SYBASE®

An SAP® Company

Release Bulletin

Replication Server® 15.6

Windows

DOCUMENT ID: DC72890-01-1560-02

LAST REVISED: July 2011

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

Contents

Electronic Bug Fix (EBF) Downloads for Security Fixes

Due to possible security vulnerabilities, Sybase® strongly recommends that you use the appropriate EBF to fix these issues.

For instructions, see *Urgent Customer Notification Web Page*.

Known issues described in this release bulletin may have been fixed in other EBFs released for this product.Get EBFs and maintenance reports describing latest fixes from the *Sybase Support Page*.

Product Summary

This release bulletin provides late-breaking information about Replication Server® version 15.6. A more recent version may be available on the Web.

Supported Platforms and Operating Systems

Check the system requirements and system patches for the server on which you will install Replication Server.

Microsoft Windows (32-bit and 64-bit):

- Windows XP Professional – Service Pack 2 or later
- Windows 2003 R2 – Service Pack 2 or later
- Windows Vista
  - Vista Business – Service Pack 1
  - Vista Enterprise – Service Pack 1
- Windows Server 2008 R2
- Windows 7

You must install the relevant Microsoft Visual Studio 2005 patch for your x86 or x64 Windows platform before you start Replication Server. See the Microsoft Visual C++ 2005 Service Pack 1 Redistributable Package ATL Security Update at *http://www.microsoft.com/downloads/details.aspx?familyid=766a6af7-ec73-40ff-b072-9112bab119c2&displaylang=en*.

For a complete list of supported operating systems, see the Sybase® Platform Certification Web site at *http://certification.sybase.com/ucr/search.do*.

Internet Protocol Version 6

Operating systems and versions that support Internet Protocol version 6 (IPv6).

- Windows XP Professional – Service Pack 2 or later
- Windows 2003 R2 – Service Pack 2 or later
- Windows Vista
  - Vista Business – Service Pack 1
  - Vista Enterprise – Service Pack 1
- Windows Server 2008 R2
- Windows 7

Replication Server

Replication Server coordinates data replication activities for local databases and exchanges data with Replication Servers that manage data at other sites.

For detailed information about new features in Replication Server 15.6, see the *Replication Server New Features Guide*.

Replication Manager

The Replication Manager (RM) is a utility for creating, managing, and monitoring replication environments, and is available as a plug-in to Sybase Central™.

Replication Manager is not certified for use with Adaptive Server® Enterprise Cluster Edition.

For detailed information about Replication Manager, see the *Replication Server Administration Guide Volume 1*.

For information about commands used to manage replication, see the *Replication Server Reference Manual*.

Replication Monitoring Services

Replication Monitoring Services (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 is not certified for use with Adaptive Server Enterprise Cluster Edition.

For detailed information about RMS, see the *Replication Server Administration Guide Volume 1*.

ExpressConnect for Oracle

ExpressConnect for Oracle is an embedded library loaded by Replication Server for Oracle replication.

ExpressConnect for Oracle (ECO), which is available with Replication Server Options 15.5 and later, provides direct communication between Replication Server and a replicate Oracle data server. ECO eliminates the need for installing and setting up a separate gateway server, thereby improving performance and reducing the complexities of managing a replication system.

To use ECO, make sure:

- Replication Server is installed using the REP_EC_ORA license.
- The version of ECO installed is ECO 15.5 ESD #1. Replication Server 15.6 does not work with ECO 15.5.

See the *ExpressConnect for Oracle Installation and Configuration Guide* in Replication Server Options 15.5 product documentation.

Separately Licensed Products

Obtain a separate license for each Replication Server Options component.

ExpressConnect for Oracle, Replication Agent, and Enterprise Connect Data Access

ExpressConnect for Oracle, Replication Agent™, and Enterprise Connect™ Data Access are available as a product called Replication Server Options that are available separately from Replication Server. You must have Replication Server to obtain Replication Server Options.

Replication Server Options provide bidirectional replication across distributed, heterogeneous systems. You can use the Replication Server Option components to implement replication on the Microsoft Windows and UNIX platforms.

Replication Server Options are available in three data-source-specific versions: Microsoft SQL Server, Oracle, and IBM DB2 UDB. See the *Release Bulletin for Replication Server Options 15.5 for Linux, Microsoft Windows, and UNIX* for details about supported platforms and compatible Replication Server versions.

Replication Server Data Assurance Option

Replication Server® Data Assurance (DA) Option 15.6 is available as a separately licensed product for Replication Server and supports Replication Server versions 15.1 and later.

Replication Server DA Option compares row data and schema between two or more Adaptive Server® databases, and reports and optionally reconciles, discrepancies.

Replication Server Data Assurance Option is licensed through SySAM license manager and is available on multiple platforms. See Replication Server Data Assurance Option documentation for additional information.

Replication Server Product Editions and Licenses

Replication Server 15.6 is available as two separate product editions—Enterprise Edition (EE) and Real-Time Loading Edition (RTLE)—that bundle different base and optional features, and which require separate licences.

RTLE enables real-time loading in order to replicate from Adaptive Server® or Oracle to Sybase® IQ. Replication from Oracle requires Replication Agent for Oracle (RAO). Though real-time loading is supported on all platforms that Replication Server supports, RAO is not available on Linux on POWER nor on Sun Solaris x64. However, you can run RAO on any of the platforms it supports. For platforms on which RAO is available, see the *Installation Guide for Replication Agent* as a part of the RSO documentation.

The documentation for RTLE includes Replication Server Options product documentation in addition to Replication Server product documentation.

See the *Replication Server New Features Guide*.

To purchase licensed options, contact your Sybase sales representative.

Product Compatibility

Learn about Replication Server compatibility with database servers, drivers, and products.

Adaptive Server

Learn about Replication Server compatibility with Adaptive Server platforms and other products.

Replication Server version 15.6 is fully compatible with both 32-bit and 64-bit versions of Adaptive Server Enterprise version 15.0 and later and Adaptive Server Enterprise version 12.5.4 on Windows.

A replication system can include Adaptive Servers, Replication Servers, DirectConnect™ products, and RepAgents on various operating systems.

Note: SQL Server versions 11.0.*x* and Adaptive Server 12.5 and earlier are no longer supported.

Replication Server Interoperability

List of products and platforms that are interoperable with Replication Server.

Table 1. Interoperability Between Adaptive Server, Open Client/Server, Replication Server, and Sybase IQ

Product Name	Version	Supported Platforms
Adaptive Server	15.5 ESD #2 15.5 ESD #1 15.5 15.0.*x*	• HP-UX Itanium 64-bit • IBM AIX POWER 64-bit • Linux x86 32-bit • Linux x86-64 64-bit • Linux on POWER 64-bit • Sun Solaris SPARC 64-bit • Sun Solaris x86-64 64-bit • Microsoft Windows x86 32-bit • Microsoft Windows x86-64 64-bit
	12.5.4	• HP-UX Itanium 64-bit • IBM AIX POWER 64-bit • Linux x86 32-bit • Linux x86-64 64-bit • Linux on POWER 64-bit • Sun Solaris SPARC 64-bit • Sun Solaris x86 64-bit • Microsoft Windows x86 32-bit
Open Client/ Server™	15.5 15.0	• HP-UX Itanium 64-bit • IBM AIX POWER 64-bit • Linux x86 32-bit • Linux x86-64 64-bit • Linux on POWER 32-bit • Linux on POWER 64-bit • Sun Solaris SPARC 64-bit • Sun Solaris x86-64 64-bit • Microsoft Windows x86 32-bit • Microsoft Windows x86-64 64-bit

Product Name	Version	Supported Platforms
	12.5.1	• HP-UX Itanium 64-bit • IBM AIX POWER 64-bit • Linux x86 32-bit • Linux x86-64 64-bit • Linux on POWER 32-bit • Linux on POWER 64-bit • Sun Solaris SPARC 64-bit • Sun Solaris x86-64 64-bit • Microsoft Windows x86 32-bit
Replication Server	15.6 15.5	• HP-UX Itanium 64-bit • IBM AIX POWER 64-bit • Linux x86 32-bit • Linux x86-64 64-bit • Linux on POWER 64-bit • Sun Solaris SPARC 64-bit • Sun Solaris x86-64 64-bit • Microsoft Windows x86 32-bit • Microsoft Windows x86-64 64-bit
	15.2 15.1	• HP-UX Itanium 64-bit • IBM AIX POWER 32-bit/64-bit • Linux x86 32-bit/64-bit • Linux on POWER 64-bit • Sun Solaris SPARC 32-bit/64-bit • Microsoft Windows x86 32-bit/64-bit
	15.0.1	• Linux x86 32-bit • Microsoft Windows x86 32-bit
	12.6	• Linux x86 32-bit • Microsoft Windows x86 32-bit
Sybase IQ	15.2 15.1	• HP-UX Itanium 64-bit • IBM AIX POWER 64-bit • Linux x86 32-bit • Linux x86-64 64-bit • Linux on POWER 64-bit • Sun Solaris SPARC 64-bit • Sun Solaris x86-64 64-bit • Microsoft Windows x86 32-bit • Microsoft Windows x86-64 64-bit
	12.7	• HP-UX Itanium 64-bit • IBM AIX POWER 64-bit • Linux x86 32-bit • Linux x86-64 64-bit • Linux on POWER 64-bit • Sun Solaris SPARC 64-bit • Sun Solaris x86-64 64-bit • Microsoft Windows x86 32-bit • Microsoft Windows x86-64 64-bit

Replication Server is available as either a 32-bit application or a 64-bit application. The 32-bit versions of Replication Server on Linux and Windows have been certified on both the 32-bit and 64-bit versions of Linux and Windows operating systems. The 64-bit versions of Replication Server on Linux and Windows are not certified on the 32-bit version of Linux and Windows operating systems.

Even though two or more products may be interoperable, features introduced in a newer version of a product are not likely to be supported by older versions of the same products.

Installation and Upgrade

Get last-minute information about installation and upgrading that was omitted from or incorrect in your installation guide, or that needs special emphasis.

For detailed information about installing and upgrading, see the *Replication Server Installation Guide*.

Special Installation Instructions

Learn about the additional installation requirements for Replication Server.

Requirements for Using SySAM

Read about the installation requirements you need to know in order to use SySAM.

SySAM Sub-Capacity Licensing

When using SySAM sub-capacity licensing, you need to set an environment variable before installation or you can copy in the license key after installation.

If you are using sub-capacity licensing, do one of:

- Set the SYBASE_SAM_CAPACITY environment variable before starting your installer. Follow the instructions provided in the *SySAM Users Guide > Configuring SySAM Sub-capacity*. However, start your installer instead of starting the sub-capacity-enabled Sybase product. The installer includes the **sysamcap** utility located in `sysam_utilities/bin` .
- Select the **Continue installation without a license key** in the Sybase Software Asset Management License pane during installation. After installation, copy your license key to *`installed_directory`*`/SYSAM-2_0/licenses` directory where *`installed_directory`* is the location in which you have installed your components into.

Coexisting with Older Versions of SySAM

Replication Server 15.6 uses SySAM 2. You may use an earlier version of SySAM, but it must be modified.

Replication Server version 15.6 uses a newer version of Sybase Software Asset Management System (SySAM) and an updated license format. You can run only one instance of a license server on a computer. To use earlier versions of Sybase products with Replication Server 15.6, see *SySAM Users Guide for details.*

For instructions on migrating a license server, see the *SySAM Users Guide*.

Special Upgrade and Downgrade Instructions

Learn about special upgrade and downgrade instructions for Replication Server.

Replication Server Configuration Guide for Windows contains detailed upgrade and downgrade instructions. Sybase strongly recommends that you read this information before you upgrade or downgrade Replication Server.

Warning! Before upgrading the user database to support Replication Server 15.5 and later, upgrade Adaptive Server Enterprise to version 12.5 or later. Otherwise, the upgrade fails.

Upgrading to Replication Server Version 15.6 from Version 15.5 Using Existing Directories

Upgrade from version 15.5 to version 15.6 using the existing version 15.5 directories. You need not upgrade the RSSD, ERSSD, and user databases.

1. Shut down Replication Server 15.5.

   See *Replication Server Configuration Guide for Windows > Start or Stop a Replication Server > Stopping a Replication Server Using isql.*
2. Back up the entire set of files and directories under `%SYBASE%`.

   Note: Keep this backup in case you need to downgrade from Replication Server version 15.6 to version 15.5.
3. Use the *Replication Server 15.6 Installation Guide for Windows* to install Replication Server 15.6 into the existing version 15.5 directories under `%SYBASE%`.

   Note: Do not install the SAMPLE_RS sample Replication Server from version 15.6 if you are using the SAMPLE_RS from version 15.5.

   The installation process:

   - Updates all subdirectories of `%SYBASE%` that share directory names in both version 15.5 and 15.6

- Backs up existing `SYBASE.*` files, where * can be `.env` or `.bat`, by adding a "save" suffix to the file. For example, `Sybase.env` is backed up as `Sybase.env.save`

4. Copy the files that you backed up in step 2 such as, the `sql.ini` file, resource files, `.cfg` files, the `run_`*`repservername`* runserver files, and ERSSD files, to the same directories which now contain Replication Server 15.6 files.

   Copy any other files that were added or changed after installation to the corresponding directories. If you use an ERSSD, copy the `dbfile`, `translog`, `errorlog`, and `backup` directories to the corresponding locations. If you use SAMPLE_RS, copy all files and directories under `%SYBASE%\REP-15_5\samp_repserver` to the corresponding locations.
5. Set the relevant environment variables in the `SYBASE.env` file in the Replication Server 15.6 installation directory.

   See *Replication Server Installation Guide for Windows > Postinstallation Tasks > Environment Variables on Windows*.
6. Change the runserver file you copied from the backup to use the Replication Server 15.6 executable, and then start Replication Server.

   See *Replication Server Configuration Guide for Windows > Start or Stop a Replication Server > Runserver File*.

Downgrading to Version 15.5 from a Previous Upgrade to Version 15.5 Directories

You can revert to Replication Server version 15.5 from version 15.6 if you backed up your version 15.5 installation before upgrading to the existing version 15.5 directories.

1. Shut down Replication Server 15.6.

   See *Replication Server Configuration Guide for Windows > Start or Stop a Replication Server > Stopping a Replication Server Using isql*.
2. Back up the entire set of files and directories under `%SYBASE%`.

   These contain the version 15.6 files.
3. Restore the Replication Server 15.5 files and directories, from the backup you made beforen you upgraded from version 15.5 to version 15.6, to overwrite the directory structure under `%SYBASE%`
4. Verify that the `.cfg`, `sql.ini`, and `run_`*`repservername`* runserver files are configured to run Replication Server 15.5.
5. If there was replication activity on the upgraded Replication Server prior to the downgrade process, copy the ERSSD from the back up of version 15.6 files in step 2, to the corresponding directories.

   If you use an ERSSD, copy the `dbfile`, `translog`, `errorlog`, and `backup` directories to the corresponding locations . If you use SAMPLE_RS, copy all files and directories under `%SYBASE%\REP-15_5\samp_repserver` to the corresponding locations.

6. Set the relevant environment variables in the `SYBASE.env` file.

   See *Replication Server Installation Guide for Windows > Postinstallation Tasks > Environment Variables on Windows*.
7. Start Replication Server 15.5 with the runserver file.

   See *Replication Server Configuration Guide for Windows > Start or Stop a Replication Server > Runserver File*.

Upgrading to Replication Server Version 15.6 from Version 15.5 Using Unique Directories

Upgrade from version 15.5 to unique directories that do not contain version 15.5 installation files. You do not need to upgrade the RSSD, ERSSD, and user databases.

1. Shut down Replication Server 15.5.

   See *Replication Server Configuration Guide for Windows > Start or Stop a Replication Server > Stopping a Replication Server Using isql*.
2. Use the *Replication Server 15.6 Installation Guide for Windows* to install Replication Server 15.6 into a unique directory.
3. Set the relevant environment variables in the `SYBASE.env` file in the Replication Server 15.6 installation directory.

   See *Replication Server Installation Guide for Windows > Postinstallation Tasks > Environment Variables on Windows*.
4. Modify the instructions in the runserver file to use the Replication Server 15.6 executable.

   See *Replication Server Configuration Guide for Windows > Start or Stop a Replication Server > Runserver File*.
5. Start Replication Server 15.6 with the runserver file you modified.

   See *Replication Server Configuration Guide for Windows > Start or Stop a Replication Server > Runserver File*.

Downgrading to Version 15.5 from a Previous Upgrade to Unique Directories

You can revert to Replication Server version 15.5 from version 15.6 if you previously upgraded to version 15.6 using unique directories.

1. Shut down Replication Server 15.6.

   See *Replication Server Configuration Guide for Windows > Start or Stop a Replication Server > Stopping a Replication Server Using isql*.
2. Set the relevant environment variables in the `SYBASE.env` in the `REP-15_5` directory to the appropriate values for Replication Server 15.5.

See *Replication Server Installation Guide for Windows > Postinstallation Tasks > Environment Variables on Windows*.

3. Modify the instructions in the runserver file to use the Replication Server 15.5 executable.

   See *Replication Server Configuration Guide for Windows > Start or Stop a Replication Server > Runserver File*.

4. Start Replication Server 15.5 with the runserver file you modified.

   See *Replication Server Configuration Guide for Windows > Start or Stop a Replication Server > Runserver File*.

ExpressConnect for Oracle Upgrade Requirement

Replication Server 15.6 does not work with ECO 15.5. When upgrading to Replication Server 15.6, install ECO 15.5 ESD #1 or a more recent version.

See the *ExpressConnect for Oracle Installation and Configuration Guide*.

Known Issues

Learn about known issues and workarounds for Replication Server.

Find issues by Change Request (CR) number.

Note: You can search the Sybase Web site for solved cases. Choose **Support > Solved Cases** or go to *http://search.sybase.com/search/simple.do?mode=sc*. You need a MySybase account to view solved cases in the archive.

Known Issues for Replication Server

Learn about known issues and apply workarounds for Replication Server.

Table 2. Replication Server Issues

CR#	Description
677341	**Changes to the LOB data are not always replicated when replicating columns of LOB types to Oracle using ExpressConnect for Oracle (ECO) in bulk copy mode**. **Workaround**: Configure the connection using ECO with **dsi_bulk_copy** set to off.

CR#	Description

653626	**DSI may shutdown when creating connections to:**
	• **Oracle using profile rs_ase_to_oracle**
	• **DB2 using profile rs_ase_to_db2**
	• **UDB using profile rs_ase_to_udb**
	At the time of shutting down, Replication Server generates an error message. You see:
	`Message from server: Message: 2601, State 6, Severity 14 --`
	`'Attempt to insert duplicate key row in object 'rs_translation' with unique index 'rs_key_translation'`
	The `rs_translation` system table has unique index on (`classid`, `source_dtid`). The two translation (ASE binary to *rs_oracle_binary* and ASE timestamp to *rs_oracle_binary*) generates the duplicate key insert error.
	Workaround: For Replication Server versions 15.5, 15.5 ESD #1, and 15.6, you have to manually remove the translations when you encounter this error. Open the SQL scripts and search for `timestamp` to find the related SQL statements.
	For ase_to_oracle:
	`delete from rs_profdetail where profid = 0x000000000e000000`
	`and pdetailid = 0x000000000e00003f`
	`delete from rs_systext where parentid = 0x000000000e00003f`
	`and sequence = 1`
	`delete from rs_profdetail where profid = 0x000000000e000000`
	`and pdetailid = 0x000000000e000040`
	`delete from rs_systext where parentid = 0x000000000e000040`
	`and sequence = 1`
	For ase_to_oracle_eco:
	`delete from rs_profdetail where profid = 0x000000000e010000`
	`and pdetailid = 0x000000000e010042`
	`delete from rs_systext where parentid = 0x000000000e010042`
	`and sequence = 1`
	`delete from rs_profdetail where profid = 0x000000000e010000`
	`and pdetailid = 0x000000000e010043`
	`delete from rs_systext where parentid = 0x000000000e010043`
	`and sequence = 1`
	For ase_to_oracle_ecda:
	`delete from rs_profdetail where profid = 0x000000000e020000`
	`and pdetailid = 0x000000000e02003f`
	`delete from rs_systext where parentid = 0x000000000e02003f`
	`and sequence = 1`
	`delete from rs_profdetail where profid = 0x000000000e020000`
	`and pdetailid = 0x000000000e020040`
	`delete from rs_systext where parentid = 0x000000000e020040`
	`and sequence = 1`
	For ase_to_udb:

CR#	Description

	`delete from rs_profdetail where profid = 0x000000000e00000c`
	`and pdetailid = 0x000000000e000c27`
	`delete from rs_systext where parentid = 0x000000000e000c27`
	`and sequence = 1`
	`delete from rs_profdetail where profid = 0x000000000e00000c`
	`and pdetailid = 0x000000000e000c28`
	`delete from rs_systext where parentid = 0x000000000e000c28`
	`and sequence = 1`
	For ase_to_db2:
	`delete from rs_profdetail where profid = 0x000000000e000012`
	`and pdetailid = 0x000000000e001231`
	`delete from rs_systext where parentid = 0x000000000e001231`
	`and sequence = 1`
	`delete from rs_profdetail where profid = 0x000000000e000012`
	`and pdetailid = 0x000000000e001232`
	`delete from rs_systext where parentid = 0x000000000e001232`
	`and sequence = 1`

643174	**`timestamp` columns are replicated in a warm standby and multisite availability (MSA) setup.** `timestamp` columns are sent to the replicate database even if High Volume Adaptive Replication (HVAR) is set to on, **send_timestamp_to_standby** is set to off, and there is no replication definition defined. **Workaround**: Set HVAR to off, or **create replication definition** without the `timestamp` columns.

625227	**SPDC-generated served partition-level license may not work.**
	When you use a served partition-level license generated from the Sybase Product Download Center to start a SySAM server, you see a message that you are using an invalid license key, similar to:
	`(SYBASE) Invalid license key (inconsistent authentication code)`
	`(SYBASE) License server system started on hpiabou`
	`(SYBASE) No features to serve, exiting`
	`(SYBASE) EXITING DUE TO SIGNAL 49 Exit reason 4`
	`(lmgrd) SYBASE exited with status 49 (No features to serve)`
	`(lmgrd) SYBASE daemon found no features.  Please correct`
	`(lmgrd) license file and re-start daemons.`
	`(lmgrd)`
	`(lmgrd) This may be due to the fact that you are using`
	`(lmgrd) a different license file from the one you expect.`
	`(lmgrd) Check to make sure that:`
	`(lmgrd) /remote/cat_fc/nli/iq152.hpia/SYSAM-2_0/licenses/18965_hpiabou_`
	`(lmgrd) is the license file you want to use.`
	Workaround: Use the old host IDs for the license running on the license server.

CR#	Description
620380	**Use rs_init to configure Replication Server with an existing RSSD.** When configuring Replication Server 15.5 or later using the **rs_init** with an existing RSSD, an error occurs in **rs_init**. **Workaround**: Drop the RSSD before configuring the Replication Server.
618624	**Insufficient number of threads in Replication Server.** If you increase the number of client connections and did not increase the number of Open Server™ threads that the Replication Server can use, Replication Server may shut down. **Workaround**: **1.** Log in to RSSD. **2.** Increase the value of **num_threads**. **3.** Restart Replication Server.

616941	**Stack trace error at start-up when sort order is not in the `objectid.dat` file.**
	If the sort order is not set correctly in the [*collate*] section of `%SYBASE%\ini\objectid.dat`, a stack trace error occurs during Replication Server start-up.
	Workaround:
	1. Make sure that the character set and set order have been set correctly in `Rep_Server_name.cfg` file. For example:
	• RS_charset=cp850
	• RS_sortorder=scannocp
	2. Go to `%SYBASE%\charsets\`*`character set`*`;` for example, `%SYBASE%\charsets\cp850.`
	3. Locate and open corresponding sort order file; for example, `scannocp.srt.`
	4. Check the line that has an "id" and find the sort order ID, for example,
	`id = 0x30; Unique ID # (48) for the sort order`
	The sort order ID is 48.
	5. In `%SYBASE%\ini\objectid.dat`, check that there is a line for that sort order under the [*collate*] section:
	`1.3.6.1.4.1.897.4.9.3.48 = scannocp`
	The last number is the sort order ID (48), all other numbers are identical for all lines. If you have a custom sort order, add this line for it.
	6. Restart Replication Server.

616158	**Differing character sets on Replication Server and Adaptive Server.** If Replication Server uses the utf8 character set and the target Adaptive Server uses the gb18030 character set, an error occurs if a table has `text` and `image` column with **dsi_bulk_copy** or **dsi_compile_enable** parameter set to on. The DSI shuts down. **Workaround**: Set the **dsi_bulk_copy** and **dsi_compile_enable** parameters to off.

CR#	Description
614717	**Issues with parallel_dsi parameter** When you change the value of the **parallel_dsi** parameter using the **alter connection** command or **configure replication server** command, Replication Server changes the existing configuration values of these parameters: • **dsi_num_threads** • **dsi_num_large_xact_threads** • **dsi_serialization_method** • **dsi_sqt_max_cache_size** This may impact your replication performance. **Workaround**: Set the **parallel_dsi** parameter first, and then manually set the desired values for the above-mentioned parameters for your environment if you do not want to use the default values.
607273	**When the RSSD is loaded in an Adaptive Server 12.5.4 *x* server, rs_helprep fails when the *repdef_name* is greater than 30 characters.** **Workaround**: • Use single quote (‘) or double quote(“) around the long *repdef_name*. • Use a truncated *repdef_name*. Truncate the *repdef_name* to the first 29 characters when issuing the **rs_helprep** request. The **rs_helprep** then appends the % wildcard at the end when it queries the RSSD tables. • Upgrade RSSD data server to Adaptive Server 15.*x*.
571435	**During subscription materialization, using quoted identifiers with a custom function string that includes a quoted constant causes a query failure. The replicate data server identifies the quoted constant as a column instead of a constant.** **Workaround**: Create the subscription without a quoted constant or create the subscription without materialization.
558253	**Creating replication definition fails when using gb18030 character set and the ERSSD.** On Windows, when you are creating the replication definition table using **create replication definition**, the gb18030 character set does not work correctly if you are: • Using the ERSSD, and • Using multibyte characters in the replication definition name or the table name. **Workaround**: Set the environment variable SACHARSET to GBK before starting the Replication Server.

CR#	Description
452806	**An application deadlock involving Replication Server and Adaptive Server may occur when Replication Server is configured to use parallel DSI while applying transactions to a table containing `text` and `image` columns.** **Workaround**: Suspend and resume the DSI connection. If the DSI thread does not suspend, restart the Replication Server.

Known Issues for Replication Agent

Learn about known issues and apply workarounds for Replication Agent.

Table 3. Replication Agent Issues

CR#	Description

596321,
596320
	Replication Server does not support specifying owner information of stored procedures in the function replication definition for Oracle.
	Therefore, Replication Agent cannot send that information to Replication Server. Because the owner information is not available, the replicated stored procedure fails to execute at the Oracle standby database.
	Workaround: For each stored procedure that is replicated from an active to a standby database, create a corresponding function string. Specify owner information in the target stored procedure.
	To customize the function-string class of the standby connection, which is inherited from **rs_oracle_function_class**, enter:
	`alter connection to dco2stb.ordb (standby connection)`
	`set function string class to my_oracle_function_class`
	`go`

Known Issues for Sybase Central

Learn about known issues and apply workarounds for Sybase Central.

Table 4. Sybase Central Issue

CR#	Description
343973	**Users must have write privileges on home directory** Sybase Central writes a registry file in the user's home directory, and uses this registry file to store environment information from the Replication Manager. You must have write privileges on the user's home directory, or Sybase Central cannot save the environment information. **Workaround**: Ensure that each user has write privileges on the user's home directory.

Known Issues for Sybase Control Center

Learn about known issues and apply workarounds for Sybase Control Center (SCC).

Table 5. Sybase Control Center Issue

CR#	Description
642275	**Sybase Control Center (SCC) 3.1 and earlier versions do not support Replication Server 15.6.** **Workaround**: Upgrade to 3.1.1 or later.

Known Issues for Replication Manager

Learn about known issues and apply workarounds for Replication Manager.

Table 6. Replication Manager Issues

CR#	Description
619049	**Simultaneously switching multiple warm standby pairs causes Replication Manager plug-in to fail.** In a Replication Manager plug-in environment that contains multiple warm standby pairs, if you select multiple logical connections and choose **switch active**, all warm standby pairs attempt to switch simultaneously. You see: `“Internal Error - Sybase Central”` **Workaround**: Switch each warm standby pair individually. Wait until the **switch active** operation is complete on the first pair before you issue **switch active** on the next pair.
616057	**The Replication Manager plug-in does not support configuration of the new block_size parameter. Configuring the parameter generates:** `“Option WITH SHUTDOWN required for block size change”` **Workaround**: None.
611031	**Java does not support the roman8 character set; therefore, you cannot select this character set in Replication Manager.** **Workaround**: When adding a server to an environment: • Enter a user name and password that will let you access the environment. • Do not select roman8 in the Character Set field for a server connection. • Do not select “default” in the Character Set field for a server connection if the server default is roman8. Select another character set such as iso_1, utf8, or cp850, that is compatible with the server and Replication Manager.
606691	**In a three-tier replication environment, the Replication Manager plug-in does not display the newer Replication Server parameters.** **Workaround**: • Add the replication environment in a normal two-tier Replication Manager environment. • Add the servers for which you want to access the configuration parameters in this new environment.

Known Issues for RMS

Learn about known issues and apply workarounds for RMS.

Table 7. RMS Issue

CR#	Description
616051	**RMS does not support dynamic configuration of parameters from Replication Server.** Executing the **configure server** command or the **configure component** command may not return all parameters from a Replication Server. The new parameters are also not displayed correctly in the Replication Manager plug-in when using a three-tier environment. **Workaround**: When configuring a Replication Server or its components, use a two-tier environment in the Replication Manager plug-in to connect and configure.

Known Issues for Replicating to Sybase IQ

Learn about known issues and apply workarounds when replicating to Sybase IQ from Adaptive Server using Real-Time Loading (RTL) in Replication Server.

Table 8. Replicating to Sybase IQ Issues from Adaptive Server Using RTL

CR#	Description
643578	**Automatic materialization to Sybase IQ fails.** If you use atomic and nonatomic materialization methods, materialization of tables to Sybase IQ using RTL fails for multiple datatypes. **Workaround**: Either: • Use the no materialization or the bulk materialization methods when replicating to Sybase IQ using RTL • Upgrade to Replication Server 15.6 ESD #1
642212	**`rssetup.sql` is missing from the scripts directory within the Sybase IQ installation directory on Sun Solaris AMD x64 platforms.** **Workaround**: Obtain this script from another platform or contact Technical Support.

CR#	Description

641373	**Lock table failure when replicating in Sybase IQ multiplex environment.**
	Because Replication Server connects and issues the **lock table** command, you must make the connection from the Replication Server to Sybase IQ in a multiplex environment to the coordinator node. Otherwise, you see:
	`E. 2010/09/14 08:51:13. ERROR #1028 DSI EXEC(104(1) pocmpx.iqdb) -`
	`dsiqmint.c(4234)`
	`       Message from server: Message: -1004015, State 0, Severity 14 --`
	`'SQL Anywhere Error -1004015: Permission denied: Command not allowed on`
	`Multiplex Writer servers. (saint_iqthresholdddl.cxx 14936)'.`
	Workaround: Change the `sql.ini` file entry for Sybase IQ that the Replication Server uses, to connect to the coordinator node.

620097	**Data corruption when replicating data to Sybase IQ 15.*x* in RTL.** When RTL replicates data to Sybase IQ 15.*x*, data inserted into `time` and `timestamp` columns may be corrupted. There is no warning nor error message indicating that the data is corrupted. **Workaround**: Select one of: • If there is no `text` or `image` in the table, mark the table to not be compiled. RTL uses the function string to process instead of **insert-location**. Issue this command: `alter connection to` *`data_server.database`* `for replicate table named` *`table_name`* `set` **dsi_compile_enable** `"off"` This will effectively slow down the process. • If there is `text` or `image` in the table, apply Sybase IQ 15.1 ESD #3 or use Sybase IQ 12.7 ESD #4 and later.
619358	**`Text` and `image` column values are truncated when inserted into a Sybase IQ 15.1 database. This problem occurs in Sybase IQ 15.1 GA, ESD #1, and ESD #2.** **Workaround**: Apply Sybase IQ 15.1 ESD #3 or use Sybase IQ 12.7 ESD #4 and later.
594620	**When inserting data from Replication Server to Sybase IQ 15.0 and later using the insert-location, you may encounter the right truncation of string data error.** **Workaround**: In Sybase IQ, apply this command: `set option public.STRING_RTRUNCATION = “OFF”` By default, the STRING_RTRUNCATION option is set to on in Sybase IQ 15.0 and later.

Known Installation Issues for Replication Server and Replication Manager

Learn about known installation issues and apply workarounds for Replication Server and Replication Manager.

Table 9. Installer issues

CR#	Description
642076	**If you use the 32-bit version of Replication Server on a 64-bit operating system, a license installation-grace period may not be provided. Replication Server does not start until a license is deployed.** **Workaround**: If you are evaluating the product, you need to request for an evaluation license from Sybase. Otherwise, you need to generate your licenses from SPDC and deploy them.
638452	**The Replication Manager plug-in does not get deployed to Sybase Central when you install a 32-bit version Replication Manager on a 64-bit Windows system.** **Workaround**: Install the 64-bit version Replication Manager plug-in on 64-bit Windows systems.
635915	**If you install Adaptive Server 15.5 or Adaptive Server 15.5 ESD #1 after Replication Server 15.6 and uninstall Adaptive Server 15.5 ESD #1, Replication Server 15.6 may not function correctly.** **Workaround**: Apply Adaptive Server 15.5 ESD #2.
619817	**If the df command fails, the installer stops responding before the preinstallation summary pane appears.** **Workaround**: Execute **strace -e statfs, statfs64 df** to identify the NFS mounts that have a problem. Then execute **umount -l <path>** to unmount all trouble NFS mounts. Re-run the installer.
619813	**The console mode installer cannot correctly validate licenses entered during installation. This results in license validation errors during installation.** **Workaround**: Use the GUI installer instead.
619807	**On Microsoft Windows 2008, the installer can create the SDK installation directory only when you are logged in as an administrator.** **Workaround**: Create the installation directory before running the installer.

CR#	Description
619805	**The installer fails on Microsoft Windows Vista and Windows 2008 on x86-64 64-bit because the installer cannot set up the environment variables.** **Workaround**: In Windows Explorer, right-click `setup.exe` or `setupConsole.exe`, and select **Properties**. On the Compatibility tab, select **Windows Vista Compatibility Mode**.
619793	**You cannot use the Tab and arrow keys to navigate the Choose Product Features window of the installation or uninstallation program.** **Workaround**: • Uninstaller – use the mouse to click the Choose Product Features window. This sets the focus on the window and allows you to use your keyboard. • Installer – use the mouse to select the features in the Choose Product Features window.
619771	**The uninstallation program stops responding when you select Delete from the Delete User Files screen and then click Previous from the Uninstall Complete screen.** **Workaround**: Do not return to the Delete User Files screen after you have selected **Delete**.
618722	**Interactive installation using response file does not work.** When installing Replication Server in an interactive mode using a response file, the installer does not use the values stored in the file. In silent mode, the response file works as expected. **Workaround**: None.
613393	**UAF issue when installing Adaptive Server into the same directory as Replication Server.** If you install Adaptive Server 15.0.3 before you install Replication Server 15.6, and install it into the same top directory, the environment variable SYBASE_UA is set to `%SYBASE%\UAF-2_0` instead of `%SYBASE%\UAF-2_5`. To use the Replication Monitoring Service for Replication Server 15.6, the environment variable must be set to `%SYBASE%\UAF-2_5`. **Workaround**: Select one of: • Install Replication Server 15.6 last. • Reinstall Replication Server. • Reset the SYBASE_UA environment variable to `%SYBASE%\UAF-2_5` in these files: • `%SYBASE%\SYBASE.bat` • `%SYBASE%\SYBASE.env`

Known Issues for Replication Server and Sybase IQ InfoPrimer Integration

These known issues concern the integration of Replication Server and Sybase IQ InfoPrimer.

Table 10. Replication Server and Sybase IQ InfoPrimer Integration Known Issues

CR#	Description

668621	**The Replication Server materialization scripts do not execute correctly on Windows server.**
	Workaround: Before running the Replication Server materialization scripts:
	1. Install MKS Toolkit 9.4. You can download the MKS Toolkit environment at *http://mkssoftware.com*.
	2. Set the TK_DUALCASE_IN_CASE_STATEMENTS environment variable to 1.
	3. Start Sybase IQ InfoPrimer Server from the `%SYBASE%\SybaseIQIPServer\scripts\replication` directory:
	`GridNode`
	`GridNode --port 5500`
	4. In the Processing tab of your EL project, insert "sh " at the beginning of the command line. For example:
	`sh {ScriptPath}\replication\mat_api.sh --env-pds`
	`"{env_pds}" --env-pdb "{env_pdb}"`
	`--env-pdb_user ....`

668152	**Unexpected column mapping may occur in SQL Transformation projects.** By default, staging table columns may not map to base table columns as expected if a base table column is: • An included attribute (column) in the Target tab in the Generic Transformation editor • A key attribute (column) • Has an expression specified in the Target tab of the Generic Transformation editor **Workaround**: For an update staging table, all transformations must be described in the Target tab of the Generic Transformation editor. All base table columns that are included attributes (columns) in the Target tab in the Generic Transformation editor must have expressions specified in the Target tab.

CR#	Description
668147	**Incorrect SQL may be generated for update operations.** A SQL transformation for an update must check the bitmap column or columns of the update staging table to determine if transformations must be applied to base table columns. These transformations are specified in the Function column of the Attributes tab of the Generic Transformation editor and should be wrapped in CASE statements in the Generated Transformation. SQL statements, constant or calculated values, and functions that take more than one argument may not be properly wrapped in CASE statements. **Workaround**: Manually edit the Generated Transformation.
665408	**Tables with Java-type or encrypted columns are not supported when no primary key is specified.** In constructing the delete and update staging tables, Sybase IQ InfoPrimer includes only primary-key columns as published in the replication definition for the primary table. If no primary-key columns are specified, Sybase IQ InfoPrimer uses all published columns in the delete and update staging table schema, excluding LOB columns, Java columns, encrypted columns, and floating-point columns. However, Sybase IQ InfoPrimer cannot distinguish between Java columns and columns that have user-defined datatypes, nor can it determine which columns might be encrypted. Sybase IQ InfoPrimer therefore does not support primary tables containing Java-type or encrypted columns and for which you specify no primary key.

Known Issues for SySAM License

Learn about known issues and apply workarounds for SySAM License.

Table 11. SySAM License Issues

CR#	Description
640129	**When Replication Server requires multiple licenses but the license server has insufficient licenses, the error message displays an incorrect number for the licenses obtained.** For example, if Replication Server requires 4 licenses but has obtained only 2, the error message displays: `Replication Server requires 4 REP_SERVER licenses but only 4 could be obtained.` **Workaround**: Determine the number of licenses available, execute: **sysam status -a**

CR#	Description
633179	**If you are installing Replication Server on a machine that has a previously installed Sybase product, you may not be provided a license installation-grace period. Replication Server does not start until a license is deployed.** **Workaround**: If you are evaluating the product, you need to request for an evaluation license from Sybase. Otherwise, you need to generate your licenses from SPDC and deploy them.

Known Issues for Replication Server Unsupported Operations

Learn about the known issues when using unsupported operations.

These Adaptive Server operations may cause incorrect replication:

- Disabling the secondary truncation point with **dbcc settrunc** and then truncating the log can cause lost transactions.
- Replication Server does not support nested transactions within replicated stored procedures.

  When you enable replication for a stored procedure using **sp_setrepproc** or **sp_setreplicate**, Adaptive Server always runs the stored procedure within a transaction. If you have not explicitly run the replicated stored procedure within a transaction, Adaptive Server places an implicit **begin transaction** command at the start of the procedure.

  If the replicated stored procedure contains nested transaction commands such as **begin transaction**, **commit transaction**, or **rollback transaction**, you might get errors when you run the procedure. For example, a **rollback transaction** command rolls back to the start of the stored procedure, rather than to the nested **begin transaction** command, which was the intended rollback point.
- Data that is inserted into a primary table using an unlogged bulk copy operation is not replicated.
- To use the atomic method of subscription materialization:
  - The user who enters the **create subscription** command or the database owner must own the primary table. Alternatively, you must use user-defined function strings for **select** operations at the primary database.
  - If the database owner or maintenance user does not own the replicate table, use user-defined function strings for **select** operations at the replicate database. If the owner of the replicate table is different from the owner of the primary table, create a unique function string by using a distinct function-string class.

Known Issues with Language and Globalization

There is a limitation when using Japanese character sets in Replication Server. Neither the eucjis nor the sjis character set can be converted; this issue affects both Adaptive Server and Open Client™ and Open Server libraries.

Hankaku Katakana conversion

In general, Japanese character sets are compatible. However, Hankaku Katakana characters, although they exist in both the eucjis and sjis character sets, cannot be converted. Converting data that contains Hankaku Katakana characters between eucjis and sjis does not work. This conversion problem occurs with `character` datatypes and the `text` datatype and is documented in the *Adaptive Server Enterprise System Administration Guide Volume 1 > Configuring Client/Server Character Set Conversions*.

This conversion problem affects both Adaptive Server and the Sybase Open Client and Open Server libraries. Because Replication Server uses these libraries for all conversions, this problem also affects Replication Server.

In Replication Server, this type of failure is treated in the same way as is the case of a single character missing from the target character set. The remainder of the conversion succeeds and replication proceeds, and problem characters are replaced by question marks in the target data area. There is currently no way to escape this restriction with the Sybase connectivity libraries. However, in Adaptive Server, if you turn on trace flag number 2402, you can remove this restriction.

Using trace flag 2402

Generally, Sybase recommends that you set up your replication system so that Replication Server handles all character set conversions at the replicate Replication Server and prevents the replicate data server from performing any conversions. In this case, you can work around the Hankaku Katakana restriction if you set up your system so that the replicate data server performs the conversion.

The following table shows how this might look if the primary data server used the sjis character set and the replicate data server used eucjis. Communication in this system is between each data server and its Replication Server and between the two Replication Servers.

Primary Replication Server	sjis
Replicate Replication Server	sjis
Primary data server	sjis
Replicate data server	eucjis

The primary and replicate Replication Servers are configured to use the same character set as the primary data server. (If only one Replication Server manages the primary and replicate data servers, configure it with the character set of the primary data server.)

In this configuration, when the replicate Replication Server connects to the replicate data server with character set sjis, the replicate data server detects this condition and converts data into its own character set, eucjis. If trace flag 2402 is activated in the replicate data server, then the conversion includes the Hankaku Katakana characters.

Setting up this workaround

1. Configure your system as suggested.
2. Turn on trace flag 2402 in the replicate data server (Adaptive Server) by including **-T2402** on the command line when you start Adaptive Server.

Changing default date format for a language

If you modify the `common.loc` file to change the default date format for a given language, make the corresponding change to the `syslanguages` table on all affected Adaptive Servers.

Documentation Changes

Read about updates, corrections, and clarifications to the documentation released with Replication Server 15.6.

SQL Anywhere Replication Support

Read about updates, corrections, and clarifications for SQL Anywhere® documentation.

For information on SQL Anywhere support as a primary or a replicate database, see the SQL Anywhere documentation.

Replication Server Administration Guide: Volume 1 Documentation Changes

Read about updates, corrections, and clarifications for Replication Server Administration Guide: Volume 1.

Usage Section in Schema Cache

The Usage section displayed in the Sample Output from **sp_sysmon** of RepAgent activity report in Chapter 5, “Managing RepAgent and Supporting Adaptive Server” is incorrect.

The Usage should be a subsection in Schema Cache. The corrected output is:

Sample Output from sp_sysmon of RepAgent Activity

View a sample output of RepAgent counter activity from **sp_sysmon**.

Sample output from sp_sysmon

```
Replication Agent
-----------------
Replication Agent: pubs2
Replication Server: NY_RS
```

```

                        per sec     per xact    count    % of total
                        -------     --------    -----    ----------
```

```

Log Scan Summary
 Number of Log Scans       n/a        n/a           1        n/a
 Amount of Time
   for Log Scan (ms)       n/a        n/a        3822        n/a
   Longest Time
   for Log Scan (ms)       n/a        n/a        3822        n/a
   Average Time
   per Log Scan (ms)       n/a        n/a        3822        n/a
```

```
Log Scan Activity
   Updates                 n/a        n/a           5        n/a
   Inserts                 n/a        n/a           5        n/a
   Deletes                 n/a        n/a           5        n/a
   Store Procedures        n/a        n/a           0        n/a
   DDL Log Records         n/a        n/a           0        n/a
   Writetext Log Records   n/a        n/a           0        n/a
   Text/Image Log Records  n/a        n/a          10        n/a
   CLRs                    n/a        n/a           0        n/a
   Checkpoints Processed   n/a        n/a           0        n/a
```

```
Transaction Activity
    Opened                 n/a        n/a           7        n/a
    Commited               n/a        n/a           7        n/a
    Aborted                n/a        n/a           0        n/a
    Delayed Commit         n/a        n/a           0        n/a
    Maintenance User       n/a        n/a           0        n/a
```

```
Log Extension Wait
    Count                  n/a        n/a           3        n/a
    Amount of time (ms)    n/a        n/a        7822        n/a
    Longest Wait (ms)      n/a        n/a        5110        n/a
    Average Time (ms)      n/a        n/a      2607.3        n/a
```

```
Schema Cache

  Usage
    Max Ever Used          n/a        n/a                0        n/a
    Schemas reused         n/a        n/a                0        n/a

  Forward Schema Lookups
    Count                  n/a        n/a                0        n/a
    Total Wait (ms)        n/a        n/a                0        n/a
    Longest Wait (ms)      n/a        n/a                0        n/a
    Average Time (ms)      n/a        n/a              0.0        n/a

  Backward Schema Lookups
    Count                  n/a        n/a                0        n/a
    Total Wait (ms)        n/a        n/a                0        n/a
    Longest Wait (ms)      n/a        n/a                0        n/a
    Average Time (ms)      n/a        n/a              0.0        n/a
```

```
Truncation Point Movement
    Moved                  n/a        n/a                0        n/a
    Gotten from RS         n/a        n/a                0        n/a
```

```
Connections to Replication Server
    Success                n/a        n/a                0        n/a
    Failed                 n/a        n/a                0        n/a
```

```
Network Packet Information
    Packets Sent           n/a        n/a                6        n/a
    Full Packets Sent      n/a        n/a                2        n/a
    Largest Packet         n/a        n/a             2048        n/a
    Amount of Bytes Sent   n/a        n/a             7695        n/a
    Average Packet         n/a        n/a           1282.5        n/a
```

```
I/O Wait from RS
    Count                  n/a        n/a                6        n/a
    Amount of Time (ms)    n/a        n/a              766        n/a
    Longest Wait (ms)      n/a        n/a              206        n/a
    Average Wait (ms)      n/a        n/a            127.7        n/a
```

```


-------------------------------------------------------------------
```

Correction to the Description of Schema Cache

The description for the Usage section should be included in the description for Schema cache in the “RepAgent Counter Activity” in Chapter 5, “Managing RepAgent and Supporting Adaptive Server.”

Schema Cache

When the structure of an object marked for replication is modified—by **alter table**, for example—Adaptive Server must log special records in the transaction log that later on will help RepAgent identify the correct schema for the object.

This section reports schema activity and RepAgent activity scanning forward and backward in the transaction log looking for object schema changes.

- Usage

  This section reports:

  - The maximum number of active schemas in the Schema Cache since the last restart of Replication Agent
  - The number of times that a schema had to be removed from the Schema Cache to free space for a new one
- Forward Schema Lookups

  This section reports:

  - The number of times RepAgent performed forward scans
  - The total amount of time, in ms, that RepAgent spent performing forward scans
  - The longest amount of time, in ms, that RepAgent spent performing a forward scan
  - The average amount of time, in ms, that RepAgent spent performing a forward scan
- Backward Schema Lookups

  RepAgent performs a backward scan when DDL is performed inside a transaction. This section reports:

  - The number of times RepAgent spent performing backward scans
  - The total amount of time, in ms, that RepAgent performed backward scans
  - The longest amount of time, in ms, that RepAgent spent performing a backward scan
  - The average amount of time, in ms, that RepAgent spent performing a backward scan

Configuration Parameters Affecting Database Connections

Update Table 7-1, “Configuration parameters affecting database connections” in Chapter 7, “Managing Database Connections.”

- **dsi_cmd_prefetch** – the default value has been changed to off.
- **dsi_proc_as_rpc** – new parameter as described in this table:

Parameter *(database_param)*	**Value *(value)***

Parameter ***(database_param)***	**Value *(value)***
dsi_proc_as_rpc	Specifies how Replication Server applies stored procedure replication. • Set on to use remote procedure call (RPC) calls. • Set off to use language calls. Default: off When Adaptive Server is the replicate database, set **dsi_proc_as_rpc** either on or off. When the replicate database is Oracle: • Set on if you use ExpressConnect for Oracle (ECO). ECO only supports stored procedure replication using RPC. By default, Replication Server sets **dsi_proc_as_rpc** on if you use one of the Oracle ECO connection profiles when you create the connection to the Oracle database from Replication Server. See *Replication Server Options 15.5 > Installation and Configuration Guide ExpressConnect for Oracle 15.5 > Configuring ExpressConnect for Oracle*. • Set off if you use ECDA Option for Oracle. ECDA does not support RPC for stored procedure replication.

Restrictions to Replicating Encrypted Columns

The second bulleted item in “Replicating encrypted columns” in Chapter 9, “Managing Replicated Tables” is incorrect.

Replace with:

- You cannot use encrypted columns in a **where** clause of a subscription or article. Replication Server receives the value in cipher text and cannot compare an encrypted value to a clear text value. The encrypted columns cannot be searchable columns.

Disabling Secondary Truncation Point in Subscription with Bulk Materialization

Replace the numbered steps in “Subscription with bulk materialization” in Chapter 12, “Managing Replicated Objects Using Multisite Availability.”

1. Dump *PDS.pdb*. The DSI connection to the replicate database is suspended when the dump marker reaches the replicate Replication Server. It is suspended so that no data will be replicated until you finished step 2. Replication Server activates and validates the subscription automatically when the dump marker is replicated.

   Warning! Do not execute **activate subscription** or it will override the wait for dump marker at the Replication Server.

2. Load *PDS.pdb* to *RDS.rdb*.

3. In *RDS.rdb*, disable the secondary truncation point, if necessary:

   dbcc settrunc('ltm', 'ignore')

4. Resume the DSI connection to the *RDS.rdb*.

Output Display in Displaying Schedules

The output that appears in `Displaying schedules` example in Chapter 13, “Scheduling Replication Tasks,” is incorrect.

The corrected output is:

```
Schedule  Schedule    Status  Type  Owner  Sequence    Command
Name      Time
--------  ----------  ------  ----  -----  --------    -------
s1        27 * * * *    1       0     sa      1      conn_suspend.sh
```

Replication Server Administration Guide: Volume 2 Documentation Changes

Read about updates, corrections, and clarifications for Replication Server Administration Guide: Volume 2.

Replication Server Parameters That Affect Performance

Update Table 4-2, “Replication Server configuration parameters” in Chapter 4, “Performance Tuning.”

- **dist_direct_cache_read**:
  - Replace text for Enhanced Distributor Thread Read Efficiency with:

    Enables the distributor (DIST) thread to read SQL statements directly from the Stable Queue Thread (SQT) cache. This reduces the workload from SQT and the dependency between the two, and improves the efficiency of both SQT and DIST.
  - The default value of **dist_direct_cache_read** has been changed to off.
- **dsi_cmd_prefetch** – the default value has been changed to off.
- **nrm_thread** – the default value has been changed to off.

Connection Parameters That Affect Performance

In Table 4-3, “Connection parameters that affect performance” in Chapter 4, “Performance Tuning,” the default setting for **dsi_cmd_prefetch** has been changed to off.

Configuring HVAR

Replace the examples that show how to set the configuration parameters with **alter connection** in “High Volume Adaptive Replication to Adaptive Server” in Chapter 4, “Performance Tuning.”

For example, after you enable HVAR, you can set different values for the parameters:

-
  ```
  alter connection to IQSRVR.iqdb
  set dsi_compile_max_cmds to ‘50000’
  go
  ```

-
  ```
  alter connection to IQSRVR.iqdb
  set dsi_bulk_threshold to ‘15’
  go
  ```

-
  ```
  alter connection to IQSRVR.iqdb
  set dsi_command_convert to ‘i2di,u2di’
  go
  ```

Note: You must execute a separate **alter connection** command for each parameter you want to change. Do not enter more than one parameter after entering **alter connection**.

Replication Server—Advanced Services Option

There are changes in “Replication Server—Advanced Services Option” in Chapter 4, “Performance Tuning.”

- Enhanced DSI Efficiency – the default setting for **dsi_cmd_prefetch** has been changed to off.
- Enabling NRM Thread in Enhanced RepAgent Executor Thread Efficiency – the default setting for **nrm_thread** has been changed to off.
- Enhanced Distributor Thread Read Efficiency – replace text with:

  Enables the distributor (DIST) thread to read SQL statements directly from the Stable Queue Thread (SQT) cache. This reduces the workload from SQT and the dependency between the two, and improves the efficiency of both SQT and DIST. Use the **dist_direct_cache_read** parameter with **configure replication server** to use this enhancement:

  ```
  configure replication server
  set dist_direct_cache_read to ‘on’
  ```

  By default, **dist_direct_cache_read** is set to off. The distributor thread requests SQL statements from SQT through the message queue.

  dist_direct_cache_read is a server-level static parameter. You must restart Replication Server after you enable or disable the parameter.

Replication Server Heterogeneous Replication Guide Documentation Changes

Read about updates, corrections, and clarifications for Replication Server Heterogeneous Replication Guide.

Specifying How Replication Server Replicates Stored Procedures

Add a new section to “Replicate database connectivity for Oracle” in Chapter 10, “Oracle Replicate Data Server.”

If you use ExpressConnect for Oracle, set **dsi_proc_as_rpc** on. ECO only supports stored procedure replication using remote procedure calls. By default, Replication Server sets **dsi_proc_as_rpc** on if you use one of the Oracle ECO connection profiles when you create the connection to the Oracle database from Replication Server. See *Replication Server Options 15.5 > Installation and Configuration Guide ExpressConnect for Oracle 15.5 > Configuring ExpressConnect for Oracle*.

If you use ECDA Option for Oracle, set **dsi_proc_as_rpc** off. ECDA does not support remote procedure calls for stored procedure replication.

Configuring RTL

Replace the examples that show how to set the configuration parameters with **alter connection** in “Sybase IQ replicate database configuration issues” in Chapter 11, “Sybase IQ Replicate Data Server Issues.”

For example, after you enable RTL, you can set different values for the parameters:

-
  ```
  alter connection to IQSRVR.iqdb
  set dsi_compile_max_cmds to ‘50000’
  go
  ```

-
  ```
  alter connection to IQSRVR.iqdb
  set dsi_bulk_threshold to ‘15’
  go
  ```

-
  ```
  alter connection to IQSRVR.iqdb
  set dsi_command_convert to ‘i2di,u2di’
  go
  ```

Note: You must execute a separate **alter connection** command for each parameter you want to change. Do not enter more than one parameter after entering **alter connection**.

Creating Interfaces File Entries

Add a new section in “Scenario for replication to Sybase IQ” in Chapter 11, “Sybase IQ Replicate Data Server Issues.”

Create an entry in the `interfaces` files of the replicate Replication Server and the Sybase IQ data server for each other.

1. Create an entry for the replicate Replication Server in the interfaces file of the Sybase IQ data server.
2. Create an entry for the Sybase IQ data server in the interfaces file of the replicate Replication Server.

   See also `“Replicate database connectivity for Sybase IQ” in Chapter 11, “Sybase IQ Replicate Data Server Issues.”`

Creating Replication Definitions and Subscriptions

Add new information at the end of step 1 in the task “Creating replication definitions and subscriptions” in Chapter 11, “Sybase IQ Replicate Data Server Issues.”

Create *repdef_testtab* replication definition:

```
create replication definition repdef_testtab
with primary at ASE_DS.pdb1
with primary table named ‘testtab’
with replicate table named dbo.‘testtab’
(c1 int, c2 int, c3 char(10))
primary key(c1)
go
```

Add any required referential constraint clauses to the replication definitions to support RTL.

Map the datatypes in primary Adaptive Server tables to replicate Sybase IQ tables:

- Adaptive Server `timestamp` maps to IQ `varbinary(8)`.
- Adaptive Server `bigdatetime` maps to IQ `timestamp`.

Warm Standby Requirements and Restrictions for Oracle

Add a new bullet item to the list in “Warm standby requirements and restrictions” in Chapter 12, “Managing Heterogeneous Warm Standby for Oracle.”

- Replication Server does not support warm standby replication between different platforms.

Oracle Database Resynchronization Procedures

The sequence of steps in “Database resynchronization scenarios,” in Chapter 13, “Resynchronizing Oracle Replicate Databases” are incorrect.

The corrected procedures are:

Resynchronizing Directly from a Primary Database

Resynchronize a replicate database from a primary database.

1. Stop replication processing by Replication Agent. Do not alter the truncation point. In Replication Agent, execute:

```
suspend
```

2. Suspend the Replication Server DSI connection to the replicate database:

   ```
   suspend connection to dataserver.database
   ```

3. Instruct Replication Server to remove data from the replicate database outbound queue and wait for a resync marker from the primary database Replication Agent:

   ```
   resume connection to data_server.database skip to
   resync marker
   ```

4. If the truncation point has not been moved, proceed to step 5. Otherwise, reinitialize the Replication Agent repository before you obtain a dump of the primary database contents. In the Replication Agent, execute:

   ```
   ra_init force
   go
   ```

5. Obtain a dump of the primary database contents following the instructions in your database documentation. If you use the Recovery Manager (RMAN) for Oracle, use the Oracle **list backup** command to obtain the last System Change Number (SCN) of the RMAN backup. Then, in Replication Agent, set this SCN as the value of **lr_dump_marker**:

   ```
   lr_dump_marker oracle scn
   ```

6. Start your Replication Agent in resync mode and send a resync marker to Replication Server:

   ```
   resume resync
   go
   ```

7. In the Replication Server system log, verify that DSI has received and accepted the resync marker from Replication Agent by looking for this message:

   ```
   DSI for data_server.database received and processed
   Resync Database Marker. Waiting for Dump Marker.
   ```

   After DSI processes the resync marker for the replicate database, you can apply the dump to the replicate database.

   Note: If you are resynchronizing multiple databases, verify that the DSI connection for each database you are resynchronizing has accepted the resync marker.

8. Apply the dump of the primary database to the replicate database following the instructions in your database documentation.
9. Verify that Replication Server has processed the dump database marker by looking for this message in the Replication Server system log:

   ```
   DSI for data_server.database received and processed
   Dump Marker. DSI is now suspended. Resume after database has been
   reloaded.
   ```

   When Replication Server receives the dump marker, the DSI connection automatically suspends.

10. After you apply the dump to the replicate database, resume DSI using:

    ```
    resume connection to data_server.database
    ```

Resynchronizing After Using a Third-Party Utility to Create a Dump

1. Stop replication processing by Replication Agent. Do not alter the truncation point. In Replication Agent, execute:

   ```
   suspend
   ```

2. Suspend the Replication Server DSI connection to the replicate database:

   ```
   suspend connection to dataserver.database
   ```

3. Instruct Replication Server to remove data from the replicate database outbound queue and wait for a resync marker from the primary database Replication Agent:

   ```
   resume connection to data_server.database skip to
   resync marker
   ```

4. If the truncation point has not been moved, proceed to step 5. Otherwise, reinitialize the Replication Agent repository before you obtain a dump of the primary database contents. In the Replication Agent, execute:

   ```
   ra_init force
   go
   ```

5. Use the third-party utility to obtain a dump of the primary database contents.
6. Determine the dump point based on information from the primary database when you took the dump, or information from the third-party utility. With a third-party utility, you are responsible for determining the dump point. For example, if you are using a disk replication tool, you can temporarily halt activity at the primary database to eliminate in-progress transactions from the disk snapshot, and then use the “end of transaction log” point as the dump database marker.
7. To mark the end of the dump position that you obtained in step 5, execute the stored procedure on the primary database for Replication Agent:

   ```
   lr_dump_marker oracle scn
   ```

8. Restart Replication Agent in resync mode and send a resync marker to Replication Server:

   ```
   resume resync
   go
   ```

   Replication Agent automatically generates a dump database marker at a time based on the end of dump position that you obtained in step 6 and set in step 7, and sends the dump database marker to Replication Server.

9. Verify that DSI has received and accepted the resync marker from Replication Agent by looking for this message in the Replication Server system log:

   ```
   DSI for data_server.database received and processed
   Resync Database Marker. Waiting for Dump Marker.
   ```

10. Apply the dump of the primary database from the third-party tool to the replicate database, following the instructions in the database and third-party utility documentation.
11. Verify that Replication Server has processed the dump database marker by looking for this message in the Replication Server system log:

```
DSI for data_server.database received and processed
Dump Marker. DSI is now suspended. Resume after
database has been reloaded.
```

When Replication Server receives the dump marker, the DSI connection automatically suspends.

12. After you apply the dump to the replicate database, resume DSI:

    ```
    resume connection to data_server.database
    ```

Resynchronizing the Active and Standby Databases When the Warm Standby Environment Is the Replicate Site

1. Stop replication processing by both the primary database Replication Agent and the warm standby active database Replication Agent. Do not alter the truncation point. In Replication Agent, execute:

   ```
   suspend
   ```

2. Suspend the Replication Server DSI connection to the active and standby databases:

   ```
   suspend connection to dataserver.database
   ```

3. Instruct Replication Server to move data from the outbound queue of the active and standby databases, and wait for a resync marker from the primary database Replication Agent:

   ```
   resume connection to data_server.database skip to
   resync marker
   ```

4. If the truncation point has not been moved, proceed to step 5. Otherwise, reinitialize the Replication Agent repository before you obtain a dump of the primary database contents. In the primary Replication Agent, execute:

   ```
   ra_init force
   go
   ```

5. Obtain a dump of the primary database contents following the instructions in your database documentation. If you use the Recovery Manager (RMAN) for Oracle, use the Oracle **list backup** command to obtain the last System Change Number (SCN) of the RMAN backup. Then, in Replication Agent, set this SCN as the value of **lr_dump_marker**:

   ```
   lr_dump_marker oracle scn
   ```

6. Start the primary Replication Agent in resync mode and send a resync marker to Replication Server:

   ```
   resume resync
   go
   ```

7. Verify that DSI for the active database has received and accepted the resync marker from the primary database Replication Agent by looking for this message in the Replication Server system log:

   ```
   DSI for data_server.database received and processed
   Resync Database Marker. Waiting for Dump Marker.
   ```

8. Verify that the Replication Server DSI for the active database has processed the dump database marker by looking for this message from the active database In the Replication Server system log:

   ```
   DSI for data_server.database received and processed
   Dump Marker. DSI is now suspended. Resume after
   database has been reloaded.
   ```

9. Apply the dump of the primary database to the active database following the instructions in database documentation.
10. Move the truncation point to the end of the transaction log for the active database. In Replication Agent, execute:

    ```
    pdb_xlog move_truncpt
    go
    ```

11. Reinitialize Replication Agent repository based on the latest system data from the active database:

    ```
    ra_init force
    go
    ```

12. Start Replication Agent for the active database in resync mode with the **init** option. In Replication Agent, execute:

    ```
    resume resync, init
    ```

13. Verify that DSI for the standby database has received and accepted the resync marker from the active database Replication Agent by looking for this message in the Replication Server system log:

    ```
    DSI for data_server.database received and processed
    Resync Database Marker. DSI is now suspended. Resume
    after database has been reloaded.
    ```

    When Replication Server receives and processes the resync database with init marker, the DSI connection suspends.

14. Obtain a dump of the active database contents and apply the dump to the standby database. You can also apply the dump of the primary database from step 5 if the dump does not include database configuration information.
15. Resume DSI to the active and standby databases:

    ```
    resume connection to data_server.database
    ```

Replication Server New Features Guide Documentation Changes

Read about updates, corrections, and clarifications for Replication Server New Features Guide.

Enhanced Distributor Thread Read Efficiency

Correction to the description for Enhanced Distributor Thread Read Efficiency in Replication Server 15.5.

In Replication Server 15.5, the description for this topic reads as:

With Replication Server 15.5, the distributor (DIST) thread reads SQL statements from the Stable Queue Transaction thread (SQT) cache directly. This reduces contention between inbound and outbound queues, and improves Replication Server performance.

Replace with:

With Replication Server 15.5, the distributor (DIST) thread reads SQL statements directly from the Stable Queue Transaction (SQT) thread cache. This reduces the workload from SQT and the dependency between the two, and improves the efficiency of both SQT and DIST.

Replication Server Reference Manual Documentation Changes

Read about updates, corrections, and clarifications for Replication Server Reference Manual.

admin schedule

In **admin schedule** in Chapter 3, “Replication Server Commands,” the example output is incorrect.

The corrected output is:

```
Schedule  Schedule    Status  Type  Owner  Sequence   Command
Name      Time
--------  ----------  ------  ----  -----  --------   -------
s1        27 * * * *    1       0     sa       1      conn_suspend.sh
```

alter connection

Update Table 3-15, “Parameters affecting database connections" in Chapter 3, “Replication Server Commands.”

- **dsi_cmd_prefetch** – the default value has been changed to off.
- **dsi_proc_as_rpc** – new parameter as described in this table.

database_param	Description and Value

database_param	Description and Value
dsi_proc_as_rpc	Specifies how Replication Server applies stored procedure replication. • Set on to use remote procedure call (RPC) calls. • Set off to use language calls. Default: off When Adaptive Server is the replicate database, set **dsi_proc_as_rpc** either on or off. When the replicate database is Oracle: • Set on if you use ExpressConnect for Oracle (ECO). ECO only supports stored procedure replication using RPC. By default, Replication Server sets **dsi_proc_as_rpc** on if you use one of the Oracle ECO connection profiles when you create the connection to the Oracle database from Replication Server. See *Replication Server Options 15.5 > Installation and Configuration Guide ExpressConnect for Oracle 15.5 > Configuring ExpressConnect for Oracle*. • Set off if you use ECDA Option for Oracle. ECDA does not support RPC for stored procedure replication.

drop schedule

The syntax for Example 1 for **drop schedule** in Chapter 3, “Replication Server Commands” is incorrect.

To delete schedule1, enter:

```
drop schedule schedule1
```

Trace Option for alter connection

The syntax for the *trace* parameter in **alter connection** in Chapter 3, “Replication Server Commands” is incorrect.

The corrections:

- Name of *trace_value* option is *value*.
- Revised description for *value*.

alter connection

Syntax

```
alter connection to data_server.database {...
    set trace [to] 'value'}
```

Parameters

trace

Allows ExpressConnect tracing at the DSI level.

value

A character string containing a new value for the option.

If you are using the **trace** option, the syntax for *value* takes the form "*module*, *condition,[on/off]"*, where:

- *module* – specifies the module type. Valid value is *econn*.
- *condition* – specifies the trace condition to set.
- *on* or *off* – specifies the state of the condition that is desired.

Note: The **trace** parameter in the **alter connection** command allows an empty string. For example:

```
alter connection to data_server.database
set trace to ''
```

An empty string disables ExpressConnect tracing values after the connection or when the Replication Server is restarted.

Value Parameter for alter connector

The **value** parameter for the **trace** option in **alter connector** in Chapter 3, "Replication Server Commands" is incorrect.

Update the description for the **value** parameter with:

Parameters

value

A character string containing a new value for the option.

If you are using the **trace** option, the syntax for *value* takes the form "*module*, *condition,[on/off]"*, where:

- *module* – specifies the module type. Valid value is *econn*.
- *condition* – specifies the trace condition to set.
- *on* or *off* – specifies the state of the condition that is desired.

configure replication server

Update Table 3-19, "Replication Server configuration parameters" in Chapter 3, "Replication Server Commands."

- **dist_direct_cache_read** – replace text for enhanced distributor thread read efficiency; default value has been changed to off.

  Replace text with:

  Enables the distributor (DIST) thread to read SQL statements directly from the Stable Queue Thread (SQT) cache. TThis reduces the workload from SQT and the dependency between the two, and improves the efficiency of both SQT and DIST.

- **nrm_thread** – the default value has been changed to off.

Replication Server System Functions

Add a new function string, **rs_set_timestamp_insert**, in Chapter 4, “Replication Server System Functions.”

rs_set_timestamp_insert

Enables replication of timestamp columns to an Adaptive Server table.

Examples

- **Example 1 –** Alters **rs_set_timestamp_insert** for non-Adaptive Server databases that do not support **set timestamp_insert on**:

```
alter function string rs_set_timestamp_insert
    for some_function_string_class
    output language
    ''
```

Usage

- **rs_set_timestamp_insert** is called after **rs_usedb** for any user database connection. Replication Server does not call this function string for Replication Server connections and RSSD connections.
- **rs_set_timestamp_insert** has function-string class scope.
- **rs_set_timestamp_insert** maps to **set timestamp_insert on** for Adaptive Server replicate databases. For all non-Adaptive Server databases, **rs_set_timestamp_insert** maps to null.
- Adaptive Server 15.0.1 and earlier databases do not support **set timestamp_insert on**.
- If executing **rs_set_timestamp_insert** fails, Replication Server continues running and does not report back to the user.

rs_subcmp

There are several changes to the **rs_subcmp** command in Chapter 7, “Executable Programs.”

- The syntax now includes **-g**, **-h**, **-H** parameters:

```
rs_subcmp [-R | -r] [-v] [-V] [-z[1 | 2] [-g] [-h]]
 [-f config_file] [-F]
 -S primary_ds [-D primary_db]
 -s replicate_ds [-d replicate_db]
 -t table_name [-T primary_table_name]
 -c select_command [-C primary_select_command]
 -u user [-U primary_user]
 [-p passwd] [-P primary_passwd]
 [-B primary_init_batch]
 [-b replicate_init_batch]
 [-n num_iterations] [-w wait_interval]
 [-e float_precision] [-E real_precision]
 [-k primary_key_column [-k primary_key_column]...]
```

```
[-i identity_column]
[-l text_image_column_name
[-l text_image_column_name]...]
[-L text_image_length_in_kilobytes]
[-N text_image_column_name
[-N text_image_column_name]...]
[-Z language]
[-o sort_order]
[-O sort_order]
[-J rs_subcmp_charset]
[-j rep_charset]
[-a replicate_column_name primary_column_name
[-a replicate_column_name primary_column_name]...]
[-q unicode_sort_order]
[-Q unicode_sort_order]
[-x schema_flag]
[-X filter_flag]
[-I interface_file]
[-H normalization_option]
```

- These are the changes to the Examples section:
  - In Example 3, this is the corrected text to introduce the example:

    Compares the schema of the `authors` table between two databases with the same name using a configuration file called `config.cfg`.
  - Corrected commands for examples 4, 5, and 6:
    - Corrected code for Example 4:

      ```
      rs_subcmp -Spds -srds -Dpdb -drdb -Usa -Psa_pwd -usa
                -psa_pwd -x1
      ```
    - Corrected code for Example 5:

      ```
      rs_subcmp -Spds -srds -Dpdb -drdb -Usa -Psa_pwd -usa
                -psa_pwd -x1 -XitD
      ```
    - Corrected code for Example 6:

      ```
      rs_subcmp -Spds -srds -Dpdb -drdb -Usa -Psa_pwd -usa
                -psa_pwd -x1 -X+TU
      ```
- The third bulleted item in the Usage section should read:

  For schema comparison to work, **rs_subcmp** must be able to locate and successfully run the `ddlgen` executable file. You can use the DDLGENLOC environment variable to set the location of `ddlgen`. If DDLGENLOC is not set, **rs_subcmp** looks for `ddlgen` at its default location, which is at `%SYBASE%\ASEP\bin\ddlgen.` To ensure that `ddlgen` runs successfully, the environment variables that `ddlgen` uses must be set correctly.

  The SYBROOT environment variables must also be set to the SYBASE environment variable.

Documentation Updates

Read about updates and corrections to the documentation released with Replication Server 15.6. Included are updated topics with the correct information.

Summary Sheet of Updated Topics

Read a summary of the changes to the topics that have been updated to fix documentation errors in version 15.6.

Guide Name (Primary Source of Information)	Updated Topic	Summary of Changes
Heterogeneous Replication Guide	See *Heterogeneous Replication Guide > Replication System Overview > Replication System Non-ASE Configurations > ASE Server Primary to Non-ASE Server Replicate*.	With Replication Server 15.5 and later, use ExpressConnect for Oracle to replicate to Oracle databases.
Administration Guide Volume 2	See: • *Administration Guide Volume 2 > Performance Tuning > DSI Bulk-Copy-in*. • *Administration Guide Volume 2 > Performance Tuning > DSI Bulk Copy-in > Limitations for Bulk Copy-in*.	DSI bulk-copy-in supports Oracle as well as Adaptive Server.
	See *Reference Manual > Replication Server Commands > alter connection*.	The correct description of **dsi_bulk_copy**: Turns the bulk-copy-in feature on or off for a connection. If **dynamic_sql** and **dsi_bulk_copy** are both on, Replication Server applies bulk copy-in when appropriate and uses dynamic SQL if Replication Server cannot use bulk copy-in.

Guide Name (Primary Source of Information)	Updated Topic	Summary of Changes
Reference Manual	See *Reference Manual > Replication Server Commands > alter connection*.	Included the valid values for **dsi_max_cmds_to_log**. Default: -1 (all commands) Valid values: 0 to 2147483647
Administration Guide Volume 2	See *Administration Guide Volume 2 > Performance Tuning > Dynamic SQL for Enhanced Replication Server Performance > replicate minimal columns with Dynamic SQL*.	Clarified the impact of **replicate_minimal_columns** when the update does not change values in the updated row. This parameter can affect trigger processing if you expect triggers to fire even if there is no change in values to any columns in the row.
Reference Manual	See *Reference Manual > Replication Server Commands >* ***create connection using profile***.	**version** is a mandatory attribute in **create connection using profile**.
Reference Manual	See **dsi_command_convert** in *Reference Manual > Replication Server Commands >* ***alter connection***.	You need not set **dsi_compile_enable** on to use **dsi_command_convert**.
Heterogeneous Replication Guide	See *Heterogeneous Replication Guide > Oracle Replicate Databases Resynchronization > Configuring Database Resynchronization > Reinitializing the Replicate Database*.	After you apply the dump from the primary database or dump source to the replicate database, reinitialize the replicate database to restore users, tables, and permissions that the dump removed.
Reference Manual	See *Reference Manual > RSSD Stored Procedures >* ***rs_ticket***.	You cannot replicate **rs_ticket**.
Administration Guide Volume 2	See *Administration Guide Volume 2 > Replication System Recovery > Replicate Database Resynchronization for Adaptive Server > Database Resynchronization Scenarios > Resynchronizing Using a Third-party Dump Utility*.	You must specify a valid hexadecimal value for *oqid* to mark the end of the dump database position in step 6.
Configuration Guide for Windows	See *Configuration Guide for Windows > Upgrade or Downgrade Replication Server > Upgrading Replication Server > Upgrading an RSSD or an ERSSD*	In step 2, the command to revoke the "sa" role after the upgrade process is finished, is incorrect. It should be: `sp_role 'revoke', sa_role, primary_user`

Guide Name (Primary Source of Information)	**Updated Topic**	**Summary of Changes**
Reference Manual	See **replicate minimal columns** in *Reference Manual > Replication Server Commands > alter connection*. Also see: - *Reference Manual > Replication Server Commands > create replication definition* - *Reference Manual > Replication Server Commands > alter replication definition*	If your replication definition has **replicate all columns** and the **replicate_minimal_ columns** connection property is set to on, the connection replicates minimal columns and overrides your replication definition. To replicate all columns to your target database, even if there is no change to the column values for the row, set **replicate_minimal_ columns** values for the DSI connection to off.
Reference Manual	See *Reference Manual > Topics > Datatypes > Exact Numeric (Decimal) Datatypes*.	Numeric datatypes in **where** clauses of replication definition must include decimal precision information.
Release Bulletin	See *Release Bulletin > Installation and Upgrade > Special Upgrade and Downgrade Instructions > Upgrading from Replication Server Version 15.5 to Version 15.6*.	Upgrading from Replication Server version 15.5 to version 15.6 has been simplified. See *Upgrading from Replication Server Version 15.5 to Version 15.6* on page 8.
Administration Guide Volume 2	See *Administration Guide Volume 2 > Performance Tuning > Replication Server - Advanced Services Option > High Volume Adaptive Replication to Adaptive Server > HVAR Processing and Limitations*. This change also applies to real-time loading (RTL). See: - *New Features Guide > New Features in Replication Server 15.6 > Replication from Oracle to Sybase IQ Using Real-Time Loading > Real-Time Loading Solution > RTL Processing and Limitations*. - *Heterogeneous Replication Guide > Sybase IQ Replicate Data Server Issues > Real-Time Loading Solution > RTL Processing and Limitations*.	The description of trigger actions in the first bulleted item in *HVAR Processing and Limitations* in the *Administration Guide Volume 2*, and in *RTL Processing and Limitations* in the New Features Guide and Heterogeneous Replication Guide, may be misleading. **Insert** triggers do not fire as the HVAR and RTL process performs a bulk load of net new rows into the table directly. **update** and **delete** triggers continue to fire when Replication Server applies the net results of compilation to the replicate database. However, row modifications that Replication Server compiles, and that are no longer in the net results, are invisible to the triggers. Triggers can detect only the final row images.

ASE Server Primary to Non-ASE Server Replicate

A simple heterogeneous replication scenario replicates one-way from an Adaptive Server primary database to a non-ASE replicate.

The only unique requirements are a component to apply transaction data to the replicate database, and the application of the HDS feature of Replication Server to translate Adaptive Server datatypes to the native datatypes of the replicate database.

For more detailed information about HDS, see *Replication Server Administration Guide Volume 1 > Manage Replicated Tables > Translating Datatypes Using HDS*.

Replication System Components

The components required for an Adaptive Server primary to non-ASE replicate configuration are:

- Adaptive Server primary database
- Replication Server
- ExpressConnect for Oracle to replicate to Oracle or a relevant ECDA database gateway designed for the replicate data server, such as ECDA Option for ODBC for Microsoft SQL Server
- Non-ASE replicate data server. For example, Microsoft SQL Server

Replication System Issues

Iin an Adaptive Server primary to non-ASE replicate configuration, consider:

- The Replication Server database connection for the replicate database must include a valid user ID and password (the maintenance user) for the replicate database. This user ID must have authority to apply replicate transactions in the replicate database.
- Create the Replication Server replicate database connection using the correct profile for the replicate database. The connection profile specifies the correct function-string class and error class for the replicate database, and additionally may contain class-level translation definitions and replicate database object creation, to support replication.

alter connection

Changes the attributes of a database connection.

Syntax

```
alter connection to data_server.database {
     [for replicate table named [table_owner.]table_name
     [set table_param [to] ‘value’]] |
     set function string class [to] function_class |
     set error class [to] error_class |
     set replication server error class [to] rs_error_class |
     set password [to] passwd |
     set log transfer [to] {on | off} |
     set database_param [to] 'value' |
     set security_param [to] 'value' |
     set security_services [to] 'default']
     set dataserver and database name [to] new_ds.new_db |
     set trace [to] 'value'}
```

Parameters

- **data_server –** The data server that holds the database whose connection is to be altered.
- **database –** The database whose connection is to be altered.
- **for replicate table named –** Specifies the name of the table at the replicate database. *table_name* is a character string of up to 200 characters. *table_owner* is an optional qualifier for the table name, representing the table owner. Data server operations may fail if actual table owners do not correspond to what you specify in the replication definition.
- **table_param –** The table-level parameter that affects a table you specify with **for replicate table name**.

  Valid values: **dsi_compile_enable** and **dsi_command_convert**. See *Table 12. Parameters Affecting Database Connections* for descriptions.
- **function_class –** The function-string class to use with the data server. See "Function string variable modifiers" for a list of function classes that Replication Server provides for database connections.
- **error_class –** The error class that handles database errors. See "Error and function classes" for a list of error classes that Replication Server provides for database connections.
- **rs_error_class –** The error class that handles Replication Server errors for a database. See "Error and function classes" for a list of Replication Server error classes.
- **passwd –** The new password to use with the login name for the database connection. You must specify a password if network-based security is not enabled.
- **log transfer on –** Allows the connection to send transactions from a RepAgent to the Replication Server.
- **log transfer off –** Stops the connection from sending transactions from a primary database RepAgent.
- **database_param –** The parameter that affects database connections from the Replication Server.
- **value –** A character string containing a new value for the option.

  Note: Parameters and values are described in *Table 12. Parameters Affecting Database Connections*.

Table 12. Parameters Affecting Database Connections

database_param	Description and *value*
batch	Specifies how Replication Server sends commands to data servers. When **batch** is “on,” Replication Server may send multiple commands to the data server as a single command batch. When **batch** is “off,” Replication Server sends commands to the data server one at a time. Default: on
batch_begin	Indicates whether a **begin transaction** can be sent in the same batch as other commands (such as **insert**, **delete**, and so on). Default: on
command_retry	The number of times to retry a failed transaction. The value must be greater than or equal to 0. Default: 3
db_packet_size	The maximum size of a network packet. During database communication, the network packet value must be within the range accepted by the database. Default: 512-byte network packet for all Adaptive Server databases Maximum: 16,384 bytes
deferred_name_resolution	Enable deferred name resolution in Replication Server to support deferred name resolution in Adaptive Server. Deferred name resolution is only supported in Adaptive Server 15.5 and later. You must ensure that deferred name resolution is supported in the replicate Adaptive Server before you enable deferred name resolution support in Replication Server. After you execute **deferred_name_resolution** with **alter connection** or **alter logical connection**, suspend and resume the connection. Default: off
disk_affinity	Specifies an allocation hint for assigning the next partition. Enter the logical name of the partition to which the next segment should be allocated when the current partition is full. Default: off

database_param	**Description and *value***
dist_sqt_max_cache_size	The maximum Stable Queue Transaction (SQT) cache size for the inbound queue. The default, 0, means the current setting of the **sqt_max_cache_size** parameter is used as the maximum cache size for the connection. Default: 0 For 32-bit Replication Server: • Minimum – 0 • Maximum – 2147483647 For 64-bit Replication Server: • Minimum – 0 • Maximum – 2251799813685247
dist_stop_unsupported_cmd	When **dist_stop_unsupported_cmd** is on, DIST suspends itself if a command is not supported by downstream Replication Server. If it is off, DIST ignores the unsupported command. Regardless of **dist_stop_unsupported_cmd** parameter's setting, Replication Server always logs an error message when it sees the first instance of a command that cannot be sent over to a lower-version Replication Server. Default: off
dsi_alt_writetext	Controls how large-object updates are sent to the replicate database. The values are: • dcany – generates a **writetext** command that includes primary key columns. This setting prevents full table scans when populating non-ASE replicate databases using DirectConnect Anywhere™ as an interface. • off – generates an Adaptive Server **writetext** command that includes a text pointer. Default: off **Note:** If you are using ExpressConnect to connect non-ASE replicate databases, then you are not required to configure the **dsi_alt_writetext** database parameter.
dsi_bulk_copy	Turns the bulk-copy-in feature on or off for a connection. If **dynamic_sql** and **dsi_bulk_copy** are both on, Replication Server applies bulk-copy-in when appropriate and uses dynamic SQL if Replication Server cannot use bulk-copy-in. Default: off

database_param	**Description and *value***
dsi_bulk_threshold	The number of consecutive **insert** commands in a transaction 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 **dsi_large_xact_size**. Replication Server uses **dsi_bulk_threshold** for real-time loading (RTL) replication to Sybase IQ and high volume adaptive replication (HVAR) to Adaptive Server. If the number of commands for an **insert**, **delete**, or **update** operation on one table is less than the number you specify after compilation, RTL and HVAR use language instead of bulk interface. Minimum: 1 **Note:** Do not set to ‘1’ when you enable RTL or HVAR as this detrimental to performance. Default: 20 Configuration level: Server, database For setting, use **configure replication server** for server-level or **alter connection** for database-level. **Note:** You must set **dsi_compile_enable** to ‘on’ to use **dsi_bulk_threshold** for RTL or HVAR.
dsi_charset_convert	The specification for handling character-set conversion on data and identifiers between the primary Replication Server and the replicate Replication Server. This parameter applies to all data and identifiers to be applied at the DSI in question. The values are: • on – convert from the primary Replication Server character set to the replicate Replication Server character set; if character sets are incompatible, shut down the DSI with an error. • allow – convert where character sets are compatible; apply any unconverted updates to the database, as well. • off – do not attempt conversion. This option is useful if you have different but compatible character sets and do not want any conversion to take place. During subscription materialization, a setting of “off” behaves as if it were “allow.” Default: on
dsi_cmd_batch_size	The maximum number of bytes that Replication Server places into a command batch. Default: 8192 bytes

database_param	Description and *value*
dsi_cmd_prefetch	Allows DSI to pre-build the next batch of commands while waiting for the response from data server, and therefore improves DSI efficiency. If you also tune your data server to enhance performance, it is likely that you will gain an additional performance increase when you use this feature. Default: off When you set **dsi_compile_enable** to 'on', Replication Server ignores what you set for **dsi_cmd_prefetch**. License: Separately licensed under the Advanced Services Option. See *Replication Server Administration Volume 2 > Performance Tuning > Replication Server – Advanced Services Option* .
dsi_cmd_separator	The character that separates commands in a command batch. Default: newline (\n) **Note:** You must update this parameter in an interactive mode, not by executing a DDL-generated script, or any other script. You cannot reset **dsi_cmd_separator** by running a script.
dsi_command_convert	Specifies how to convert a replicate command. A combination of these operations specifies the type of conversion: • **d** – delete • **i** – insert • **u** – update • **t** – truncate • **none** – no operation Combinations of operations for **dsi_command_convert** include **i2none**, **u2none**, **d2none**, **i2di**, **t2none**, and **u2di**. You must type the number 2. The operation before conversion precedes the 2 and the operations after conversion are after the 2. For example: • **d2none** – do not replicate the **delete** command. • **i2di**,**u2di** – convert both **insert** and **update** to **delete** followed by **insert**, which is equivalent to an autocorrection. • **t2none** – do not replicate **truncate table** command. Default: none You can also configure this parameter at the table level. For setting, use **alter connection** for database-level, or **alter connection** with the **for replicate table named** clause for table-level configuration. Set **dsi_command_convert** to **none** to remove the current **dsi_command_convert** setting for a connection or a table.

database_param	**Description and *value***
dsi_commit_check_locks_intrvl	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. Default: 1000ms (1 second) Minimum: 0 Maximum: 86,400,000 ms (24 hours)
dsi_commit_check_locks_log	The number of times the DSI executor thread executes the **rs_dsi_check_thread_lock** function string before logging a warning message. Used with parallel DSI. Default: 200 Minimum: 1 Maximum: 1,000,000
dsi_commit_check_locks_max	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. Default: 400 Minimum: 1 Maximum: 1,000,000
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). Default: on

database_param	**Description and *value***
dsi_compile _enable	Set to 'on' to enable RTL or HVAR at the server-level, database-level, or table-level. Default: • off – server and database-level. Replication Server uses continuous log order row by row change replication. • on – table-level For setting, use **configure replication server** for server-level, **alter connection** for database-level, or **alter connection** with the **for replicate table named** clause for table-level configuration. Set **dsi_compile_enable** to 'off' for an affected table if replicating new row changes causes problems, such as when there is a trigger on the table which requires all the operations on that table to be replicated in log order, and therefore compilation is not allowed. **Note:** Set **dsi_compile _enable** to 'on' at the server or database-level before you set **dsi_compile_enable** to 'off' at the table-level. When you set **dsi_compile_enable** to 'on', Replication Server ignores what you set for **replicate_minimal_columns** and **dsi_cmd_prefetch**. After you execute **dsi_compile_enable** at the server, database, or table-level, suspend and resume the connection.
dsi_compile_max_cmds	Specifies, in number of commands, the maximum size of a group of transactions. When HVAR or RTL reaches the maximum group size for the current group that it is compiling, HVAR or RTL starts a new group. If there is no more data to read, and even if the group does not reach the maximum number of commands, HVAR or RTL completes grouping the current set of transactions into the current group. Minimum: 100 Default: 100,000 You can configure the parameter at the server or database levels For setting, use **configure replication server** for server-level or **alter connection** for database-level. **Note:** You must set **dsi_compile_enable** to 'on' to use **dsi_compile_max_cmds**.
dsi_compile_retry_threshold	Specifies a threshold value for the number of commands in a group.

database_param	**Description and *value***
dsi_connector_type	Specifies the database driver technology used for implementing the connector. This parameter along with **dsi_dataserver_make** is used to identify the connector that is associated with the connection. If you are replicating to ASE or IQ, set this parameter value to *ctlib* or if replicating to Oracle, set the value to *oci*. Default: ctlib. Valid values: ctlib, oci.
dsi_dataserver_make	Specifies the data server type that contains the replicate database that you want to connect to. Possible values are: ASE, IQ, and ORA. Use **dsi_dataserver_make** and **dsi_connector_type** to identify the connector that is associated with the connection. Set to IQ to replicate to Sybase IQ. Set to ASE to replicate to Adaptive Server, and ORA to replicate to Oracle. You can configure **dsi_dataserver_make** at the database level. If you do not specify this parameter, Replication Server deduces the data server type from the function-string class name of the database connection. If the functions-string class is customized, Replication Server cannot deduce the data server type and defaults to ‘ASE’.
dsi_exec_request_sproc	Turns on or off request stored procedures at the DSI of the primary Replication Server. Default: on
dsi_fadeout_time	The number of seconds of idle time before a DSI connection is closed. A value of “-1” indicates that a connection will not close. Default: 600 seconds
dsi_ignore_underscore_name	When the transaction partitioning rule is set to “name,” specifies whether or not Replication Server ignores transaction names that begin with an underscore. Values are “on” and “off.” Default: on

database_param	Description and *value*
dsi_isolation_level	Specifies the isolation level for transactions. The ANSI standard and Adaptive Server supported values 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. **Note:** Data servers supporting other isolation levels are supported as well through the use of the **rs_set_isolation_level** function string. Replication Server supports all values for replicate data servers. The default value is the current transaction isolation level for the target data server.
dsi_keep_triggers	Specifies whether triggers should fire for replicated transactions in the database. Set off to cause Replication Server to set triggers off in the Adaptive Server database, so that triggers do not fire when transactions are executed on the connection. Set on for all databases except standby databases. Default: on (except standby databases)
dsi_large_xact_size	The number of commands allowed in a transaction before the transaction is considered to be large. Minimum: 4 Default: 100
dsi_max_cmds_to_log	The number of commands to write into the exceptions log for a transaction. Default: -1 (all commands) Valid values: 0 to 2147483647
dsi_max_xacts_in_group	Specifies the maximum number of transactions in a group. Larger numbers may improve data latency at the replicate database. Range of values: 1 – 1000. Default: 20

database_param	**Description and *value***
dsi_max_text_to_log	The number of bytes to write into the exceptions log for each **rs_writetext** function in a failed transaction. Change this parameter to prevent transactions with large *text, unitext, image* or *rawobject* columns from filling the RSSD or its log. Default: -1 (all *text*, *unitext*, *image*, or *rawobject* columns)
dsi_non_blocking_commit	The number of minutes that Replication Server saves a message after a commit. A 0 value means that non-blocking commit is disabled. **Note:** You cannot use this parameter with **alter connection** to configure an active database connection in a standby environment. Default: 0 Maximum: 60
dsi_num_large_xact_threads	The number of parallel DSI threads to be reserved for use with large transactions. The maximum value is one less than the value of **dsi_num_threads**. Default: 0
dsi_num_threads	The number of parallel DSI threads to be used. The maximum value is 255. Default: 1
dsi_partitioning_rule	Specifies the partitioning rules (one or more) the DSI uses to partition transactions among available parallel DSI threads. Values are **origin**, **ignore_origin**, **origin_sessid**, **time**, **user**, **name**, and **none**. See the *Replication Server Administration Guide Volume 2* for detailed information. Default: none

database_param	Description and *value*
dsi_proc_as_rpc	Specifies how Replication Server applies stored procedure replication. - Set on to use remote procedure call (RPC) calls. - Set off to use language calls. Default: off When the replicate database is Adaptive Server , **dsi_proc_as_rpc** can be on or off. When the replicate database is Oracle: - Set on if you use ExpressConnect for Oracle (ECO). ECO only supports stored procedure replication using RPC. By default, Replication Server sets **dsi_proc_as_rpc** on if you use one of the Oracle ECO connection profiles when you create the connection to the Oracle database from Replication Server. See *Replication Server Options 15.5 > Installation and Configuration Guide ExpressConnect for Oracle 15.5 > Configuring ExpressConnect for Oracle*. - Set off if you use ECDA Option for Oracle. ECDA does not support RPC for stored procedure replication.
dsi_quoted_identifier	Enables or disables quoted identifier support in the Data Server Interface (DSI). Default: off
dsi_replication	Specifies whether or not transactions applied by the DSI are marked in the transaction log as being replicated. When **dsi_replication** is set to “off,” the DSI executes **set replication off** in the Adaptive Server database, preventing Adaptive Server from adding replication information to log records for transactions that the DSI executes. Since these transactions are executed by the maintenance user and, therefore, not usually replicated further (except if there is a standby database), setting this parameter to “off” avoids writing unnecessary information into the transaction log. **dsi_replication** must be set to “on” for the active database in a warm standby application for a replicate database, and for applications that use the replicated consolidated replicate application model. Default: on (“off” for standby database in a warm standby application)

database_param	**Description and *value***
dsi_replication_ddl	Supports bidirectional replication by specifying whether or not transactions are to be replicated back to the original database. 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 DDL transaction replication in bidirectional MSA replication environment. Default: off
dsi_row_count_validation	Enables or disables row count validation. Default: on
dsi_rs_ticket_report	Determines whether to call function string **rs_ticket_report** or not. **rs_ticket_report** function string is invoked when **dsi_rs_ticket_report** is set to on. Default: on

database_param	**Description and *value***
dsi_serialization_method	Specifies the method used to determine when a transaction can start, while still maintaining consistency. In all cases, commit order is preserved. These methods are ordered from most to least amount of parallelism. Greater parallelism can lead to more contention between parallel transactions as they are applied to the replicate database. To reduce contention, use the **dsi_partition_rule** option. • **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. • **wait_for_commit** – specifies that a transaction cannot start until the transaction scheduled to commit immediately preceding it is ready to commit. • **wait_after_commit** – specifies that a transaction cannot start until the transaction scheduled to commit immediately preceding it has committed completely. **Note:** You can only set **dsi_serialization_method** to **no_wait** if **dsi_commit_control** is set to “on”. These options are retained only for backward compatibility with older versions of Replication Server: • **none** – same as **wait_for_start**. • **single_transaction_per_origin** – same as **wait_for_start** with **dsi_partitioning_rule** set to **origin**. **Note:** The **isolation_level_3** value is no longer supported as a serialization method but it is the same as setting **dsi_serialization_method** to **wait_for_start** and **dsi_isolation_level** to 3. Default: **wait_for_commit**

database_param	**Description and *value***
dsi_sqt_max_cache_size	Maximum SQT (Stable Queue Transaction interface) cache size for the outbound queue, in bytes. The default, “0,” means that the current setting of **sqt_max_cache_size** is used as the maximum cache size for the connection. Default: 0 For 32-bit Replication Server: • Minimum – 0 • Maximum – 2147483647 For 64-bit Replication Server: • Minimum – 0 • Maximum – 2251799813685247
dsi_text_convert_multiplier	Changes the length of *text* or *unitext* datatype columns at the replicate site. Use **dsi_text_convert_multiplier** when *text* or *unitext* datatype columns must expand or contract due to character set conversion. Replication Server multiplies the length of primary *text* or *unitext* data by the value of **dsi_text_convert_multiplier** to determine the length of *text* or *unitext* data at the replicate site. Its type is *float*. • If the character set conversion involves expanding *text* or *unitext* datatype columns, set **dsi_text_convert_multiplier** equal to or greater than 1.0. • If the character set conversion involves contracting *text* or *unitext* datatype columns, set **dsi_text_convert_multiplier** equal to or less than 1.0. Default: 1

database_param	**Description and *value***
dsi_timer	Use the **dsi_timer** configuration parameter to specify a delay between the time transactions commit at the primary database and the time transactions commit at the standby or replicate database. Replication Server processes transactions in the outbound queue in commit order after the period of delay is over. After you execute **dsi_timer** with **alter connection** or **alter logical connection**, suspend and resume the connection. Specify the delay in the hh:mm format. • Maximum: 24 hours. • Default: 00:00, which means there is no delay. **Note:** Replication Server does not support time zone differences between the RepAgent or Replication Agent at the primary database and the Replication Server with the DSI connection where you want to execute **dsi_timer**.
dsi_xact_group_size	The maximum number of bytes, including stable queue overhead, to place into one grouped transaction. A grouped transaction is multiple transactions that the DSI applies as a single transaction. A value of –1 means no grouping. Sybase recommends that you set **dsi_xact_group_size** to the maximum value and use **dsi_max_xacts_in_group** to control the number of transactions in a group. **Note:** Obsolete for Replication Server version 15.0 and later. Retained for compatibility with older Replication Servers. Maximum: 2,147,483,647 Default: 65,536 bytes
dump_load	Set to "on" at replicate sites only to enable coordinated dump. See the *Replication Server Administration Guide Volume 2* for details. Default: off
dynamic_sql	Turns dynamic SQL feature on or off for a connection. Other dynamic SQL related configuration parameters will take effect only if this parameter is set to on. **Note:** If **dynamic_sql** and **dsi_bulk_copy** are both on, DSI applies bulk copy-in. Dynamic SQL is used if bulk copy-in is not used. Default: off

database_param	**Description and *value***
dynamic_sql_cache_management	Manages the dynamic SQL cache for a connection. Values: • mru – specifies that once **dynamic_sql_cache_size** is reached, the old dynamic SQL prepared statements are deallocated to give room for new statements. • fixed – specifies that once the **dynamic_sql_cache_size** is reached, allocation for new dynamic SQL statements stops. Default: fixed
dynamic_sql_cache_size	Allows Replication Server to estimate how many database objects can use dynamic SQL for a connection. You can use **dynamic_sql_cache_size** to limit resource demand on a data server. Default: 100 Minimum: 1 Maximum: 65,535
exec_cmds_per_timeslice	Specifies the number of LTL commands an LTI or RepAgent executor thread can process before yielding the CPU. By increasing this value, you allow the RepAgent executor thread to control CPU resources for longer periods of time, which may improve throughput from RepAgent to Replication Server. Set this parameter at the connection level using **alter connection**. See "Controlling the number of commands the RepAgent executor can process," in the *Replication Server Administration Guide Volume 2*. Default: 2,147,483,647 Minimum: 1 Maximum: 2,147,483,647

database_param	**Description and *value***
exec_nrm_request_limit	Specifies the amount of memory available for messages from a primary database waiting to be normalized. Set **nrm_thread** to 'on' with **configure replication server** before you use **exec_nrm_request_limit**. Minimum: 16,384 bytes Maximum: 2,147,483,647 bytes Default for: • 32-bit – 1,048,576 bytes (1MB) • 64-bit – 8,388,608 bytes (8MB) After you change the configuration for **exec_nrm_request_limit**, suspend and resume the Replication Agent. License: Separately licensed under the Advanced Services Option. See "Replication Server – Advanced Services Option," in the *Replication Server Administration Guide Volume 2*.
exec_sqm_write_request_limit	Specifies the amount of memory available for messages waiting to be written to an inbound queue. Default: 1MB Minimum: 16KB Maximum: 2GB
md_sqm_write_request_limit	Specifies the amount of memory available to the Distributor for messages waiting to be written to the outbound queue. **Note:** In Replication Server 12.1, **md_sqm_write_request_limit** replaces **md_source_memory_pool**. **md_source_memory_pool** is retained for compatibility with older Replication Servers. Default: 1MB Minimum: 16KB Maximum: 2GB

database_param	Description and *value*
parallel_dsi	Provides a shorthand method for configuring parallel DSI threads. A setting of "on" configures these values: • **dsi_num_threads** to 5 • **dsi_num_large_xact_threads** to 2 • **dsi_serialization_method** to "wait_for_commit" • **dsi_sqt_max_cache_size** to 1 million bytes (on 32-bit platform) and 20 million bytes (on 64-bit platform). A setting of "off" configures these parallel DSI values to their defaults. You can set this parameter to "on" and then set individual parallel DSI configuration parameters to fine-tune your configuration. Default: off
rep_as_standby	When the database is marked with **sp_reptostandby** and **rep_as_standby** is on, tables with a table replication definition not covered by a database replication definition are replicated. To replicate the tables, set: • **rep_as_standby** to on • **send maint xacts to replicate** to false • **send warm standby xacts** to true Default: off

database_param	**Description and *value***
replicate_minimal_columns	Specifies whether Replication Server should send all replication definition columns for all transactions, or only those needed to perform update or delete operations at the replicate database. Values are On and Off. Replication Server uses this connection-level parameter when a replication definition does not contain the **replicate minimal columns** clause, or if there is no replication definition at all. **Note:** If your replication definition has **replicate all columns** and the **replicate_minimal_columns** connection property is set to 'on', the connection replicates minimal columns. If you want to replicate all columns to your target database, even if there is no change to the column values for the row, set **replicate_minimal_ columns** values for the DSI connection to "off". You can use **admin config** to display **replicate_minimal_columns** configuration information. When you set **dsi_compile_enable** to 'on', Replication Server ignores what you set for **replicate_minimal_columns**. See "Using replicate_minimal_columns with dynamic SQL" in the *Replication Server Administration Guide Volume 2*.
save_interval	The number of minutes that the Replication Server saves messages after they have been successfully passed to the destination data server. See the *Replication Server Administration Guide Volume 2* for details. Default: 0 minutes
sub_sqm_write_request_limit	Specifies the memory available to the subscription materialization or dematerialization thread for messages waiting to be written to the outbound queue. Default: 1MB Minimum: 16KB Maximum: 2GB
use_batch_markers	Controls the processing of function strings **rs_batch_start** and **rs_batch_end**. If **use_batch_markers** is set to on, the **rs_batch_start** function string is prepended to each batch of commands and the **rs_batch_end** function string is appended to each batch of commands. Set **use_batch_markers** to on only for replicate data servers that require additional SQL to be sent at the beginning or end of a batch of commands that is not contained in the **rs_begin** function string. Default: off

- **security_param –** A parameter that affects network-based security for connections. See "Parameters affecting network-based security" table in **create route** for a list of parameters and a description of values.
- **set security_services to 'default' –** Resets all network-based security features for the connection to match the global settings of your Replication Server.
- **new_ds and new_db –** Name of the new data server and database for the connection.

Note: The *new_ds* and *new_db* parameters can have the same values that you have defined for *data_server* and *database* parameters.

- **trace –** Allows ExpressConnect tracing at the DSI level.
- **value –** A character string containing a new value for the option.

  If you are using the trace option, the syntax for value takes the form “module, condition, [on|off]”, where:

  - *module* – Specifies the module type. Valid value is *econn*.
  - *condition* – Specifies if a trace option is set to *on* or *off*.
  - *on* or *off* – Specifies the state of the condition that is desired.

Note: The **trace** parameter in the **alter connection** command allows empty string. For example:

```
alter connection to data_server.database
set trace to ''
```

An empty string disables ExpressConnect tracing values after the connection or when the Replication Server is restarted.

Examples

- **Example 1 –** Changes the function-string class for the *pubs2* database in the TOKYO_DS data server to *sql_derived_class*:

```
suspend connection to TOKYO_DS.pubs2

alter connection to TOKYO_DS.pubs2b
set function string class to sql_derived_class

resume connection to TOKYO_DS.pubs2
```

- **Example 2 –** Changes the number of LTL commands the LTI or RepAgent Executor thread can process before it must yield the CPU to other threads:

```
suspend connection to TOKYO_DS.pubs2
alter connection to TOKYO_DS.pubs2b
set exec_cmds_per_timeslice to ’10’
resume connection to TOKYO_DS.pubs2
```

Usage

- Use **suspend connection** to suspend activity on the connection before altering it.
- Execute **alter connection** at the Replication Server where the connection was created.
- Before you use **log transfer off** to stop data transfer from a primary database, be sure there are no replication definitions defined for data in the database.
- To change the route to a Replication Server, use **alter route**.
- Use **set function string class [to]** *function_class* to activate class-level translations for non-Sybase data servers.
- You can set connection parameters using the **alter connection** parameter.
- Execute **alter connection** at the Replication Server where the connection was created.

Database connection parameters

- Use **alter connection** to change the configuration parameters of a DSI or a database connection. To change a DSI configuration value, suspend the connection to the DSI, change the value, and then resume the connection to the DSI. This procedure causes the new value to take effect.
- Replication Server configuration parameters are stored in the *rs_config* system table. Some parameters can be modified by updating rows in the table. See the *Replication Server Administration Guide Volume 1* for more information.
- See the *Replication Server Administration Guide Volume 2* for more information about configuring parallel DSI threads.
- Use **assign action** to enable retry of transactions that fail due to specific data server errors.
- Before you change the function-string class, make sure that the class and all the required function strings exist for the new class.
- Before you change the error class, make sure the new class exists.
- Change the character for data servers that require a command separator to recognize the end of a command.

  If you have specified a different separator character and want to change it back to a newline character, enter the **alter connection** command as follows:

  ```
  alter connection to data_server.database
    set  to '<Return>'
  ```

  where you press the Return key, and no other characters, between the two single-quote characters.

The **dsi_bulk_copy** parameter

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.

The **dsi_partitioning_rule** parameter

You can specify more than one partitioning rule at a time. Separate values with a comma, but no spaces. For example:

```
alter connection to data_server.database
  set dsi_partitioning_rule to ‘origin,time’
```

The dataserver and database name parameter

Using **dataserver and database name** parameter you can switch the connection from using one connector to using another connector. For example, if you replicating to Oracle using the ASE/CT-Lib connector and DirectConnect for Oracle and you want to switch your connection to use the Oracle/OCI connector, you may be required to use a new data server and database name. Because the name given to the DirectConnect/Oracle in the Sybase interfaces file may not be the same as the Oracle data server name in the Oracle TNS Names file. To change:

1. Suspend the connection.
2. Alter the connection setting **dsi_dataserver_make** to *ora* and **dsi_connector_type** to *oci*.
3. Alter the connection setting **dataserver and database name** to **new_ds** and **new_db** where:
   - *new_ds* – name of the data server in the Oracle `tnsnames.ora` file
   - *new_ds* – name of the database

   Note: The *new_ds* and *new_db* parameters can have the same values that you have defined for *data_server* and *database* parameters.
4. Resume the connection.

The **dump_load** parameter

Before setting **dump_load** to “on,” create function strings for the **rs_dumpdb** and **rs_dumptran** functions. Replication Server does not generate function strings for these functions in the system-provided classes or in derived classes that inherit from these classes.

The **save_interval** configuration parameter

Set **save_interval** to save transactions in the DSI queue that can be used to resynchronize a database after it has been restored from backups. Setting a save interval is also useful when you set up a warm standby of a database that holds replicate data or receives replicated functions. You can use **sysadmin restore_dsi_saved_segments** to restore backlogged transactions.

Network-based security parameters

- Both ends of a connection must use compatible Security Control Layer (SCL) drivers with the same security mechanisms and security features. The data server must support **set proxy** or an equivalent command.

  It is the replication system Administrator's responsibility to choose and set security features for each server. Replication Server does not query the security features of remote servers before attempting to establish a connection. Connections fail if security features at both ends of the connection are not compatible.
- **alter connection** modifies network-based security settings for an outgoing connection from Replication Server to a target data server. It overrides default security parameters set with **configure replication server**.
- If **unified_login** is set to "required," only the replication system Administrator with "sa" permission can log in to the Replication Server without a credential. If the security mechanism should fail, the replication system Administrator can log in to Replication Server with a password and disable **unified_login**.
- A Replication Server can have more than one security mechanism; each supported mechanism is listed in the `libtcl.cfg` file under SECURITY.
- Message encryption is a costly process with severe performance penalties. In most instances, it may be wise to set **msg_confidentiality** "required" only for certain connections. Alternatively, choose a less costly security feature, such as **msg_integrity**.

Using alter connection to change maintenance passwords

- You can change the maintenance user password of any DSI connection using the **alter connection** command:

  ```
  alter connection to data_server.database
  set password to password
  ```
- If your Replication Server is using ERSSD and the *data_server.database* match the ERSSD names, using **alter connection** and **set password** updates the *rs_maintusers* table, issues **sp_password** at ERSSD, and updates the configuration file line *RSSD_maint_pw*.

Permissions

alter connection requires "sa" permission.

DSI Bulk Copy-in

Replication Server supports bulk-copy-in, which improves performance when replicating large batches of **insert** statements on the same table in the replicate database.

In normal replication, when replicating to a replicate database, Replication Server forms a SQL **insert** command, sends the command to the replicate database, and waits for the replicate database to process the row and send back the result of the operation. This affects Replication Server performance when large batches of data are being replicated, such as in end-of-day batch processing or trade consolidation.

Database Support

Bulk-copy-in is supported for Adaptive Server databases, and Oracle replicate databases that are updated by ExpressConnect for Oracle. If you turn on DSI bulk=copy-in and the replicate database is not supported, DSI shuts down with an error. See *Replication Server Options > ExpressConnect for Oracle Installation and Configuration Guide > System Requirements*.

DSI Bulk Copy-in Configuration Parameters

Database connection parameters that control bulk operations in DSI.

Parameter	Description
dsi_bulk_copy	Turns the bulk copy-in feature on or off for a connection. If **dynamic_sql** and **dsi_bulk_copy** are both on, Replication Server applies bulk copy-in when appropriate and uses dynamic SQL if Replication Server cannot use bulk copy-in. Default: off.
dsi_bulk_threshold	The number of consecutive **insert** commands in a transaction 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 **dsi_large_xact_size**. Minimum: 1 Default: 20

replicate minimal columns Clause and Dynamic SQL

Replication processing uses dynamic SQL when the replication definition contains replicate minimal columns or, when you set **replicate_minimal_columns** on for a connection.

You can use **replicate_minimal_columns** for physical connections and warm standby environments. DSI can use the parameter to determine whether to use minimal columns when there is no replication definition, or when the replication definition does not contain the **replicate minimal columns** clause.

By default, **replicate_minimal_columns** is on for all connections. The **replicate_minimal_columns** setting for a connection overrides replication definitions set with the **replicate all columns** clause.

With custom function strings, the behavior of the current replication environment may change when you set **replicate_minimal_columns** on for a connection. If the application is relying on a command to be sent to the replicate database for trigger processing, the default **replicate_minimal_columns** setting of on does not send the command when there are no changes to any columns in the row. To restore the original behavior, set **replicate_minimal_columns** off for the connection.

For example, to enable **replicate_minimal_columns** for the connection to the `pubs2` database in the SYDNEY_DS data server:

```
alter connection to SYDNEY_DS.pubs2
set replicate_minimal_columns to ‘on’
```

replicate_minimal_columns can affect trigger processing if you expect triggers to fire even if there is no change in values to any columns in the row.

You can use **admin config** to display the **replicate_minimal_columns** setting.

Note: When you set **dsi_compile_enable** ‘on’, Replication Server ignores the **replicate_minimal_columns** setting.

create connection using profile

create connection using profile clause uses predefined information to configure the connection between Replication Server and a non-Adaptive Server database, and, if needed, to modify the RSSD and the named *data_server.database*. To create a connection to Adaptive Server, see **create connection**.

Syntax

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

Parameters

- **data_server –** The data server that holds the database to be added to the replication system.
- **database –** The database to be added to the replication system.
- **connection_profile –** Indicates the connection profile that you want to use to configure a connection, modify the RSSD, and build replicate database objects.
- **version –** Specifies the connection profile version to use.
- **user –** The login name of the Replication Server maintenance user for the database. Replication Server uses this login name to maintain replicated data. You must specify a user name if network-based security is not enabled.
- **other_create_connection_options –** Use the 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. For a complete list of the other **create connection** options, see **create connection**.

- **display_only –** Use **display_only** with the **using profile** clause to display the commands that will be executed and the names of the servers upon which the commands will be executed. See the client and Replication Server logs for the result 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;standard
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 the *my_msss_error_class* error class:

```
create connection to msss_server.msss_db
using profile rs_ase_to_msss
set username to msss_maint;standard
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;standard
set username to ora_maint
set password to ora_maint_pwd
display_only

go
```

```
Display only using Connection Profile rs_ase_to_oracle;standard.

Command(s) intended for: prs01
create connection to oracle.instance
   set error class to rs_oracle_error_class
   set function string class to rs_oracle_function_class
   set username to ora_maint
   set password to ********
   set batch to off

Command(s) intended for 'edsprs01.edbprs01':
delete from rs_translation where classid = 0x0000000001000007 and
```

```
                                 source_dtid = 0x000000000000000c

Command(s) intended for 'edsprs01.edbprs01':
insert rs_translation (prsid, classid, type, source_dtid,
target_dtid,
                       target_length, target_status, rowtype)
values (0, 0x0000000001000007, 'D', 0x000000000000000c,
0x0000000000010200,
        19, 0, 0)

Command(s) intended for 'edsprs01.edbprs01':
delete from rs_translation where classid = 0x0000000001000007  and
                                 source_dtid = 0x000000000000000d

Command(s) intended for 'edsprs01.edbprs01':
insert rs_translation (prsid, classid, type, source_dtid,
target_dtid,
                       target_length, target_status, rowtype)
values (0, 0x0000000001000007, 'D', 0x000000000000000d,
0x0000000000010200,
        19, 0, 0)

Command(s) intended for 'edsprs01.edbprs01':
delete from rs_translation where classid = 0x0000000001000007 and
                                 source_dtid = 0x0000000000000001

Command(s) intended for 'edsprs01.edbprs01':
insert rs_translation (prsid, classid, type, source_dtid,
target_dtid,
                       target_length, target_status, rowtype)
values (0, 0x0000000001000007, 'D', 0x0000000000000001,
0x0000000000010202,
        0, 0, 0)
Command(s) intended for 'edsprs01.edbprs01':
delete from rs_translation where classid = 0x0000000001000007 and
                                 source_dtid = 0x0000000000000013

Command(s) intended for 'edsprs01.edbprs01':
insert rs_translation (prsid, classid, type, source_dtid,
target_dtid,
                       target_length, target_status, rowtype)
values (0, 0x0000000001000007, 'D', 0x0000000000000013,
0x0000000000010202,
        0, 0, 0)

Command(s) intended for 'edsprs01.edbprs01':
delete from rs_translation where classid = 0x0000000001000007 and
                                 source_dtid = 0x000000000000000E

Command(s) intended for 'edsprs01.edbprs01':
insert rs_translation (prsid, classid, type, source_dtid,
target_dtid,
                       target_length, target_status, rowtype)
values (0, 0x0000000001000007, 'D', 0x000000000000000E,
0x0000000000010205,
        136, 0, 0)
```

```
Command(s) intended for 'edsprs01.edbprs01':
delete from rs_translation where classid = 0x0000000001000007 and
                                 source_dtid = 0x000000000000000F

Command(s) intended for 'edsprs01.edbprs01':
insert rs_translation (prsid, classid, type, source_dtid,
target_dtid,
                       target_length, target_status, rowtype)
values (0, 0x0000000001000007, 'D', 0x000000000000000f,
0x0000000000010205,
        136, 0, 0)

Command(s) intended for 'edsprs01.edbprs01':
delete from rs_translation where classid = 0x0000000001000007 and
                                 source_dtid = 0x000000000000001b

Command(s) intended for 'edsprs01.edbprs01':
insert rs_translation (prsid, classid, type, source_dtid,
target_dtid,
                       target_length, target_status, rowtype)
values (0, 0x0000000001000007, 'D', 0x000000000000001b,
0x0000000000010201,
        9, 0, 0)

Command(s) intended for 'edsprs01.edbprs01':
delete from rs_translation where classid = 0x0000000001000007  and
                                 source_dtid = 0x000000000000001c

Command(s) intended for 'edsprs01.edbprs01':
insert rs_translation (prsid, classid, type, source_dtid,
target_dtid,
                       target_length, target_status, rowtype)
values (0, 0x0000000001000007, 'D', 0x000000000000001c,
0x0000000000010200,
        19, 0, 0)

Command(s) intended for 'oracle.instance':
drop table rs_info

Command(s) intended for 'oracle.instance':
commit

Command(s) intended for 'oracle.instance':
create table rs_info (rskey varchar2 (20), rsval varchar2 (20))

Command(s) intended for 'oracle.instance':
commit

Command(s) intended for 'oracle.instance':
insert into rs_info values ('charset_name', 'iso_1')

Command(s) intended for 'oracle.instance':
insert into rs_info values ('sortorder_name', 'bin_iso_1')

Command(s) intended for 'oracle.instance':
```

```
commit

Command(s) intended for 'oracle.instance':
drop public synonym rs_lastcommit

Command(s) intended for 'oracle.instance':
commit

Command(s) intended for 'oracle.instance':
drop table rs_lastcommit

Command(s) intended for 'oracle.instance':
commit

Command(s) intended for 'oracle.instance':
create table rs_lastcommit(origin  number(8),origin_qid char(72),
                        secondary_qid char(72),origin_time  date,
                           dest_commit_time date)

Command(s) intended for 'oracle.instance':
commit

Command(s) intended for 'oracle.instance':
grant all on rs_lastcommit to public

Command(s) intended for 'oracle.instance':
commit

Command(s) intended for 'oracle.instance':
create public synonym rs_lastcommit for rs_lastcommit

Command(s) intended for 'oracle.instance':
commit

Command(s) intended for 'oracle.instance':
CREATE OR REPLACE PROCEDURE
     RS_UPDATE_SEQUENCE(SequenceName VARCHAR2, SequenceValue
NUMBER,
                        Increment NUMBER)
    AS CurrentID NUMBER; LastID NUMBER; SeqCursor INTEGER; SQLStmt
        VARCHAR2(1024);
     Result NUMBER;
     BEGIN
     SQLStmt := 'SELECT ' || SequenceName || '.NEXTVAL FROM DUAL';
     SeqCursor := DBMS_SQL.OPEN_CURSOR;
     DBMS_SQL.PARSE(SeqCursor,SQLStmt,DBMS_SQL.NATIVE);
     DBMS_SQL.DEFINE_COLUMN(SeqCursor, 1, LastID);
     Result := DBMS_SQL.EXECUTE_AND_FETCH(SeqCursor);
     DBMS_SQL.COLUMN_VALUE(SeqCursor,1,CurrentID);
     LOOP
       IF ( Increment < 0 ) THEN EXIT WHEN CurrentID <=
SequenceValue;
         EXIT WHEN CurrentID > LastID;
       ELSE EXIT WHEN CurrentID >= SequenceValue;
         EXIT WHEN CurrentID < LastID;
       END IF;
       LastID := CurrentID;
       Result := DBMS_SQL.EXECUTE_AND_FETCH(SeqCursor);
       DBMS_SQL.COLUMN_VALUE(SeqCursor,1,CurrentID);
     END
     LOOP;
       DBMS_SQL.CLOSE_CURSOR(SeqCursor);
     END;

Command(s) intended for 'oracle.instance':
grant all on RS_UPDATE_SEQUENCE to public

Command(s) intended for 'oracle.instance':
DROP sequence rs_ticket_seq

Command(s) intended for 'oracle.instance':
CREATE sequence rs_ticket_seq

Command(s) intended for 'oracle.instance':
Drop table rs_ticket_history

Command(s) intended for 'oracle.instance':
CREATE TABLE rs_ticket_history(cnt numeric(8,0, h1 varchar(10,
   h2 varchar(10), h3 varchar(10), h4 varchar(50), pdb
varchar(30),
   prs varchar(30), rrs varchar(30), rdb varchar(30), pdb_t date,
   exec_t date, dist_t date, rsi_t date, dsi_t date,
   rdb_t date default current_date, exec_b int, rsi_b int, dsi_tnx
int,
   dsi_cmd int, ticket varchar(1024))

Command(s) intended for 'oracle.instance':
create unique index rs_ticket_idx on rs_ticket_history(cnt)

Command(s) intended for 'oracle.instance':
create or replace trigger rs_ticket_tri
   before insert on rs_ticket_history
   for each row
   begin
     if :new.cnt is null then
       select rs_ticket_seq.nextval into :new.cnt  from dual;
     end if;
   end rs_ticket_tri;Command(s) intended for 'oracle.instance':
grant all on rs_ticket_history to public

Command(s) intended for 'oracle.instance':
commit
```

Usage

- Connection profiles specify the function-string class and the error class. Connection profiles can also specify other connection options such as whether commands should be batched and what command separator to use. In addition to connection settings, connection profiles can specify class-level translations to install in the RSSD and objects, such as the *rs_lastcommit* table, to be created in the replicate database.

- 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.
- Always specify the **set username** clause right after the **using profile** clause.

Reinitializing the Replicate Database

After you apply the dump from the primary database or dump source to the replicate database, reinitialize the replicate database to restore users, tables, and permissions that the dump removed.

1. If maintenance and DDL users do not exist in the primary database, add them to the replicate database after you apply the dump from the primary database.
2. Run the `hds_oracle_new_setup_for_replicate.sql` script on the replicate database to add the relevant Replication Server system tables to the replicate database.

   The script also inserts relevant values and grants the required permissions in the replicate database.

rs_ticket

A stored procedure in the primary database that monitors Replication Server performance, module heartbeat, replication health and table-level quiesce.

Syntax

```
rs_ticket h1 [, h2 [, h3 [, h4]]]
```

Parameters

- **h1 [, h2 [, h3 [, h4]]] –** Header information in short *varchar* strings.

Examples

- **Example 1 –** Executes **rs_ticket** at regular intervals:

  ```
  Exec rs_ticket 'heartbeat', 'beat-sequence-number'
  ```
- **Example 2 –** To measure performance, execute the following from the primary database:

  ```
  Exec rs_ticket 'start'
  Execute replication benchmarks
  Exec rs_ticket 'stop'
  ```

Usage

- The **rs_ticket** stored procedure has a ticket version number V=2 and a ticket size of 1024 bytes.
- If your application understands only version 1 ticket, call **rs_ticket_v1** to generate ticket in version 1 format. The **rs_ticket_v1** syntax is:

```
rs_ticket_v1 h1 [, h2 [, h3 [, h4]]]
```

- **rs_ticket** executes the following command:

  ```
  rs_marker 'rs_ticket rs_ticket_param'
  ```

  To avoid issuing wrongly formatted **rs_marker** and to enforce the *rs_ticket_param* standard, you should invoke **rs_ticket** instead of **rs_marker**. If you call **rs_marker** directly and form an incorrect **rs_marker** subcommand, the Replication Server refuses the **rs_marker** and shuts down the RepAgent connection. In this case, you must skip **rs_marker** from the transaction log, which may cause data loss.
- The Replication Server EXEC, DIST, RSI, and DSI modules parse and process **rs_ticket** subcommand:
  - When EXEC processes **rs_ticket**, it appends a timestamp, and then the total bytes received from RepAgent after *rs_ticket_param*. An EXEC timestamp takes the form "EXEC(spid)=mm/dd/yy hh:mm:ss.ddd". The byte information is "B(spid)=ddd". EXEC writes **rs_ticket** back to inbound queue.
  - When DIST processes **rs_ticket**, it appends another timestamp to *rs_ticket_param*. A DIST timestamp takes the form "DIST(spid)=mm/dd/yy hh:mm:ss.ddd".
  - When RSI processes **rs_ticket**, it appends yet another timestamp to *rs_ticket_param*. An RSI timestamp takes the form "RSI(spid)=mm/dd/yy hh:mm:ss.ddd".
  - When DSI processes **rs_ticket**, it appends yet another timestamp to *rs_ticket_param*. A DSI timestamp takes the form "DSI(spid)=mm/dd/yy hh:mm:ss.ddd".
- There are no subscriptions for **rs_ticket**. DIST does not send **rs_ticket** to DSI unless there is at least one subscription from the replicate site.
- **rs_ticket** is lightweight and nonintrusive and can be used in test environments as well as production environments.
- **rs_ticket** lets you know, without quiescing the Replication Server, when the data has been completely flushed out of replication path.
- The movement of **rs_ticket** is tracked by the EXEC, DIST, RSI, and DSI threads through RSTicket counter. Each thread has one RSTicket counter which is increased by one whenever the corresponding thread receives **rs_ticket**. This counter is never reset.

  You can monitor the module that **rs_ticket** has reached by sampling the RSTicket counters. RMS or other Replication Server monitoring tool uses these counters to produce EXEC, DIST, RSI, and DSI heartbeat.

  You can also monitor the health of the replication path by sending an **rs_ticket** at primary and checking the RSTicket counters. If RSTicket counter of a module is not increasing, it shows that replication path at this stage is broken.
- You must not mark **rs_ticket** for replication.
- Use **rs_ticket** only when Replication Server is 15.0 or higher.

Resynchronizing Using a Third-Party Dump Utility

Coordinate resynchronization after you dump the primary database using a third-party dump utility, such as a disk snapshot.

Third-party tools do not interact as closely with the primary database as native database dump utilities. If your third-party tool does not record anything in the primary database transaction log that RepAgent can use to generate a dump database marker, generate your own dump database markers to complete the resynchronization process. See your third-party tool documentation.

1. Stop replication processing by RepAgent. In Adaptive Server, execute:

   ```
   sp_stop_rep_agent database
   ```
2. Suspend the Replication Server DSI connection to the replicate database:

   ```
   suspend connection to dataserver.database
   ```
3. Instruct Replication Server to remove data from the replicate database outbound queue and wait for a resync marker from the primary database RepAgent:

   ```
   resume connection to data_server.database skip to
   resync marker
   ```
4. Obtain a dump of the primary database contents using the third-party dump utility.
5. Determine the dump point based on information from the primary database when you took the dump, or information from the third-party tool. With a third-party tool, you are responsible for determining the dump point. For example, if you are using a disk replication tool, you can temporarily halt activity at the primary database to eliminate transactions in progress from the disk snapshot, and then use the “end of transaction log” point as the dump database marker.
6. Execute the **rs_marker** stored procedure on the primary database for RepAgent to mark the end of the dump position that you obtained in step 5:

   ```
   rs_marker “dump database database_name 'current date' oqid"
   ```

   where *current date* is any value in `datetime` format and *oqid* is any valid hexadecimal value. See *Replication Server Reference Manual > Topics > Datatypes > Date/time, and Date and Time Datatypes > Entry Format for Date/Time Values*.

   For example, you can mark the end of the dump position on the `rdb1` database with a date and time value of "20110915 14:10:10" and a value of 0x0003 for *oqid*:

   ```
   rs_marker “dump database rdb1 '20110915 14:10:10' 0x0003"
   ```

   RepAgent automatically generates a dump database marker for the point you marked in step 6, and sends the dump database marker to Replication Server.
7. Instruct RepAgent to start in resync mode and send a resync marker to Replication Server:

   - If the truncation point has not been moved from its original position, execute this command in Adaptive Server:

```
sp_start_rep_agent database, 'resync'
```

- If the truncation point has been moved from its original position, execute this command in Adaptive Server:

  ```
  sp_start_rep_agent database, 'resync purge'
  ```

8. Verify that DSI has received and accepted the resync marker from Replication Agent by looking for this message in the Replication Server system log:

   ```
   DSI for data_server.database received and processed
   Resync Database Marker. Waiting for Dump Marker.
   ```

9. Verify that Replication Server has processed the dump database marker by looking for this message in the Replication Server system log:

   ```
   DSI for data_server.database received and processed
   Dump Marker. DSI is now suspended. Resume after
   database has been reloaded.
   ```

   When Replication Server receives the dump marker, the DSI connection automatically suspends.

10. Apply the dump of the primary database from the third-party tool to the replicate database. See your Adaptive Server and third-party tool documentation.
11. After you apply the dump to the replicate database, resume DSI:

    ```
    resume connection to data_server.database
    ```

Exact Numeric (Decimal) Datatypes

Learn about exact numeric (decimal) datatypes.

Replication Server supports the following exact numeric (decimal) datatypes:

- *decimal* – exact decimal numbers between -10^{38} and 10^{38} -1, inclusive.
- *numeric* – exact decimal numbers between -10^{38} and 10^{38} -1, inclusive.

When you create a replication definition, omit the length and precision from *numeric* datatype declarations. Replication Server processes *numeric* values without affecting precision.

Note: If you are using a numeric datatype in a **where** clause in a replication definition, the value must include the precision information.

Identity columns use *numeric* as the underlying datatype, with exact decimal numbers of scale 0 between 1 and 10^{38} -1, inclusive.

When you create a replication definition for a table that contains an *identity* column, specify “identity” as the datatype for the column.

This command is applied to the replicated table before an **insert** command:

```
set identity_insert table_name on
```

This command is applied to the replicated table after an **insert** command:

```
set identity_insert table_name off
```

Identity columns are never updated by the **update** command.

If the replicate data server is Adaptive Server and a table contains an *identity* column, the maintenance user must be the owner of the table (or must be the "dbo" user or aliased to the "dbo" login name) at the replicate database in order to use the Transact-SQL **identity_insert** option.

HVAR Processing and Limitations

HVAR applies only the net-row changes of a transaction while maintaining the original commit order, and guarantees transactional consistency even as it skips intermediate row changes.

This has several implications:

- **Insert** triggers do not fire, as the HVAR process performs a bulk load of net new rows directly into the table. **Update** and **delete** triggers continue to fire when Replication Server applies the net results of compilation to the replicate database. However, row modifications that Replication Server compiles, and that are no longer in the net results, are invisible to the triggers. Triggers can detect only the final row images.

  Suppose you use Replication Server to audit user updates using a `last_update_user` column in a table schema with a trigger logic that associates a user to any column in the table modified by the user. If userA modifies `colA` and `colC` in the table and then userB modifies `colB` and `colD`, when the trigger fires, the trigger logic can detect only the last user who modified the table, and therefore the trigger logic associates userB as the user that modified all four columns. If you define triggers that contain similar logic where every individual row modification must be detected, you may have to disable HVAR compilation for that table.
- HVAR does not apply row changes in the same order in which the row changes are logged. To apply changes to a replicated table in log order, disable HVAR compilation for that table.
- If there are referential constraints on replicate tables, you must specify the constraints in replication definitions. To avoid constraint errors, HVAR loads tables according to replication definitions.
- Replication Server does not support customized function strings or any parallel DSI serialization methods, except for the default **wait_for_commit** method, when you enable HVAR. HVAR treats customized function strings as noncompilable commands.
- Replication Server reverts to log-order row-by-row continuous replication when it encounters:
  - Noncompilable commands – stored procedures, SQL statements, data definition language (DDL) transactions, system transactions, and Replication Server internal markers.
  - Noncompilable transactions – a transaction that contains noncompilable commands.
  - Noncompilable tables – tables with HVAR disabled, with modified function strings, and with referential constraint relationships with tables that HVAR cannot compile.

- If the replication definition does not include the **replicate minimal columns** clause, HVAR automatically changes a primary-key **update** to a **delete** followed by an **insert**. A primary-key update is either one of:
  - An update that affects the primary key of a table where the primary key is defined in the replication definition of the table, or,
  - An update that affects any column, except for `text` and `image` columns, when no replication definition exists. In this case, Replication Server assumes all the columns are part of the primary key since there is no specific primary-key definition from a replication definition.
- If the replication definition includes the **replicate minimal columns** clause, HVAR automatically identifies the table as noncompilable. The **update** operation applied to the table is noncompilable because HVAR cannot transform the **update** to a pair of operations consisting of a **delete** and an **insert**. Within the transaction group that HVAR is processing, HVAR can successfully compile into the net-change database all operations that HVAR processed before HVAR encountered the noncompilable primary-key **update** operation. However, within the transaction group, HVAR marks as noncompilable, the initial noncompilable primary-key **update** and all operations that follow it. The noncompilable state of the table is transient and lasts only for the duration of the same transaction group that HVAR is processing .
- HVAR ignores parameters, such as **dsi_partition_rule** that can stop transaction grouping.
- If errors occur during HVAR processing, Replication Server retries compilation with progressively smaller transaction groups until it identifies the transaction that failed compilation, then applies the transaction using continuous replication.
- To realize performance benefits, keep the primary and replicate databases synchronized to avoid the overhead of additional processing by Replication Server when errors occur. You can set **dsi_command_convert** to **i2di,u2di** to synchronize the data although this also incurs a processing overhead. If the databases are synchronized, reset **dsi_command_convert** to **none**.
- HVAR performs row-count validation to ensure replication integrity. The row-count validation is based on compilation. The expected row count is the number of rows remaining after compilation.
- When there are columns with `identity` datatype in a replication definition, Replication Server executes these commands in the replicate database:
  - **set identity_insert_***table_name* **on** before identity column inserts and **set identity_insert_***table_name* **off** after identity column inserts.
  - **set identity_update_***table_name* **on** before identity column updates and **set identity_update_***table_name* **off** after identity column updates.

Obtaining Help and Additional Information

Use the Sybase Getting Started CD, Product Documentation site, and online help to learn more about this product release.

- The Getting Started CD (or download) – contains release bulletins and installation guides in PDF format, and may contain other documents or updated information.
- Product Documentation at *http://sybooks.sybase.com/* – is an online version of Sybase documentation that you can access using a standard Web browser. You can browse documents online, or download them as PDFs. In addition to product documentation, the Web site also has links to EBFs/Maintenance, Technical Documents, Case Management, Solved Cases, Community Forums/Newsgroups, and other resources.
- Online help in the product, if available.

To read or print PDF documents, you need Adobe Acrobat Reader, which is available as a free download from the *Adobe* Web site.

Note: A more recent release bulletin, with critical product or document information added after the product release, may be available from the Product Documentation Web site.

Technical Support

Get support for Sybase products.

If your organization has purchased a support contract for this product, then one or more of your colleagues is designated as an authorized support contact. If you have any questions, or if you need assistance during the installation process, ask a designated person to contact Sybase Technical Support or the Sybase subsidiary in your area.

Downloading Sybase EBFs and Maintenance Reports

Get EBFs and maintenance reports from the Sybase Web site.

1. Point your Web browser to *http://www.sybase.com/support*.
2. From the menu bar or the slide-out menu, under **Support**, choose **EBFs/Maintenance**.
3. If prompted, enter your MySybase user name and password.
4. (Optional) Select a filter from the **Display** drop-down list, select a time frame, and click **Go**.
5. Select a product.

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

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

Sybase Product and Component Certifications

Certification reports verify Sybase product performance on a particular platform.

To find the latest information about certifications:

- For partner product certifications, go to *http://www.sybase.com/detail_list?id=9784*
- For platform certifications, go to *http://certification.sybase.com/ucr/search.do*

Creating a MySybase Profile

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

1. Go to *http://www.sybase.com/mysybase*.
2. Click **Register Now**.

Accessibility Features

Accessibility ensures access to electronic information for all users, including those with disabilities.

Documentation for Sybase products is available in an HTML version that is designed for accessibility.

Vision impaired users can navigate through the online document with an adaptive technology such as a screen reader, or view it with a screen enlarger.

Sybase HTML documentation has been tested for compliance with accessibility requirements of Section 508 of the U.S Rehabilitation Act. 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 may 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 the Sybase Accessibility site: *http://www.sybase.com/products/accessibility*. The site includes links to information about Section 508 and W3C standards.

You may find additional information about accessibility features in the product documentation.