
	rs_subcmp enhancement.....	63
	Schema comparison .....	65
	Manual data reconciliation .....	68
<b>CHAPTER 5</b>	<b>New Features in Replication Manager 15.0.1 .....</b>	<b>71</b>
	Support for dynamic configuration .....	71
	Support for heterogeneous data servers.....	72
	Non-Sybase data servers support.....	72
	Replication Agents and Mirror Replication Agents support.....	73
	Replication Manager plug-in enhancements .....	74
<b>CHAPTER 6</b>	<b>New Features in Replication Server Version 15.0.....</b>	<b>75</b>
	Support for longer identifiers .....	75
	New datatype: bigint.....	76
	New unsigned integer datatypes.....	77
	New Unicode datatype: unitext .....	78
	Replicating computed columns .....	79
	Replicating encrypted columns .....	80
	Replicating partitioned tables .....	80
	Larger disk partitions.....	82
	Larger text and image size.....	83
	Embedded Replication Server System Database (ERSSD) enhancement.....	83
	New password encryption algorithm .....	84
	Mixed-version enhanced support .....	85
	New interface for monitors and counters .....	86
	Support for isolation levels .....	87
	Bidirectional replication support for DDL in MSA .....	89
	Batching of commands for non-ASE servers .....	89

	SySAM license management .....	90
<b>CHAPTER 7</b>	<b>New Features in Replication Manager 15.0 .....</b>	<b>93</b>
	New user interface features .....	93
	Two-tier management solution .....	93
	Three-tier management solution .....	94
	Replication Manager plug-in replaces Replication Server plug-in 94	
	Online help .....	94
	Visual monitoring of status .....	95
	Event Log pane .....	95
	Background processing.....	96
	Replication Manager logging enhancement.....	96
	Script editors .....	97
	Replication Manager features supported .....	98
	Support for new datatypes .....	98
	Support for DirectConnect.....	98
	Replication support.....	99
	Routes .....	99
	Troubleshooting tools.....	100
	Connection status hide options .....	103
	Warm standby wizards .....	104
	Thread management .....	107
<b>CHAPTER 8</b>	<b>Introducing Replication Monitoring Services .....</b>	<b>109</b>
	Introducing Replication Monitoring Services .....	109
	Monitoring servers in the replication environment.....	110
	Software requirements and compatibilities.....	111
	Installation .....	111
	Starting and stopping RMS .....	111
	Connecting to RMS in Sybase Central.....	112
	Monitoring a replication environment using RMS.....	113
	Adding and dropping servers for monitoring in Sybase Central... 113	
	Viewing monitored objects in Sybase Central .....	115
	Setting configuration parameters for monitored replication objects 115	
	Monitoring a logical group of servers .....	116
	Suspending or resuming components in the replication environment.....	117
	Shutting down monitored servers.....	118
	Generating rollup status for servers .....	118
	Generating latency and heartbeat information .....	119

	Adding event triggers .....	119
<b>CHAPTER 9</b>	<b>New Features in Replication Server Version 12.6.....</b>	<b>121</b>
	MultiSite availability (MSA).....	121
	Support for symmetric multiprocessors (SMP).....	123
	The embedded RSSD (ERSSD) .....	124
	Performance enhancements .....	124
	Better management of empty transactions.....	124
	Internal commit control for parallel processing.....	125
	New Replication Server configuration parameters .....	125
	New database configuration parameters.....	125
	Changed database configuration parameters .....	126
	date and time datatypes.....	127
	Support for sending encrypted passwords.....	128
	New bulk materialization method .....	129
	Chinese character set (GB18030) support.....	129
<b>Index .....</b>		<b>131</b>





# About This Book

## **Audience**

This book is intended for customers who are installing and using Replication Server® 15.2 and the subsequent EBFs.

## **How to use this book**

This book describes the new Sybase® Replication Server version 15.2 features.

This book also provides information about the features in Replication Server versions 15.1, 15.0.1, 15.0, and 12.6.

- Chapter 1, “New Features in Replication Server Version 15.2,” describes the Replication Server version 15.2 features.
- Chapter 2, “New Features in Replication Server Version 15.1,” describes the Replication Server version 15.1 features.
- Chapter 3, “New Features in Replication Manager 15.1,” describes the Replication Manager version 15.1 features.
- Chapter 4, “New Features in Replication Server Version 15.0.1,” describes the Replication Server version 15.0.1 features.
- Chapter 5, “New Features in Replication Manager 15.0.1,” describes new features introduced in Replication Manager version 15.0.1.
- Chapter 6, “New Features in Replication Server Version 15.0,” describes the Replication Server version 15.0 features.
- Chapter 7, “New Features in Replication Manager 15.0,” describes the new features introduced in Replication Manager version 15.0.
- Chapter 8, “Introducing Replication Monitoring Services,” introduces the new component in Replication Server 15.0. The Replication Monitoring Services is a middle-management layer for monitoring large and complex replication environments.
- Chapter 9, “New Features in Replication Server Version 12.6,” describes the new and changed features in Replication Server version 12.6.

## **Related documents**

The Sybase Replication Server documentation set consists of:

- 
- The release bulletin for your platform – contains last-minute information that was too late to be included in the books.

A more recent version of the release bulletin may be available on the World Wide Web. To check for critical product or document information that was added after the release of the product CD, use the Sybase Product Manuals at <http://www.sybase.com/support/manuals/>.

- *Installation Guide* for your platform – describes installation and upgrade procedures for all Replication Server and related products.
- *Configuration Guide* for your platform – describes configuration procedures for all Replication Server and related products, and explains how to use the `rs_init` configuration utility.
- *Getting Started with Replication Server* – provides step-by-step instructions for installing and setting up a simple replication system.
- *New Features Guide* (this book) – describes the new features in Replication Server versions 15.2, 15.1, 15.0.1, 15.0, and 12.6.
- *Administration Guide* – contains an introduction to replication systems. This manual includes information and guidelines for creating and managing a replication system, setting up security, recovering from system failures, and improving performance.
- *Design Guide* – contains information about designing a replication system and integrating heterogeneous data servers into a replication system.
- *Heterogeneous Replication Guide* and the Replication Server Options documentation set – describes how to use Replication Server to replicate data between databases supplied by different vendors.
- *Reference Manual* – contains the syntax and detailed descriptions of Replication Server commands in the Replication Command Language (RCL); Replication Server system functions; Sybase Adaptive Server® commands, system procedures, and stored procedures used with Replication Server; Replication Server executable programs; and Replication Server system tables.
- *Troubleshooting Guide* – contains information to aid in diagnosing and correcting problems in the replication system.
- *System Tables Diagram* – illustrates system tables and their entity relationships in a poster format. Available only in print version.
- Replication Manager plug-in help, which contains information about using Sybase Central™ to manage Replication Server.

**Other sources of information**

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

- The Getting Started CD contains release bulletins and installation guides in PDF format, and may also contain other documents or updated information not included on the SyBooks CD. It is included with your software. To read or print documents on the Getting Started CD, you need Adobe Acrobat Reader, which you can download at no charge from the Adobe Web site using a link provided on the CD.
- The SyBooks CD contains product manuals and is included with your software. The Eclipse-based SyBooks browser allows you to access the manuals in an easy-to-use, HTML-based format.

Some documentation may be provided in PDF format, which you can access through the PDF directory on the SyBooks CD. To read or print the PDF files, you need Adobe Acrobat Reader.

Refer to the *SyBooks Installation Guide* on the Getting Started CD, or the *README.txt* file on the SyBooks CD for instructions on installing and starting SyBooks.

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

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

**Sybase certifications on the Web**

Technical documentation at the Sybase Web site is updated frequently.

❖ **Finding the latest information on product certifications**

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

❖ **Finding the latest information on component certifications**

- 1 Point your Web browser to Availability and Certification Reports at <http://certification.sybase.com/>.

- 
- 2 Either select the product family and product under Search by Base Product; or select the platform and product under Search by Platform.
  - 3 Select Search to display the availability and certification report for the selection.

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

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

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

**Sybase EBFs and software maintenance**

❖ **Finding the latest information on EBFs and software maintenance**

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

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

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

**Conventions**

The following style conventions are used in this book:

- In a sample screen display, commands you should enter exactly as shown are in:

`this font`

- In a sample screen display, words that you should replace with the appropriate value for your installation are shown in:

*this font*

- In the regular text of this document, the names of files and directories appear in italics:

*/usr/w/sybase*

- The names of programs, utilities, procedures, and commands appear in this type:

bcp

The conventions for syntax statements in this manual are as follows:

**Table 1: Syntax conventions**

Key	Definition
command	Command names, command option names, utility names, utility flags, and other keywords are in Arial.
<i>variable</i>	Variables, or words that stand for values that you fill in, are in italics.
{ }	Curly braces indicate that you choose at least one of the enclosed options. Do not include braces in your option.
[ ]	Brackets mean choosing one or more of the enclosed options is optional. Do not include brackets in your option.
( )	Parentheses are to be typed as part of the command.
	The vertical bar means you can select only one of the options shown.
,	The comma means you can choose as many of the options shown as you like, separating your choices with commas to be typed as part of the command.

## Accessibility features

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

---

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

---

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

---

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

**If you need help**

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

# New Features in Replication Server Version 15.2

This chapter describes the enhancements that have been made to Replication Server 15.2.

<b>Topic</b>	<b>Page</b>
Support for DSI bulk copy-in	1
Non-blocking commit	6
Quoted identifiers	9
Replication Server gateway	15
SQL statement replication	18
Non-Adaptive Server replication support enhancements	26
Non-Adaptive Server error class support	30

## Support for DSI bulk copy-in

In versions 15.1 and earlier, when Replication Server replicates data to Adaptive Server, Replication Server forms a SQL insert command, sends the command to Adaptive Server, and waits for Adaptive Server to process the row and send back the result of the operation. This process affects Replication Server performance when large batches of data are replicated, such as in end-of-day batch processing or trade consolidation.

Replication Server version 15.2 introduces support for bulk copy-in to improve performance when replicating large batches of insert statements on the same table in Adaptive Server® Enterprise 12.0 and later. Replication Server implements bulk copy-in in Data Server Interface (DSI), the Replication Server module responsible for sending transactions to replicate databases, using the Open Client™ Open Server™ Bulk-Library.

---

**Note** Bulk copy-in is supported only for Adaptive Server databases. If you turn on DSI bulk copy-in and the replicate database is not Adaptive Server, DSI shuts down with an error.

---

For information about the Open Client Open Server Bulk-Library, see the *Open Client and Open Server Common Libraries Reference Manual*.

## Setting up bulk copy-in

These database connection parameters control bulk operations in DSI:

Parameter	Description
dsi_bulk_copy	Turns the bulk copy-in feature on or off for a connection. If <code>dynamic_sql</code> and <code>dsi_bulk_copy</code> are both on, DSI applies bulk copy-in. Dynamic SQL is used if bulk copy-in is not used. Default: off.
dsi_bulk_threshold	The number of insert commands that, when reached, triggers Replication Server to use bulk copy-in. When Stable Queue Transaction (SQT) encounters a large batch of insert commands, it retains in memory the number of insert commands specified to decide whether to apply bulk copy-in. Because these commands are held in memory, Sybase suggests that you do not configure this value much higher than the configuration value for <code>dsi_large_xact_size</code> . Minimum: 1 Default: 20

To set the values of `dsi_bulk_copy` and `dsi_bulk_threshold`, use:

- alter connection to change the bulk copy-in connection parameters at the connection level:
 

```
alter connection to dataserver.database
set {dsi_bulk_copy | dsi_bulk_threshold} to value
```
- configure replication server to change the server defaults:



```
configure replication server
set {dsi_bulk_copy | dsi_bulk_threshold} to value
```

To check the values of `dsi_bulk_copy` and `dsi_bulk_threshold`, use admin config.

When `dsi_bulk_copy` is on, SQT counts the number of consecutive insert statements on the same table that a transaction contains. If this number reaches the `dsi_bulk_threshold`, DSI:

- 1 Bulk-copies the data to Adaptive Server until DSI reaches a command that is not insert or that belongs to a different replicate table.
- 2 Continues with the rest of the commands in the transaction.

Adaptive Server sends the result of bulk copy-in at the end of the bulk operation, when it is successful, or at the point of failure.

---

**Note** The DSI implementation of bulk copy-in supports multistatement transactions, allowing DSI to perform bulk copy-in even if a transaction contains commands that are not part of the bulk copy.

---

## Changes to subscription materialization

Bulk copy-in also improves the performance of subscription materialization. When `dsi_bulk_copy` is on, Replication Server uses bulk copy-in to materialize the subscriptions if the number of insert commands in each transaction exceeds `dsi_bulk_threshold`.

---

**Note** In normal replication, bulk operation is disabled for a table if autocorrection is on. However, in materialization, bulk operation is applied even when autocorrection is enabled, if `dsi_bulk_threshold` is reached and the materialization is not a nonatomic subscription recovering from failure.

---

For more information about subscription materialization, see *Replication Server Administration Guide Volume 1*.

## New counters for bulk copy-in

New counters have been added for bulk copy-in:

Counter	Description
DSINoBulkDatatype	The number of bulk operations skipped due to the data containing datatype is incompatible with bulk copy-in.
DSINoBulkFstr	The number of bulk operations skipped due to tables that have customized function strings for rs_insert or rs_writetext.
DSINoBulkAutoc	The number of bulk operations skipped due to tables that have autocorrection enabled.
DSIEBFBulkNext	The number of batch flushes that executed because the next command is a bulk copy.
DSIEBulkSucceed	The number of times the Data Server Interface executor (DSI/E) invoked blk_done(CS_BLK_ALL) at the target database.
DSIEBulkCancel	The number of times DSI/E invoked blk_done(CS_BLK_CANCEL) at the target database.
DSIEBulkRows	The number of rows that DSI/E sent to the replicate data server using bulk copy-in.
BulkTime	The amount of time, in milliseconds, that DSI/E spent in sending data to the replicate data server using bulk copy-in.

## Limitations

The Replication Server DSI does not use bulk copy-in when:

- Autocorrection is on and the data is not part of subscription materialization.
- rs\_insert has a user-defined function string.
- text column has a user-defined function string for rs\_writetext with output none or rpc.
- The data row contains opaque datatype or a user-defined datatype (UDD) that has an rs\_datatype.canonic\_type value of 255.
- The data row contains an image or a Java datatype.

The bulk copy-in feature is not supported under the conditions listed below. In these instances, disable bulk copy-in.

- The replicate database is not Adaptive Server. In this case, if the DSI bulk copy-in is enabled, DSI terminates with an error message.

- The data size changes between Replication Server and the replicate Adaptive Server character sets, and the datarow contains text columns. In this case, if the DSI bulk copy-in is enabled, DSI terminates with this message:

```
Bulk-Lib routine 'blk_textxfer' failed.  
Open Client Client-Library error: Error: 16843015,  
Severity 1 -- 'blk_textxfer(): blk layer: user  
error: The given buffer of xxx bytes exceeds the  
total length of the value to be transferred.'
```

- The owner.tablename length is larger than 255 bytes and the replicate database is earlier than version Adaptive Server 15.0.3 Interim Release. If the DSI bulk copy-in is enabled, Replication Server terminates with this message:

```
Bulk-Lib routine 'blk_init' failed.
```

To specify not to use bulk copy-in when owner.tablename length is larger than 255 bytes:

- a Turn trace on:

```
trace "on", rsfeature, ase_cr543639
```

- b Add this to the Replication Server configuration file:

```
trace=rsfeature,ase_cr543639
```

Other limitations:

- Unlike the insert command, bulk copy-in does not generate timestamps; NULL values are inserted to the timestamp column if the timestamp column is not included in the replication. Either disable bulk copy-in, or set up your replication definition to include the timestamp column.
- Text and image columns are always logged, even if you change the writetext function string to no log.
- Bulk copy does not invoke insert trigger in Adaptive Server.
- The configuration parameter send\_timestamp\_to\_standby has no effect on bulk copy-in. timestamp data is always replicated.

## Non-blocking commit

Replication Server 15.2 includes non-blocking commit, which uses the delayed commit feature in Adaptive Server to improve replication performance.

---

**Note** To use non-blocking commit, you must use Sybase Enterprise Connect™ Data Access 15.0 ESD #3 (ECDA) or later.

---

## Adaptive Server delayed commit feature

Adaptive Server 15.0 and later includes the delayed commit feature designed to improve performance by delaying the commit phase of a transaction. The commit phase includes writing log records of the transaction to disk and then notifying the client application of the transaction status. With delayed commit, Adaptive Server notifies the client application of a successful commit before writing the corresponding transaction log to disk. This delay in writing to disk reduces contention on the last and active log page and thus improving performance.

However, the last page of a transaction log can be lost, if Adaptive Server terminates or if you shut down Adaptive Server using shutdown with no wait.

See “Using delayed\_commit to determine when log records are committed,” in Chapter 11, “Developing a Backup and Recovery Plan” in the *Adaptive Server Enterprise 15.0 System Administration Guide: Volume 2* and the delayed\_commit parameter of the set command in “set,” in Chapter 1, “Commands” in the *Adaptive Server Enterprise 15.0 Reference Manual: Commands*.

## Configuring non-blocking commit

The dsi\_non\_blocking\_commit configuration parameter extends the period of time Replication Server saves messages after a commit.

Extending the save period requires a larger stable queue. See “Stable queues,” in Chapter 2, “Replication Server Technical Overview” in the *Replication Server Administration Guide Volume 1*.

Use alter connection to configure dsi\_non\_blocking\_commit for a database connection:

```
alter connection to data_server.database  
    set dsi_non_blocking_commit to 'value'
```

---

**Note** You cannot use this parameter with alter connection to configure an active database connection in a standby environment.

---

Use configure replication server to configure dsi\_non\_blocking\_commit as a server default:

```
configure replication server  
    set dsi_non_blocking_commit to 'value'
```

where *value* is the number of minutes, to a maximum of 60, to extend the save period. The default is zero, which disables non-blocking commit.

Use admin config to check the current *value* of dsi\_non\_blocking\_commit.

## System functions that support non-blocking commit

rs\_set\_non\_blocking\_commit and rs\_set\_non\_blocking\_commit\_flush support the non-blocking commit feature.

### ***rs\_set\_non\_blocking\_commit***

rs\_set\_non\_blocking\_commit executes every time DSI connects to the replicate data server, if the dsi\_non\_blocking\_commit value is from 1 to 60. If the value of dsi\_non\_blocking\_commit is zero, rs\_set\_non\_blocking\_commit does not execute.

rs\_set\_non\_blocking\_commit has function-string class scope.

rs\_set\_non\_blocking\_commit function maps to the “set delayed\_commit on” function string in Adaptive Server 15.0 and later, and to the corresponding “alter session set commit\_write = nowait;” function string in Oracle 10g v2 and later. For all other non-Sybase databases, rs\_set\_non\_blocking\_commit maps to null.

### ***rs\_set\_non\_blocking\_commit\_flush***

rs\_set\_non\_blocking\_commit\_flush executes at intervals equal to any number of minutes from 1 to 60 that you specify with dsi\_non\_blocking\_commit.

rs\_set\_non\_blocking\_commit\_flush does not execute if the value of dsi\_non\_blocking\_commit is zero.

`rs_set_non_blocking_commit_flush` has function-string class scope.

`rs_set_non_blocking_commit_flush` maps to the corresponding function string in Adaptive Server 15.0 and later, and Oracle 10g v2 and later. For all other non-Sybase databases, `rs_set_non_blocking_commit_flush` maps to null.

Example

**Example 1** Create an instance of an `rs_set_non_blocking_commit_flush` function string for Adaptive Server:

```
create function string rs_set_non_blocking_commit_flush
  for sqlserver_derived_class
  output language
  'set delayed_commit off; begin tran; update rs_lastcommit set
  origin_time = getdate() where origin = 0; commit tran;
  set delayed_commit on'
```

**Example 2** Create an instance of an `rs_set_non_blocking_commit_flush` function string for Oracle:

```
create function string rs_set_non_blocking_commit_flush
  for oracle_derived_class
  output language
  'alter session set commit_write = immediate; begin tran;
  update rs_lastcommit set origin_time = getdate() where
  origin = 0; commit tran; alter session set commit_write = nowait'
```

## Support for non-Adaptive Server databases

Replication Server 15.2 with non-blocking commit enabled supports replication into Oracle 10gV2 and later because Oracle 10g v2 supports functionality similar to delayed commit.

Replication Server 15.2 heterogeneous datatype support (HDS) scripts have new function strings that support the non-blocking commit feature. Sybase Enterprise Connect Data Access for Oracle supports these function strings. See the *Replication Server Heterogeneous Replication Guide*.

## Product version requirements

You can use `dsi_non_blocking_commit` only with Adaptive Server 15.0 and later, and Oracle 10g v2 and later. Replication Server disables the `dsi_non_blocking_commit` configuration parameter for the connection for unsupported versions of Adaptive Server, Oracle, or other databases.

For details about the commands discussed in this section, see the *Replication Server Reference Manual*.

## Quoted identifiers

Object names that contain special characters such as spaces and non-alphanumeric characters, start with a character other than an alphabetic character, or that correspond to a reserved word, must be enclosed in double quote characters to be parsed correctly. These object names are referred to as quoted identifiers. Although Replication Server version 15.1 and earlier can accept quoted identifiers, forwarding quoted identifiers to data servers is not supported in these versions.

---

**Note** To use quoted identifiers, you must use ECDA 15.0 ESD #3 or later.

---

As of Replication Server 15.2, quoted identifier support allows you to:

- Mark identifiers in a replication definition as quoted.
- Create a connection where you can forward quoted identifiers to data servers.

Embedded double quote characters in identifiers is not currently supported.

Data servers such as Adaptive Server, SQL Anywhere®, Microsoft SQL Server, Universal Database (UDB), and Oracle handle quoted identifiers differently in terms of supported length, special characters, and reserved words. In a heterogeneous environment, ensure that the quoted identifiers being replicated are valid on both the primary and replicate data servers.

## Enabling the quoted identifier feature

The `dsi_quoted_identifiers` configuration parameter enables or disables quoted identifier support in the Data Server Interface (DSI). Use the `create connection` or the `alter connection` command to set `dsi_quoted_identifiers` on or off for a data server connection. The default value of `dsi_quoted_identifiers` is off.

To check the value of `dsi_quoted_identifiers`, use the `admin config` command.

## Marking identifiers as quoted

create replication definition and alter replication definition commands allow you to mark quoted identifiers using the new parameter `quoted`. When an identifier is marked and the `dsi_quoted_identifiers` is set to on, the replicate servers that subscribe to the replication definition receives the marked identifier as a quoted identifier. If the `dsi_quoted_identifiers` is off, the markings are ignored and the replicate server does not receive quoted identifiers.

Modified *create replication definition* syntax

The modified create replication definition syntax is:

```
create replication definition replication_definition
with primary at data_server.database
[with all tables named [table_owner.] 'table_name' [quoted] |
[with primary table named [table_owner.] 'table_name'
with replicate table named [table_owner.] 'table_name' [quoted]]
(column_name [as replicate_column_name] [datatype [null | not null]
[map to published_datatype]] [quoted]
[, column_name [as replicate_column_name]
[datatype [null | not null] computed]
[map to published_datatype]] [quoted]...)
primary key (column_name [, column_name]...)
[searchable columns (column_name [, column_name]...)]
[send standby [{all | replication definition} columns]]
[replicate {minimal | all} columns]
[replicate_if_changed (column_name [, column_name]...)]
[always_replicate (column_name [, column_name]...)]
[with dynamic sql | without dynamic sql]
```

For example, to create a table `foo` with column `foo_col1` as a quoted identifier, enter:

```
create replication definition repdef
with primary at primaryDS.primaryDB
with all tables named "foo"
("foo_col1" int quoted, "foo_col2" int)
primary key ("foo_col1")
```

Modified *alter replication definition* syntax

The modified alter replication definition syntax is:

```
alter replication definition replication_definition
{with replicate table named table_owner.'table_name' |
add column_name [as replicate_column_name]
[datatype [null | not null]]
[map to published_datatype] [quoted],... |
alter columns with column_name
[as replicate_column_name] [quoted | not quoted],... |
alter columns with column_name
datatype [null | not null]
[map to published_datatype],... |
alter columns column_name {quoted | not quoted}}
```



```
add primary key column_name [, column_name]... |
drop primary key column_name [, column_name]... |
add searchable columns column_name [, column_name]... |
drop searchable columns column_name [, column_name]... |
send standby [off | {all | replication definition} columns] |
replicate {minimal | all} columns |
replicate_if_changed column_name [, column_name]... |
always_replicate column_name [, column_name]... |
{with | without} dynamic sql
alter replicate table name {quoted | not quoted}
```

For example, to mark as quoted the table named foo, enter:

```
alter replication definition repdef
alter replicate table name "foo" quoted
```

To unmark the column foo\_col1, enter:

```
alter replication definition repdef
with replicate table named "foo"
alter columns "foo_col1" not quoted
```

---

**Note** When replicating to a warm standby database and to replication definition subscribers, and the primary table name is marked as quoted but the replicate table name is not, or vice-versa, Replication Server sends both the primary table name and the replicate table name as quoted.

---

## Forwarding quoted identifiers to data servers

Data servers receive quoted identifiers differently. Adaptive Server, SQL Anywhere, and Microsoft SQL Server do not expect quoted identifiers, and require a special command to configure the connection for quoted identifiers. Oracle and UDB do not require the connection to be configured to accept quoted identifiers.

Use `rs_set_quoted_identifier` function string to set the DSI connection appropriately for each data server type. When `dsi_quoted_identifiers` is on, Replication Server sends `rs_set_quoted_identifier` to the replicate data server to signal the data server to expect quoted identifiers. If the replicate data server is Adaptive Server, SQL Anywhere, or Microsoft SQL Server, `rs_set_quoted_identifier` is set to set `quoted_identifiers` on command. Otherwise, `rs_set_quoted_identifier` is set to "".

`rs_set_quoted_identifier` has function-string-class scope.

## Changes to rs\_helprep

rs\_helprep stored procedure displays quoted identifiers.

Example

**Example 1** Given this table and replication definition:

```
create table t1 (c1 int, c2 int)
create replication definition r1
  with primary at ost_wasatch_08.pdb1
  with all tables named t1
  (c1 int, "c2" int quoted)
  primary key (c1)
```

rs\_helprep r1 displays c2 as a quoted identifier:

Replication Definition Name	PRS	Type	Creation Date
r1	ost_wasatch_09	Tbl	Nov 11, 2008 2:28PM

PDS.DB	Primary Owner	Primary Table
ost_wasatch_08.pdb1		t1

Replicate Owner	Replicate Table
	t1

Send	Min Cols. Used by Standby	Min Vers	Dynamic SQL	SQL Stmt. Rep.
No	No	1000	On	None

Col. Name	Rep. Col. Name	Datatype	Len.	Pri. Col.	Searchable
c1	c1	int	4	1	0
"c2"	"c2"	int	4	0	0

Function Name	FString Class	FString Source	FString Name
rs_delete	rs_sqlserver_function_class	Class Default	rs_delete
rs_insert	rs_sqlserver_function_class	Class Default	rs_insert
rs_select	rs_sqlserver_function_class	Class Default	rs_select
rs_select_with_lock	rs_sqlserver_function_class	Class Default	rs_select_with_lock
rs_truncate	rs_sqlserver_function_class	Class Default	rs_truncate
rs_update	rs_sqlserver_function_class	Class Default	rs_update

Subscriptions known at this Site 'ost\_wasatch\_09'.

```
Subscription Name  Replicate DS.DB  Owner  Creation Date
-----
(return status = 0)
```

**Example 2** Given the table and replication definition defined in example 1, when you define t1 as a quoted identifier:

```
alter replication definition r1
alter replicate table name "t1" quoted
```

rs\_helprep r1 displays c2 and t1 as quoted identifiers:

```
Replication Definition Name  PRS  Type  Creation Date
-----
r1  ost_wasatch_09  Tbl  Nov 11, 2008 2:28PM
```

```
PDS.DB  Primary Owner  Primary Table
-----
ost_wasatch_08.pdb1  "t1"
```

```
Replicate Owner  Replicate Table
-----
"t1"
```

```
Send Min Cols. Used by Standby Min Vers Dynamic SQL SQL Stmt. Rep.
-----
No  No  1000  On  None
```

```
Col. Name  Rep. Col. Name  Datatype  Len.  Pri. Col.  Searchable
-----
c1  c1  int  4  1  0
"c2"  "c2"  int  4  0  0
```

```
Function Name  FString Class  FString Source  FString Name
-----
rs_delete  rs_sqlserver_function_class  Class Default  rs_delete
rs_insert  rs_sqlserver_function_class  Class Default  rs_insert
rs_select  rs_sqlserver_function_class  Class Default  rs_select
rs_select_  rs_sqlserver_function_class  Class Default  rs_select_
with_lock  with_lock
rs_truncate  rs_sqlserver_function_class  Class Default  rs_truncate
rs_update  rs_sqlserver_function_class  Class Default  rs_update
```

```
Subscriptions known at this Site 'ost_wasatch_09'.
Subscription Name  Replicate DS.DB  Owner  Creation Date
-----
(return status = 0)
```

**Example 3** Given the replication definition defined in example 2, when you define c2 as not quoted:

```
alter replication definition r1
alter columns c2 not quoted
```

rs\_helprep r1 displays t1 as the only quoted identifier:

```
Replication Definition Name PRS                               Type Creation Date
-----
r1                          ost_wasatch_09       Tbl  Nov 11, 2008 2:28PM

PDS.DB                      Primary Owner        Primary Table
-----
ost_wasatch_08.pdb1        "t1"

Replicate Owner            Replicate Table
-----
                            "t1"

Send Min Cols. Used by Standby Min Vers Dynamic SQL SQL Stmt. Rep.
-----
No                          No                    1000    On          None

Col. Name  Rep. Col. Name  Datatype  Len.  Pri. Col.  Searchable
-----
c1         c1              int       4     1          0
c2         c2              int       4     0          0

Function Name  FString Class                FString Source  FString Name
-----
rs_delete     rs_sqlserver_function_class  Class Default  rs_delete
rs_insert     rs_sqlserver_function_class  Class Default  rs_insert
rs_select     rs_sqlserver_function_class  Class Default  rs_select
rs_select_    rs_sqlserver_function_class  Class Default  rs_select_
with_lock     rs_sqlserver_function_class  Class Default  with_lock
rs_truncate   rs_sqlserver_function_class  Class Default  rs_truncate
rs_update     rs_sqlserver_function_class  Class Default  rs_update

Subscriptions known at this Site 'ost_wasatch_09'.
Subscription Name  Replicate DS.DB  Owner  Creation Date
-----
(return status = 0)
```

## Product version requirements

For replication of a quoted identifier to succeed, the primary Replication Server and the Replication Server that connects to the replicate data server version must be 15.2. However, intermediate Replication Servers in a route can be earlier versions.

For details about the commands discussed in this section, see the *Replication Server Reference Manual*.

## Replication Server gateway

In managing a replication system, the replication system administrator (RSA) must log in to multiple replication servers, ID servers, and the corresponding Replication Server System Database (RSSD). The RSA must also frequently switch logins between Replication Server and the RSSD. In version 15.2, Replication Server introduces Replication Server gateway, which minimizes explicit login into the different servers.

The Replication Server gateway uses your RSSD primary user name and password to log in to RSSD, your ID server user name and password to log into ID Server, your remote server identification (RSI) to log into a remote Replication Server, and your maintenance user ID to log into the remote Adaptive Server. You do not need to supply this information more than once, when you access Replication Server itself.

## Cascading connection

The Replication Server gateway also supports cascading connections, which allow your Replication Server to communicate with servers that it is not directly connected to. It also allows you to manage a replication domain using a single client connection. For example, you can connect to an ID Server, and then to the ID Server's RSSD. In this case, both the primary, controlling Replication Server and the ID server are gateways; commands pass through to the ID server's RSSD, and result sets pass back through to you.

## Enabling the Replication Server gateway

The connect command has been added to turn Replication Server into a gateway to its RSSD, ID server, or to a remote Replication Server.

---

**Note** Issuing the connect command requires an sa role for the first login to Replication Server.

---

### Syntax

```
connect [to] [rssd | idserver | srv_name | ds_name.db_name]
```

### Parameters:

- *rssd* – turns Replication Server into a gateway to its RSSD. Allows the gateway to use *RSSD\_primary\_user* and *RSSD\_primary\_pw* entries in its configuration file.
- *idserver* – turns Replication Server into a gateway to its ID server, providing that the Replication Server itself is not the ID server. Allows the gateway to use *ID\_user* and *ID\_pw* entries in the configuration file.
- *srv\_name* – the name of the remote Replication Server you want the gateway to connect to. The Replication Server gateway uses RSI to log into the remote server, and requires a direct route to the remote server.

---

**Note** Replication Server cannot directly connect to itself. However, you can work around this by using a cascading connection.

---

- *ds\_name.db\_name* – the name of the remote data server and database that you want the gateway to connect to. The Replication Server gateway uses the maintenance user to log into the remote data server. This allows you to perform tasks that maintenance users of the designated database are permitted to do. However, you cannot access the other databases defined in the data server you connected to.

Replication Server gateway can directly connect to Adaptive Server, and to Sybase® IQ data servers that do not require Enterprise Connect Data Access (ECDA). For other data servers, Replication Server gateway has to use the ECDA to connect the Replication Server and the remote data server.



## SQL statement replication

Replication Server 15.2 supports SQL statement replication which complements log-based replication and addresses performance degradation caused by batch jobs.

In SQL statement replication, Replication Server receives the SQL statement that modified the primary data, rather than the individual row changes from the transaction log. Replication Server applies the SQL statement to the replicated site. RepAgent sends both the SQL data manipulation language (DML) and individual row changes. Depending on your configuration, Replication Server chooses either individual row change log replication or SQL statement replication.

SQL statement replication includes row count verification to ensure that the number of rows changed in the primary and replicate databases match after replication. If the number of rows do not match, you can specify how Replication Server handles this error.

To enable SQL statement replication:

- Configure the primary database to log SQLDML.
- Configure Replication Server to replicate SQLDML:
  - a Create replications definitions with SQLDML for table and multisite availability (MSA) replication.
  - b In Replication Server, set `WS_SQLDML_REPLICATION` parameter on for warm standby replication.

See the *Adaptive Server 15.0.3 New Features Guide* for more information on SQL statement replication.

## Modifications to system configuration

Several Adaptive Server stored procedures support SQL statement replication.

### Database level

`sp_setrepdbmode` has been added to support SQL statement replication.

`sp_setrepdbmode`

`sp_setrepdbmode` allows you to enable SQL statement replication for a specific DML operation.

The DML operations that apply to SQL statement replication include:



- U – update
- D – delete
- I – insert select
- S – select into

When the database replication mode is set to any combination of UDIS the RepAgent sends both individual log records and the information needed by Replication Server to build the SQL statement.

For example, to replicate delete statements as SQL statement replication and also enable replication of select into, enter:

```
sp_setrepdbmode pdb, 'DS', 'on'
```

You can set SQL statement replication at the database level only when the database has been marked for replication by setting `sp_reptostandby` to ALL or L1.

See “`sp_setrepdbmode`” in the “System Changes” chapter in the *Adaptive Server Enterprise 15.0.3 New Features Guide*.

## Table level

`sp_setrepdefmode` has been enhanced to support the SQL statement replication.

`sp_setrepdefmode`

`sp_setrepdefmode` includes options to:

- Enable or disable SQL statement replication for a specific DML operation
- Configure the threshold that must be reached to activate SQL statement replication

The DML operations that apply to SQL statement replication include:

- U – update
- D – delete
- I – insert select

When the table replication mode is set to any combination of UDI, the RepAgent sends additional information to enable SQL statement replication for the specified DML operation.

For example, to enable SQL statement replication for update, delete, and insert select operations on table `t`, enter:

```
sp_setrepdefmode t, 'UDI', 'on'
```

```
go
```

See “sp\_setrepdefmode” in the “System Changes” chapter in the *Adaptive Server Enterprise 15.0.3 New Features Guide*.

## Session level

Use session option set repmode to set replication mode to SQL statement replication. You can specify session-level settings either when you log in, or at the beginning of a batch job. Session-level settings override both database-level and object-level settings.

Use set repmode on to enable SQL statement replication for the DML operation specified, for the duration of the session. Use set repmode off to remove all SQL statement replication settings at the session level. For example, to replicate only select into and delete as SQL statements for the duration of the session, enter:

```
set repmode on 'DS'
```

See “set repmode” in the “System Changes” chapter in the *Adaptive Server Enterprise 15.0.3 New Features Guide*.

## SQL statement replication configuration

You can change replication options at database and table levels.

### Database replication definition

To replicate SQL statements in an MSA environment, you must include the replicate SQLDML clause with the create replication definition or alter replication definition commands.

This code segment displays the syntax for create and alter database replication definitions:

```
[[not] replicate setname [in (table list)] ]
```

where:

setname = DDL | tables | functions | transactions | system procedures | SQLDML | 'options'.

The 'options' parameter is a combination of:

- U – update

- D – delete
- I – insert select
- S – select into

The SQLDML parameter is also defined as a combination of U, D, I, and, S statements.

This example shows how to use the 'options' parameter to replicate SQLDML on tables tb1 and tb2:

```
replicate 'UDIS' in (tb1,tb2)
```

This example shows how to use the SQLDML parameter that produces the same result as the 'options' parameter in the previous example:

```
replicate SQLDML in (tb1,tb2)
```

You can use multiple replicate clauses in a create database replication definition. However, for an alter database replication definition, you can use only one clause.

If you do not specify a filter in your replication definition, the default is the not replicate clause. Apply alter database replication definition to change the SQLDML filters. You can either specify one or multiple SQLDML filters in a replicate clause.

#### Examples

This example shows how to filter out the select into statement for all tables. The second clause, not replicate 'U' in (T), filters out updates on table T:

```
create database replication definition dbrepdef
with primary at ds1.pdb1
not replicate 'S'
not replicate 'U' in (T)
go
```

This example enables update and delete statements on all tables using the replicate 'UD' clause:

```
create database replication definition dbrepdef_UD
with primary at ds2.pdb1
replicate 'UD'
go
```

You can use multiple clauses to specify a table multiple times in the same definition. However, you can use each of U, D, I, and S only once per definition.

```
create database replication definition dbrepdef
with primary at ds2.pdb1
replicate tables in (tb1,tb2)
```

```
        replicate 'U' in (tb1)
        replicate 'I' in (tb1,tb2)
go
```

This example applies update and delete statements for tables tb1 and tb2:

```
alter database replication definition dbrepdef
with primary at ds1.pdb1
replicate 'UD' in (tb1,tb2)
go
```

### Table replication definition

To support SQL statement replication, you must include the replicate SQLDML clause for a create table replication definition. This code segment displays the syntax for a create table replication definition:

```
[replicate {SQLDML ['off'] | 'options'}]
```

The 'options' parameter is a combination of these statements:

- U – update
- D – delete
- I – insert select

---

**Note** If your replication definition has the [replicate {minimal | all} columns] clause, then the [replicate {minimal | all} columns] clause must always precede the [replicate {SQLDML ['off'] | 'options'}] clause.

---

This is a sample create replication definition for a table:

```
create replication definition repdef1
with primary at ds3.pdb1
with all tables named 'tb1'

(id_col int,
str_col char(40))

primary key (id_col)
replicate all columns
replicate 'UD'
go
```

A table replication definition with the `send standby` clause can specify a replicate 'I' statement. You can replicate an insert select statement as a SQL replication statement only in warm standby or MSA environments. A table replication definition without a `send standby` clause cannot replicate the insert select statement.

## SQL statement replication restrictions

SQL statement replication is not supported when:

- A replicate database has a different table schema than the primary database.
- Replication Server must perform data or schema transformation.
- Subscriptions or articles include `where` clauses.
- Updates include one or more text or image columns.
- Function strings `rs_delete`, `rs_insert`, and `rs_update` are customized.

## SQL statement replication does not support autocorrection

SQL statement replication cannot perform autocorrection. If Data Server Interface (DSI) encounters a DML command for SQL statement replication and autocorrection is on, by default, DSI is suspended and stops replication. Use the `assign action` command with error number 5193 to specify how Replication Server handles this error.

Replication Server does not replicate SQLDML until the table level subscription is validated.

## Row count verification

In Replication Server 15.2, you can specify how Replication Server responds to SQLDML row count errors that may occur during SQL statement replication. SQLDML row count errors occur when the number of rows changed in the primary and replicate databases do not match after SQL statement replication. The default error action is to stop replication.

Syntax

Use the `assign action` command at the primary site for the Replication Server error class to specify other error actions for SQLDML row count errors:

```
assign action  
{ignore | warn | retry_log | log | retry_stop | stop_replication}
```

```
for error_class
to server_error1 [, server_error2]..
```

*error\_class* is the error class name for which the action is being assigned. With Replication Server 15.2, you can specify Replication Server error classes such as the default `rs_repserver_error_class` error class.

*server\_error* is the error number. With Replication Server 15.2, you can specify error numbers for Replication Server:

<b>server_error</b>	<b>Description</b>	<b>Default error action</b>
5186	Row count verification error for SQL statement replication if the affected number of rows is different from what is expected.	stop_replication
5193	Cannot enable autocorrection if SQL statement replication is enabled. Either enable SQL statement replication only or disable SQL statement replication before you enable autocorrection.	stop_replication

#### Example

To execute the `assign action` command using the `rs_repserver_error_class` to stop replication if Replication Server encounters error number 5186, enter:

```
assign action stop_replication for rs_repserver_error_class to 5186
```

If there is a row count error, this is an example of the error message that displays:

```
DSI_SQLDML_ROW_COUNT_INVALID 5186
Row count mismatch for SQLDML command executed on
'mydataserver.mydatabase'.
The command impacted 1000 rows but it should impact 1500
rows.
```

## Configuring warm standby database for SQL replication

By default, warm standby applications do not replicate the DML commands that support SQL statement replication. To use SQL replication, you can:

- Create table replication definitions using `replicate SQLDML` and send standby clauses.

- Set the `WS_SQLDML_REPLICATION` parameter to on. The default value is `UDIS`. However, `WS_SQLDML_REPLICATION` has a lower precedence than the table replication definition for SQL replication. If your table replication definition contains a `send standby` clause for a table, the clause determines whether or not to replicate the DML statements, regardless of the `WS_SQLDML_REPLICATION` parameter setting.

## Replication Server System Database (RSSD) modifications

These system tables in the RSSD support SQL statement replication:

- `rs_dbreps` – `status` column includes 4 new sets of 2-bit sets, each of which corresponds to a DML filter. The first bit of a set indicates if it is an empty filter and the second bit indicates if it is a negative statement set.
- `rs_dbsubsets` – `type` column includes four new types: `U`, `L`, `I`, and `S` corresponding to the DML `UDIS` filters. In this case, `L` is used for delete instead of `D`.
- `rs_objects` – `attributes` column includes five new bits; one for each `U`, `D`, `I`, or `S` operation, and one to indicate if a table replication definition has fewer columns than the number of incoming data rows.

A system function replication definition, `rs_sqldml`, also supports SQL statement replication.

## Product and mixed-version requirements

SQL statement replication requires Adaptive Server version 15.0.3 and later, primary and replicate Replication Server version 15.2 and later, and route version 15.2 and later.

For details about the commands discussed in this section, see the *Replication Server Reference Manual*.

## Non-Adaptive Server replication support enhancements

Replication Server 15.2 includes several enhancements to installation, configuration, and overall usability of setting up replication environments that include actively supported non-Adaptive Server Enterprise (non-ASE) data servers. These enhancements automate the installation and configuration process by providing a pre-configured Replication Server environment that enables replication involving actively supported non-ASE data servers to be up and running quickly.

Actively supported data servers are data servers for which Sybase provides all the required software, documentation, and support for the data servers to serve as both a primary or a replicate data server. See the *Replication Agent Release Bulletin* for your platform for the list of actively supported non-ASE data servers.

### Simplified installation and configuration

With Replication Server 15.2, you do not need to edit and execute scripts to install datatype definitions, function strings, and class-level translations for heterogeneous (non-ASE) datatype support. The functions provided by the scripts are included as part of the Replication Server 15.2 installation, or are included in connection profiles that are installed with Replication Server 15.2. These enhancements simplify installation and configuration for non-ASE environments. Follow the simplified configuration instructions in Chapter 8, “Installing and Implementing Non-ASE Support Features” in the *Replication Server 15.2 Configuration Guide* for your platform.

In addition, Replication Server 15.2 provides error class support for non-ASE replicate databases. See “Non-Adaptive Server error class support” on page 30.



## Connection profiles

With Replication Server 15.2, you can use connection profiles that contain connection configurations and replicate database object definitions relevant to each type of actively supported non-ASE data server. You can use these connection profiles and simple syntax to create connections between actively supported data servers such as Adaptive Server Enterprise, IBM DB2, Microsoft SQL Server, and Oracle. Replication Server uses the connection profile to configure the connection and create replicate database objects for you.

Connection profiles specify the function-string class, error class, and class-level translations to be installed. You can also use connection profile options to specify other actions such as whether commands should be batched and the command separator to use.

---

**Note** When you create a connection using a connection profile, the system table services (STS) caches are refreshed so that you do not need to restart Replication Server.

---

## Using connection profiles

Use the using profile clause with the create connection command to create a connection between a non-ASE database and Adaptive Server using a connection profile.

### Syntax

Here is a portion of the create connection syntax that shows the using profile and display\_only clauses:

```
create connection to data_server.database
using profile connection_profile[:version]
set username [to] user
[other_create_connection_options]
[display_only]
```

Parameters:

- *connection\_profile* – provide the connection profile you want to use to configure a connection, modify the Replication Server System Database (RSSD), and build replicate database objects.
- *version* – specify the particular version of a connection profile.

- *other\_create\_connection\_options* – use *other\_create\_connection\_options* to set connection options not specified in the profile, such as setting your password, or to override options specified in the profile, such as specifying a custom function string class to override the function string class provided in Replication Server. See “create connection,” in Chapter 3, “Replication Server Commands” in the *Replication Server Reference Manual* for all the parameters you can use with create connection.
- *display\_only* – use with a connection profile to display but not execute commands and the servers where the commands would be executed. Use the client and Replication Server logs to see the results of using *display\_only*.

## Examples

**Example 1** Creates a connection to an Oracle replicate database:

```
create connection to oracle.instance
using profile rs_ase_to_oracle
set username to ora_maint
set password to ora_maint_pwd
```

**Example 2** Creates a connection to a Microsoft SQL Server replicate database that is also a primary database. In this example, the command replaces any error class setting provided by the connection profile with another error class—*my\_msss\_error\_class*:

```
create connection to msss_server.msss_db
using profile rs_ase_to_msss
set username to msss_maint
set password to msss_maint_pwd
set error class to my_msss_error_class
with log transfer on
```

**Example 3** Creates a connection to a DB2 replicate database using a specific version of the profile—*v9\_1*. In this example, the command overrides the command batch size provided by the connection profile with a new value—16384:

```
create connection to db2.subsys
using profile rs_ase_to_db2:v9_1
set username to db2_maint
set password to db2_maint_pwd
set dsi_cmd_batch_size to '16384'
```

**Example 4** Use the *display\_only* option to show the commands that will be executed if you use a particular profile. The commands and the command output display on your screen and are also written to the Replication Server log:

```
create connection to oracle.instance
using profile rs_ase_to_oracle
```

```
set username to ora_maint
set password to ora_maint_pwd
display_only
go
```

## Listing available connection profiles

Use the `admin show_connection_profiles` command to list the profile name, version, and comments for each profile defined in Replication Server.

**Syntax** `admin show_connection_profiles[, "match_string"]`

Use the `match_string` option to display only the connection profiles whose names contain the string you provide in the option.

**Examples** **Example 1** Lists the names of all connection profiles currently defined in Replication Server:

```
admin show_connection_profiles
go
```

Extract of output is:

Profile Name	Version	Comments
-----	-----	-----
rs_ase_to_db2	Standard	Standard ASE to DB2 replication connection profile.
rs_ase_to_udb	Standard	Standard ASE to DB2 replication connection profile.
...		
rs_db2_to_ase	Standard	Standard DB2 to ASE replication connection profile.
rs_db2_to_msss	Standard	Standard DB2 to Microsoft SQLServer connection profile.
...		

There is a connection profile for each primary and replicate database combination, such as Adaptive Server and Oracle, Oracle and Adaptive Server, and IBM DB2 and /Microsoft SQL Server, and so on.

See “`admin show_connection_profiles`,” in Chapter 3, “Replication Server Commands” in the *Replication Server Reference Manual* for the list of connection profiles.

**Example 2** Lists the names of all connection profiles currently defined in Replication Server that have the string “oracle” in the connection profile name:

```
admin show_connection_profiles, "oracle"
go
```

Output is:

Profile Name	Version	Comments
-----	-----	-----
rs_ase_to_oracle	Standard	Standard ASE to Oracle replication connection profile.

## System tables for connection profiles

Two system tables in the RSSD support connection profiles:

- rs\_profile – stores currently defined profiles in Replication Server.
- rs\_profdetail – stores the profile details.

For more information on support for non-ASE data servers, see the *Replication Server Heterogeneous Replication Guide* and the *Replication Server Administration Guide Volume 1*. For details about the commands discussed in this section, see the *Replication Server Reference Manual*. For the updated configuration process, see the *Replication Server Configuration Guide* for your platform.

## Non-Adaptive Server error class support

Replication Server 15.2 provides support for error classes and error action mapping for non-Adaptive Server Enterprise (non-ASE) replication databases. You can use the default non-ASE error classes included in Replication Server 15.2. You can also create and alter your own error classes for non-ASE replicate databases.

### Default non-ASE error classes

The default non-ASE error classes provided with Replication Server 15.2 are listed in Table 1-1. You cannot modify these default error classes.

**Table 1-1: Non-ASE error classes**

Database	Class name
IBM DB2	rs_db2_error_class
IBM UDB	rs_udb_error_class
Microsoft SQL Server	rs_mssql_error_class
Oracle	rs_oracle_error_class

## Creating error classes

Replication Server 15.2 includes the set template to option for the create error class command. Use create error class, and set template to, and another error class as a template, to create your own error classes. create error class copies the error actions from the template error class to the new error class.

Syntax `create error class error_class  
[set template to template_error_class]`

Example This example creates the my\_error\_class error class for an Oracle database based on rs\_oracle\_error\_class as a template:

```
create error class my_error_class set template to
rs_oracle_error_class
```

## Altering error classes

Use the alter error class command, and another error class as a template, to alter error classes. alter error class copies error actions from the template error class to the error class you want to alter and overwrites error actions that have the same error code

Syntax `alter error class error_class  
set template to template_error_class`

Example For example, to alter my\_error\_class for an Oracle database based on rs\_sqlserver\_error\_class as a template:

```
alter error class my_error_class set template to
rs_sqlserver_error_class
```

## **Creating and altering connections**

In Replication Server 15.2, you can use the create connection and alter connection commands to assign non-ASE error classes to specific connections on non-ASE replication databases.

## **Native error codes**

When Replication Server establishes a connection to a non-ASE replicate server, Replication Server verifies if the option to return native error codes from the non-ASE replicate server is enabled for the connection. If the option is not enabled, Replication Server logs a warning message that the connection works but error action mapping may not be correct.

For more information about error classes and error handling, see the *Replication Server Administration Guide Volume 2*. For details about the commands discussed in this section, see the *Replication Server Reference Manual*.

# New Features in Replication Server Version 15.1

This chapter describes the enhancements that have been made to Replication Server 15.1.

<b>Topic</b>	<b>Page</b>
Dynamic SQL enhancements	33
Function replication enhancements	35
Adaptive Server shared-disk cluster support	36
Monitor and counter enhancements	36
Improved stable queue management	38
Changed locales directory	41
Extended password encryption support	42
rs_ticket stored procedure version 2	42
New Replication Server counters	43
Extended support for large-object (LOB) datatypes	44
Partial update of LOB datatypes	45
Extended timestamp support	46
New opaque datatype	46
Dump transaction enhancement	47
Distributor status recording	49
Enhanced text update	49
Adaptive Server integer identity support	50
Stable Queue Manager performance enhancements	50

## Dynamic SQL enhancements

Dynamic SQL in Replication server enhances replication performance by allowing Replication Server Data Server Interface (DSI) to prepare dynamic SQL statements at the target user database and to run them repeatedly.

Dynamic SQL now supports heterogeneous replicate databases including Oracle, Universal Database (UDB), DB2, and Microsoft SQL. create/alter replication definition commands allow you to control the application of dynamic SQL on each table through replication definition.

See the *Replication Server Reference Manual* for information about create/alter replication definition commands.

You can change the dynamic SQL execution at the table level for a specific replicate database by using:

```
set dynamic_sql {on | off}  
for replication definition with replicate at  
data_server.database
```

To check for dynamic SQL usage, turn on stats\_sampling and run admin stats, dsi command and look for DSIEDsqlPrepared, DSIEDsqlExecuted and other dynamic SQL related counters.

Use stored procedures rs\_helprep, rs\_helpsub, and rs\_helppubsub to display dynamic SQL setting for each replication definition.

See “rs\_helprep”, “rs\_helpsub”, and “rs\_helppubsub,” in Chapter 6, “Adaptive Server Stored Procedures” in the *Replication Server Reference Manual* for information about using these stored procedures.

### Limitations

Dynamic SQL commands support the data within Sybase range. If data falls outside Sybase ranges that cause dynamic SQL to fail, DSI logs an error message and resends dynamic SQL using the language command. DSI shuts down only if the language command also fails.

If this condition happens frequently, disable dynamic SQL from the table replication definition or use the set dynamic\_sql off command.

Use any of these commands to turn dynamic\_sql off:

- alter connection... set dynamic\_sql off – turns dynamic SQL off for all commands replicating to this connection.
- create/alter replication definition...without dynamic\_sql – turns dynamic SQL off for all commands using this replication definition.
- set dynamic\_sql off for replication definition with replicate at... – turns dynamic SQL off for all commands using this replication definition at this replicate connection.



## Function replication enhancements

In Replication Server 15.1, you can create a function replication definition that has a different name than the function name.

Replication Server enforces different primary and replicate function names for the request function replication definition. The `maint_user` runs the transaction at the replicate database if the function is replicated through applied function replication definition. The `origin_user` runs the transaction if the function is replicated through request function replication definition at the replicate database.

These enhancements let you:

- Replicate multiple functions with the same name from different databases.
- Have multiple replication definitions for one primary function and each specifies a different replicate function for a different replicate site.

To manage function replication definition, use:

- create applied function replication definition
- create request function replication definition
- alter applied function replication definition
- alter request function replication definition

### Mixed-version support

This enhancement supports mixed version environments. However, function replication definitions that have different primary function name and replication definition name are not replicated to Replication Servers earlier than 15.1.

---

**Warning!** If your system has an earlier version of a request function replication definition, drop the earlier-version definition before creating a 15.1 replication definition for the same primary function.

---

### Warm standby and multisite availability (MSA) support

In a warm standby or MSA environment, there is only one parameter list for all the function replication definitions of the same primary function. If you alter one function replication definition to add a parameter, the new parameter is added to all the function replication definitions created for this function.

See Chapter 3, “Replication Server Commands” in the *Replication Server Reference Manual* for detailed information about these commands.

### Limitations

Enhanced function replication has these limitations:

- All the function replication definitions created for the same function must have the same parameter list with the same name and datatype.
- If you created a function replication definition with differing primary function name and replication definition name in version 15.1, any earlier version of the request function replication definition for the same primary function is disabled.
- You cannot have both applied function replication definitions and request function replications for a primary function. The function replication definition created by using create function replication definition command is considered an applied function at the function primary site.
- For each applied and request function replication definition, you must create a corresponding subscription to replicate a function.

## **Adaptive Server shared-disk cluster support**

Replication Server and RepAgent thread both support the Adaptive Server shared-disk cluster environment, which is where many Adaptive Servers share a single set of disks or databases. In a Sybase shared-disk cluster, a database can be either a replication source or a replication destination. You can perform all tasks, such as configuring RepAgent or marking tables for replication, from any instance in the cluster.

See Chapter 5 “Setting Up and Managing RepAgent” in the *Replication Server Administration Guide Volume 1*.

## **Monitor and counter enhancements**

The enhanced monitors and counters feature allows you to collect the information of the most active tables, procedures and related statistics, and store this information into the Replication Server System Database (RSSD) rs\_statdetail table and related tables. You can use this information to diagnose the replicate database performance issues such as missing indexes on the primary keys, and latency problems in the Replication Agent™ and Stable Queue Transaction interface (SQT)/Distributor (DIST) processing.

## Active object identification

New counters have been added to count the statement execution time on tables and procedures:

- AOBJInsertCommand
- AOBJUpdateCommand
- AOBJDeleteCommand
- AOBJWritetextCommand
- AOBJExecuteCommand

To flush the active object counter metrics to the RSSD, run any of these commands:

- admin stats, "all", save
- admin stats, dsi, save
- admin stats, sysmon, save

See "rs\_helpcounter" in the *Replication Server Reference Manual* for details on displaying information about counters.

## New procedures interface

To bring out the most active tables and procedures, and related statistics information, Replication Server 15.1 introduces the stored procedures `rs_stat_populate` and `rs_stat_genreport`. `rs_stat_populate` reads data from `rs_statdetail`, summarizes, augments, denormalizes, and saves result into `rs_statreport`, where `rs_stat_genreport` reads data and generates report.

Load this script into the RSSD after upgrading to Replication Server 15.1:

```
$$SYBASE/$SYBASE_REP/scripts/rs_install_statreport_v1510_[ase/asa].sql
```

After loading the script, run the stored procedures `rs_stat_populate` and `rs_stat_genreport`. Running these stored procedures produces this information:

- Replication Server performance overview – overview information about your Replication Servers, such as DIST processing, DSI processing, and so on.
- Replication Server performance analysis – performance analysis and tuning suggestions based on critical Replication Server counters. The detailed description is available in the script file.

- Active object identification result – lists the active table and procedure names, owner names, execution times, and so on.

For more information about `rs_stat_populate`, `rs_stat_genreport`, `rs_statreport`, and `rs_statdetail`, see the script file, which contains syntax, examples, and so on.

## Improved stable queue management

Replication Server 15.1 simplifies stable queue management. Enhanced queue dump commands provide flexibility in identifying the stable queues, controlling the stable queue contents to dump, and supporting additional output file options. Replication Server 15.1 also includes new commands that allow you to delete and restore specific transactions from the Stable Queue Manager (SQM).

For more information about the stable queue management, see the *Replication Server Administration Guide Volume 1*. For details about the commands discussed in this section, see the *Replication Server Reference Manual*.

## Changes to `sysadmin dump_queue`

Enhancements to `sysadmin dump_queue` include:

- an option to specify the server or database name instead of the queue number when identifying the stable queue to dump.
- an option to specify the number of commands to dump.
- filtering options such as dumping only the begin and end commands of a transaction and dumping everything in the queue as comments except SQL statements.
- the option to direct the output to the Replication Server log or to a user-defined log file.
- an option to start the data dump from where the previous `sysadmin dump_queue` command stopped for that particular queue and session.

The modified `sysadmin dump_queue` syntax is:

```
sysadmin dump_queue {, q_number | server [,database]},  
                    {q_type,seg, blk, cnt  
                    [, num_cmds]
```

```
[, {L0 | L1 | L2 | L3}]  
[, {RSSD | client | "log" | file_name}] |  
"next" [, num_cmds}]
```

## Changes to *sysadmin sqt\_dump\_queue*

Enhancements to `sysadmin sqt_dump_queue` include:

- an option to specify the server or database name instead of the queue number when identifying the stable queue to dump.
- an option to dump all committed transactions and read transactions found in the SQT cache.
- an option to specify the number of commands to dump.
- filtering options such as dumping only the begin and end commands of a transaction and dumping everything in the queue as comments except SQL statements.
- the option to direct the output to the Replication Server log or to a user-defined log file.

The modified `sysadmin sqt_dump_queue` syntax is:

```
sysadmin sqt_dump_queue {, q_number | server [,database]},  
q_type, reader  
[, {open | closed | read}]  
[, num_cmds]  
[, {L0 | L1 | L2 | L3}]  
[, {RSSD | client | "log" | file_name}]
```

## Changes to *resume connection*

The `resume connection skip transaction` option has been enhanced to support skipping a specified number of transactions in the connection queue before resuming the connection. Skipped transactions are written to the database exception log, and to either the Replication Server log or the alternative log file specified by the `sysadmin dump_file` command. The maximum number of transactions that this command can skip is the number of transactions in the Data Server Interface (DSI) outbound queue.

The modified `resume connection` syntax is:

```
resume connection to data_server.database  
[skip [n] transaction | execute transaction]
```

## Changes to `sysadmin log_first_tran`

A new option *n* has been added to the `sysadmin log_first_tran` command. Use the new option to specify the number of transactions to write to the database exceptions log, and to either the Replication Server log or the alternative log file specified by the `sysadmin dump_file` command.

The modified `sysadmin log_first_tran` syntax is:

```
sysadmin log_first_tran [,n], data_server, database
```

## New `sysadmin sqm_zap_tran` command

`sysadmin sqm_zap_tran` deletes a specific transaction from the stable queue and returns a message stating the number of deleted commands. The transaction is identified through the local queue ID (LQID).

The `sysadmin sqm_zap_tran` syntax is:

```
sysadmin sqm_zap_tran {, q_number, | server [,database]},  
q_type, lqid [, {L0 | L1 | L2 | L3}]  
[, {RSSD | client | "log" | file_name}]
```

---

**Note** Replication Server must be in standalone mode to use this command.

---

## New `sysadmin sqm_unzap_tran` command

`sysadmin sqm_unzap_tran` restores a transaction in the stable queue and returns a message stating the number of restored transaction commands. The transaction is identified through the LQID.

The `sysadmin sqm_unzap_tran` syntax is:

```
sysadmin sqm_unzap_tran {, q_number, | server [,database]},  
q_type, lqid [, {L0 | L1 | L2 | L3}]  
[, {RSSD | client | "log" | file_name}]
```

---

**Note** Replication Server must be in standalone mode to use this command.

---

## New `sysadmin dump_tran` command

Use `sysadmin dump_tran` to dump the statements of a specific stable queue transaction into a log file. The transaction is identified through the LQID.

The `sysadmin dump_tran` syntax is:

```
sysadmin dump_tran {{, q_number, | server [,database]},
                    q_type, lqid
                    [, num_cmds]
                    [, {L0 | L1 | L2 | L3}}
                    [, {RSSD | client | "log" | file_name}} |
                    "next" [, num_cmds]}
```

## Changed `locales` directory

The Replication Server release area and localization directory structure have been modified.

You can now install or uninstall multiple Sybase products, such as Replication Server and Adaptive Server, on the same computer and same directory. You can also install multiple versions of Replication Server in the same directory.

The changes to the `locales` directory include:

- Replication Server-specific `locales` files have been moved from `$$SYBASE/locales` to a new directory `$$SYBASE/$$SYBASE_REP/locales`.
- All `<charset>` subdirectories have been consolidated into one `<utf8>` subdirectory for each language in the new directory `$$SYBASE/$$SYBASE_REP/locales`.

The UTF-8 character set that is used for all supported languages instead of using different character sets. You can convert UTF-8 into other characters and vice versa.

Replication Server reads messages from a localization file, and converts the messages to the specified character set format during runtime.

- `rs_init` locale files are now located in `$$SYBASE/$$SYBASE_REP/locales/<language>/utf8/sybinit`.

## Extended password encryption support

Replication Server 15.1 uses Sybase Common Security Infrastructure (CSI) for server or client authentication, cryptography for encryption and decryption of passwords that are stored in the RSSD tables, and key-pair generation to support extended password encryption.

Extended password encryption uses asymmetric key encryption, which allows Open Client applications with connection property `CS_SEC_EXTENDED_ENCRYPTION` enabled to connect to Replication Server. It also allows Replication Server to enable `CS_SEC_EXTENDED_ENCRYPTION` when connecting to other servers.

Asymmetric key encryption uses a public key to encrypt the password and a private key to decrypt the password. The private key is not shared across the network, and is therefore secure.

---

**Note** To use the extended password encryption feature, you must have a server that supports extended password encryption, such as Adaptive Server 15.0.2 ESD #2 or later. Additionally, this feature is not supported in HP-Itanium platform in 15.1 release.

---

## *rs\_ticket* stored procedure version 2

`rs_ticket` is now on its version 2 and provides support for non-ASE databases. You can use `rs_ticket` stored procedure without additional configuration and administration. Tickets are automatically inserted into the `rs_ticket_history` table, which is located in the replicate database. Tickets are sharable among multiple applications, where they are issued without obstruction from tickets of other applications.

With `rs_ticket` version 2, more ticket information is provided for better usability, such as the Replication Server Interface (RSI) timestamp, which you can retrieve without writing complex queries. You can directly query the `rs_ticket_history` table for Replication Server performance. If the computer time or tickets are not synchronized across multiple time zones, you can change the timestamp columns to adjust the ticket date.



The earlier version of `rs_ticket` has been renamed `rs_ticket_v1`. To use the earlier version, alter `rs_ticket_report` function string with your old content or with the default, `exec rs_ticket_report ?rs_ticket_param!param?`.

---

**Note** If you previously disabled `dsi_rs_ticket_report` and upgraded to Replication Server 15.1, the `dsi_rs_ticket_report` setting is reenabled after the upgrade process has finished.

---

These are the format changes that have been made to `rs_ticket`:

- Changed version number to 2, `V=2`; if a ticket has version number equal to 1, Replication Server does not write date to tickets.
- The ticket size has been increased from 255 to 1024 bytes.
- The timestamp format has been changed to include the date. The new timestamp format is `mm/dd/yy hh:mm:ss:mmm`.
- A Replication Server Interface (RSI) module timestamp that forces the RSI sender to parse RSI messages has been added. Tickets pass more than one RSI modules. However, the `rs_ticket_history` table keeps only the last RSI timestamp.
- Primary and target Replication Server names have been added to identify where a ticket comes from and where it goes to.
- Primary and replicate database names have been added.
- Two DSI counters have been added:
  - `DSI_T=xxx` – total transactions that the Data Server Interface (DSI) reads.
  - `DSI_C=xxx` – total commands that the DSI reads.

See the *Replication Server Reference Manual* for detailed information about using `rs_ticket` version 2.

## New Replication Server counters

New counters have been added for `REPAGENT`, `RSIUSER`, `SQM`, `DSI`, and `DSIEXEC`:

Counter	Description
RepAgentExecTime	The amount of time, in milliseconds, that the RepAgent user thread is scheduled by Open Client/Server™.
RSIUExecTime	The amount of time, in milliseconds that the RSI user thread is scheduled by Open Client/Server.
SQMWaitSegTime	The amount of time waiting for allocating segments.
DSINoDsqlNULL	Number of commands that cannot use dynamic SQL statements because of NULL value in where clauses.
DSINoDsqlDatatype	Number of commands that cannot use dynamic SQL statements because of text, image, java and ineligible UDDs
DSINoDsqlRepdef	Number of commands excluded from dynamic SQL by replication definition.
DSINoDsqlColumnCount	Number of commands excluded from dynamic SQL because the number of parameters exceeds 255.
DSINoDsqlMissingCols	Number of commands excluded from dynamic SQL because some columns are not available at DSI. This can be caused by minimal columns feature.
DSIEDsqlPrepared	Dynamic SQL statements prepared at target database by a Data Server Interface executor (DSI/E).
DSIEDsqlDealloc	Dynamic SQL statements deallocated at target database by a DSI/E.
DSIEDsqlExecuted	Dynamic SQL statements executed at target database by a DSI/E.
DSIEDsqlDeallocSchema	Dynamic SQL statements deallocated at replicate database by a DSI/E because of schema change.
DSIEDsqlDeallocExecFail	Dynamic SQL statements deallocated at replicate database.

See “rs\_helpcounter” in Chapter 6 “Adaptive Server Stored Procedures” in the *Replication Server Reference Manual* for commands to retrieve information about counters.

## Extended support for large-object (LOB) datatypes

Replication Server 15.1 supports the replication of Microsoft SQL Server 2005 datatypes varchar(max), nvarchar(max), and varbinary(max). These datatypes can each store up to 2,147,483,647 bytes of data.

Replication Server introduces LOB types as user-defined datatypes (UDDs) in the table-level replication environment. Replication Server also supports database-level replication for new LOB types. The new LOB types are directly mapped to text, unitext, and image datatypes.

The base type of UDDs is:

New LOB datatype	Base type
varchar(max)	text
nvarchar(max)	unitext
varbinary(max)	image

#### Limitations

The limitations of the new LOB datatypes are:

- You cannot define as a primary key a LOB column in the table replication definition.
- You cannot define as searchable a LOB column in the table replication definition or function replication definition.
- You cannot replicate stored procedures that include one of the new LOB datatypes as a parameter.
- You cannot use text pointers to manipulate the data of the new LOB datatypes.

In a mixed-version environment, the primary and replicate Replication Server must have a site version of 15.1 and an LTL version of 710.

For more information about the new LOB datatypes, see the *Replication Server Reference Manual*.

## Partial update of LOB datatypes

A partial-update transaction allows you to directly insert a character string or overwrite an existing character string of a table column without issuing a delete and replace commands.

Replication Server 15.1 supports replication of partial-update transaction to supported LOB datatypes.

To implement partial update, use the new `rs_updatetext` LTL command:

```
{distribute|_ds} command_tags {applied|_ap} 'table'.rs_updatetext
{partialupd|_pu} [{first|_fi}] [last] [{changed|_ch}] [with log]
[{withouttp|_wo}] [{offset|_os}=offset {deletelen|_dln}=deletelength]
[{textlen|_tl}=length] text_image_column
```

#### Limitations

Partial update:

- Does not support multiple character set conversion.

- Support is restricted to Microsoft SQL Server 2005.

For more information about partial update, see the *Replication Server Design Guide*.

## Extended *timestamp* support

A new datatype, timestamp has been added to Replication Server 15.1. timestamp is defined as varbinary(8) with a status bit indicator that differentiates it from varbinary. The timestamp datatype allows the replication of timestamp columns to replicate, standby, and MSA databases. You can also define timestamp as a primary key in a replication definition, and as a searchable column in a replication definition and a function replication definition.

The `send_timestamp_to_standby` configuration parameter has been added to support timestamp replication. When `send_timestamp_to_standby` is enabled and there are no replication definitions, timestamp columns are sent to the replicate database.

---

**Note** The replicate Adaptive Server must be version 15.0.2 or later to support timestamp in replication definition.

---

See the *Replication Server Reference Manual* for more information about the timestamp datatype. See the *Replication Server Administration Guide Volume 1* for information about replicating timestamp columns.

## New *opaque* datatype

The opaque datatype handles replication of datatypes that Replication Server does not support. RepAgent provides formatting data that can be directly applied in the target database.

The opaque datatype handles replication of datatypes that can store unspecified or inconsistent values, such as anydata datatype and the Microsoft SQL Server `sql_variant` datatype.

### Limitations

Limitations of the opaque datatypes include:

- You cannot use opaque datatypes in searchable columns and where clauses of replication definitions, subscriptions, and articles.
- You cannot use a map to clause with opaque datatypes.
- You cannot use dynamic SQL when an opaque datatype column or parameter exists in your replication definition.
- You cannot use the opaque datatype if your function string has a remote procedure call (RPC).
- You cannot apply character set conversion or byte-order conversion to opaque datatypes.

In a mixed-version environment, the primary and replicate Replication Server must have a site version of 15.1 and an LTL version of 710.

For more information about opaque datatypes, see the *Replication Server Reference Manual*.

## Dump transaction enhancement

The Log Transfer Language (LTL) dump subcommand and `rs_dumptran` function string have been enhanced to support replication of dump transaction commands with the `with standby_access` parameter.

## LTL *dump* subcommand enhancement

`standby`, `stdb`, and `status` parameters have been added to the dump subcommand syntax to support `with standby_access`:

```
{distribute|_ds} command_tags dump [ database |  
  {transaction | tran | _tr}[standby | stdb | status]]  
  database_name, 'dump_label' id
```



## Distributor status recording

A distributor (DIST) thread reads transactions from the inbound queue and writes replicated transactions into the outbound queue. A DIST thread is created when Replication Server connects to the primary database, and can be suspended or resumed manually, or through a Replication Server configuration. Resuming and suspending a DIST thread modifies the DIST status of the thread.

With Replication Server 15.1, you can now save the DIST status of a distributor thread in the RSSD. This allows the DIST thread to retain its status even after a Replication Server is shut down.

For more information about DIST status recording, see the *Replication Server Reference Manual*.

## Enhanced text update

Replication Server supports the replication of large objects such as text and image to non-ASE servers by passing a writetext command to DirectConnect Anywhere™, where it is converted to an update statement. The writetext command includes large-object pointers that an update statement uses to search and propagate the replicate database. Most data servers have their own unique implementation of updating large objects. Therefore, large-object replication to these servers becomes slow and inefficient, often requiring a full table scan of the replicate database for a single update.

Replication Server 15.1 provides an option to include primary keys with writetext commands sent to DirectConnect Anywhere. With the primary keys, DirectConnect Anywhere can create update statements that can efficiently search and replicate the replicate database.

Replication Server 15.1 introduces the Data Server Interface (DSI) configuration parameter `dsi_alt_writetext`. Use `dsi_alt_writetext` to instruct the Replication Server to include a text pointer or a set of primary keys with the writetext command.

---

**Note** You need ECDA 15.0 ESD #2 to use this feature.

---

For more information, see the *Replication Server Reference Manual*.

## Adaptive Server integer *identity* support

The Adaptive Server 15.0 allows you to use these datatypes as identity values:

- bigint
- int
- numeric
- smallint
- tinyint
- unsigned bigint
- unsigned int
- unsigned smallint

Replication Server 15.1 supports replication of the above datatypes. When you create a replication definition for a table that contains an identity column, specify *identity* as the datatype for the column.

## Stable Queue Manager performance enhancements

The Stable Queue Manager (SQM) performance has been enhanced to include:

- Stable queue caching
- Segment preallocation
- Support for direct I/O file access

### Stable queue caching

Replication Server uses a simple caching mechanism to optimize I/O. This mechanism reduces write latency and improves reader speed, since data can usually be read quickly from the cache.

A cache is made up of multiple pages and each page is made up of multiple adjoining blocks. A cache is allocated for each queue at start-up time. Changing the page size changes the size of I/O in the stable queue devices. When a page is full, the entire page is written in one single write operation.



Configuring stable  
queue cache  
parameters

In stable queue caching, the page pointer moves forward and rotates back at the end of the cache. SQM flushes the current page if the writer has filled the message queue and is blocked when waiting for messages. Only blocks with data are written to a disk when flushing a page that is not full.

Set the server-wide caching default value using:

```
configure replication server set sqm_cache_enable to
"on|off"
```

Enable or disable the caching for a queue and override the server-level setting using:

```
alter queue q_number, q_type, set sqm_cache_enable to
"on|off"
```

When `sqm_cache_enable` parameter is disabled, SQM module returns back to the earlier mechanism, which maintains a fixed 16K; one-block buffer.

Set the server-wide page size default value using:

```
configure replication server set sqm_page_size to
"num_of_blocks"
```

Set the page size for a specified queue using:

```
alter queue q_number, q_type, set sqm_page_size to
"num_of_blocks"
```

`num_of_blocks` specifies the number of 16K blocks in a page. Configuring the page size also sets the I/O size of Replication Server. For example, if you set the page size to 4, this instructs the Replication Server to write to stable queue in 64K chunks.

Set the server-wide cache size default value using:

```
configure replication server set sqm_cache_size to
"num_pages"
```

Set the cache size for a specified queue using:

```
alter queue q_number, q_type, set sqm_cache_size to
"num_pages"
```

`num_pages` specifies the number pages in the cache.

All SQM configuration commands are static, thus you must restart the server for these commands to take effect.

See the *Replication Server Reference Manual* for detailed information about these configuration parameters.

## Segment preallocation

Replication Server 15.1 preallocates segments in the background to reduce segment allocation latency. Segment allocation imposes significant latency to writer threads especially when the RSSD is on a remote Adaptive Server.

When a writer thread needs a new segment, it checks whether the preallocated segment is available, if it is not, the thread requests to allocate the segment. Once the writer thread gets the new segment, a preallocation request is made so that the segment is allocated in the background. By the time the writer thread needs a new segment, it is already available.

Enable or disable segment preallocation using:

```
configure replication server set sqm_seg_prealloc to
"on|off"
```

This command is static, which means you must restart the server for it to take effect. It supports only the server-level configuration.

## Support for direct I/O file access

For file system partitions, direct I/O reduces the I/O latency as compared to the synchronous I/O, DSYNC.

---

**Note** Direct I/O is supported only on Sun Solaris SPARC.

---

Adjust the stable queue cache properly. A proper cache size ensures that most read transactions are completed within the cache. Configure direct I/O using:

```
configure replication server set sqm_write_flush to
"dio"
```

This command enables direct /IO and is effective only when the stable queue is on the file system. The direct I/O method allows the Replication Server to read or write directly to the disk without the buffering the file system.

This command is static, which means you must restart the server for it to take effect.

# New Features in Replication Manager 15.1

This chapter describes the Replication Server 15.1 features supported by Replication Manager.

Topic	Page
Enhanced support for dynamic SQL	53
Enhanced support for function replication definitions	54
Support for large-object datatypes	54
Sybase Central 6.0	55
Support for opaque datatypes	55
Support for timestamp datatypes	56

## Enhanced support for dynamic SQL

With Replication Manager 15.1, you can enable dynamic SQL replication in table replication definitions in mixed-version replication environments where the Replication Server version is 15.0.1 or later.

The Replication Manager GUI has been modified. The General tab of the Create/Alter Replication Definition dialog box now includes a Replicate dynamic SQL option. This option is selected by default. If you do not require dynamic SQL replication, unselect the option. If you are working with a version of Replication Server for which Replication Manager does not support dynamic SQL, the Replicate dynamic SQL option is disabled.

---

**Note** If you modify the dynamic SQL replication property of a replication definition, the changes apply only to the modified replication definition. Other replication definitions for the same table remain unchanged.

---

See “Dynamic SQL for enhanced Replication Server performance” in the *Replication Server Administration Guide Volume 2* for detailed information about dynamic SQL replication in Replication Server.

## Enhanced support for function replication definitions

With Replication Manager 15.1, you can create multiple function replication definitions with names that differ from the corresponding primary function names. To support this feature, the following changes have been incorporated in the Replication Manager GUI:

- On the General tab of the Add New Function Replication Definition dialog box, after you create a function replication definition, the Add New Function Replication Definition icon is not replaced by the Function Replication Definitions folder. The Add New Function Replication Definition icon and the Function Replication Definitions folder coexist so that you can create multiple function replication definitions.
- On the General tab of the Add New Function Replication Definition dialog box, the Replication Definition text field is now available for you to create your own function replication definition name. You can create a function replication definition name that differs from the selected stored procedure in the Replication Definition field.
- The Replicate Stored Procedure option available on the General tab instead of the Advanced tab.

See the *Replication Server Administration Guide Volume 1* for detailed information about working with function replication.

## Support for large-object datatypes

With Replication Manager 15.1, you can manage large-object (LOB) datatypes that are defined in your replication environment. Replication Manager supports Microsoft SQL Server LOB datatypes of `varchar(max)`, `nvarchar(max)`, and `varbinary(max)`. However, if a column contains any of these LOB datatypes, the column cannot be a primary key, or marked as searchable.

The following changes have been made in the Replication Manager GUI:

- If you have LOB datatypes in your replication environment, the Replication Definition Datatype and Published Datatype lists in the Create Replication Definition dialog box display LOB.

- On the Columns tab in the Add New Table Replication Definition dialog, when you select a column that contains varchar(max), nvarchar(max), and varbinary(max) datatypes, the Primary Key and Searchable options are disabled.

See the *Replication Server Administration Guide Volume 1* for detailed information about working with LOB datatypes.

## Sybase Central 6.0

Replication Manager 15.1 runs in Sybase Central 6.0.

## Support for opaque datatypes

In Replication Manager 15.1, you can replicate opaque data. Opaque data can store unspecified or inconsistent values such as the anydata datatype of Oracle or the sql\_variant datatype of Microsoft SQL Server. This is how Replication Manager supports opaque data:

- In a mixed-version replication environment, you can replicate opaque data when the Replication Server is version 15.0.1 or later.
- The Replication Manager GUI has been modified. On the Columns tab of the Add New Table Replication Definition window, a new option “opaque” appears on the Replication Definition list in the Datatypes area.
- You cannot make columns with opaque data searchable.
- The opaque datatype is supported both in function and table replication definitions.

For detailed information about replication of supported datatypes, see the *Replication Server Reference Manual*.

## Support for timestamp datatypes

In Replication Manager 15.1, you can replicate timestamps in your data. This is how Replication Manager supports timestamp replication:

- In a mixed-version replication environment, you can replicate timestamps where the Replication Server version is 15.0.1 or later.
- The Replication Manager GUI has been modified. On the Columns tab of the Add New Table Replication Definition window, a new option “time stamp” appears on the Replication Definition list in the Datatypes area.
- Timestamp datatype is supported both in function and table replication definitions.

For detailed information about replication of supported datatypes, see the *Replication Server Reference Manual*.

# New Features in Replication Server Version 15.0.1

This chapter describes the new features introduced in Replication Server 15.0.1. Several enhancements have been made to enable more efficient configuration and tuning of Replication Server.

Topic	Page
Configuration and tuning enhancements	57
Monitor and counter enhancements	58
Dynamic SQL for enhanced Replication Server performance	60
Master database replication	62
rs_subcmp enhancement	63
Schema comparison	65
Manual data reconciliation	68

## Configuration and tuning enhancements

Replication Server 15.0.1 now supports dynamic configuration, which simplifies Replication Server customization and performance tuning by:

- Providing the ability to some of the Replication Server configuration parameters to handle dynamic modification of values.
- Introducing admin config, a Replication Command Language (RCL) that retrieves the server, connection, logical connection, and route parameters and their runtime values.

### Configuring dynamic parameters

Several Replication Server configuration parameters are changed to dynamic, allowing you to change their values using the configure replication server command. You no longer need to restart the Replication Server for the new values to take effect. Table 4-1 lists the dynamic configuration parameters.

**Table 4-1: Dynamic configuration parameters**

init_sqm_write_delay	init_sqm_write_max_delay
memory_limit	num_concurrent_subs
queue_dump_buffer_size	sqm_recover_segs
sqm_warning_thr_ind	sqm_warning_thr1
sqm_warning_thr2	sqt_max_cache_size
sqt_init_read_delay	sqt_max_read_delay
sts_cachesize	sts_full_cache_system_table_name

Use the new admin config command to retrieve the values of these parameters.

*admin config*

admin config lets you retrieve the values of the configuration parameters used to customize and tune the Replication Server. In Replication Server 15.0.1, these parameters are categorized as:

- Server parameters
- Connection parameters
- Logical connection parameters
- Route parameters

Syntax

The admin config syntax is:

```
admin config [,"connection" |,"logical_connection" |,"route" ] [,server
[,database]] [,configuration_name]
```

See Chapter 3, “Replication Server Commands” of the *Replication Server Reference Manual*, for detailed information in using admin config command.

## Monitor and counter enhancements

The process of monitoring the counters that provide performance information has been improved in Replication Server 15.0.1. Sampling and saving statistics to the RSSD now runs asynchronously, allowing you to execute other Replication Server commands or terminate the session while the sampling continues in the background.

You can check the progress of asynchronous commands to prevent overlaps. When a command is denied due to overlapping, you can cancel the commands in progress to execute another command.



Enhanced *admin stats* command      The *admin stats* command has been modified. The modified command syntax is:

```
admin { stats | statistics } [, sysmon | "all"
    | module_name [, inbound | outbound] [, display_name] ]
    [, server[, database] | instance_id]
    [, display |, save [, obs_interval] ]
    [, sample_period]
```

The behavior for the *save* parameter with a *sample\_period* has also changed. The *admin stats* returns the session immediately. Its original *num\_obs* option has been changed to *obs\_interval*, which can be a numeric value in seconds, or a quoted time format string hh:mm[:ss].

Example      To start sampling and saving statistics to the RSSD for one hour and thirty minutes at 20-second intervals, enter:

```
admin stats, "all", save, 20, "01:30:00"
```

*cancel* command      The new command syntax also supports the *cancel* option, which stops the currently running command.

```
admin { stats | statistics } , cancel
```

---

**Note** For multiple observation intervals (*obs\_interval*), the data that is saved at the time of cancellation is not deleted.

---

Replication Server checks the unfinished command to avoid corrupted sampling. If there is any command in progress, running either in the background or in other user sessions, the following commands cannot be executed:

- *admin stats* with any variation of *save*. For example:

```
admin stats, "all", save, 30
```

- *admin stats* with a sampling period. For example:

```
admin stats, "all", display, 120
```

- Commands that resets counters. For example:

```
admin stats, reset
```

When a command is denied, the following error message is returned, and *admin stats* with the *cancel* option is required to stop the unfinished command before executing another command.

```
Admin command failed. Existing command is currently in
progress
```

Commands that display current statistics of your session are allowed to be executed even if there is another command in progress. For example:

```
admin stats, sqm, inbound, display
```

Checking the status of asynchronous commands

The output of the existing `admin stats, status` command has been modified to include the status of the sampling command:

```
1> admin stats, status
2> go
Command in progress, sampling period 00:30:00, time
elapsed 00:02:32
```

```
Sybase Replication Server Statistics Configuration
=====
Configuration                Default                Current
-----
stats_sampling                off                   on
stats_show_zero_counters      off                   off
stats_reset_rssd              on                    on
```

Keeping previously saved data in RSSD

Replication Server 15.0.1 has a new configuration parameter, `stats_reset_rssd`, which allows you to optionally keep the old sampling data in RSSD by setting its value to off. By default, `stats_reset_rssd` is turned on.

```
configure replication server
set stats_reset_rssd to {"on", "off"}
```

See Chapter 3, “Replication Server Commands” of the *Replication Server Reference Manual*, for detailed information in using `admin stats` command.

## Dynamic SQL for enhanced Replication Server performance

Dynamic SQL in Replication Server enhances replication performance by allowing Replication Server Data Server Interface (DSI) to prepare dynamic SQL statements at the target user database and to execute them repeatedly. Instead of sending SQL language commands to the target database, only the literals are sent on each execution, thereby eliminating the overheads brought by SQL statement syntax checks and optimized query plan builds.

You can use dynamic SQL in a user database connection for a language command if:

- The command is insert, update, or delete.
- There are no text, image, unitext, or java columns in the command.
- There are no NULL values in the where clause of an update or delete command.
- There are no more than 255 parameters in the command:
  - insert commands can have no more than 255 columns.
  - update commands can have no more than 255 columns in the set clause and where clause combined.
  - delete commands can have no more than 255 columns in the where clause.
- The command does not use user-defined function strings.

Setting up the configuration parameters to use dynamic SQL

Configure dynamic SQL at a server or a connection level by issuing the following commands:

```
configure replication server
set { dynamic_sql |
      dynamic_sql_cache_size |
      dynamic_sql_cache_management }
to value

alter connection to server.db
set { dynamic_sql |
      dynamic_sql_cache_size |
      dynamic_sql_cache_management }
to value
```

The server-level configurations provide the default values for the connections created or started in the future. For database level configurations:

- `dynamic_sql` – turns dynamic SQL on or off for a connection. Other dynamic SQL related configuration parameters take effect only if this parameter is set to on.
- `dynamic_sql_cache_size` – tells the Replication Server how many database objects may use the dynamic SQL for a connection. This parameter is provided to limit the resource demand on the data server.

- `dynamic_sql_cache_management` – manages the dynamic SQL cache for a connection. Once the dynamic SQL statements reaches `dynamic_sql_cache_size` for a connection, it either stops allocating new dynamic SQL statements if the value is fixed, or keeps the most recently used statements and deallocates the rest to make room for the new statements if the value is mru.

#### Limitations

Dynamic SQL has these limitations:

- If a table is replicated to a standby or MSA connection using an internal replication definition and dynamic SQL is enabled for the connection, any new replication definition for the table should define the column order consistent with the column order in the primary database. Otherwise, the existing prepared statements may be invalidated, and may require the standby or MSA connection to be restarted.
- Dynamic SQL requires ASE or DirectConnect™ 12.6.1 ESD#2 for UDB as target database.

## Master database replication

The Adaptive Server RepAgent thread now supports master database replication in Adaptive Server 15.0, ESD #2 and later. The master database can be replicated using warm standby with any version of Replication Server that supports warm standby, including Replication Server version 12.0 and later.

In addition, the master database can be replicated using multisite availability (MSA) with any version of Replication Server that supports MSA.

Replication of the master database is limited to DDL commands and system procedures used to manage logins and roles.

The supported DDL commands and system procedures are:

- `alter role`
- `create role`
- `drop role`
- `grant role`
- `revoke role`
- `sp_addlogin`

- `sp_displaylevel`
- `sp_droplogin`
- `sp_locklogin`
- `sp_modifylogin`
- `sp_password`
- `sp_passwordpolicy`
- `sp_role`

## ***rs\_subcmp*** enhancement

In Replication Server 15.0.1, `rs_subcmp` uses a hash algorithm to improve its performance. Hash algorithm compresses the data in primary and replicated tables. The compressed data is then fetched by `rs_subcmp`.

Instead of taking the entire row of data during comparison between the primary table and replicated table, `rs_subcmp` now transfers only the compressed data of each data row from the primary or replicated tables, and then verifies or reconciles inconsistencies between them.

With the hash algorithm, the amount of data to be transferred from the primary and replicated tables to `rs_subcmp` is reduced, decreasing the total running time of `rs_subcmp`.

New command line  
and configuration file  
parameters

Table 4-2 and Table 4-3 describe parameters that are new to `rs_subcmp` that support the use of the hash algorithm.

**Table 4-2: New command line parameters for hash algorithm**

Parameter	Description
-h	Perform faster comparison.
-H	Normalize data when using hash algorithm across different platforms or character sets.

**Table 4-3: New configuration file parameters for hash algorithm**

Parameter	Description
FASTCMP	The same as command line parameter -h. The syntax is: FASTCMP = Y N Values are: Y – perform fast comparison using compressed data. N (default) – perform normal comparison as before.
HASH_OPTION	The same as command line parameter -H. The syntax is: HASH_OPTION = <i>lsb</i>   <i>msb</i>   <i>unicode</i>   <i>unicode_lsb</i>   <i>unicode_msb</i> If this parameter does not exist in the configuration file, by default, rs_subcmp uses the native byte order and character set.

**Limitation**

The enhanced rs\_subcmp requires ASE 15.0.2 or later and cannot handle case sensitive comparison. It also cannot handle text, unitext, or image datatypes and does not allow you to specify the precision for the float datatype (maximum precision is used).

---

**Note** Sybase suggests to set the ASE parameter default data cache to 128M or higher to get a better comparison performance.

---

**Example**

**Example 1** To instruct rs\_subcmp to get the compressed data and reconcile any inconsistencies between the primary and replicated tables, enter:

```
rs_subcmp [-h] [...]
```

**Example 2** To instruct rs\_subcmp to normalize data when using hash algorithm, enter the following command, where options can be *lsb*, *msb*, *unicode*, *unicode\_lsb*, and *unicode\_msb*:

```
rs_subcmp [-h -H options] [...]
```

Table 4-4 describes the string options to use with rs\_subcmp.

**Table 4-4: rs\_subcmp string options**

Option	Description
<i>lsb</i>	All byte-order dependent data is normalized to lsb-first (little-endian) before data compression.
<i>msb</i>	All byte-order dependent data is normalized to msb-first (big-endian) byte order before data compression.
<i>unicode</i>	Character data is normalized to unicode (UTF-16) before data compression.

---

**Note** UTF-16 string looks very much like an array of short integers and is byte-order dependent, thus Sybase suggests that you use *lsb* and *msb* in conjunction with *unicode* for platform independence. For example, use *unicode\_lsb* or *unicode\_msb*.

---

See Chapter 7, “Executable Programs” of the *Replication Server Reference Manual*, for more information on using these parameters with *rs\_subcmp*.

## Schema comparison

In Replication Server 15.0.1, the *rs\_subcmp* function has been expanded to include schema comparison between tables and databases. This is useful in comparing schema between two databases that may have the same data but different schemas.

Table 4-5 and Table 4-6 enumerate the schema types and subtypes supported by *rs\_subcmp*.

**Table 4-5: Schema types supported by rs\_subcmp**

Type	Description
A	All aliases in the database.
D	All defaults in the database.
E	All user-defined datatypes in the database.
G	All groups in the database.
R	All rules in the database.
T	All user tables in the database. Includes table elements such as indexes, keys, constraints, and triggers.
U	All users in the database.
V	All views in the database.
P	All procedures in the database.

**Table 4-6: Schema sub-types supported by rs\_subcmp**

Type	Description
c	Constraint
d	Bind default
f	Foreign key
g	Grant
i	Index
m	Procedure mode
p	Primary key
r	Bind rule
t	Trigger

The command line and configuration file parameter set has been expanded to support the rs\_subcmp schema comparison. Table 4-7 describes the new rs\_subcmp command line parameters.



**Table 4-7: Command line parameters for schema comparison**

Parameter name	Description	Valid values
-x	comparison flag	0 (default) – data comparison. 1 – database schema comparison. 2 – table schema comparison.
-X	filter flag	If the value starts with “+”, only the schema types are selected for comparison, and subtypes are ignored. Otherwise, the schema types and subtypes are both unselected and not used for comparison.
-l	interface file	Interface file location.

Table 4-8 describes the new configuration file parameters.

**Table 4-8: Configuration file parameters for schema comparison**

Item name	Description	Valid values
SCHEMAFLAG	comparison flag	0 (default) – data comparison. 1 – database schema comparison. 2 – table schema comparison.
FILTER	filter flag	If the value starts with “+”, only the schema types are selected for comparison, and sub-types are ignored. Otherwise, the schema types and subtypes are both unselected and not used for comparison.
IFILE	interface file	Interface file location.

## Examples

**Example 1** Compares all schemas between two databases using the *config.cfg* file:

```
rs_subcmp -f config.cfg
```

The configuration file contains the following:

```
PDS = PASE
RDS = R2ASE
PDB = pubs2
PTABLE = authors
RTABLE = authors
PUSER = sa
RUSER = sa
PPWD =
RPWD =
SCHEMAFLAG = 1
```

**Example 2** Compares schema between two databases without a configuration file:

```
rs_subcmp -Spds -srds -Dpdb -drdb -Usa -usa -Psa_pwd
          -psa_pwd -x1
```

**Example 3** Compares schema of two databases excluding index, trigger, and datatype:

```
rs_subcmp -Spds -srds -Dpdb -drdb -Usa -usa -Psa_pwd
          -psa_pwd -x1 -XitD
```

**Example 4** Compares all table schemas and user schemas:

```
rs_subcmp -Spds -srds -Dpdb -drdb -Usa -usa -Psa_pwd
          -psa_pwd -x1 -X+TU
```

Report and reconciliation files

A report file which details the comparison result between two tables or two databases is created after every schema comparison. The report file is named *reportPROCID.txt*. If inconsistencies exist, *rs\_subcmp* creates a reconciliation script named *reconcilePROCID.sql*. The report file and the reconciliation script are saved in the same directory where you issued the *rs\_subcmp*.

## Manual data reconciliation

The enhanced *rs\_subcmp* command now supports manual reconciliation of inconsistent data through the creation of a reconciliation file. This allows you to verify the reconciliation statements before execution.

New *rs\_subcmp* parameters

Replication Server 15.0.1 introduces two new parameters for this feature. Table 4-9 describes these parameters.

**Table 4-9: New parameters for *rs\_subcmp* manual reconciliation**

Parameter	Description
-g	A command line parameter indicating to <i>rs_subcmp</i> to create a reconciliation file.
RECONCILE_FILE	A configuration file parameter. Values are: Y – create a reconciliation file. N (default) – do not create a reconciliation file.

Limitation

The reconciliation file's SQL statements cannot contain text, unitext, or image, data.

See Chapter 7, “Executable Programs” of the *Replication Server Reference Manual*, for more information on using these parameters with `rs_subcmp`.



# New Features in Replication Manager 15.0.1

This chapter describes the Replication Server 15.0.1 features supported by Replication Manager.

Topic	Page
Support for dynamic configuration	71
Support for heterogeneous data servers	72

## Support for dynamic configuration

The Replication Manager (RM) plug-in has been modified to support the dynamic configuration feature of Replication Server 15.0.1. Versions of Replication Server earlier than 15.0.1 do not support dynamic configuration.

This feature allows the RM to dynamically retrieve all parameters that can be configured for the Replication Server, including the parameters for connections and routes.

Enhanced properties dialog box

To support dynamic configuration, the Parameters tab of the Properties dialog box in RM has been enhanced to accurately display the parameter status information returned by the Replication Server. It includes:

- The Status field replaces the Restart Required field in RM 15.0. Its associated text field now has a wider range of information. The remaining fields and functions of the properties dialog box remain the same.

- The parameter information in the Run Value column may now contain the value “<server default>”. This may be used for parameters in the connections, logical connections, and routes to indicate that a parameter at that component level has not been set, and the Replication Server default value will be used. These types of parameters can have a default setting at the server level, or can have the parameter set for an individual connection, logical connection, or route.
- There may be some parameters for which the legal values are considered nonstandard so the Legal values field may be left blank. This is usually due to a parameter that may contain a textual setting and a numerical setting. In these cases, the Explanation field may contain further information regarding legal values for the parameter.

## Support for heterogeneous data servers

The RM plug-in now supports heterogeneous data servers, which provides you the ability to create connections, replication definitions, and subscriptions for non-Sybase data servers. Non-Sybase data servers include Oracle, Microsoft SQL Server, and IBM UDB.

---

**Note** This version of RM does not support IBM DB2 data servers or Replication Agents, although Sybase provides a replication solution for IBM DB2 on mainframe.

---

## Non-Sybase data servers support

To provide support for heterogeneous data servers, the RM displays the databases, tables, and stored procedures for non-Sybase data servers in a two-tier environment, the same as in the ASE data servers.

See Chapter 3, “Managing Replication Server with Sybase Central” of the *Replication Server Administration Guide Volume 1*, for more information about two- and three-tier management solutions.

The RM allows you to create, update, and delete both primary and replicate connections. It also allows you to create, update, and delete databases, tables, and function replication definitions and subscriptions. The RM user interface for non-Sybase data servers is identical to the current ASE server interface.

---

**Note** RM supports table, and function replication definition, and subscription, but does not support database replication definition, and subscription for non-Sybase data servers. RM also supports all connections for non-Sybase data servers except for logical connection.

---

For non-Sybase data servers, the RM uses Replication Agents and DirectConnect to communicate with the data servers and to act as an interface for the RM. The RM uses an interface server to retrieve the metadata from the data servers. The Replication Agent serves as the interface server for the primary data servers, while DirectConnect serves as the interface server for the replicate data servers.

The interface server is not necessarily the server used to replicate data in the environment. For example, when defining an Oracle server that will be both a primary and a replicate data server, you only identify a Replication Agent as the interface server. This Replication Agent can be used to extract transactions from the Oracle server but a DirectConnect may also be needed to replicate data to the Oracle server. Also, if you are replicating from a Microsoft SQL Server, you can identify a Replication Agent as the interface server, but may require several Replication Agents to replicate data from several databases.

## Replication Agents and Mirror Replication Agents support

In version 15.0.1, the RM provides the ability to initialize the Sybase Replication Agent and displays its performance statistics in a two-tier environment. Additionally, the connection associated with the Replication Agent is displayed in the tree view hierarchy upon clicking the Replication Agent.

The RM also provides a feature that tests the connection between the primary data server and the Replication Agent, and the connection between the Replication Agent and the Replication Server.

The context menu for the Replication Agent now provides commands to test connections, display statistics, and initialize the Replication Agent. In the Sybase Central object tree, right-click the Replication Agent icon to display the context menu.

## Replication Manager plug-in enhancements

To provide heterogeneous support, the following items have been modified in the RM:

- The Add Server wizard to add non-Sybase data servers to a two-tier environment.
- The Add Connection wizard to create connections to non-Sybase databases.
- The Replication Agent context menu provides commands to initialize the agent, displays performance statistics, and tests network connections.
- The tree view hierarchy displays the connection associated with the Replication Agent under the actual agent.
- The Create Replication Definition dialog to create replication definitions for tables in a non-Sybase database. You can use user-defined datatypes to define non-Sybase datatypes.



# New Features in Replication Server Version 15.0

This chapter describes the new features introduced with Replication Server 15.0.

Topic	Page
Support for longer identifiers	75
New datatype: bigint	76
New unsigned integer datatypes	77
New Unicode datatype: unitext	78
Replicating computed columns	79
Replicating encrypted columns	80
Replicating partitioned tables	80
Larger disk partitions	82
Larger text and image size	83
Embedded Replication Server System Database (ERSSD) enhancement	83
New password encryption algorithm	84
Mixed-version enhanced support	85
New interface for monitors and counters	86
Support for isolation levels	87
Bidirectional replication support for DDL in MSA	89
Batching of commands for non-ASE servers	89
SySAM license management	90

## Support for longer identifiers

The limit of 30 bytes for selected database and replication object names (identifiers) is extended to 255 bytes for Replication Server version 15.0. Longer identifiers are supported for these objects:

- Tables.

- Columns.
- Stored procedures.
- Parameters – for Replication Server functions and Adaptive Server stored procedures.
- Functions – for function replication definitions.
- Function strings.
- Replication definitions – including table replication definitions, function replication definitions, and database replication definitions.
- Publications.
- Articles.

All other database and replication object identifiers, login names and passwords, transaction names, subscription names, database names, and server names retain the existing 30-byte limit.

Parameter names for Replication Server functions and Adaptive Server stored procedures are the only identifiers that can begin with the @ character.

- Replication Server function parameter names can be as many as 256 bytes including the @ character.
- Adaptive Server stored procedure parameter names can be as many as 255 bytes including the @ character.

---

**Note** rs\_subcmp supports long identifiers for both table and column names.

---

#### Limitation

The create function, alter function, and drop function commands do not support long identifiers. The name of the function and the parameters of these commands cannot exceed 30 bytes.

## New datatype: *bigint*

Replication Server 15.0 adds support for the *bigint* datatype. *bigint* is a fixed-width, 8-byte datatype. Like *int*, *smallint*, and *tinyint*, *bigint* is a signed integer datatype. *bigint* can hold whole numbers between  $-2^{63}$  and  $+(2^{63} - 1)$ .

Table 6-1 describes the range and storage size for all integer datatypes.

**Table 6-1: Signed integer datatypes**

Integer datatypes	Range	Storage (bytes)
bigint	Whole numbers between $-2^{63}$ and $+(2^{63} - 1)$ , inclusive ( $-9,233,372,036,854,775,808$ and $+9,233,372,036,854,775,807$ )	8
int	Whole numbers between $-2^{31}$ and $+(2^{31} - 1)$ , inclusive ( $-2,147,483,648$ , and $+2,147,483,647$ )	4
smallint	Whole numbers between $-2^{15}$ and $+(2^{15} - 1)$ , inclusive ( $-32,768$ and $+32,767$ )	2
tinyint	Positive whole numbers between 0 and 255, inclusive	1

**Mixed-version issues**

To fully support bigint, the primary and replicate Replication Server must have a site version of 15.0, and the LTL version must be 700. If the LTL version is less than 700 at connect-source time, RepAgent converts bigint columns to numeric.

---

**Note** rs\_subcmp supports the new bigint datatype.

---

## New unsigned integer datatypes

Replication Server version 15.0 supports four new unsigned integer datatypes. You can use these unsigned datatypes in the same way as their signed equivalents:

- unsigned tinyint – tinyint
- unsigned smallint – smallint
- unsigned int – int
- unsigned bigint – bigint

Signed integers are whole positive or negative numbers. Unsigned integers are only whole positive numbers. The storage sizes of the signed and unsigned integers datatypes are the same. Table 6-2 shows the range of positive numbers supported for each unsigned datatype.

**Table 6-2: Unsigned integer datatypes**

Integer datatypes	Range	Storage (bytes)
unsigned bigint	Whole numbers between 0 and 18,446,744, 073, 709,551,615, inclusive	8
unsigned int	Whole numbers between 0 and 4,294,967,295, inclusive	4

Integer datatypes	Range	Storage (bytes)
unsigned smallint	Whole numbers between 0 and 65535, inclusive	2
unsigned tinyint	Positive whole numbers between 0 and 255, inclusive	1

Mixed-version issues To fully support the unsigned integer datatypes, the primary and replicate Replication Server must have a site version of 15.0, and the LTL version must be 700.

If the LTL version is less than 700 at connect-source time, RepAgent makes these datatype conversions:

- unsigned bigint → numeric
- unsigned int → numeric
- unsigned smallint → int
- unsigned tinyint → tinyint

Replication definitions created with unsigned integer datatypes are not sent to Replication Servers version 12.5.x and earlier. If a replication definition is subscribed by Replication Server version 12.5.x, it cannot be altered to add unsigned integer columns.

---

**Note** rs\_subcmp supports the new unsigned integer datatypes.

---

## New Unicode datatype: *unitext*

Replication Server version 15.0 adds support for the *unitext* datatype. *unitext* is a variable-width, nullable Unicode datatype. Although independent of the text datatype, *unitext* mirrors its behavior, and can be used wherever text is used.

Like the data in the other Unicode datatypes—*unichar* and *univarchar*—*unitext* data is encoded in UTF-16, which is essentially a 2-byte, fixed-width encoding of Unicode. *unitext* can hold as many as 1,073,741,823 Unicode characters, or the equivalent of 2,147,483,647 bytes. *unitext* has no connection to the default character set ID or the default sort order.

The main advantage of Unicode datatypes is efficiency. The UTF-16 character types are approximately 33% more space efficient than UTF-8 for Asian characters.

Mixed-version issues To fully support unitext, the primary and replicate Replication Server must have a site version of 15.0, the route version must be 15.0, and the LTL version must be 700. If the LTL version is less than 700 at connect-source time, RepAgent converts unitext columns to image.

---

**Note** rs\_subcmp supports the new unitext datatype.

---

## Replicating computed columns

Computed columns allow you to create an expression and place the result of the expression in a table column. A computed column is:

- **Materialized** – when its value is computed for each insert or update. Materialized computed columns are stored in the same way as regular columns.
- **Virtual** – when its value is computed only when referenced in a query. Virtual computed columns are not stored in the table or index page.

A computed column expression is:

- **Deterministic** – when its value is the same each time it is evaluated.
- **Nondeterministic** – when its value may be different each time it is evaluated (for example, a date stamp).

Replication Server replicates materialized computed columns in DML statements in the same way it replicates other columns; it does not replicate virtual computed columns.

The replication of computed columns is supported by function strings. In Replication Server version 15.0, the class-level function string `rs_set_dml_on_computed` is applied at the replicate database DSI when a connection is established. It issues `set dml_on_computed "on"` after the use database statement. If the replicate Adaptive Server is version 12.5.x or earlier, the command is ignored.

Since Replication Server does not distinguish between computed and regular columns, there are no changes to the syntax for creating or altering replication definitions.

When creating or altering replication definitions for tables containing:

- Deterministic columns – you can choose whether to include those columns in the replication definition. Since deterministic columns always realize the same value, you can create the replication definition without them and allow each replicated insert and update to compute values at the replicate database.
- Nondeterministic columns – you must include nondeterministic computed columns in the replication definition to ensure that the primary and replicate databases remain synchronized.

rs\_subcmp support for  
computed columns

rs\_subcmp supports comparison and reconciliation of materialized computed columns; it does not support virtual columns.

If the supported Adaptive Server provides the set `dml_on_computed` “on” command, `rs_subcmp` inserts and updates deterministic and nondeterministic materialized columns as regular columns.

## Replicating encrypted columns

Replication Server 15.0 supports replication of encrypted columns in Adaptive Server 15.0. Similar to Adaptive Server, Replication Server does not support encrypted columns that contain text and image data.

Replication Server replicates encrypted data as well as the encryption keys. For more information about this feature, see the *New Features Adaptive Server Enterprise 15.0 with Encrypted Columns*.

---

**Note** `rs_subcmp` supports replication of encrypted columns in Adaptive Server.

---

## Replicating partitioned tables

In Replication Server 15.0, partitioned tables introduced in Adaptive Server® Enterprise 15.0 are replicated in a way that is similar to nonpartitioned tables. The `rs_truncate` system function in LTL, and the `rs_truncate` function and function string in Replication Server have been extended to support partitioned tables.

rs\_truncate changes in LTL

In LTL, the rs\_truncate function has been extended to include partition names, as shown in the following syntax:

```
distribute command_tags applied [owner=owner_name]
table.rs_truncate [partition_name,
[partition_name]...] yielding
```

You must assign a *partition\_name* for each partition specified in the truncate table partition command.

rs\_truncate function changes in Replication Server

In Replication Server, the rs\_truncate function has been extended to accept parameters to support a new truncate table partition command, shown as follows:

```
truncate table table_name partition partition_name
```

When a truncate partition command is issued, the RepAgent sends the following LTL:

```
applied [owner=owner_name] table_name.rs_truncate,
partition_name _yd
```

rs\_truncate function string changes in Replication Server

The partition names are passed as parameters to the rs\_truncate function. The rs\_truncate function string accepts position-based function-string parameters. The following is a position-based variable that specifies the parameter position in the function, in the LTL command:

```
?n!param?
```

A sample function string for rs\_truncate with the position-based variable is as follows:

```
truncate table publishers partition ?1!param?,
?2!param?
```

Examples

**Example 1** To replicate truncate table partition as a delete command, alter the function string in the following way:

```
alter function string publisher.rs_truncate
for rs_sqlserver_function_class
output language
'begin transaction
if (?1!param? = '') /* NO parameter */
delete publisher
if (?1!param? = 'A')
delete publishers where c1 < 1000
if (?1!param? = 'B')
delete publishers where c1 >= 1000
commit transaction'
```

**Example 2** : To not to truncate table partitions at the replicate server, alter the function string to do nothing if there is a parameter, in the following way:

```
alter function string publisher.rs_truncate
for rs_sqlserver_function_class
output language
'if (?!param? = '') delete publisher'
```

Mixed-version issues

For the RepAgent to send an `rs_truncate` applied subcommand with parameters, the site version must be 15.0 and the LTL version must be 700. If the LTL version is below 700, RepAgent skips the `rs_truncate` portion of the distribute command.

To replicate the `rs_truncate` function with a parameter to the replicate Replication Server, the route version must be 1500. If the route version is lower than 1500, the `rs_truncate` command with the parameter is skipped.

## Larger disk partitions

Replication Server version 15.0 extends the maximum disk partition size from 2GB to 1TB. The new limit is applicable after the Replication Server site version is updated to 15.0.

Two new RCL commands support larger disk partitions:

- `create partition` – makes a partition available to Replication Server. This command replaces the existing `add partition` command. `add partition` is still supported, for backward compatibility with earlier versions. The syntax and usage of the two commands are identical. The command name has been changed to be consistent with other Replication Server command names.
- `alter partition` – increases the size of a partition. The syntax is:

```
alter partition logical_partition_name [ expand [ size = size ] ]
```

For example, to increase the size of the logical partition `p2` by 50MB, enter:

```
alter partition p2 expand size = 50
```



## Larger text and image size

In Replication Server 15.0, the `rs_subcmp` utility extends the maximum text and image size from 32KB to 2GB. You can specify the new value as length in kilobytes, doing either of:

- Using the `-L text_image_length_in_kilobytes` parameter to set the new text and image length in kilobytes. For example, to set the new length to 64KB, enter:

```
rs_subcmp -L 64 -f subcmp.cfg
```

- Providing the new value for text and image length to the parameter `TXT_IMG_LEN` in the configuration file. Start `rs_subcmp` using the modified configuration file:

```
rs_subcmp -f subcmp.cfg
```

*subcmp.cfg* file includes the following configuration parameter and its value:

```
TXT_IMG_LEN = 64
```

---

**Note** If the value is specified both in the command line and the configuration file, the value in the command line overwrites the value in the configuration file.

---

## Embedded Replication Server System Database (ERSSD) enhancement

ERSSDs were introduced in Replication Server version 12.6. For more information, see Chapter 9, “New Features in Replication Server Version 12.6.”

You can create a route from a Replication Server with an ERSSD, as long as both the source and the destination servers are version 15.0 or later.

To create a route from Replication Server with an ERSSD, use the `create route` command. Verify that the Replication Agent name is in the Replication Server interfaces file; an ERSSD Replication Agent is started as an open server during `create route`. If the Replication Agent name does not appear in the interfaces file, the command fails.

The default ERSSD Replication Agent name is *erssd\_name\_ra*. To replace the default name with that of your Replication Agent server, enter:

```
configure replication server
set erssd_ra to <value>
```

---

**Note** Sybase provides ERSSD in SQL Anywhere (SA) as an option, as well as continues to support the traditional RSSD in Adaptive Server Enterprise.

---

## New password encryption algorithm

Replication Server 15.0 uses the FIPS-certified Advanced Encryption Standard (AES) algorithm to encrypt new Replication Server user passwords. The AES algorithm uses the 128-bit encryption key and can be obtained from the Certicom Security Builder library.

Migrating existing  
encrypted passwords

Use the information in Table 6-3 to migrate existing encrypted passwords in the Replication Server configuration file, and the *rs\_users* and *rs\_maintusers* tables.

**Table 6-3: Commands to encrypt passwords in new algorithm**

To	Command
Migrate existing user passwords to the new algorithm	<pre>alter user user set password password</pre> <p>where:</p> <ul style="list-style-type: none"> <li>• <i>user</i> is the login name of the existing user.</li> <li>• <i>password</i> is the existing password you want to encrypt using the new algorithm.</li> </ul>
Migrate existing database maintenance user passwords to the new algorithm	<pre>alter connection to data_server.database set password to password</pre> <p>where <i>password</i> is the existing password you want to encrypt using the new algorithm.</p>
Migrate existing route user passwords to the new algorithm	<pre>alter route to dest_replication_server set password to passwd</pre> <p>where:</p> <ul style="list-style-type: none"> <li>• <i>dest_replication_server</i> is the name of the kdestination Replication Server.</li> <li>• <i>passwd</i> is the existing password you want to encrypt using the new algorithm.</li> </ul>
Migrate existing user passwords in the configuration file to the new algorithm	Use <code>rs_init</code> to encrypt the passwords using the new algorithm.

**Mixed-version issues**

To fully support the new password encryption algorithm, both the Replication Server and the `rs_init` utility must have a site version of 15.0. If the site version is lower than 15.0, an error message displays and encryption is disabled.

## Mixed-version enhanced support

In mixed-version environments, interaction between Replication Servers of different versions is restricted to the capabilities of the oldest version. Information associated with new features may not be available to Replication Servers of earlier versions.

Features in the new version, when sent to downstream Replication Servers of earlier versions, can break the multisite availability (MSA) replication and must be filtered out. In Replication Server 15.0, mixed-version support has been enhanced to provide a mechanism in which the later-version feature data is filtered out by the primary Replication Server before sending it out to the downstream Replication Servers of earlier versions.

New configuration parameter

To enable the primary Replication Server to block certain new feature commands not supported by earlier versions of Replication Server, a new Replication Server configuration parameter, `dist_stop_unsupported_cmd`, is available for all server and connection-level commands.

Replication Server commands that include the new configuration parameter are described in Table 6-4.

**Table 6-4: Modified commands for Mixed-version support**

Replication Server command	Syntax
<code>configure replication server</code>	<code>configure replication server set dist_stop_unsupported_cmd to [on   off]</code>
<code>alter connection</code>	<code>alter connection srv.db set dist_stop_unsupported_cmd to [on   off]</code>
<code>alter logical connection</code>	<code>alter logical connection lsrv.ldb set dist_stop_unsupported_cmd to [on   off]</code>

By default, `dist_stop_unsupported_cmd` is set to off. When this parameter is set to on, DIST suspends itself if a command cannot be sent to some destination Replication Servers. You have to resume the DIST by skipping the entire transaction or by resetting this parameter to off.

When the parameter is set to off, the distributor (DIST) thread skips the newer commands to earlier version Replication Servers.

## New interface for monitors and counters

Replication Server 15.0 provides a new, simpler interface for monitoring the counters that provide performance information. By default, all counters are inactive until you turn them on using either:

- The `admin stats` (admin statistics) command, which activates counters for a specific time period, or
- The Replication Server configuration parameter `stats_sampling`, which is a toggle that activates or deactivates counters.

Using `admin stats`, you can specify which counter statistics to report, whether to display those statistics on the screen or save them to the RSSD, and how many seconds to collect the statistics. `admin stats` lets you specify statistics for:

- Individual counters
- Individual modules
- The `sysmon` counters, which are a set of counters Sybase has identified as those most valuable for monitoring performance
- All counters

Viewing statistics on screen provides a point-in-time benchmark. Saving statistics to the RSSD lets you accumulate data, so you can see changes in statistics over time, and perform averages and other calculated values. You can specify a sampling period and the number of observations during that sampling period.

When you turn on counter activity using `stats_sampling`, the counters stay active until you turn them off.

New `admin stats` options are:

- `admin stats, backlog` – reports the current backlog in the inbound and outbound stable queues.
- `admin stats, { tps | cps | bps }` – reports throughput in transactions per second, commands per second, or bytes per second.
- `admin stats, status` – reports configuration information for counters.

## Support for isolation levels

Isolation levels let you control the degree to which data can be accessed by other users during a transaction. With version 15.0, Replication Server decouples isolation levels and serialization methods for the replicate data server, and enables all isolation levels for replicate data servers that Replication Server supports. In versions earlier than 15.0, Replication Server supported only isolation level 3.

Through the use of custom function strings, Replication Server supports all isolation levels the replicate data servers may use. Support is not limited to the ANSI standard only.

Each isolation level specifies the types of actions that are not permitted while concurrent transactions are processing. Higher levels include the restrictions imposed by lower levels.

You can set the isolation level with the database configuration parameter `dsi_isolation_level`. The ANSI standard levels supported by Adaptive Server are:

- 0 – ensures that data written by one transaction represents the actual data.
- 1 – prevents dirty reads and ensures that data written by one transaction represents the actual data.
- 2 – prevents nonrepeatable reads and dirty reads, and ensures that data written by one transaction represents the actual data.
- 3 – prevents phantom rows, nonrepeatable reads, and dirty reads, and ensures that data written by one transaction represents the actual data.

The default value is the current transaction isolation level for the target data server.

For example, to select isolation level 2 for the Replication Server connection to the TOKYO\_DS data server and pubs2 database, enter:

```
alter connection to TOKYO_DS.pubs2
set dsi_isolation_level to '2'
```

In versions earlier than 15.0, you set isolation level 3 and the serialization method at the same time:

```
alter connection to TOKYO_DS.pubs2
set dsi_serialization_method to 'isolation_level_3'
```

With version 15.0, the equivalent is:

```
alter connection to TOKYO_DS.pubs2
set dsi_serialization_method to 'wait_for_start'
```

```
alter connection to TOKYO_DS.pubs2
set dsi_isolation_level to '3'
```

## Bidirectional replication support for DDL in MSA

You can configure multisite availability (MSA) to set up a two-way replication of data definition language (DDL) transactions between two Adaptive Server databases.

In version 15.0, bidirectional DDL replication support in MSA environments is supported only for non-warm standby databases.

Replication Server 15.0 supports this bidirectional replication using a new configuration parameter `dsi_replication_ddl`. When `dsi_replication_ddl` is set to on, DSI sends set replication off to the replicate database, which instructs it to mark the succeeding DDL transactions available in the system log not to be replicated. Therefore, these DDL transactions are not replicated back to the original database, which enables the DDL transactions replication in bidirectional MSA replication environment.

### ❖ Setting up bidirectional replication

- 1 Create a bidirectional MSA replication environment. For steps, see Chapter 12, “Managing Replicated Objects Using MultiSite Availability,” in the *Replication Server Administration Guide Volume 1*.
- 2 Grant “set session authorization” privilege to a maintenance user on the destination database, as shown in the following example:

```
grant set session authorization to maint_user
```

- 3 In the destination database, set `dsi_replication_ddl` to on to enable bidirectional DDL replication, as shown in the following example:

```
alter connection to dataserver.database set dsi_replication_ddl on
```

- 4 Replicate DDL transactions.

## Batching of commands for non-ASE servers

Replication Server 15.0 allows you to batch commands for non-ASE database servers. By batching commands, you may be able to achieve improved performance in Replication Server.

New function strings	Support for command batching to non-ASE servers is achieved through the use of two function strings, <code>rs_batch_start</code> and <code>rs_batch_end</code> . These function strings store the SQL translation needed for marking the beginning and end of command batches. Use of these function strings is not necessary for Adaptive Server Enterprise or any other data server where the function strings <code>rs_begin</code> and <code>rs_commit</code> already support the needed functionality.
New DSI connection parameter	A DSI connection parameter, <code>use_batch_markers</code> , is used to control the processing of the two function strings, <code>rs_batch_start</code> and <code>rs_batch_end</code> . You can set <code>use_batch_markers</code> with <code>alter connection</code> and <code>configure connection</code> commands. If <code>use_batch_markers</code> is set to on the function strings <code>rs_batch_start</code> and <code>rs_batch_end</code> are executed. The default is off.
New DSI configuration parameters	<p>DSI must be correctly configured for batching of commands to the replicate data server. There are three DSI configuration parameters to consider for each connection that will be batching commands:</p> <ul style="list-style-type: none"><li>• <code>batch</code></li><li>• <code>batch_begin</code></li><li>• <code>use_batch_markers</code></li></ul> <p>For more information about configuring DSI for non-ASE data servers, see Chapter 5, “Replication Server Issues,” in the <i>Replication Server Heterogeneous Replication Guide</i>.</p>

## SySAM license management

The Sybase Software Asset Management (SySAM) implementation has changed for this version of Replication Server. The changes include:

- Asset management and reporting tools are provided with SySAM version 2.0. These tools allow you to monitor license usage and compliance.
- A single installation method supports all editions of Replication Server.
- SySAM configuration is no longer optional.
- Flexible SySAM configuration options.
- SySAM licenses are no longer shipped along with order fulfillment. You must obtain license certificates from the Sybase Product Download Center (SPDC).



- SySAM license keys include information about the support plan you purchased. You must update these licenses whenever you renew your support plan.
- Licensing policies are strictly and consistently enforced.
- Replication Server can function under grace periods if cannot obtain a license. These grace periods allow you reasonable time to respond to the issues causing license failure. During the grace period, the Replication Server continues to function normally. However, the Replication Server or any of its features, will shut down at the end of the grace period if licensing issues remain unresolved.
- You can receive real-time e-mail notifications about licensing events.
- Licenses issued from SPDC include information about the host machine where the licenses will be deployed. These licenses cannot be used on another machine without being reissued from SPDC.

---

**Note** Replication Server 15.0 does not require the REP\_SSL license, as SSL now comes as a part of the basic REP\_SERVER license.

---

These changes affect the Replication Server installation and configuration process. For pre-installation and SySAM installation information, see the *Replication Server Installation Guide* for your platform.

Plan your SySAM deployment before installing Replication Server.



# New Features in Replication Manager 15.0

This chapter describes new features in the 15.0 version of the Replication Manager, a plug-in to Sybase Central.

<b>Topic</b>	<b>Page</b>
New user interface features	93
Replication Manager features supported	98

## New user interface features

The Replication Manager supports several new interface features that promote ease-of-use and productivity.

## Two-tier management solution

Replication Manager can manage replication environments by connecting directly to servers in the environment without communicating through a management server layer. This two-tier management solution lets you manage small, simple replication environments with fewer than ten servers.

To set up a replication environment in Sybase Central, you select the servers contained in the environment from a drop-down list, which is derived from the local interfaces file, then provide a user name and password that the Replication Manager uses to connect to these servers. These passwords are encrypted when stored in the Sybase Central repository.

## **Three-tier management solution**

If you are managing a large or complex replication environment, you might want to install the Replication Monitoring Services (RMS) server. In a three-tier management solution, RMS server is a middle-management layer that monitors the status of the servers and other components in the replication environment. Replication Manager connects to the servers in the environment through RMS.

The Replication Manager provides the client interface that displays the information provided by RMS.

For more information about RMS and its functionality, see Chapter 8, “Introducing Replication Monitoring Services.”

## **Replication Manager plug-in replaces Replication Server plug-in**

With Replication Server 15.0, the Replication Manager plug-in replaces the Replication Server plug-in as the complete management tool for developing, managing, and monitoring a Sybase Replication Server environment for a two-tier management solution.

In earlier versions, the Sybase Central Replication Server Manager included a Replication Server plug-in and a Replication Server Manager Server (RSM Server) as the software tools to monitor, analyze, troubleshoot, and administer a replication system.

The Replication Server plug-in included a graphical user interface (GUI) integrated with Sybase Central. The Replication Manager plug-in has an interface that is similar to the Replication Server plug-in, but that runs within the Sybase Central, Java Edition framework.

In addition, Replication Manager does not require the RSM Server to manage servers in a replication environment. It communicates directly with the Replication Server and Adaptive Servers in a two-tier management solution.

## **Online help**

Online help for Replication Manager is now available. The online help contains extensive topic-level help that provides a quick reference for all Replication Manager concepts and tasks.

Select Help from the Sybase Central main menu and then select Replication Manager online help.

## Visual monitoring of status

The state of each object displays on the object icon, in the parent object Details list, and on the Properties dialog box for that object. You can monitor the status of servers, connections, routes, and queues.

## Using the Details list

When you select an object in the left pane that contains subcomponents or function components, one or more tabs display in the right pane of the Sybase Central window with lists of information. For most objects, a single tab called “Details” displays, which contains a list of general information about the object.

The Details list displays:

- Subcomponents, which are other replication or database objects that are contained in another object.
- Function components, which are components that invoke a wizard when double-clicked. For example, the Add Connection object is a function component that invokes the Add Connection wizard.

## Event Log pane

Replication Manager displays an event log in a pane at the bottom of the Sybase Central window. The event log displays:

- Component state changes for connections, routes, and queues
- Server availability changes
- Background thread completion
- RMS event trigger execution

---

**Note** For more information about RMS and event triggers, see Chapter 8, “Introducing Replication Monitoring Services.”

---

To display or hide the event log, select Event Log from the View menu.

## Background processing

Several tasks performed by Replication Manager can be very time-consuming, such as creating a subscription that also materializes the table. These tasks are now performed in the background, allowing Sybase Central to continue to function. When you start a time-consuming task, Replication Manager displays a message window indicating that a process is running. You can click Stop Process in this window to stop the process.

---

**Note** The process continues even if the Background Process window is closed.

---

When a background task completes, the Replication Manager places an event entry in the event log.

## Using the Background Processes dialog box

To see the status of a background process, you can open the Background Processes dialog box, which displays a list of all of the currently running background threads.

To access the Background Processes dialog box, select Search | Background Processes. The Background Processes dialog box opens, displaying the following:

- Process – the name of the process.
- Start time – the start time of the process.
- Status – the status of the process.

## Replication Manager logging enhancement

Replication Manager now uses the Sybase Central message logging feature to provide a log of all commands sent by the Replication Manager to any server. Because the log may contain passwords (for example, passwords are needed to create a connection, and these passwords are saved in the log), the Replication Manager provides the ability to turn off command logging.

### ❖ Turning command logging on and off

- 1 Right-click the Replication Manager object (at the top of the tree).
- 2 Select Properties.

- 3 In the Replication Manager properties dialog box, select Write SQL Commands to Log.

To turn logging off, return to the Replication Manager properties dialog box and unselect the Write SQL Commands to Log check box.

---

**Warning!** Turning on command logging can fill up the Sybase Central log, causing it to crash. If you turn command logging on, monitor the log closely.

---

## Script editors

Replication Manager provides two script editors, the Replication Command Language (RCL) script editor and the Structured Query Language (SQL) script editor. These editors operate the same way, except the RCL script editor highlights RCL keywords while the SQL script editor highlights SQL keywords.

You can use the script editor to view generated RCL commands, which include syntax to create connection and configuration parameters that can be used to create connections outside the Replication Manager.

With the script editors, you can:

- Select several Replication Server objects and generate RCL for all objects selected.
  - Edit and save the generated RCL script.
  - Load an RCL script from a file and add it to the current script.
  - Resubmit an RCL script to Replication Server.
- ❖ **Accessing the script editor**
- 1 Select the Replication Server object for which you want to generate RCL.
  - 2 Right-click that object.
  - 3 Select Generate RCL from the context menu. The RCL Script Editor opens and contains the commands needed to create the object.

## Replication Manager features supported

This section describes the Replication Server 15.0 features that the Replication Manager also supports.

### Support for new datatypes

Replication Manager supports the following new datatypes supported by Replication Server 15.0:

- bigint
- unsigned integer datatypes
  - unsigned tinyint
  - unsigned smallint
  - unsigned int
  - unsigned bigint
- unitext

For more details of the datatypes supported by Replication Server 15.0, see Chapter 6, “New Features in Replication Server Version 15.0.”

### Support for DirectConnect

In version 15.0, the Replication Manager manages a component that represents a Sybase DirectConnect data access server. The DirectConnect server acts as an Open Server gateway by converting the Open Client/Server protocol used by Replication Server to the native communication protocol used by the non-Sybase replicate database.

In Sybase Central, the DirectConnect data access server is managed as any other object in the replication environment. The Replication Manager displays the state on the icon of the DirectConnect data access server and on the parent object’s Detail list. The state of the server reflects whether the server is available and the state of the back-end data servers.

#### Limitations

In version 15.0, the Replication Manager neither displays the error log nor sets configuration parameters for DirectConnect.



## Replication support

Replication Server provides replication at the database, table, and stored procedure levels. Replication Manager allows you to create a replication definition for a database, a table, or a stored procedure. Note that a replication definition for a stored procedure is called a function replication definition. You can create, alter, and delete function replication definitions and function subscriptions.

For more information about creating, altering, and deleting replication definitions and subscriptions, see the Replication Manager online help.

## Routes

A route is a one-way message stream from a source Replication Server to a destination Replication Server. Routes carry replication data. You can use Replication Manager to manage routes.

For more information on managing routes using Replication Manager, see the Replication Manager online help.

## Upgrading routes

With Replication Server 15.0, you can upgrade routes using the Replication Manager. In earlier versions, the RSM Server provided the route upgrade capability.

The Replication Manager displays a set of all routes in the Replication Server that are eligible to be upgraded. You can select one single route at a time for upgrade. Upon selection of a route, the Replication Manager runs the upgrade process in the background and reports any errors or warnings in the event log.

The new route upgrade implementation provides the following additional functionalities:

- Canceling a route upgrade process – the ability to cancel a route upgrade process. However, canceling can leave the replicate Replication Server in an unstable state.
- Recovering from an unsuccessful upgrade – a recovery procedure to reset the replicate RSSD. An unsuccessful upgrade can result from either an abnormal termination of the upgrade process or a cancellation of the upgrade process.

❖ **Upgrading a route using Replication Manager**

- 1 In the object tree, select a Replication Server object.
- 2 Right-click the Replication Server object and select Upgrade Route. A dialog box lists all the routes that can be upgraded.
- 3 From the list, select a route and click the Upgrade button.

A background process starts, which on completion indicates whether the upgrade is successful or not.

Limitations

The route upgrade implementation has these limitations:

- You cannot currently upgrade multiple routes at the same time.
- Route upgrade is supported only in an environment domain using the two-tier solution.

## Troubleshooting tools

Occasionally, an environment stops replicating data. This can happen when a transaction is not formatted correctly, or when a server generates an error. To troubleshoot the situation, you can view the Replication Server queue data and exceptions log.

### Accessing the exception log

Use the exceptions log to troubleshoot a problem with replication.

❖ **Using the exceptions log**

When a transaction is not processed because of a SQL error, you can:

- 1 Issue a resume command with a skip transaction clause to the queue.
- 2 Right-click the connection and select View Exceptions.  
  
The View Exceptions dialog box displays the exceptions log in the Exceptions Log table.
- 3 View the skipped transaction and the erroneous SQL in the exceptions log.
- 4 Filter the display of the exceptions log by selecting the columns in the table: Origin Data Server, Origin Database, and Previous Hours.
- 5 To see the command language associated with a transaction, select it from the Exceptions Log table. The Command Editor text box displays the SQL commands in the transaction.

- 6 Edit the transaction in the text box.
- 7 To resubmit the transaction to the replicate data server, select Resubmit.

## Accessing queue data

Data that is passed between servers (Adaptive Server, Replication Server, and so on) is stored in stable queues within Replication Server. The Replication Manager displays the statistics of queue usage and displays the content of the queues.

### Using the View Queue Data dialog box

The View Queue Data dialog box lets you filter and sort the data from a queue as an aid in troubleshooting transactions in the queue. You can also edit, delete, or undelete a given command, or purge the first transaction in the queue.

The View Queue Data dialog box contains the following fields:

- Filter fields, which let you select the type of filters that the Replication Manager uses to display data from the queue. These filters include:
  - Column
  - Column value
  - Segment
  - Starting block
  - Number of blocks displayed
  - Number of rows displayed
  - Whether to start at the first active segment or not
  - Whether to include all data to the end of the segment
  - Whether to include all rows or not
  - Whether you want to show deleted data
  - Whether to view all data to the end of the queue
- General buttons, which let you:
  - Display the queue data with the current filters
  - Close the dialog box
  - Purge the first transaction from the queue

- Edit transactions
- Delete transactions
- Undelete transactions
- Group transactions, which returns the Queue Data scrolling list display back to grouped transactions
- Queue Data scrolling list, which contains rows of data from the current queue. Each column contains specific information about the command and transaction contained in each row. For example, to sort the queue data by a specific column, select that column name. The Queue Data scrolling list refreshes, sorting the data according to that column. An arrow displays next to the column name to show that you have sorted the data by that column. The columns you can sort by include:
  - Segment
  - Transaction Name
  - Command
  - Origin Site
  - Origin Commit Time
  - Origin User
  - Transaction ID
  - Origin QID

---

**Note** You can only delete, undelete, or purge queue transactions when Replication Server is in standalone mode. For more information, see Chapter 3, “Managing Replication Server with Sybase Central,” in the *Replication Server Administration Guide Volume 1*.

---

❖ **Viewing queue data**

- 1 Right-click the queue whose data you want to view.
- 2 Select View Data. The View Queue Data dialog box opens.
- 3 To filter data shown, select one of the filter fields.

For more information, see “Using the View Queue Data dialog box” on page 101.

- 4 To sort the data, select segment, transaction, origin, size, status, commit time, or user.

## Connection status hide options

You can hide (or filter out) the status of connections if you do not want to see the connection status either on the individual connection icon or as part of the rollup status for Replication Server.

Because the filtering state of the connection status is stored locally by the Replication Manager, different instances of the Replication Manager do not share filtering states. For example, if you create a connection using one instance of the Replication Manager, and then set the Replication Agent status to “hide” for that connection, another Sybase Central plug-in instance monitoring the same environment does not filter the connection status because the filtering information is available only to the original Replication Manager instance.

In addition, any connection created outside of Sybase Central (by `rs_init` or from the command line) is not filtered automatically by the Replication Manager. You must set the filtering manually from within Sybase Central.

## Filtering connection status in warm standby environments

If you are creating a warm standby environment, the Replication Manager automatically sets the filtering state for the active Data Server Interface (DSI) thread and standby RepAgent thread connections. You must set filtering for the physical connection manually by selecting one of the connection status hide options from the context menu.

## Using connection status hide options

The options for hiding connection status are as follows:

- Hide the State of the Replication Agent – hides the state of the Replication Agent thread in the Details list, on the Connection Properties dialog box, and in the rollup status for the Replication Server to which that Replication Agent thread is connected.
- Hide the State of the DSI Thread – hides the state of the DSI thread in the Details list, on the Connection Properties dialog box, and in the rollup status for the Replication Server to which the DSI thread is associated.

❖ **Hiding connection status**

- 1 Right-click the connection whose status you want to hide.
- 2 Select Hide Connection Status from the drop-down menu.  
A dialog box shows options for hiding the connection status.
- 3 Select an option.

The state for that connection now reads “Hidden.” The state on the Connection Properties dialog box and in the rollup status for the Replication Server is also hidden. The Event Log records this change.

## Warm standby wizards

To create a warm standby environment, you must create the following components in order:

- A logical connection
- A connection to the active database
- A connection to the standby database

Previously, these steps were only part of the Configure Replication Environment wizard, which enables you to build a warm standby environment in one step. With this version of Replication Manager, you can use a separate wizard for each step in the process, which lets you drop and re-create connections as needed.

The three wizards are:

- Add Logical Connection – add a logical connection whether you are basing this connection on an existing physical connection or not.
- Add Active Database – add an active database connection to an existing logical connection.
- Add Standby Database – add a standby database to an existing logical connection that already has an active database connection.

## Using the Add Logical Connection wizard

Using the Add Logical Connection wizard, you can add a logical connection whether you are basing this connection on an existing physical connection or not.

**❖ Creating a logical connection**

- 1 In the tree view, select the Logical Connection folder under the Replication Server object.
- 2 In the Details view, double-click Add Logical Connection. The Add Logical Connection wizard starts.
- 3 On the Convert Existing Connection wizard page, verify that the Use an Existing Connection as the Active Connection check box is not selected.
- 4 Enter the database name and data server name for the logical connection.
- 5 Review the summary information for the logical connection.
- 6 If everything looks correct, click Finish. Otherwise, click Back to return to an earlier page in the wizard and change the logical connection information. Then return to the final wizard page and click Finish.

Replication Manager creates the logical connection object.

---

**Note** You must create the connections to the active and standby databases before you have a working warm standby environment. For more information, see “Using the Add Active Database wizard” on page 105 and “Using the Add Standby Database wizard” on page 106.

---

## Using the Add Active Database wizard

Using the Add Active Database wizard, you can add an active database connection to an existing logical connection.

**❖ Creating an active database connection**

- 1 In the Details view, right-click the logical connection object you created using the Add Logical Connection wizard.
- 2 Select Add Active Database. The Add Active Database wizard starts.
- 3 Select the active server and active database.
- 4 Select the Replication Server that will manage the database connections.
- 5 Enter the user name and password of the maintenance user.
- 6 Select the user name and password that the RepAgent will use to connect to Replication Server.

If the RepAgent user does not exist, the wizard creates one for you and gives it a default name and password. Accept the defaults, or enter your own values.

- 7 Select the Enable all objects in the active database check box to enable database objects.
- 8 Review the summary information about the replication environment.
- 9 If everything looks correct, click Finish. Otherwise, click Back to return to an earlier page in the wizard and change the replication environment information. Then return to the final wizard page and click Finish.

Replication Manager creates the active database connection.

## Using the Add Standby Database wizard

Using the Add Standby Database wizard, you can add a standby database to an existing logical connection which already has an active database connection.

### ❖ Creating a standby database connection

- 1 In the Details view, right-click the logical connection object you created using the Add Logical Connection wizard.
- 2 Select Add Standby Database. The Add Standby Database wizard starts.
- 3 Select the standby server and standby database.
- 4 Select the Replication Server that will manage the database connections.
- 5 Enter the user name and password of the maintenance user.
- 6 Select the user name and password that the RepAgent will use to connect to Replication Server.

If the RepAgent user does not exist, the wizard creates one for you and gives it a default name and password. Accept the defaults, or enter your own values.

- 7 Select the materialization method.
- 8 Select one of the following options:
  - Initialize Standby Database with Dump Load, which initializes the standby database with the current data dump from the active database.
  - Use Dump Marker in Transaction Log, which replicates transactions that are executed between the time the active database is enabled and the time the data is dumped.



- 9 Review the summary information about the replication environment.
- 10 If everything looks correct, click Finish. Otherwise, click Back to return to an earlier page in the wizard and change the replication environment information. Then return to the final wizard page and click Finish.

Replication Manager creates the standby database connection.

---

**Note** Before you use your warm standby environment, resume the DSI thread on the replicate database connection, if necessary.

---

## Thread management

Replication Manager displays the state of the threads in the Replication Server. If the thread is directly related to another Replication Server component, such as a connection, route, or queue, then the thread component is represented in Sybase Central by the related component and its features. For example, the Replication Agent thread (RepAgent) or DSI threads are represented by the associated connection component. The DSI EXEC thread, however, is not represented by another component.

## Using thread context menus

The Thread context menus give you access to the menus of a related component, if a related component exists for a particular thread. For example, if you right-click a DIST thread, the context menu displays Connection and Copy. If you select Connection, a context submenu displays, showing you the commands from the connection context menu.

---

**Note** Some thread objects are not related to a component and, therefore, do not give access to a submenu. For example, if you right-click a DSI EXEC thread, the context menu displays only the Copy command.

---

The following table shows the mapping of thread objects to related components:

Thread	Related component
DSI	Connection
DIST	
REP AGENT	

Thread	Related component
SQM	Queue
SQT	
RSI	Route
USER	User

## Viewing the Details list and other thread information

When you select the Threads folder in the left pane, the Details list and tab display, as well as several other tabs. These other tabs give you access to additional information about each type of thread, as follows:

- DIST – displays information about distributor threads.
- DSI – displays information about DSI threads.
- RSI – displays information about RSI threads.
- SQM – displays information about SQM threads.
- SQT – displays information about SQT threads.

See the *Replication Server Administration Guide Volume 1* for more information about Replication Server threads.

## Using thread information

You can copy information about a thread from any of the thread information lists to the system clipboard. The columns of data in each list are separated by tabs so that you can paste the columns directly into a columnar format such as a spreadsheet.

### ❖ Copying thread information

- 1 Right-click the thread in the right pane.
- 2 Select Copy.
- 3 Go to the destination file and paste the information as you would any text.

For example, if you select the DIST thread number 19 for a Replication Server called “myRepServer,” then select Copy from the menu and paste that information into a file, the results look similar to the following:

```
19 Awaiting Wakeup 102 myRepServer.emb2 102 P Normal 0 1 0 21787 43856 0 0 0 14
```

# Introducing Replication Monitoring Services

This chapter introduces a new Replication Server 15.0 component called the Replication Monitoring Services (RMS). RMS is the new middle-management monitoring layer that replaces the existing Replication Server Manager and provides monitoring services for large and complex replication environments.

Name	Page
Introducing Replication Monitoring Services	109
Monitoring a replication environment using RMS	113

## Introducing Replication Monitoring Services

Replication Monitoring Services (RMS) replaces the functionality of the existing Replication Server Manager Server (RSM Server). RMS monitors the servers and components in a replication environment, provides the ability to control the flow of data in the replication environment, and sets the configuration parameters.

RMS functionality is available through the Replication Manager plug-in to Sybase Central and the command line API. Replication Manager provides commands to create, modify, or delete replication objects, while RMS provides an API to monitor, manage, and configure the replication environment.

RMS is applicable only for a three-tier management solution.

Three-tier management solution

A three-tier management solution is for large and complex replication environments consisting of ten or more Replication Servers. The Replication Manager connects to the servers in the environment through RMS. RMS provides the monitoring capabilities for the replication environment.

RMS monitors the status of servers and other components in the replication environment, and the Replication Manager plug-in provides the client interface that displays the status information provided by the RMS.

Table 8-1 lists the features supported in the Replication Server Manager, a component in the earlier versions, and in the Replication Monitoring Services, the new component in version 15.0.

**Table 8-1: Difference in features supported in RSM and RMS**

Feature	Replication Server Manager	Replication Monitoring Services
Allows you to manage, monitor, and configure replication system components	X	X
Monitors the availability of servers and the state of all connections and routes	X	X
Manages a warm standby environment	X	X
Supports multisite availability		X
Supports the Embedded RSSD	X	X
Provides support for SSL and network-based security (DCE/Kerberos) security between servers in a replication environment	X	
Runs on Microsoft Windows platforms as well as all UNIX platforms supported by Replication Server		X
Provides a server-centric view of the replication environment	X	X
Enables administration of a logical group of servers		X
Sets configuration parameters of Replication Servers, Replication Agents, the Adaptive Server Enterprise Replication Agent thread (RepAgent), connections, and routes	X	X
Monitors the latency and the state and performance of a replication path	X	X
Provides commands to create, alter, and delete replication objects	X	
Provides logging and tracing of server commands	X	X
Generates a rollup status for Replication Servers and Adaptive Servers	X	X
Executes user-defined scripts for events in the replication environment	X	X

## Monitoring servers in the replication environment

Using RMS, you can monitor the following servers in your replication environment:

- Adaptive Server Enterprise
- SQL Anywhere and IQ

- Replication Agent
- Mirror Replication Agent
- DirectConnect
- Open Server
- Replication Server
- Remote RMS servers

## Software requirements and compatibilities

Replication Monitoring Services requires the following software:

- JRE version 1.4.2.03
- Sybase Unified Agent Framework (UAF) version 1.5.0.97
- jConnect™ for JDBC™ version 6.0

## Installation

You can use InstallShield to install the Replication Monitoring Services component at the same time you install Replication Server.

If you specify a Typical or Full installation when you install Replication Server 15.0, InstallShield automatically installs the Replication Monitoring Services component along with Sybase Central and Replication Manager.

## Starting and stopping RMS

RMS is a Java application built using the Sybase Unified Agent Framework (UAF). To start or stop RMS you must start or stop UAF.

### ❖ Starting RMS

- 1 Navigate to the Replication Server installation directory, *%SYBASE%* on Windows, or *\$SYBASE* on UNIX.
- 2 Set the environment variables by executing *SYBASE.bat* (Windows) or by sourcing *SYBASE.csh* (UNIX).

- 3 Change to the `%SYBASE-UA%bin` directory (Windows) or the `$SYBASE-UA/bin` directory (UNIX).
- 4 In Windows, execute `agent.bat` or add this command to run Unified Agent (UA) in the background. Similarly on UNIX, execute `agent` or add this command to run UA in the background.

❖ **Stopping RMS**

- 1 Navigate to the Replication Server installation directory, `%SYBASE%` on Windows, or `$SYBASE` on UNIX.
- 2 Set the environment variables by executing `SYBASE.bat` (Windows) or by sourcing `SYBASE.csh` (UNIX).
- 3 Stop RMS:

- If an Adaptive Server Unified Agent is running, enter one of these commands:

```
shutdown [-port rmi-port] [-U username]  
[-P password]
```

Or

```
shutdown [-port rmi-port] [-user username]  
[-password password]
```

where:

- *rmi-port* value is 9999.
- *username* and *password* are the values configured for the UA.
- If RMS is the only Unified Agent running, enter `shutdown.bat` (Windows) or `shutdown` (UNIX).

## Connecting to RMS in Sybase Central

To connect to RMS in Sybase Central:

- 1 Select the Connect icon from the toolbar. The Connect to a Replication Domain window opens.
- 2 Select the RMS Server radio button.
- 3 Enter the user name and password required to connect to RMS.
- 4 Select RMS from the list of servers in the drop-down list or click the Options button to provide the connection information for the RMS.

5 Enter a server name, host, and port number.

6 Click OK.

The RMS server is added to the object tree in Sybase Central.

## Monitoring a replication environment using RMS

To monitor a replication environment, you must set up a RMS domain. This domain consists of the servers in your environment you want to monitor using the RMS. In a three-tier management solution, the Replication Manager connects to the RMS, which then connects to the various servers in your replication environment.

To set up the RMS domain and monitor servers in this domain, you can either:

- Use the Replication Manager plug-in graphical interface, or
- Use command line API

In a three-tier replication environment, you can perform some monitoring through Replication Manager, and additional monitoring through the RMS API at the command line.

The following sections provide information on how to monitor RMS using the Replication Manager plug-in and the API. For those tasks for which you can use both interfaces, only the Replication Manager plug-in is described in detail, while the name of API is identified. For those tasks that can be performed using only the API, detailed API information is given.

## Adding and dropping servers for monitoring in Sybase Central

Add and drop servers in a three-tier environment in the same way as in a two-tier environment.

### Adding a server

The servers that you add for monitoring in a three-tier solution can be from anywhere in your network. This allows you to monitor replication systems that are distributed worldwide.

You can use Replication Manager wizard to add a server to a RMS domain. Depending on the type of server you are adding (Adaptive Server, Replication Server, RepAgent, Open Server, Sybase Replication Agents, DirectConnect), the wizard prompts for different information.

Before you add a server, have this information available:

- The server's name.
- The type of server you are adding.
- The user name and password used to administer the server. The login must have System Administrator privileges on the server.
- Replication Server only – the user name and password of the RSSD primary user.

To add servers to the RMS server folder in Sybase Central:

- 1 Select the RMS.
- 2 Select File | New | Servers, or double-click the Add Servers icon in the right pane of the main window.
- 3 Enter the required information in the wizard dialog boxes.

The servers are added to the Sybase Central viewer under the RMS server in the object tree.

API command to add server to monitor

Use `add server` to add a server to be monitored by RMS. For detailed information about this API, see Chapter 9, “Replication Monitoring Services API,” in the *Replication Server Reference Manual*.

## Dropping a server

To drop a server from the RMS domain:

- 1 Select the server you want to drop.
- 2 Do one of the following:
  - Click the Delete icon from the toolbar.



- Right-click the selected server and select Delete from the context menu.

---

**Note** Although Sybase Central removes the server from the RMS server folder, the server is not actually removed from your replication system. Therefore, the server name may still appear in the dialog boxes because there are routes or database connections associated with it.

---

API command to drop server from being monitored

Use `drop server` to drop a server that is being monitored by RMS. For detailed information about this API, see Chapter 9, “Replication Monitoring Services API,” in the *Replication Server Reference Manual*.

## Viewing monitored objects in Sybase Central

Viewing monitored objects in Replication Manager is the same as viewing objects in a two-tier environment.

In the object tree, double-click or expand the RMS icon to view the replication objects managed by RMS. Under RMS, you can view the monitored servers and its components such as connections, routes, queues, and threads. When you select a particular replication object such as the Routes folder, you can view the list of created routes. You can manage these replication objects using Replication Manager.

## Setting configuration parameters for monitored replication objects

You can set the configuration parameters for the following replication objects monitored by RMS:

- Replication Server
- Replication Agent thread
- Remote RMS server
- Database connections and logical connections
- Routes

To set the configuration properties of an object, in Replication Manager

- 1 Select the object and choose File | Properties. The property sheet for the object opens.

- 2 Select the parameters tab in the Properties dialog box. The Parameters page displays a list of all configuration parameters for the selected server or component.
- 3 Select the parameter you want to modify and click the Edit Parameter button.
- 4 In the Edit dialog, enter the new parameter and click OK.
- 5 You can modify any other parameter in the list. When you are finished, click OK in the Properties dialog box.

API commands to set configuration parameters

RMS API commands to set configuration parameters for the replication objects are:

- `configure server` – returns configuration parameter information for a Replication Server or Replication Agent, or sets the value of a specified configuration parameter. `configure server` also retrieves and sets RMS-specific parameters.
- `configure component` – returns configuration parameters for a component or sets the value of the configuration parameter specified.
- `configure RMS` – returns the configuration parameter information for the RMS, or sets the value of a specified RMS configuration parameter

For detailed information about these APIs, see Chapter 9, “Replication Monitoring Services API,” in the *Replication Server Reference Manual*.

## Monitoring a logical group of servers

RMS enables you to define a set of servers as a logical group and monitor the group as a single entity. You can have many logical groups in your replication environment. The servers in a group can belong to different logical groups. However, all servers in one logical group must be of the same type. For example, you can have one logical group of all Replication Servers and another for all Adaptive Servers. When you issue any command to a group, it affects all the servers contained in this logical group.

You can delete logical groups. When you delete a logical group, only the group is deleted, not the servers within the group.

RMS also returns a rollup status for each group or each server in a group. Roll-up status shows the lowest status reported, for example, if any server in a group is not UP, then the group status is reported as SUSPECT.

API commands to create/delete/list logical groups

RMS API commands to manage logical groups are:

- `create group` – defines a logical group of servers.
- `delete group` – deletes a logical group that was added using the `create group` command.
- `get group` – returns a result set that contains either a list of the groups and a rollover status for each group or for each server.

For detailed information about these APIs, see Chapter 9, “Replication Monitoring Services API,” in the *Replication Server Reference Manual*.

## Suspending or resuming components in the replication environment

You can suspend or resume the following monitored components in the replication environment:

- RepAgent in Adaptive Server
- Replication Agent and DSI connection in Replication Server
- Routes
- Queues (resume only)
- Replication in a Replication Agent

In Sybase Central, select the monitored connection or route from the RMS Server folder in the object tree:

- To suspend a monitored connection or route, select the connection or route object from the RMS Server folder and select Suspend from the context menu.
- To resume a monitored connection or route, select the connection or route object from the RMS Server folder and select Resume from the context menu.

API commands to suspend/resume replication components

RMS API commands to suspend or resume components in the replication environment are:

- `suspend component` – suspends a component in a specified server.
- `suspend replication` – suspends replication in a Replication Agent.
- `resume component` – resumes a component in a specified server.
- `resume replication` – resumes replication in a Replication Agent.

For detailed information about these APIs, see Chapter 9, “Replication Monitoring Services API,” in the *Replication Server Reference Manual*.

## Shutting down monitored servers

RMS enables you to shut down any monitored Replication Server, Replication Agent, or Mirror Replication Agents.

In Sybase Central, select the monitored server or RMS in the object tree and select Shutdown from the context menu. To shut down a group of servers, select the logical group folder and select Shutdown All from the context menu.

API command to shut down Replication Server, group, or RMS

Use shutdown *server* to shut down one or more, or group of monitored Replication Servers or RMS. For detailed information about this API, see Chapter 9, “Replication Monitoring Services API,” in the *Replication Server Reference Manual*.

## Generating rollup status for servers

The RMS status shows the lowest status reported, for example, if the status of any server in the list is not UP, then the status for the RMS is reported as SUSPECT.

For example, the status of the Adaptive Server reflects the status of its Replication Agent threads. If any of the monitored Replication Agent threads in the Adaptive Server are not UP, then the state of the Adaptive Server is set to SUSPECT.

For more information about RMS server and component states, see Appendix C, “RMS Server and Component States,” in the *Replication Server Reference Manual*.

API commands to view rollup status information

RMS API commands to view rollup status information and the various server states are:

- get servers – returns a list of servers that are monitored by the RMS, and the status of the RMS environment.
- get status description – retrieves the list of status descriptions for a server or component.

For detailed information about these APIs, see Chapter 9, “Replication Monitoring Services API,” in the *Replication Server Reference Manual*.

## Generating latency and heartbeat information

The RMS heartbeat feature uses the stored procedure `rs_ticket` to generate latency information, which is the amount of time it takes for a transaction to move from the primary database to the replicate database. You can use this information to monitor Replication Server performance, module heartbeat, replication health, and table-level quiesce.

At a specified interval, RMS executes `rs_ticket` at the primary database. The generated latency information is stored in a table in the replicate database. You can use RMS to set up the heartbeat process to retrieve the latency information from the replicate database.

For more information about `rs_ticket` and its functionality, see Chapter 6, “Adaptive Server Stored Procedures,” in the *Replication Server Reference Manual*.

API commands to monitor heartbeat

RMS API commands to monitor heartbeat are:

- `get heartbeat` – retrieves a list of the heartbeat processes that have been defined in the RMS.
- `get heartbeat ticket` – retrieves a set of tickets from the `rms_ticket_history` table, for the heartbeat process and date and time range specified.
- `start heartbeat` – sets up and starts a heartbeat process from a specified primary connection to a specified replicate connection.
- `stop heartbeat` – stops the heartbeat process between the primary and replicate databases. Optionally truncates the `rms_ticket_history` table.

For detailed information about these APIs, see Chapter 9, “Replication Monitoring Services API,” in the *Replication Server Reference Manual*.

## Adding event triggers

Replication Monitoring Services is designed to monitor the replication environment. When something happens in your environment, server and component status changes. These changes are displayed in the event log. RMS allows you to create event triggers to monitor these changes.

Event triggers notify you when some events occur in the replication environment. RMS executes the script when the specified event occurs. For example, a user can set up a script to be notified with an e-mail message when a connection suspends. You can create an event trigger for any server or component that the RMS monitors.

❖ **Creating an event trigger for a Replication Server**

- 1 In the object tree, select the Replication Server.
- 2 On the right side of the desktop, select the event log pane.
- 3 Double-click the Add Server Event Trigger icon.
- 4 Select the status change that will trigger the event.
- 5 As an option, enter a “Wait before executing” value. This notifies RMS to wait for the event to change before executing the trigger.
- 6 To execute a trigger at each monitoring interval, rather than only once, select “Execute at Each Interval.”
- 7 Enter the name of the script for RMS to execute when the event occurs.
- 8 Click OK. The new event displays in the Event Log pane.

API commands to  
add/drop/get triggers

RMS API commands for trigger-related tasks are:

- `add event trigger` – sets up a trigger, such as a process or a script, that is executed by the RMS when a specific event occurs.
- `drop event trigger` – removes a trigger that the RMS is monitoring.
- `get triggers` – displays information about the triggers that are monitored by the RMS.

For detailed information about these APIs, see Chapter 9, “Replication Monitoring Services API,” in the *Replication Server Reference Manual*.

# New Features in Replication Server Version 12.6

This chapter describes the new features introduced with Sybase Replication Server version 12.6 and 12.5 EBF.

Topic	Page
MultiSite availability (MSA)	121
Support for symmetric multiprocessors (SMP)	123
The embedded RSSD (ERSSD)	124
Performance enhancements	124
date and time datatypes	127
Support for sending encrypted passwords	128
New bulk materialization method	129
Chinese character set (GB18030) support	129

## MultiSite availability (MSA)

MSA extends Replication Server replication capabilities and can make the process of setting up a replication system both faster and easier.

Some of the features that MSA provides include:

- A simple replication methodology that requires only one replication definition for the primary database and only one subscription for each subscribing database.
- A replication filtering strategy that lets you choose whether or not to replicate individual tables, transactions, functions, system stored procedures, and data definition language (DDL).
- Replication of DDL to any replicate database—including non-warm standby databases.
- Replication to multiple replicate sites—for standby as well as nonstandby databases.

You can overlay MSA scenarios onto your existing replication structure. The procedures for implementing MSA are similar to those you already use to replicate to warm standby or replicate databases.

Database replication

When you use table and function replication, you describe each piece of data that is to be replicated using individual table and function replication definitions and subscriptions. This methodology allows you to transform data and provides fine-grained control over the information being entered in the replicate database. However, you must mark each table or function to be replicated, create a replication definition for each replicated table or function, and create subscriptions for each replication definition at each replicate database.

MSA lets you identify specific database objects: tables, functions, transactions, DDL, and system stored procedures in a single replication definition. You can choose to replicate the entire database; or you can choose to replicate—or not replicate—specific tables, functions, transactions, DDL, and system stored procedures in that database. If you do not need to replicate partial tables, MSA can provide replication while affording the advantages of simple setup and maintenance.

When the replicate is a warm standby database

In the non-MSA warm standby scenario, changes to the primary database are copied directly to the warm standby database without alteration. This methodology allows replication of DDL. To change or qualify the data sent, you must add table and function replication definitions. Each primary database can have one, and only one, standby database.

MSA provides all the features of traditional Sybase warm standby. In addition, MSA:

- Enables replication to multiple standby databases
- Provides the option to replicate or not replicate specific database objects

MSA supports the use of logical connections.



## Support for symmetric multiprocessors (SMP)

Replication Server 12.6 lets you run Replication Server on either symmetric multiprocessor (SMP) or single-processor platforms. Replication Server multithreaded architecture supports both hardware configurations.

---

**Note** The SMP option for Linux Edition is not available.

---

On a single processor platform, Replication Server threads run serially. On a multiprocessor platform, Replication Server threads can run in parallel, thereby improving performance and efficiency.

Replication Server support for multiple processors is based on Open Server support for multiple processors. Both servers use the POSIX thread library on UNIX platforms and the WIN32 thread library on Windows NT platforms. For detailed information about Open Server support for multiple processing machines, see the *Open Server Server-Library/C Reference Manual*.

When Replication Server is in single-processor mode, a server-wide mutual exclusion lock (mutex) enforces serial thread execution. Serial thread execution safeguards global data, server code, and system routines, ensuring that they remain thread-safe.

When Replication Server is in multiprocessor mode, the server-wide mutex is disengaged and individual threads use a combination of thread management techniques to ensure that global data, server code, and system routines remain secure.

### Configuring SMP

To enable SMP on a multiprocessor machine, use configure replication server with the `smp_enable` option. For example:

```
configure replication server set smp_enable to 'on'
```

### Monitoring thread status

You can verify Replication Server thread status using these commands and procedures:

- `admin who` – provides information on all Replication Server threads
- `admin who_is_up` or `admin who_is_down` – lists Replication Server threads that are running, or not running.
- `sp_help_rep_agent` – provides information on the RepAgent thread and the RepAgent User thread. This is an Adaptive Server stored procedure.

### Monitoring performance

Replication Server provides monitors and counters for monitoring performance. See Chapter 15, “Using Counters to Monitor Performance,” in the *Replication Server Administration Guide Volume 2*.

See the *Replication Server Administration Guide* for more information about increasing Replication Server performance using SMP.

## The embedded RSSD (ERSSD)

Replication Server can run either on an Adaptive Server Replication Server System Database (RSSD) or on an embedded RSSD (ERSSD). ERSSD is designed for users who do not want to manage the Replication Server RSSD in Adaptive Server. Replication Server is easier to install and manage with ERSSD. If you select embedded RSSD when you install Replication Server, ERSSD is automatically installed, configured, and started in the background. It is self-maintained. Backup procedures are automatic and pre-configured.

### Limitations

Currently, you cannot create a route originating from Replication Server with ERSSD. Nor can you migrate between RSSD and ERSSD.

To use the ERSSD, you must select it when you install Replication Server. For more details, see the *Replication Server Installation Guide* and the *Replication Server Administration Guide*.

## Performance enhancements

Replication Server 12.6 includes new performance enhancements. See Chapter 16, “Performance Tuning,” in the *Replication Server Administration Guide Volume 1* for detailed information about these enhancements.

## Better management of empty transactions

Transactions that contain only a begin and a commit statement can degrade the performance of warm standby connections. To enhance performance, these transactions are now deleted from the inbound queue as they are read.

To further enhance performance, Sybase recommends that you also tune your application to eliminate as many of these empty transactions as possible.

## Internal commit control for parallel processing

To resolve conflicting updates when using parallel processing, Replication Server must maintain transaction commit order and resolve commit consistency deadlocks.

Replication Server introduces a new method to maintain commit control using the function string `rs_dsi_check_thread_lock`. Replication Server uses `rs_dsi_check_thread_lock` to check whether the current DSI executor thread is blocking another replicate database process. This new method handles commit control within Replication Server thus requiring less network I/O than other methods, and may result in the rollback of only one transaction instead of many.

New database connection parameters used with internal commit control are:

- `dsi_commit_check_locks_intrl`
- `dsi_commit_check_locks_max`
- `dsi_commit_control`

## New Replication Server configuration parameters

You can use new Replication Server configuration parameter to fine-tune Replication Server performance.

- `sqt_init_read_delay` – the amount of time an SQT thread sleeps while waiting for an SQM read before checking for new instructions in its command queue.
- `sqt_max_read_delay` – the maximum amount of time an SQT thread sleeps while waiting for an SQM read before checking for new instructions in its command queue.

## New database configuration parameters

You can use new database connection configuration parameters to fine-tune Replication Server performance.

- `dsi_commit_check_locks_intrl` – specifies the number of milliseconds (ms) the DSI executor thread waits between executions of the `rs_dsi_check_thread_lock` function string. Used with parallel DSI.

- `dsi_commit_check_locks_max` – specifies the maximum number of times a DSI executor thread checks whether it is blocking other transactions in the replicate database before rolling back its transaction and retrying it. Used with parallel DSI.
- `dsi_commit_control` – specifies whether commit control processing is handled internally by Replication Server using internal tables (on) or externally using the `rs_threads` system table (off). Used with parallel DSI.

## Changed database configuration parameters

The database connection parameters `dsi_serialization_method` and `dsi_partitioning_rule` have changed.

`dsi_serialization_method` specifies how and when parallel DSI threads can start. It includes these new or changed options:

- `no_wait` – specifies that a transaction can start as soon as it is ready—without regard to the state of other transactions.
- `wait_for_start` – specifies that a transaction can start as soon as the transaction scheduled to commit immediately before it has started. Replaces the `none` option. When used with the `origin` partitioning parameter, this option replaces the `single_transaction_per_origin` serialization method.
- `isolation_level_3` – is the same as `wait_for_start`, except that DSIs will specify isolation level 3 when connecting to the replicate database.
- `wait_for_commit` – specifies that a transaction can start only when the previous transaction is ready to commit.
- `none` – maintained for backward compatibility. Replaced by `wait_for_start`.
- `single_transaction_per_origin` – maintained for backward compatibility. Replaced by `dsi_serialization_method` set to `wait_for_start` and `dsi_partitioning_rule` set to `origin`.

`dsi_partitioning_rule` specifies the partitioning rules the DSI uses to partition transactions among available parallel DSI threads. It includes these new options:

- `origin` – specifies that transactions with the same origin must be serialized when applied to the replicate database.

- `origin_sessid` – specifies that transactions with the same origin *and* the same process ID (SPID in Adaptive Server) must be serialized when applied to the replicate database. The LTL version must be 600 or later.

Sybase recommends that you try setting `dsi_partitioning_rule` to `origin_sessid`, time as this setting may provide the most efficient partitioning.

## date and time datatypes

There are two new datatypes, `date` and `time`, in Replication Server. These datatypes extend the existing `datetime` and `smalldatetime` datatypes, providing `date` and `time` columns to replicate and standby databases. Both are fixed-width 4-byte datatypes that support `rs_subcmp`, and mixed-version environments.

Replication Server version 12.6 with `date` and `time` datatype support is backward-compatible with earlier versions of Adaptive Server. However, earlier versions of Adaptive Server do not recognize `date` and `time`, and thus can send only `datetime` and `smalldatetime` data.

The new columns generated by `date` and `time` datatypes allow you to replicate `date` and `time` data to both standby and replicate databases. These columns can be part of the primary key in a replication definition, and are searchable columns in a replication definition. You can use `date` and `time` columns in the `where` clause of `define subscription`, `create subscription`, or `create article`. In the same way, the `date` and `time` columns are searchable parameters in a function replication definition, again used in the `where` clause of `define subscription`, `create subscription`, or `create article`.

**Table 9-1: Range and storage needs for date/time datatypes**

Datatype	Range	Storage needed
date	January 1, 0001 to December 31, 9999	4
time	12:00:00 AM to 11:59:59.999 PM	4
smalldatetime	January 1, 1900 to June 6, 2079; 12:00:00AM to 11:59:59.999 PM	4
datetime	January 1, 1753 to December 31, 9999; 12:00:00AM to 11:59:59.999 PM	8

## Support for sending encrypted passwords

Replication Server 12.6 supports the `-X` option in `isql` that sends encrypted passwords through the network when making a client connection.

To ensure that all Replication Server client connections—except the first connection to the RSSD—send encrypted passwords, set the Replication Server configuration parameter `send_enc_password` to `on`. For example, enter:

```
configure replication server
set send_enc_password to 'on'
```

To ensure that all Replication Server client connections, *including* the first connection to the RSSD, send encrypted passwords, set the configuration parameter `RS_enc_pw` to `on` in the `rs_name.cfg` file using a text editor.

If `RS_enc_pw` is `on`, all Replication Server connections to the RSSD send encrypted passwords, even if `send_enc_password` is `off`.

## New bulk materialization method

Replication Server 12.6 supports a new bulk materialization method for copying or moving a database from a source Adaptive Server to a destination Adaptive Server without shutting down the source Adaptive Server. The Adaptive Server quiesce database ... to *manifest\_file* and mount commands let you quiesce the server and copy or move the database.

To use this bulk materialization method, both the source and destination database servers must be Adaptive Server version 12.5.1 or later.

You can use quiesce database ... to *manifest\_file* to generate all the data storage information and then use mount to mount the data to a new database—see Chapter 10, “Managing Subscriptions,” in the *Replication Server Administration Guide*. You can use quiesce database ... to *manifest\_file* and mount when you add a warm standby database—see Chapter 13, “Managing Warm Standby,” in the *Replication Server Administration Guide Volume 1*.

For information about mount and unmount, see Chapter 22, “Database Mount and Unmount,” in the *Adaptive Server Enterprise System Administration Guide*.

## Chinese character set (GB18030) support

Replication Server supports all character sets supported by Adaptive Server Enterprise. Accordingly, Replication Server 12.6 supports the Chinese character set (GB18030).





# Index

## A

- Adaptive Servers
  - communication through Replication Manager 94
  - shared-disk cluster support 36
- adding event triggers in Replication Manager 119
- adding servers to RMS server 113
- admin config 58
- admin config** command 3
- alter connection** command 2
- alter partition command 82

## B

- background processes
  - running in background 96
  - stopping a background process 96
  - viewing in Background Processes dialog box 96
- Background Processes dialog box 96
- batching of commands
  - for non-ASE servers 89
  - using DSI connection and configuration parameters 90
  - using function strings 90
- bulk copy-in support 1–5
  - commands for 2–3
  - connection parameters 2
  - connection parameters, checking value of 3
  - connection parameters, setting value of 2
  - Data Server Interface (DSI), implementation in 2, 3
  - multi-statement transactions, support for 3
  - subscription materialization, changes to 3
- bulk insert. *See* bulk copy-in support
- bulk materialization 129

## C

- cascading connection, in Replication Server gateway 15
- character set
  - Chinese (GB18030) 129
- Chinese character set support 129
- commands
  - admin config** 3, 9
  - alter connection** 2, 6, 9
  - alter replication definition** 10
  - configure replication server** 2, 7
  - connect** 16, 17
  - create connection** 9
  - create replication definition** 10
  - create route** 83
  - disconnect** 17
  - show connection** 17
  - show server** 17
  - sysadmin dump\_tran** 41
  - sysadmin sqm\_unzap\_tran** 40
  - sysadmin sqm\_zap\_tran** 40
- commit control
  - internal 125
- computed columns
  - materialized 79
  - replicating 79
  - virtual 79
- configuration
  - stable queue cache parameters 51
- configuration parameters
  - dsi\_bulk\_copy** 2, 3
  - dsi\_bulk\_threshold** 2, 3
  - dsi\_non\_blocking\_commit** 6
- dynamic\_sql 61
- dynamic\_sql\_cache\_management 62
- dynamic\_sql\_cache\_size 61
- setting with Replication Manager 115
- stats\_reset\_rssd 60
- stats\_sampling 87

## Index

**configure replication server** command 2, 7  
connection profiles 27  
connection status  
    filtering in warm standby environments 103  
    hide options 103  
conventions, document style xii  
counter statistics  
    displaying on screen 87  
    optionally keeping in RSSD 60  
    reporting 87  
    saving to RSSD 87  
counters 43  
create partition command 82

## D

Data Server Interface 2, 3  
datatype  
    bigint 76  
    date, time 127  
    unitext 78  
    unsigned 77  
datatypes  
    opaque 46, 55  
    supported by Replication Manager 98  
    timestamp 56  
date, database 127  
Details list 95  
dialog boxes  
    Background Processes 96  
    Connection Properties 103  
    View Queue Data 101  
direct I/O file access 52  
DIST status recording 49  
dropping servers from RMS server 114  
DSI 2, 3  
**dsi\_bulk\_copy** connection parameter 2, 3  
    checking value of 3  
    *See also* bulk copy-in support  
    setting value of 2  
**dsi\_bulk\_threshold** connection parameter 2, 3  
    checking value of 3  
    *See also* bulk copy-in support  
    setting value of 2  
**dsi\_quoted\_identifiers** 9

dynamic parameters  
    configuring 57  
dynamic SQL  
    configuration parameters setup 61  
    enhancement 33, 53  
dynamic SQL parameters 58

## E

editors  
    RCL script editor 97  
    SQL script editor 97  
Embedded Replication Server System Database 124  
    **create route** 83  
    limitations 124  
    routing 83  
empty transactions 124  
encrypted passwords  
    extended support 42  
    sending 128  
enhanced text update 49  
enhancements  
    configuration and tuning 57  
    dump transaction 47  
    dynamic SQL 33, 53  
    for Replication Server performance 2, 60  
    function replication 35, 54  
    locales directory 41  
    **log\_first\_tran** 40  
    monitor and counter 36, 58  
    non-ASE error class support 30–32  
    non-ASE replicate support 26–30  
    non-blocking commit 6–9  
    quoted identifiers 9–15  
    release area 41  
    Replication Manager plug-in 74  
    Replication Server gateway 15–17  
    resume connection 39  
    rs\_subcmp 63  
    SQL statement replication 18–25  
    SQM performance 50  
    stable queue management 38  
    sysadmin dump\_queue 38  
    sysadmin sqt\_dump\_queue 39  
ERSSD

See Embedded Replication Server System Database 111

Event Log pane 95

- displaying or hiding 95
- viewing events 95

event triggers

- adding in Replication Manager 119
- creating in RMS 119

exceptions log

- troubleshooting 100
- viewing 100

## F

function replication

- enhancement 35

function string

- rs\_dsi\_check\_thread\_lock 125
- rs\_set\_dml\_on\_computed 79
- rs\_set\_non\_blocking\_commit 7
- rs\_set\_non\_blocking\_commit\_flush 7
- rs\_set\_quoted\_identifier 11

## H

hash algorithm

- command line parameters 64
- configuration file parameters 64

## I

internal commit control 125

isolation level

- kinds of 88
- setting 88

## J

Java JRE

- compatibility with Replication Monitoring Services 111

jdbcConnect for JDBC

- compatibility with Replication Monitoring Services

## L

license management

- with SySAM 90

limitations

- dynamic SQL 34
- function replication 35
- LOB datatypes 45
- opaque datatype 46
- partial update 45

list

- details 95

LOB datatypes

- partial update 45
- support 54

locales directory

- changes 41

log

- event 95

longer identifiers 75

## M

manual data reconciliation 68

marking identifiers as quoted 10–11

master database

- supported DDL commands 62

message logging

- disabling 96
- enabling 96

mixed-version

- enhanced support for 85

monitoring of status 95

monitoring replication environment

- for three-tier environment 109
- using heartbeat and latency information 119
- using Replication Manager graphical interface 113
- using RMS 113
- using RMS API 113

monitoring replication environment using RMS 113

monitoring status

- in Details list 95

## Index

- replication objects 95
- visual display 95
- mount command 129
- multiprocessors
  - enabling 123
  - monitoring 123
- multisite availability (MSA) 121
- multithreaded architecture 123

## N

- new features
  - Replication Manager 15.0 93
  - Replication Manager 15.0.1 71
  - Replication Manager 15.1 53
  - Replication Server 15.0 75
  - Replication Server 15.0.1 57
  - Replication Server 15.1 33
  - Replication Server 15.2 1
- non-ASE error class support 30–32
  - altering error classes 31
  - creating error classes 31
  - default non-ASE error classes 30
  - native error codes 32
- non-ASE replicate support 26–30
  - connection profiles 27
  - listing connection profiles 29
  - simplified installation, configuration 26
  - using connection profiles 27
- non-blocking commit 6–9
  - ASE delayed commit feature 6
  - configuring 6
  - dsi\_non\_blocking\_commit** 6
  - non-ASE databases, support for 8
  - Oracle, support for 8
  - rs\_set\_non\_blocking\_commit** 7
  - rs\_set\_non\_blocking\_commit\_flush** 7
  - system functions for 7

## O

- online help
  - in Replication Manager plug-in 94
  - invoking 95

- opaque datatypes 55

## P

- parallel processing 123
  - empty transactions 124
  - internal commit control 125
- password encryption
  - Advanced Encryption Standard algorithm 84
  - extended support 42
  - FIPS-certified 84
  - for maintenance user passwords 85
  - for route user passwords 85
  - for user passwords 85
  - for user passwords in configuration file 85
- POSIX thread library 123

## Q

- queue data
  - troubleshooting 101
  - viewing 101
- quiesce database command 129
- quoted identifiers 9–15
  - alter replication definition**, changes to 10
  - create replication definition**, changes to 10
  - dsi\_quoted\_identifiers** 9
  - embedded double quote characters 9
  - enabling in DSI 9
  - forwarding to data servers 11
  - marking identifiers as quoted 10–11
  - product version requirements 15
  - quoted** parameter 10
  - rs\_helprep** changes 12–14
  - rs\_set\_quoted\_identifier** 11

## R

- RCL command
  - admin stats 59
  - admin stats, status 60
- replicating
  - data, large batch of 2

- database objects 122
  - encrypted columns 80
  - master database 62
  - partitioned tables 80
  - to standby databases 122
  - Replication features
    - supported by Replication Manager 98
  - Replication Manager
    - adding event triggers 119
    - new features 53
    - new features 15.0 93
    - new features 15.0.1 71
    - support of Replication Server/Replication features 98
    - using to set configuration parameters 115
  - Replication Monitoring Services
    - accessing using Replication Manager or API 113
    - adding servers 113
    - compatibilities 111
    - connecting to 112
    - description 109
    - dropping servers 114
    - in three-tier management solution 109
    - installing 111
    - monitoring logical group of servers 116
    - monitoring replication environment 113
    - setting configuration parameters 115
    - shutting down monitored servers 118
    - software requirements 111
    - starting and stopping 111
    - suspending/resuming components 117
    - using Unified Agent Framework 111
    - viewing monitored objects 115
  - Replication Server
    - communication through Replication Manager 94
    - configuring using Replication Manager 115
    - dynamic configuration 57
    - new features 15.2 1
  - Replication Server features
    - supported by Replication Manager 98
  - Replication Server gateway 15–17
    - cascading connection 15
    - connection dropping 17
    - connections, tracking 17
    - enable, to 16
    - limitations 17
    - product version requirements 17
  - Replication Server System Database (RSSD) 124
  - RMS
    - See* Replication Monitoring Services
  - routes
    - upgrading 99
  - row count verification, in SQL statement replication 23
  - rs\_dsi\_check\_thread\_lock function string 125
  - rs\_helprep** 12–14
  - rs\_subcmp
    - parameters for manual reconciliation 68
    - schema sub-types 66
    - schema types 66
    - string options 65
  - rs\_ticket
    - version 2 42
- ## S
- sampling data
    - optionally keeping in RSSD 60
  - schema comparison 65
    - command line parameters 67
    - configuration file parameters 67
  - segment preallocation 52
  - send\_enc\_password configuration parameter 128
  - setting configuration parameters
    - using Replication Manager 115
  - software compatibilities of Replication Monitoring Services 111
  - software requirements for Replication Monitoring Services 111
  - SQL statement replication 18–25
    - autocorrection 23
    - configuring warm standby 24
    - database level 18
    - database replication definition 20
    - product and mixed version requirements 25
    - replicate SQLDML clause 20
    - restrictions 23
    - row count verification 23
    - RSSD modifications 25
    - session level 20
    - set repmode 20

## Index

- sp\_setrepdbmode 18
- sp\_setrepdefmode 19
- table level 19
- table replication definition 22
- WS\_SQLDML\_REPLICATION parameter 25
- stable queue caching 50
- starting and stopping RMS 111
- stats\_reset\_rssd 60
- status monitoring using RMS 95
- subscription materialization 3
- support
  - Adaptive Server integer identity 50
  - bulk copy-in. *See* bulk copy-in support
  - contacting Sybase Technical Support xiv
  - direct I/O 52
  - dynamic configuration 71
  - for Adaptive Server shared-disk cluster 36
  - for DirectConnect 98
  - for heterogeneous data servers 72
  - for non-Sybase data servers 72
  - for Replication Agents and Mirror Replication Agents 73
  - LOB datatypes 44, 54
  - opaque datatypes 55
  - timestamp datatypes 46, 56
- symmetric multiprocessors (SMP) 123
- syntax conventions xiii

## T

- threads
  - in Replication Server 107
  - using context menu 107
  - viewing details 108
- three-tier management solution 109
- time, database 127
- timestamp datatypes 56
- transactions
  - empty 124
- two-tier management solution 93, 94

## U

- unsigned datatypes 77

- unsigned bigint 77
- unsigned int 77
- unsigned smallint 78
- unsigned tinyint 78
- user documentation, for Replication Server ix
- user interface features, Replication Manager 93

## V

- visual monitoring of status 95

## W

- warm standby
  - adding active database connection 105
  - adding logical connection 104
  - adding standby database connection 106
  - filtering connection status 103
  - wizards in 104
- WIN32 thread library 123
- wizards
  - Add Active Database 105
  - Add Logical Connection 104
  - Add Standby Database 106