



ASE-to-ASE Replication Quick Start Guide

Replication Server[®] 15.7.1

SP100

DOCUMENT ID: DC01251-01-1571100-01

LAST REVISED: May 2013

Copyright © 2013 by Sybase, Inc. All rights reserved.

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

Upgrades are provided only at regularly scheduled software release dates. No part of this publication may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without the prior written permission of Sybase, Inc.

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

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world.

Java and all Java-based marks are trademarks or registered trademarks of Oracle and/or its affiliates in the U.S. and other countries.

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

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

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

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

Contents

Conventions	1
Before You Begin	5
Prerequisites	5
Using the Examples to Set Up Your Replication Environment	5
Identifying the Sybase Installation Directory	6
Downloading Replication Server from SPDC	6
Downloading Replication Server EBFs	7
Installation and Configuration	9
Mounting the Installation Media	9
Installing Replication Server in GUI Mode	10
Configuring a Replication Server	14
Verifying the Configuration	19
Multisite Availability	21
Multisite Availability Diagram	21
Adding the Primary Database to the Replication Server	21
Adding the Replicate Database to the Replication Server	24
Marking the Primary Database for Replication	26
Replicating DDL	28
Replicating DML	29
Warm Standby Applications	31
Warm Standby Application Diagrams	31
Creating a Logical Connection	32
Adding the Active Database to the Replication Server	32
Marking the Active Database for Replication	36
Adding the Standby Database to the Replication Server	37
Initializing the Standby Database	40

Switching the Active and Standby Databases	43
Replicating DDL in a Warm Standby Environment	46
Replicating DML in a Warm Standby Environment	47
Tracing the Replication Server Transactions to Target Databases	48
Replication Definitions for Improving Performance	49
Creating a Replication Definition	49
Using SQL Statement Replication for Warm Standby	50
Materialization and Resynchronization	53
Scenario 1	53
Scenario 2	60
Scenario 3	64
Scenario 4	68
Scenario 5	74
Scenario 6	75
Index	79

Conventions

These style and syntax conventions are used in Sybase® documentation.

Style conventions

Key	Definition
<code>monospaced (fixed-width)</code>	<ul style="list-style-type: none"> • SQL and program code • Commands to be entered exactly as shown • File names • Directory names
<i>italic monospaced</i>	In SQL or program code snippets, placeholders for user-specified values (see example below).
<i>italic</i>	<ul style="list-style-type: none"> • File and variable names • Cross-references to other topics or documents • In text, placeholders for user-specified values (see example below) • Glossary terms in text
bold san serif	<ul style="list-style-type: none"> • Command, function, stored procedure, utility, class, and method names • Glossary entries (in the Glossary) • Menu option paths • In numbered task or procedure steps, user-interface (UI) elements that you click, such as buttons, check boxes, icons, and so on

If necessary, an explanation for a placeholder (system- or setup-specific values) follows in text. For example:

Run:

```
installation directory\start.bat
```

where *installation directory* is where the application is installed.

Syntax conventions

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

Case-sensitivity

- All command syntax and command examples are shown in lowercase. However, replication command names are not case-sensitive. For example, **RA_CONFIG**, **Ra_Config**, and **ra_config** are equivalent.
- Names of configuration parameters are case-sensitive. For example, **Scan_Sleep_Max** is not the same as **scan_sleep_max**, and the former would be interpreted as an invalid parameter name.
- Database object names are not case-sensitive in replication commands. However, to use a mixed-case object name in a replication command (to match a mixed-case object name in the primary database), delimit the object name with quote characters. For example: **pdb_get_tables "TableName"**
- Identifiers and character data may be case-sensitive, depending on the sort order that is in effect.
 - If you are using a case-sensitive sort order, such as “binary,” you must enter identifiers and character data with the correct combination of uppercase and lowercase letters.
 - If you are using a sort order that is not case-sensitive, such as “nocase,” you can enter identifiers and character data with any combination of uppercase or lowercase letters.

Terminology

Replication Agent™ is a generic term used to describe the Replication Agents for Adaptive Server® Enterprise, Oracle, IBM DB2 UDB, and Microsoft SQL Server. The specific names are:

- RepAgent – Replication Agent thread for Adaptive Server Enterprise
- Replication Agent for Oracle

- Replication Agent for Microsoft SQL Server
- Replication Agent for UDB – for IBM DB2 on Linux, Unix, and Windows
- Replication Agent for DB2 for z/OS

Before You Begin

Learn about the prerequisites, guidelines, and assumptions for creating a simple replication environment.

Prerequisites

Become familiar with the prerequisites for setting up the replication system.

- You must be familiar with Adaptive Server Enterprise data servers and have an understanding of Sybase replication.
- You must have a valid license for Replication Server® and Adaptive Server Enterprise (ASE).
 - If you purchased your product from Sybase or an authorized Sybase reseller, go to the secure Sybase Product Download Center (SPDC) at <https://sybase.subscribenet.com> and log in to generate license keys. The license generation process may vary slightly, depending on whether you ordered directly from Sybase or from a Sybase reseller
 - If you ordered your product under an SAP® contract and were directed to download from SAP Service Marketplace (SMP), you can use SMP at <http://service.sap.com/licensekeys> to generate license keys for Sybase products that use SySAM 2-based licenses.
- Use two different servers with the same named database. However, you can substitute the replicate database hosted on a different server to a different database on the same server that hosts the primary database.
- TCP/IP connectivity must be available.

Using the Examples to Set Up Your Replication Environment

Supplied examples walk you through setting up your replication environment.

The examples explain how to:

- Install Replication Server
- Configure and verify the Replication Server installation
- Add the primary and replicate database to the Replication Server
- Replicate data and DDL from a primary to a replicate database
- Manage warm standby applications
- Manage materialization

Follow these general instructions when using the examples to set up your replication environment:

Before You Begin

- Some directories, files, executable commands, and examples are provided only for Solaris; adjust them accordingly for Windows and for other UNIX and Linux platforms.
- Do not use Replication Server-reserved words for object names and connection names. Replication Server also reserves all keywords and identifiers beginning with “rs_”. See the *Replication Server Reference Manual > Topics > Reserved Words*.
- The values in the examples can be customized to meet the needs of your replication environment.

The examples make several assumptions:

- The simple replication environment uses three Adaptive Server databases and one Replication Server. The primary database is the source of changes to be replicated; the replicate database is where changes are applied, and the Replication Server System Database (RSSD) holds the system tables for one Replication Server. Each Replication Server requires an RSSD, and in a production environment, the RSSD must be maintained like any other critical database or file.

Note: The sample scenarios do not discuss the Embedded Replication Server System Database (ERSSD) database.

- You can have many Replication Servers in your production environment. The examples discuss how to install and configure a single Replication Server, which has a RSSD database in an existing Adaptive Server.
- The replication environment uses the `pubs2` database, which is available in the standard installation of Adaptive Server. See the *Adaptive Server Enterprise Installation Guide > Post Installation Tasks > Installing Sample Databases* for information on installing `pubs2` database.

Identifying the Sybase Installation Directory

The **SYBASE** environment variable identifies the installation path for installing Replication Server.

Select a path on a host drive to be the recipient of the Sybase installation and configuration activities. For example: `/software/sybase`

Downloading Replication Server from SPDC

Download Replication Server from the Software Product Download Center (SPDC) Website.

Prerequisites

You must have a login ID and password to download software from the SPDC Website.

Task

1. Go to <https://sybase.subscribenet.com/control/sybs/login> and enter your login ID and password.
2. Select Replication Server from the product list.
3. Select Replication Server again.
4. Select the Replication Server software you need by version and platform.
The Software Terms and Conditions page appears, listing countries and regions.
5. Select your country or region to read the terms. The software licence agreement for the software appears. Read the terms and either:
 - Select I Agree to accept the terms, go to the Product Download page, and continue with step 6, or
 - Select Cancel to return to the high-level Product Information page.
6. On the Product Download page, select FTP Download or Advanced Download Options for the Replication Server software. Download and save the compressed files to your local drive.

Downloading Replication Server EBFs

Download Replication Server EBFs from the Sybase Support page or the SAP® Service Marketplace (SMP). The location you use depends on how you purchased the product. .

- If you purchased the product directly from Sybase or from an authorized Sybase reseller:
 - a) Point your Web browser to <http://www.sybase.com/support>.
 - b) Select **Support > EBFs/Maintenance**.
 - c) If prompted, enter your MySybase user name and password.
 - d) Select Replication Server.
 - e) Locate the latest ESD for Replication Server under the specific platform.
 - f) Click the **Info** icon to display the EBF/Maintenance report, or click the product description to download the software.
- If you ordered your Sybase product under an SAP contract:
 - a) Point your browser to <http://service.sap.com/swdc> and log in if prompted.
 - b) Select **Search for Software Downloads** and enter the name of your product. Click **Search**.

Before You Begin

Installation and Configuration

Installation and configuration topics describe how to set up your Replication Server and then configure it to suit your environment.

Mounting the Installation Media

If you are installing using the CD or DVD, mount the installation media.

The location of the **mount** command is site-specific and may differ from the instructions shown here. If you cannot mount the installation media in the appropriate drive using the path shown, check your operating system documentation or contact your system administrator.

- On HP-UX:

Log out, log in as “root”, and issue:

```
mount -F cdrfs -o ro device_name /mnt/cdrom
```

Then, log out as “root” and log in again as “sybase.”

- On IBM AIX:

Log in as “sybase” and issue:

```
mount -v 'cdrfs' -r device_name /mnt/cdrom
```

- On Solaris:

The operating system automatically mounts the CD or DVD. Log in as “sybase.” If CD- or DVD-reading errors occur, check your operating system kernel to make sure that the ISO 9660 option is turned on. If you have previously installed a Sybase CD or DVD on your system, the # sign interferes with the installation process. Before installing your current CD or DVD, either:

- Restart your system, or,
- Eject the CD or DVD. Delete the *Volume Label* file in `/vol/dsk`, and reinsert the CD or DVD.

- On Linux and Linux on POWER:

Log in as “sybase” and issue:

```
# mount -t iso9660 /dev/cdrom /mnt/cdrom
```

Installing Replication Server in GUI Mode

The installer creates the target directory and installs the selected components into that directory.

Prerequisites

Verify that the drive on which you install the Replication Server has enough disk space for the components being installed, and at least 100MB of extra disk space for the installation program.

Task

1. Insert the Replication Server installation media in the appropriate drive, or download and extract the Replication Server installation image from the Sybase Product Download Center (SPDC) or the SAP Service Marketplace (SMP).

2. Launch the **setup** program.

If the installer does not start automatically, double-click `setup.exe` or select **Start > Run** and enter the following, where *x* is your CD or DVD drive:

```
x:\setup.exe
```

If there is not enough disk space in the temporary disk space directory, set the environment variable `TMP` to *directory_name* before running the installer again, where *directory_name* is where the installation program writes the temporary installation files. When specifying the *directory_name*, include its full path.

3. If you downloaded the product from SPDC or SMP, go to the directory where you extracted the installation image and start the installer:

```
./setup.bin
```

If you are installing using the CD or DVD, mount it and start the installer.

- On HP-UX:

```
cd /cdrom  
./setup.bin
```

- On IBM AIX:

```
cd /device_name  
./setup.bin
```

- On Solaris:

```
cd /cdrom/Volume Label  
./setup.bin
```

- On Linux and Linux on POWER:

```
cd /mnt/cdrom  
./setup.bin
```

where:

- *cdrom* and */mnt/cdrom* is the directory (mount-point) you specified when mounting the CD or DVD drive.
- *device_name* is the name of the CD or DVD device drive.
- *setup.bin* is the executable file name to install Replication Server.

If there is not enough disk space in the temporary disk space directory, set the environment variable IATEMPDIR to *tmp_dir* before running the installer again, where *tmp_dir* is where the installation program writes the temporary installation files. When specifying the *tmp_dir*, include its full path.

4. In the Introduction window, click **Next**.
5. Specify where to install Replication Server:

Option	Description
Click Choose .	To select an installation directory. A window appears, and from that window browse for and select the directory.
Enter a new directory path.	To create a new directory.
Click Restore Default Folder .	To restore to the default directory if you do not want to use the directory that you have entered.

- If the directory you chose does not exist, click **Yes** to create it.
- If it does exist, and already contains a Replication Server installation, you are warned that you will overwrite the older version. Click **Next**.

The installer checks that the version you want to update is compatible with the version of Replication Server you are installing. If the version is incompatible, the Check Upgrade Incompatible Version dialog appears, and you see this:

```
Warning: The current "Sybase Replication Server" in your
destination directory is not compatible with this version
upgrade; some bug fixes may be unavailable
if you proceed. See the release note for more information.
```

You may see a similar message if your Replication Server is an out-of-band release, such as an emergency bug fix, one-off, controlled, or instrumental release:

```
Warning: The current "Sybase Replication Server" in your
destination directory is an out-of-band release; some bug fixes
may be unavailable if you proceed.
Verify that the bug fixes you need are in this newer version
before proceeding with the upgrade.
```

If you see such messages, click **Cancel** to stop the installation process. To override the error and continue with the installation, select **Proceed installation with incompatible version** and click **Next**.

Warning! Upgrading to an incompatible version may cause software regression. SAP recommends that you cancel the installation and obtain a compatible version of Replication Server.

Note: If you perform the installation in a silent, or unattended mode, and the versions are incompatible, the installer prompts you to rerun the installer with this argument, then quit:

```
DALLOW_UPGRADE_TO_INCOMPATIBLE_VERSION=true
```

6. Select the type of installation:

Option	Description
Typical	Installs the default components. This is recommended for most users.
Full	Installs every Replication Server component, including all the supported language modules.
Custom	Lets you select the components to install. Some components are automatically installed if they are required to run your selected components.

Click **Next**.

7. Select the geographic location, agree to the license agreement, then click **Next**.

8. In the Sybase Software Asset Management License Server window:

Option	Choose
Specify license keys	<p>Either:</p> <ul style="list-style-type: none"> • Click Browse to select the license file. • Use Shift+Click or Ctrl+Click to select multiple license files. The license pane displays the license information. • Copy and paste the license information directly in the license pane. <p>Click Next.</p> <p>If you specify a served license key, the installer prompts you to install a new SySAM license server. Select:</p> <ul style="list-style-type: none"> • Next – to install the new SySAM license server. Follow the installation prompts. • Previous – and select Use previously deployed license server if you have an existing SySAM license server on the same host.

Option	Choose
Use previously deployed license server	Enter the: <ul style="list-style-type: none"> • Host name of the machine where the license server is running • Port number if the port number you are using is not the default Click Next .
Continue installation without a license key	If you do not have licenses for any of the Replication Server components, select this option, then click Next to proceed. The installer allows you to install and use Replication Server components without a license for a grace period of 30 days. To continue using these components after the end of the grace period, obtain valid licenses, and install these licenses using the Replication Server License Installer.

Use **sysadmin lmconfig** to configure and show license management-related information in Replication Server. See *Replication Server Reference Manual > Replication Server Commands > sysadmin lmconfig*.

9. Set SySAM e-mail configuration if necessary. Click **Next**.
10. The installation summary window displays the selections you have made. Review the information, and click **Install**.
11. In the Start Sample Replication Server window, select:

Note: You cannot start the sample Replication Server on Linux on POWER because the ERSSD requires Sybase SQL Anywhere Server, which is not available on Linux on POWER.

Option	Description
Yes	To configure and start a sample Replication Server. The installer displays the configuration information for the sample Replication Server. Record this information. You can enter a maximum of 30 bytes in the password field. For a: <ul style="list-style-type: none"> • Single-byte character – enter 6 to 30 characters. • Double-byte character – enter 3 to 15 characters.

Option	Description
No	To manually configure a full-featured Replication Server and start a sample Replication Server after installation. The installer prompts you to either create a sample Replication Server directory or to proceed with the installation. Select either: <ul style="list-style-type: none"><li data-bbox="368 348 1178 409">• Yes – to create a sample Replication Server directory without starting the sample Replication Server, or,<li data-bbox="368 418 807 447">• No – to continue with the installation.

Click **Next**.

12. Click Done.

Next

Verify that the installation is valid and successful.

- View the log files in the `log` directory to check for errors. A valid installation does not contain the word “ERROR.”
- Check that the date of the `si_reg.xml` file in the `Sybase_Install_Registry` directory reflects the date of the current installation.

Configuring a Replication Server

Configure a Replication Server using the values in the example.

Prerequisites

- Allocate a disk partition of at least 20MB for each Replication Server you are installing. You can add more partitions later, if necessary.
- Ensure that the raw device or file system is available and has write permissions. If using a raw device, allocate the entire partition to the Replication Server. You can add more space for the stable device later, if necessary. If you allocate only a part of the partition to Replication Server, you cannot use the remainder for any other purpose.

Task

Each replication system has a single Replication Server that manages the overall environment information. This Replication Server is called an ID Server. A primary database is the source of replication and can only belong to one Replication Server environment. A replicate database is the destination database.

This example creates a single Replication Server, which has a Replication Server System Database (RSSD) in an existing Adaptive Server. The script, in this example, creates the data and log devices that the RSSD database will use. In this example:

- PRS – name of the Replication Server
 - sunak1505i – name of the Adaptive Server where the RSSD will reside
1. Go to the `$SYBASE` directory.
 2. Add the primary Adaptive Server, the replicate Adaptive Server, the Replication Server, and the server containing the RSSD database to the interfaces file.
 3. Set the environment variables by sourcing the `SYBASE.csh` file.
 4. Go to `$SYBASE/REP-15_5/init/rs`.
 5. Make a copy of the `install.rs` file and rename it as `PRS.rs`.
 6. Edit the parameter values in `PRS.rs` file.

Note: You need to modify the parameters described in the *Sample values for PRS.rs file* table.

Table 1. Sample Values for PRS.rs File

Parameter	Description	Values
<code>sybinit.release_directory</code>	Valid path where Replication Server instance is created (<code>\$SYBASE</code>).	<code>/opt/sybase</code>
<code>rs.rs_idserver_name</code>	Name of the Replication Server.	PRS
<code>rs.rs_id_server_is_rs_server</code>	Specifies whether the Replication Server created is the ID server or not.	yes
<code>rs.rs_idserver_user</code>	Because this is the ID server, user name that other Replication Servers will use to connect to this Replication Server. If this is not the ID server, then it is the user name of the ID server identified in the parameter, <code>rs.rs_idserver_name</code> .	<code>PRS_id_user</code> Note: The default name is the name of the Replication Server followed by <code>_id_user</code> . For example, <code>PRS_id_user</code> .
<code>rs.rs_idserver_pass</code>	Password of the ID server user.	<code>PRS_id_passwd</code>
<code>rs.rs_name</code>	The name of the Replication Server.	PRS

Parameter	Description	Values
rs.rs_rs_sa_pass	Password for the sa user ID.	sa_pass Note: You can enter a maximum of 30 bytes in the password field. For: <ul style="list-style-type: none"> single-byte character – enter 6 to 30 characters. double-byte character – enter 3 to 15 characters.
rs.rs_needs_repagent	Specifies whether the RSSD needs a RepAgent. This is needed when multiple replication servers are used to replicate data from the primary to the replicate.	no Note: If the primary database and the replicate database will be replicating through multiple Replication Servers, enter yes.
rs.rs_rssd_sqlsrvr	The name of the Adaptive Server where the RSSD will reside.	sunak1505i
rs.rs_rssd_db	Specifies the name of the RSSD database.	USE_DEFAULT Note: The default name is the name of the Replication Server followed by _RSSD. For example, PRS_RSSD.
rs.rs_create_rssd	Specifies whether the script will create the RSSD database.	yes
rs.rs_rssd_sa_login	Specifies the name of the login that has “sa” privileges.	sa
rs.rs_rssd_sa_pass	Specifies the password of the login entered in the rs.rs_rssd_sa_login parameter.	password
rs.rs_rssd_prim_user	User that the Replication Server uses to log into the RSSD.	PRS_RSSD_prim Note: The default name is the name of the RSSD database followed by _prim. For example, PRS_RSSD_prim.

Parameter	Description	Values
rs.rs_rssd_prim_pass	Specifies the password of the login entered in the rs.rs_rssd_prim_user parameter.	PRS_RSSD_prim_ps
rs.rs_rssd_maint_user	User that the Replication Server uses to log into the RSSD when getting work from other Replication Servers.	PRS_RSSD_maint Note: The default name is the name of the RSSD database followed by _maint. For example, PRS_RSSD_maint.
rs.rs_rssd_maint_pass	Specifies the password of the login entered in the rs.rs_rssd_maint_user parameter.	PRS_RSSD_maint_ps
rs.rs_rsddb_size	Specifies the size (in MB) of the system database device.	40
rs.rs_rssd_log_size	Specifies the size (in MB) of the RSSD database log device.	32
rs.rs_rssd_db_device_name	The name of the ASE device that stores the data portion of the RSSD database.	PRS_RSSD_data
rs.rs_create_rssd_data_base_dev	Specifies whether a new ASE device needs to be created for the data portion of the RSSD database.	yes
rs.rs_rssd_db_device_path	Specifies the physical path for file system and file (or raw device) for the RSSD database data device.	/opt/sybase/ PRS_RSSD_data
rs.rs_rsddb_device_size	Specifies the size (in MB) of the RSSD database device.	40 The value must be the size of the data portion of the RSSD database specified in the rs.rs_rsddb_size or greater.
rs.rs_rssd_log_device_name	Specifies the logical name of RSSD database log device.	PRS_RSSD_log

Parameter	Description	Values
rs.rs_create_rssd_log_dev	Specifies whether the device for the log will be created.	yes Note: If the same device is used for both data and log, then enter no for this parameter.
rs.rs_rssd_log_device_path	Specifies the physical path for file system and file (or raw device) for the RSSD database log device.	/opt/sybase/ PRS_RSSD_log
rs.rs_rssd_log_device_size	Specifies the size (in MB) of the RSSD database log device.	32 The value must be the size of the log portion of the RSSD database specified in the rs.rs_rsddb_log_size or greater.
rs.rs_diskp_name	Specifies the path and raw device (or file name) for the Replication Server stable device. This file or device must already exist.	/opt/sybase/ PRSpart1.dat
rs.rs_diskp_lname	Specifies the logical name of the stable device partition.	part1
rs.rs_diskp_size	Specifies the size, in MB, of the stable device partition.	20 The minimum size is 20MB.
rs.rs_rs_user	User that other Replication Servers can use when a route between Replication Servers is created .	PRS_rsi Note: The default name is the name of the Replication Server followed by _rsi. For example, PRS_rsi.
rs.rs_rs_pass	Specifies the password of the login entered in the rs.rs_rs_user parameter.	PRS_rsi_ps

Parameter	Description	Values
<code>rs.rs_ltm_rs_user</code>	Default user for Replication Agents to log into the Replication Server.	PRs_ra Note: The default name is the name of the Replication Server followed by <code>_ra</code> . For example, <code>PRs_ra</code> .
<code>rs.rs_ltm_rs_pass</code>	Specifies the password of the login entered in the <code>rs.rs_ltm_rs_user</code> parameter.	PRs_ra_ps

7. Save the file.
8. Go to the `$SYBASE/REP-15_5/install` directory.
9. Create the Replication Server and the RSSD:

```
./rs_init -r ../init/rs/PRs.rs
```

Note: If the `rs_init` command does not complete, check the log file in `$SYBASE/$SYBASE_REP/init/log`, correct the error, and resubmit the `rs_init`. If the problem occurred because the RSSD database had no space, then run the `isql` command on the data server and delete the RSSD database (for example, `PRs_RSSD`). If the `rs_init` command created the devices for the RSSD database, drop the devices (`sp_dropdevice`), and resubmit the `rs_init` command.

Verifying the Configuration

Verify that the Replication Server installation directory and subdirectories (`%SYBASE%` on Microsoft Windows platforms and `$SYBASE` on UNIX platforms) have been created, and that other required software has been installed.

1. Change to the directory where you installed Replication Server.
2. Set the environment variables by sourcing the `SYBASE.csh` file.
3. Log in to Replication Server:

```
isql -Usa -Psa_pass -SPRS
```

PRs – name of the Replication Server.

Note: The default system administrator user ID is "sa". The password to the "sa" user ID is the value that was filled in the resource file for `rs.rs_rs_sa_pass`.

4. Enter:

```
admin who
go
```

The output from the command looks similar to:

Installation and Configuration

Spid	Name	State	Info
27	DSI EXEC	Awaiting Command	
101			(1) sunak1505i.PRS_RSSD
20	DSI	Awaiting Message	101 sunak1505i.PRS_RSSD
26	SQM	Awaiting Message	101:0 sunak1505i.PRS_RSSD
21	dSUB	Sleeping	
15	dCM	Awaiting Message	
18	dAIO	Awaiting Message	
23	dREC	Sleeping	dREC
9	dDELSEG	Awaiting Message	
28	USER	Active	sa
14	dALARM	Awaiting Wakeup	
24	dSYSAM	Sleeping	

5. Verify the Replication Server version. Enter:

```
admin version
go
```

Note: The listing from the command looks similar to:

```
Version
-----
Replication Server/15.7.1/P/Sun_svr4/OS 5.8/1/OPT64/Sun
Apr 22 18:37:00 2012
```

6. Exit the `isql` session using:

```
quit
```

Multisite Availability

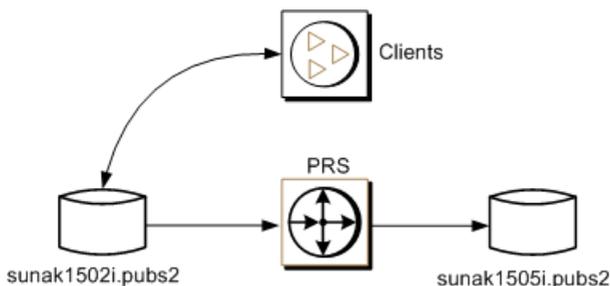
Multisite Availability (MSA) extends Replication Server replication capabilities and simplifies setting up a replication system.

Key features of MSA include:

- A simple replication methodology that requires only one replication definition for the primary database and only one subscription for each subscribing database.
- A replication filtering strategy that lets you choose whether or not to replicate individual tables, transactions, functions, system stored procedures, and data definition language (DDL).
- Replication of DDL to any replicate database—including non-warm standby databases. Replication to multiple replicate sites—for standby as well as nonstandby databases.

Multisite Availability Diagram

The diagram illustrates a simple example of multisite availability replication.



Adding the Primary Database to the Replication Server

Add a primary database to the Replication Server for MSA.

1. Add the Replication Server to the interfaces file of the data server that hosts the primary database.

Note: Restart the Adaptive Server if the interfaces file has changed.

2. Go to `$SYBASE/$SYBASE_REP/init/rs`.
3. Make a copy of the `setupdb.rs` file and rename it as `myprimary.rs`.
4. Edit the `myprimary.rs` file.

Note: You need to modify the parameters described in the "Sample values for myprimary.rs file" table.

Table 2. Sample Values for the myprimary.rs File

Parameter	Description	Value
sybinit.release_directory	Valid path of the Replication Server software (\$SYBASE).	/opt/sybase
rs.rs_name	Name of the Replication Server.	PRS
rs.rs_rs_sa_user	Specifies the user ID that has "sa" privileges on Replication Server.	sa
rs.rs_rs_sa_pass	Specifies the password of the "sa" user.	sa_pass
rs.rs_ds_name	Name of the data server that hosts the primary database.	sunak1502i
rs.rs_ds_sa_user	Specifies the user ID that has "sa" privileges on data server.	sa
rs.rs_ds_sa_password	Specifies the password of the "sa" user for the data server.	password
rs.rs_db_name	Specifies the name of the primary database.	pubs2
rs.rs_needs_repagent	Specifies whether you plan to replicate from specified primary database.	yes
rs.rs_db_maint_user	Specifies the user ID whose work is not replicated when logged on the primary database. The user is called maintenance user.	<database>_maint <hr/> Note: If the user ID does not exist, the script creates the user ID on the database. The user ID cannot be the name of an alias. <hr/>
rs.rs_db_maint_password	Specifies the password for the maintenance user.	<database>_maint_ps

Parameter	Description	Value
rs.rs_ltm_rs_user	Specifies the user that the Replication Agent will use to log into the Replication Server. The name must exist. This name typically comes from values that were set up during Replication Server creation time: rs.rs_ltm_rs_user.	PRS_ra
rs.rs_ltm_rs_pass	Specifies the password of the rs.rs_ltm_rs_user.	PRS_ra_ps
rs.rs_db_physical_for_logical	Specifies whether this is a warm standby database.	no

5. Save the file.
6. Go to `$SYBASE/$SYBASE_REP/install`.
7. Create the connection from the primary database to the Replication Server by running the resource file:

```
./rs_init -r ../init/rs/myprimary.rs
```

If the **rs_init** command fails, check the log file in `$SYBASE/$SYBASE_REP/init/logs`, correct the issue, then disable the RepAgent as follows:

- a. Log in to the primary Adaptive Server using an “sa” user role and access the primary database.
- b. Disable the RepAgent thread in the primary database:

```
sp_config_rep_agent pubs2, 'disable'
go
```

Re-run **rs_init** command

8. Validate the primary connection:

```
isql -Usa -Psa_pass -SPRS
```

9. Enter:

```
admin who
go
```

The output from the command looks similar to:

```
Spid      Name      State      Info
-----
48        DSI EXEC  Awaiting Command  102(1) sunak1502i.pubs2
33        DSI      Awaiting Message  102 sunak1502i.pubs2
35        DIST     Awaiting Wakeup   102 sunak1502i.pubs2
36        SQT      Awaiting Wakeup   102:1 DIST sunak1502i.pubs2
34        SQM      Awaiting Message  102:1 sunak1502i.pubs2
32        SQM      Awaiting Message  102:0 sunak1502i.pubs2
37        REP AGENT Awaiting Command  sunak1502i.pubs2
39        NRM      Awaiting Message  sunak1502i.pubs2
```

Multisite Availability

27	DSI EXEC	Awaiting Command	101 (1) sunak1505i.PRS_RSSD
20	DSI	Awaiting Message	101 sunak1505i.PRS_RSSD
26	SQM	Awaiting Message	101:0 sunak1505i.PRS_RSSD
21	dSUB	Sleeping	
15	dCM	Awaiting Message	
18	dAIO	Awaiting Message	
23	dREC	Sleeping	dREC
9	dDELSEG	Awaiting Message	
49	USER	Active	sa
14	dALARM	Awaiting Wakeup	
24	dSYSAM	Sleeping	

10. Exit the `isql` session.

Adding the Replicate Database to the Replication Server

Add a replicate database to the Replication Server for MSA.

1. Add the replicate Adaptive Server to the `interfaces` file of the Replication Server.

Note: Restart the Replication Server if the `interfaces` file has changed.

2. Go to `$SYBASE/$SYBASE_REP/init/rs`.
3. Make a copy of the `setupdb.rs` file and rename it as `myreplicate.rs`.
4. Edit the `myreplicate.rs` file.

Note: You need to modify the parameters described in the "Sample values for `myreplicate.rs` file" table.

Table 3. Sample Values for the `myreplicate.rs` file

Parameter	Description	Value
<code>sybinit.release_directory</code>	Valid path of the Replication Server software (<code>\$SYBASE</code>).	<code>/opt/sybase</code>
<code>rs.rs_name</code>	Name of the Replication Server.	PRS
<code>rs.rs_rs_sa_user</code>	Specifies the user ID that has "sa" privileges on Replication Server.	sa
<code>rs.rs_rs_sa_pass</code>	Specifies the password of the "sa" user.	sa_pass
<code>rs.rs_ds_name</code>	Name of the data server that hosts the replicate database.	sunak1505i
<code>rs.rs_ds_sa_user</code>	Specifies the user ID that has "sa" privileges on data server.	sa

Parameter	Description	Value
rs.rs_ds_sa_password	Specifies the password of the “sa” user for the data server.	password
rs.rs_db_name	Specifies the name of the replicate database.	pubs2
rs.rs_needs_repagent	Specifies whether you plan to replicate from specified rs.rs_db_name.	no
rs.rs_db_maint_user	Specifies the user ID who will apply the work at the replicate database. The user is called the maintenance user.	<database>_maint Note: If the user ID does not exist the script creates the user ID on the replicate database. The user ID cannot be the name of an alias.
rs.rs_db_maint_password	Specifies the password for the user specified in rs.rs_db_maint_user.	<database>_maint_ps
rs.rs_db_physical_for_logical	Specifies whether this is a warm standby database.	no

5. Save the file.
6. Go to `$SYBASE/$SYBASE_REP/install`.
7. Create the connection from the primary database to the Replication Server by running the resource file:

```
./rs_init -r ../init/rs/myreplicate.rs
```

If the **rs_init** command fails, correct the issue, and re-run the **rs_init**.

8. Validate the primary connection:

```
isql -Usa -Psa_pass -SPRS
```

9. Enter:

```
admin who
go
```

The output from the command looks similar to:

```
Spid      Name      State      Info
-----
48        DSI EXEC  Awaiting Command  102(1) sunak1502i.pubs2
33        DSI      Awaiting Message  102 sunak1502i.pubs2
35        DIST     Awaiting Wakeup   102 sunak1502i.pubs2
36        SQT      Awaiting Wakeup   102:1 DIST sunak1502i.pubs2
34        SQM      Awaiting Message  102:1 sunak1502i.pubs2
32        SQM      Awaiting Message  102:0 sunak1502i.pubs2
37        REP AGENT Awaiting Command  sunak1502i.pubs2
39        NRM      Awaiting Message  sunak1502i.pubs2
```

Multisite Availability

```
27      DSI EXEC      Awaiting Command      101(1) sunak1505i.PRS_RSSD
20      DSI           Awaiting Message      101 sunak1505i.PRS_RSSD
26      SQM           Awaiting Message      101:0 sunak1505i.PRS_RSSD
55      DSI EXEC      Awaiting Command      103(1) sunak1505i.pubs2
54      DSI           Awaiting Message      103 sunak1505i.pubs2
53      SQM           Awaiting Message      103:0 sunak1505i.pubs2
21      dSUB          Sleeping
15      dCM           Awaiting Message
18      dAIO          Awaiting Message
23      dREC          Sleeping              dREC
9       dDELSEG        Awaiting Message
56      USER          Active                sa
14      dALARM        Awaiting Wakeup
24      dSYSAM        Sleeping
```

10. Exit the `isql` session.

Marking the Primary Database for Replication

Use database replication definition and subscription to replicate the entire primary database.

1. Log in to the primary database with system administrator privileges:

```
% isql -Usa -P -Ssunak1502i
```

2. Connect to the `pubs2` database:

```
use pubs2
go
```

3. Mark the primary database for replication. For example:

```
sp_reptostandby pubs2, 'all'
go
```

4. Set the RepAgent parameter **send warm standby xacts** to true so that RepAgent sends Data Manipulation Transactions (DML) and Data Definition Language (DDL) to the replicate database. For example, at the primary data server, enter:

```
sp_config_rep_agent
pubs2, send_warm_standby_xacts, true
go
```

```
Parameter_Name Default_Value Config_Value Run_Value
-----
send warm standby xacts false true true
(1 row affected)
```

```
RepAgent configuration changed for database pubs2.
The changes will take effect the next time the
RepAgent thread is started. (return status = 0)
```

5. Stop and restart the RepAgent:

```
sp_stop_rep_agent pubs2
go
```

```
sp_start_rep_agent pubs2
go
```

6. Exit the **isql** session.

7. At the Replication Server, create a database replication definition that also replicates DDL:

```
isql -Usa -Psa_pass -SPRS
create database replication definition name
with primary at pds.pdb
replicate DDL
```

where:

- *name* – is the unique identifier for this replication definition.
- *pds* – is the name of the primary ASE data server.
- *pdb* – is the name of the primary database.

For example:

```
create database replication definition pubs2_repdef
with primary at sunak1502i.pubs2
replicate DDL
go
```

```
Database replication definition pubs2_repdef for
sunak1502i.pubs2 is created.
```

8. Create a database subscription for the replicate database. This example creates a database subscription that uses no materialization method and will replicate the truncate table command:

```
create subscription sub_name
for database replication definition name
with primary at pds.pdb
with replicate at rds.pdb
without materialization
subscribe to truncate table
```

where:

- *sub_name* – is the unique identifier for this subscription.
- *name* – is the unique identifier for the replication definition.
- *pds* – is the name of the primary ASE data server.
- *pdb* – is the name of the primary database.
- *rds* – is the name of the replicate ASE data server.

for example:

```
create subscription pubs2_sub
for database replication definition pubs2_repdef
with primary at sunak1502i.pubs2
with replicate at sunak1505i.pubs2
without materialization
subscribe to truncate table
go
```

Note: For the subscription to be successfully created or dropped, the connection to the replicate database must be available.

9. Check the subscription status at the primary and replicate data servers:

```
check subscription pubs2_sub
for database replication definition pubs2_repdef
with primary at sunak1502i.pubs2
with replicate at sunak1505i.pubs2
go
```

The status shows:

```
Subscription pubs2_sub is VALID at the replicate.
Subscription pubs2_sub is VALID at the primary.
```

The database is now ready for replication

Note: If the subscription results in errors, you can drop the subscription using the **drop subscription** command. For example:

```
drop subscription pubs2_sub
for database replication definition pubs2_repdef
with primary at sunak1502i.pubs2
with replicate at sunak1505i.pubs2
without purge
go
```

The replicate connection must be available before you can drop a subscription.

Replicating DDL

Replicate DDL.

1. Log in to the primary database using a user ID that exists on both the primary and replicate data server and that has permission to create a table on both the primary and replicate databases.

Note: Do not use the same maintenance user ID that you used to set up the primary connection.

2. Create a new table:

```
% isql -Usa -P -Ssunak1502i
use pubs2
go
create table t1 (a char(10), b integer, c text)
go
```

3. Create unique indexes to ensure data integrity:

```
create unique clustered index t1_idx1 on t1 (a,b)
go
```

4. Log in to the replicate database:

```
% isql -Usa -P -Ssunak1505i
use pubs2
go
```

5. Verify that the table and index exists in the replicate database. If the table and index do not exist, follow the instructions in steps 6 and 7 otherwise, go to step 8.
6. Check the Replication Server log file at: `$$SYBASE/REP- 15_5/install/PRS.log`. Correct the errors and restart the connection to the replicate database:

```
resume connection to rds.rdb
go
```

where:

- `rds` – is the name of the data server that hosts the replicate database.
- `rdb` – is the name of the replicate database.

Note: Make sure that the user making the corrections is not same the maintenance user that was used to set up the primary connection.

7. If you want Replication Server to skip any current transactions when trying to resume connection to the replicate database, use:

```
resume connection to rds.rdb
skip transaction
go
```

See the *Replication Server Reference Manual > Replication Server Commands > resume connection* for other available **resume connection** options.

Note: If you see this message:

```
"Message from server: Message: 2762, State 3, Severity 16 - 'The
'CREATE TABLE' command is not allowed within a multi-statement
transaction in the 'pubs2 database."
```

Ensure that the Replication Agent has been configured to send warm standby transactions and that the RepAgent has been stopped and restarted since the last time when the RepAgent configuration parameter was changed.

8. Grant **insert**, **update**, and **delete** permission for the new table to the replicate database maintenance user at the replicate database:

```
grant all on t1 to pubs2_maint
go
```

Replicating DML

Replicate DML.

1. Log in to the primary database using a user ID that has permission to **insert**, **update**, **delete** and **truncate** a table. See *Replication Server Administration Guide Volume 1 > Manage Database Connections > Manage the Maintenance User* for information on how to grant permission to the maintenance user ID.

Note: Do not use the same maintenance user ID that you have used to set up the primary connection.

2. In the primary database, insert a row to t1:

```
insert into t1 values('a',1,'this is the first row')
go
```

3. Check whether the row exists in the replicate database:

```
select * from t1
go
```

If the row does not exist, follow instructions in steps 4 and 5 otherwise, go to step 6.

4. Check the Replication Server log file at: `$$SYBASE/REP- 15_5/install/PRS.log`. Correct the errors and restart the connection to the replicate database:

```
resume connection to rds.rdb
go
```

5. If you want Replication Server to skip any current transaction when trying to resume connection to the replicate database, use:

```
resume connection to rds.rdb
skip transaction
go
```

See the *Replication Server Reference Manual > Replication Server Commands > resume connection* for other available options for the **resume connection** command.

6. Log in to the primary database and update the row:

```
update t1 set c = 'this is an update' where b = 1
go
```

7. Log in to the replicate database and verify that the row was updated:

```
select * from t1
go
```

8. Log in to the primary database and enter:

```
truncate table t1
go
```

9. Log in to the replicate database and enter:

```
select count (*) from t1
go
```

The number of rows at the replicate table, t1, should now be zero.

Warm Standby Applications

A warm standby application is a pair of Adaptive Server databases, one of which is a backup copy of the other.

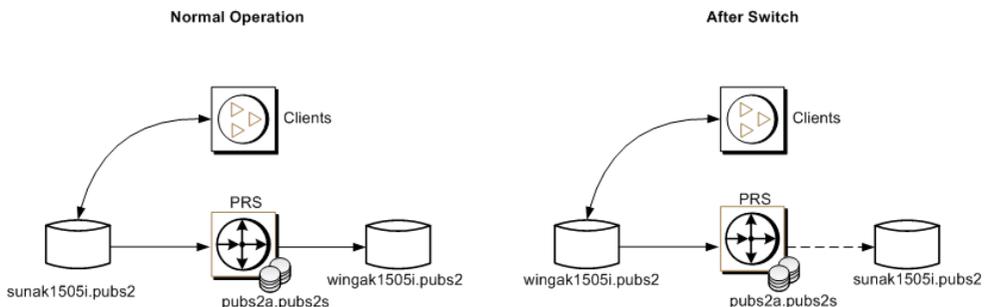
Client applications update the active database; Replication Server maintains the standby database as a copy of the active database. If the active database fails, or if you need to perform maintenance on the active database or on the data server, a switch to the standby database allows client applications to resume work with little interruption. Warm standby applications topics describe how to set up and configure a warm standby application between two Adaptive Server databases—the active database and the standby database.

See the *Replication Server Administration Guide Volume 2 > Manage Warm Standby Applications* for detailed information about warm standby applications, how it works in Replication Server, and other related topics.

Most of the times in Replication Server, databases are defined as "primary" and "replicate." However, in discussing warm standby applications databases are also defined as "active" and "standby."

Warm Standby Application Diagrams

Learn how a warm standby works. The diagrams illustrate the normal operation, and switching of active and standby databases of an example warm standby application.



Creating a Logical Connection

Create a logical connection, which establishes one symbolic name for both the active and standby databases.

In a warm standby application, the active database and the standby database appear in the replication system as a connection from the Replication Server to a single logical database. A logical connection is created to establish one symbolic name for both the active and standby databases. The names do not have to exist in the `interfaces` file nor in the replication system.

1. Log in to the Replication Server:

```
isql -Usa -Psa_pass -SPRS
```

2. Create the logical connection for the warm standby:

```
create logical connection to pubs2a.pubs2s  
go
```

3. Enable SQL statement replication:

```
alter logical connection to pubs2a.pubs2s  
set ws_sqldml_replication to "on"  
go
```

Note: SQL statement replication is only available in Adaptive Server 15.0.3 and later.

4. Exit the `isql` session.

Adding the Active Database to the Replication Server

Add the active database.

1. Add entries for Replication Server and Adaptive Server that host the primary or active database to the corresponding `interfaces` files.

Note: Restart the Adaptive Server or Replication Server if the `interfaces` file has changed.

2. Go to `$SYBASE/REP-15_5/init/rs`.
3. Make a copy of the `setupdb.rs` file and rename it as `myactive.rs`.
4. Edit the `myactive.rs` file.

Table 4. Sample values for the myactive.rs file

Parameter	Description	Value
sybinit.release_directory	Specifies the valid path of the Replication Server software (\$SYBASE).	/opt/sybase
rs.rs_name	Specifies the name of the Replication Server	PRS
rs.rs_rs_sa_user	Specifies the user ID that has “sa” privileges on Replication Server.	sa
rs.rs_rs_sa_pass	Specifies the password of the “sa” user.	sa_pass
rs.rs_ds_name	The name of the data server that hosts the primary database.	sunak1505i
rs.rs_ds_sa_user	Specifies the user ID that has “sa” privileges on data server.	sa
rs.rs_ds_sa_password	Specifies the password of the “sa” user for the data server.	password
rs.rs_db_name	Specifies the name of the primary database	pubs2
rs.rs_needs_repagent	Specifies whether you plan to replicate from specified primary database.	yes
rs.rs_db_maint_user	Specifies the user ID that performs insert , update , delete , and truncate table commands to the replicate database. The user ID must have permissions to perform DML commands at the replicate database. The user is called maintenance user. Note: Do not use a user ID that has been aliased to another user ID.	USE_DEFAULT The default value is <i>database-name_maint</i> .
rs.rs_db_maint_password	Specifies the password for the maintenance user.	<database>_maint_ps

Parameter	Description	Value
rs.rs_ltm_rs_user	The user that the Replication Agent will use to log into the Replication Server. The name must exist. This name typically comes from values that were set up during RS creation time: rs.rs_ltm_rs_user.	PRS_ra
rs.rs_ltm_rs_pass	Specifies the password of the login entered in the rs.rs_ltm_rs_user parameter.	PRS_ra_ps
rs.rs_db_physical_for_logical	Specifies whether this is a warm standby database.	yes
rs.rs_db_active_or_standby	Specifies whether the configuration is for active or standby database.	active
rs.rs_db_logical_ds_name	Specifies the data server portion of the logical connection name.	pubs2a
rs.rs_db_logical_db_name	Specifies the database portion of the logical connection name.	pubs2s

5. Save the file.
6. Go to `$SYBASE/REP-15_5/install`.
7. Create the connection from the active database to the Replication Server by running the resource file:

```
./rs_init -r ../init/rs/myactive.rs
```

If the **rs_init** command fails, correct the issue, then disable the RepAgent as follows:

- a) Log in to the primary Adaptive Server using an “sa” user role and access the primary database
- b) Disable the RepAgent thread in the primary database:

```
isql -Usa -P -Ssunak1505i
use pubs2
go
sp_config_rep_agent pubs2, 'disable'
go
```

Re-run **rs_init** command

8. Validate the primary connection:

```
isql -Usa -Psa_pass -SPRS
```

9. Enter:

```

admin who
go
The output generated from admin who is similar to
Spid      Name      State      Info
-----
36        DIST     Awaiting  Wakeup    102 pubs2a.pubs2s
37        SQT      Awaiting  Wakeup    102:1 DIST pubs2a.pubs2s
31        SQM      Awaiting  Message   102:1 pubs2a.pubs2s
30        SQM      Awaiting  Message   102:0 pubs2a.pubs2s
27        DSI EXEC Awaiting  Command   101(1) sunak1505i.PRS_RSSD
20        DSI      Awaiting  Message   101 sunak1505i.PRS_RSSD
26        SQM      Awaiting  Message   101:0 sunak1505i.PRS_RSSD
49        DSI EXEC Awaiting  Command   103(1) sunak1505i.pubs2
35        DSI      Awaiting  Message   103 sunak1505i.pubs2
38        REP AGENT Awaiting  Command   sunak1505i.pubs2
39        NRM      Awaiting  Message   sunak1505i.pubs2
21        dSUB     Sleeping
15        dCM      Awaiting  Message
18        dAIO     Awaiting  Message
23        dREC     Sleeping  dREC
9         dDELSEG Awaiting  Message
29        USER     Active    sa
14        dALARM   Awaiting  Wakeup
24        dSYSAM   Sleeping
    
```

Note: The RepAgent for the active database must be available.

10. Validate the status of the active database connection:

```

admin logical_status
go
    
```

The output generated from **admin logical_status** is similar to:

Logical Connection Name	Active Connection Name	Active Conn State	Standby Connection Name	Standby Conn State
[102] pubs2a.pubs2s	[103] sunak1505i.pubs2	Active/	None	None

Controller RS	Operation in Progress	State of Operation in Progress	Spid
[16777317] PRS	None	None	

Marking the Active Database for Replication

Use **sp_reptostandby** or **sp_setreptable** to replicate tables in the active database.

The database is enabled for both SQL statement replication (available in Adaptive Server 15.0.3 and later) and non-SQL statement replication.

You can enable replication for tables in the active database in either of these ways:

- Use **sp_reptostandby** to mark the database for replication, enabling replication of data and supported schema changes or,
- Use **sp_setreptable** to mark individual tables for replication of data changes.

In this example, we have used **sp_reptostandby** to mark the database for replication.

1. Log in to the Adaptive Server as the system administrator:

```
% isql -Usa -P -Ssunak1505i
use pubs2
go
```

2. Mark the database tables for replication for both DDL and DML commands and procedures:

```
sp_reptostandby pubs2, 'all'
go
```

```
The replication mode for database 'pubs2' has been
set to 'ALL'.
(return status = 0)
```

3. Mark the database to send SQL statements for **update**, **delete**, **insert**, and **select** into commands, if the SQL statement threshold has been met:

```
sp_setrepdbmode pubs2, 'UDIS', 'on'
go
```

```
The replication mode for database 'pubs2' is 'udis'.
(return status = 0)
```

4. Set the database threshold for SQL statement replication to 10.

```
sp_setrepdbmode pubs2, 'threshold', '10'
go
```

```
The replication threshold for 'pubs2' is '10'.
(return status = 0)
```

Note: You can also set the threshold per table:

```
sp_setrepredefmode t1, 'threshold', '10'
```

5. Exit the **isql** session.

Adding the Standby Database to the Replication Server

Add the standby database. The example procedure for configuring warm standby uses a dump of the active database to load the standby database and starts replication after the setup is complete.

1. Add entries for Replication Server and Adaptive Server that host the replicate database to the corresponding `interfaces` files.

Note: Restart the Adaptive Server and Replication Server if the `interfaces` file has changed.

2. Add the maintenance user login name for the standby connection in the standby data server.
3. Go to `$SYBASE/REP-15_5/init/rs`.
4. Make a copy of the `setupdb.rs` file and rename it as `mystandby.rs`.
5. Edit the `mystandby.rs` file.

Table 5. Sample Values for the `mystandby.rs` File

Parameter	Description	Value
<code>sybinit.release_directory</code>	Specifies the valid path of the Replication Server software (<code>\$SYBASE</code>).	<code>/opt/sybase</code>
<code>rs.rs_name</code>	Specifies the name of the Replication Server	<code>PRS</code>
<code>rs.rs_rs_sa_user</code>	Specifies the user ID that has “sa” privileges on Replication Server.	<code>sa</code>
<code>rs.rs_rs_sa_pass</code>	Specifies the password of the “sa” user.	<code>sa_pass</code>
<code>rs.rs_ds_name</code>	The name of the data server that hosts the standby database.	<code>wingak1505i</code>
<code>rs.rs_ds_sa_user</code>	Specifies the user ID that has “sa” privileges on data server.	<code>sa</code>
<code>rs.rs_ds_sa_password</code>	Specifies the password of the “sa” user for the data server.	<code>password</code>
<code>rs.rs_db_name</code>	Specifies the name of the standby database	<code>pubs2</code>

Parameter	Description	Value
rs.rs_needs_repagent	Specifies whether you plan to replicate from specified standby database.	yes
rs.rs_db_maint_user	Specifies the user ID that performs insert , update , delete , and truncate table commands to the replicate database. The user ID must have permissions to perform DML commands at the replicate database. The user is called maintenance user.	The maintenance user (rs.rs_db_maint_user) that was defined for the active database. Using the same user for the active and standby database simplifies synchronizing the server user IDs after the database dump is loaded to the standby server.
rs.rs_db_maint_password	Specifies the password for the maintenance user.	The password for the user specified rs.rs_db_maint_user
rs.rs_ltm_rs_user	The user that the Replication Agent will use to log into the Replication Server. The name must exist. This name typically comes from values that were set up during RS creation time: rs.rs_ltm_rs_user.	PRS_ra
rs.rs_ltm_rs_pass	Specifies the password of the login entered in the rs.rs_ltm_rs_user parameter.	PRS_ra_ps Warning! When creating the replicate connection, comment out the line for this parameter. Otherwise, the resource file will not work.
rs.rs_db_physical_for_logical	Specifies whether this is a warm standby database.	yes
rs.rs_db_active_or_standby	Specifies whether the configuration is for active or standby database.	standby
rs.rs_db_logical_ds_name	Specifies the data server portion of the logical connection name.	pubs2a
rs.rs_db_logical_db_name	Specifies the database portion of the logical connection name.	pubs2s
rs.rs_db_active_ds_name	Specifies the server that hosts the standby database.	sunak1505i

Parameter	Description	Value
rs.rs_db_active_db_name	Specifies the name of the active database.	pubs2
rs.rs_db_active_sa	Specifies the user ID that has “sa” privileges on the active database.	sa
rs.rs_db_active_sa_pw	Specifies the password of the “sa” user.	password
rs.rs_init_by_dump	Specifies that the standby database is initialized using a dump of the active database.	yes
rs.rs_db_use_dmp_marker	Specifies that “dump marker” option is used to notify replication when to begin forwarding transactions to the standby database.	yes

6. Save the file.
7. Go to `$SYBASE/$SYBASE_REP/install`.
8. Create the connection from the Replication Server to the standby database by running the resource file:

```
./rs_init -r ../init/rs/mystandby.rs
```

If the **rs_init** command fails, correct the issue, then disable the RepAgent as follows:

- a) Log in to the primary Adaptive Server using an “sa” user role and access the primary database.
- b) Disable the RepAgent thread in the primary database:

```
isql -Usa -P -Swingak1505i
use pubs2
go
sp_config_rep_agent pubs2,'disable'
go
```

Re-run **rs_init** command

9. Validate the replicate or standby connection:

```
isql -Usa -Psa_pass -SPRS
```

10. Enter:

```
admin who
go
The output generated from admin who is similar to
Spid    Name      State                Info
-----
36      DIST     Awaiting Wakeup     102 pubs2a.pubs2s
37      SQT      Awaiting Wakeup     102:1 DIST pubs2a.pubs2s
```

Warm Standby Applications

```

31      SQM      Awaiting Message      102:1 pubs2a.pubs2s
30      SQM      Awaiting Message      102:0 pubs2a.pubs2s
27      DSI EXEC Awaiting Command      101 (1) sunak1505i.PRS_RSSD
20      DSI      Awaiting Message      101 sunak1505i.PRS_RSSD
26      SQM      Awaiting Message      101:0 sunak1505i.PRS_RSSD
49      DSI EXEC Awaiting Command      103 (1) sunak1505i.pubs2
35      DSI      Awaiting Message      103 sunak1505i.pubs2
38      REP AGENT Awaiting Command      sunak1505i.pubs2
39      NRM      Awaiting Message      sunask1505i.pubs2
        DSI EXEC Suspended          104 (1) wingak1505i.pubs2
        DSI      Suspended          104 wingak1505i.pubs2
21      dSUB     Sleeping
15      dCM      Awaiting Message
18      dAIO     Awaiting Message
23      dREC     Sleeping dREC
9       dDELSEG  Awaiting Message
29      USER     Awaiting Command      sa
55      USER     Active                  sa
14      dALARM   Awaiting Wakeup
24      dSYSAM   Sleeping

```

Note: The RepAgent for the active database must be available.

11. Validate the status of the active database connection:

```

admin logical_status
go

```

The output generated from **admin logical_status** is similar to:

Logical Connection Name	Active Connection Name	Active Conn State	Standby Connection Name	Standby Conn State
[102] pubs2a.pubs2s	[103] sunak1505i.pubs2	Active/	[104] wingak1505i.pubs2	Suspended / Waiting for Enable Marker

Controller RS	Operation in Progress	State of Operation in Progress	Spid
[16777317] PRS	None	None	

12. Exit the isql session

Initializing the Standby Database

Use Adaptive Server commands and utilities to initialize the standby database.

This example uses the “dump marker” option to initially materialize the standby database. See the *Replication Server Administration Guide Volume 1 > Manage Database Connections* for

information on how to grant permissions to the maintenance user ID for all the tables in the database.

1. Log in to the active data server and dump the active database:

```
dump database pubs2 to '/backup/data/sybase1550/ASE-15_5/pubs2.dmp'
```

Note: Make sure that the Backup Server is running. Dumps and loads are performed through Backup Server.

2. Exit from the active database.
3. Load the standby database with the dump from the active database:

```
load database pubs2 from '/backup/data/sybase1550/ASE-15_5/pubs2.dmp'
```

4. After completing the load operations, bring the standby database online:

```
online database pubs2
go
Started estimating recovery log boundaries for
database 'pubs2'.
Database 'pubs2', checkpoint=(1564, 65),
first=(1564, 65), last=(1565, 17).
Completed estimating recovery log boundaries for
database 'pubs2'.
Started ANALYSIS pass for database 'pubs2'.
Completed ANALYSIS pass for database 'pubs2'.
Recovery of database 'pubs2' will undo incomplete
nested top actions.
Database 'pubs2' is now online
```

5. Check the “suid” for the maintenance user defined for the standby database at the server level:

```
use master
go

select suid,name from syslogins
where name ='pubs2_maint'
go

suid      name
-----
3         pubs2_maint
```

6. Check the “suid” for the maintenance user defined in the standby database:

```
use pubs2
go

select suid,name from sysusers
where name = 'pubs2_maint'
go

suid      name
-----
8         pubs2_maint
```

Note: If the `suid` for the maintenance user does not exist, add it with the `sp_addlogin` command.

- Change the “`suid`” in the `sysusers` table in the standby database to match the “`suid`” in the ASE server that hosts the standby database:

```
sp_configure "allow updates to system tables",1
go
```

Parameter	Default	Memory Used	Config Value	Run Value	Unit	Type
allow updates to system tables	0	0	1	1	switch	dynamic

```
Configuration option changed. ASE need not be rebooted
since the option is dynamic.
Changing the value of 'allow updates to system tables'
does not increase the amount of memory Adaptive Server
uses.
(return status = 0)
```

```
update sysusers set suid = 3 where name = "pubs2_maint"
go
(1 row affected)
```

Note: If there is a difference in the server user IDs (suids) assigned to the users at the active database versus the standby database, modify the `sysusers` table in the newly loaded database to match both the logins.

If the maintenance user does not exist on the database, add the user with the `sp_adduser` command and skip step 7.

- Exit the `isql` session.
- Log in to Replication Server and resume the connection to the standby database:

```
resume connection to wingak1505i.pubs2
go
Connection to 'wingak1505i.pubs2' is resumed
```

Note: Validate that the connection is not suspended or down. If the connection is down, check the Replication Server log for errors and correct the errors, and then resume the connection.

- Check the warm standby status:

```
admin logical_status
go
```

The output generated from `admin logical_status` is similar to:

Logical Connection Name	Active Connection Name	Active Conn State	Standby Connection Name	Standby Conn State
[102] pubs2a.pubs2s	[103] sunak1505i.pubs2	Active/	[104] wingak1505i.pubs2	Active`

Controller RS	Operation in Progress	State of Operation in Progress	Spid
[16777317] PRS	None	None	

11. Release the secondary truncation point of the standby database server:

```
isql -Usa -P -Swingak1505i
use pubs2
go

dbcc settrunc ('ltm','ignore')
go
```

Switching the Active and Standby Databases

Switch from the active to the standby database if the active database will be unavailable for a long time.

In general, do not switch the active and standby databases when the active data server experiences a transient failure from which the Adaptive Server recovers upon restarting with no need for additional recovery steps. You may want to switch if the active database will be unavailable for a long period of time.

1. Ensure that the RepAgent is shut down at the active database. If the RepAgent is still active, issue:

```
isql -Usa -P -Ssunak1505i
use pubs2
go
sp_stop_rep_agent pubs2
go
The Replication Agent thread for database 'pubs2' is
being stopped.
(return status = 0)
```

2. At the Replication Server, enter:

```
isql -Usa -Psa_pass -SPRS
switch active for pubs2a.pubs2s to wingak1505i.pubs2
go
Switch active to wingak1505i.pubs2 for logical
connection to pubs2a.pubs2s is in progress
```

3. To monitor the progress of a switch, use:


```
'pubs2a.pubs2s' received and processed enable
marker.
```

- Resume the standby connection to replicate the data to the standby database, if the new standby database does not need to be resynchronized with the new active database, go to step 7:

```
resume connection to sunak1505i.pubs2
go
Connection to 'sunak1505i.pubs2' is resumed.
```

- To verify whether the warm standby is operational or not, use:

```
admin logical_status
go
```

Logical Connection Name	Active Connection Name	Active Conn State	Standby Connection Name	Standby Conn State
[102] pubs2a.pubs2s	[104] wing-ak1505i.pubs2	Active/	[104] sunak1505i.pubs2	Active/

Controller RS	Operation in Progress	State of Operation in Progress	Spid
[16777317] PRS	None	None	

- If the old active database needs to be resynchronized with the new active database, first drop the standby connection from the Replication Server:

```
drop connection to sunak1505i.pubs2
go
Connection to 'sunak1505i.pubs2' is dropped.
```

- Rebuild the standby side of the warm standby connection using a dump of the new active database and synchronize with a dump marker by following the steps previously described in these topics:
 - Adding the standby database to the Replication Server
 - Initializing the standby database

See also

- Adding the Standby Database to the Replication Server* on page 37
- Initializing the Standby Database* on page 40

Replicating DDL in a Warm Standby Environment

Replicate DDL in a warm standby environment.

1. Log in to the active database using a user ID that exists on both the active and standby data server, and that has permission to create a table.

Note: Do not use the maintenance user ID with the same password that you have defined when the active connection was set up.

2. Create a new table:

```
% isql -Usa -P -Ssunak1505i
use pubs2
go
create table t1 (a integer, b char(10), c datetime)
go
```

3. Create unique indexes for better performance:

```
create unique clustered index t1_idx1 on t1 (a,b)
go
```

4. Log in to the standby database:

```
% isql -Usa -P -Swingak1505i
use pubs2
go
```

5. Verify that the table and index exists in the standby database. If the table and index do not exist, follow the instructions in steps 6 and 7 otherwise, go to step 8.

6. Check the Replication Server log file in: `$SYBASE/REP-15_5/install/PRS.log`. Correct the errors and resume the connection to the standby database:

```
resume connection to rds.rdb
go
```

where:

- `rds` – is the name of the data server that hosts the standby database.
- `rdb` – is the name of the standby database.

Note: If there are errors in the Replication Server log, make sure that the user making the corrections is not the same maintenance user used to set up the active connection.

7. If you want Replication Server to skip any current transaction when trying to resume connection to the standby database, use:

```
resume connection to rds.rdb
skip transaction
go
```

See the *Replication Server Reference Manual* > *Replication Server Commands* > **resume connection** for other available options for the **resume connection** command.

- Grant **insert**, **update**, and **delete** permission for the new table to the replicate database maintenance user at the standby database:

```
grant all on t1 to pubs2_maint
go
```

Replicating DML in a Warm Standby Environment

Replicate DML in a warm standby.

- Log in to the active database using a user ID that exists on both the active and standby data server, and that has permission to create a table.

- In the active database, insert a row to t1:

```
insert into t1 values (1,'first row',getdate())
go
```

- Check whether the row exists in the standby database:

```
select * from t1
go
```

If the row does not exist, follow instructions in steps 4 and 5 otherwise, go to step 6.

- Check the Replication Server log file at: `$SYBASE/REP-15_2/install/PRS.log`. Correct the errors and restart the connection to the replicate database:

```
resume connection to rds.rdb
go
```

- If you want Replication Server to skip any current transaction when trying to resume connection to the standby database, use:

```
resume connection to rds.rdb
skip transaction
go
```

See the *Replication Server Reference Manual* > *Replication Server Commands* > **resume connection** for other available options for the **resume connection** command.

- Log in to the active database and update the row:

```
update t1 set b = 'changed row' where a = 1
go
```

- Log in to the standby database and verify whether the row exists:

```
select * from t1
go
```

- Log in to the active database and enter:

```
truncate table t1
go
```

- Log in to the standby database and enter:

```
select count (*) from t1
go
```

Tracing the Replication Server Transactions to Target Databases

Trace transactions that Replication Server sends to all replicate databases.

Because the trace is not specific to a connection and can get quite large, Sybase recommends that you do not keep the trace function turned on.

1. Use **isql** to log in to Replication Server.
2. Turn the trace on. Output from the trace function is written to the Replication Server log.

```
trace "on", dsi, dsi_buf_dump
go
```

3. Use **isql** to insert one row in the active database:

```
insert into t1 values (1, 'first row', getdate())
go
(1 row affected)
```

4. Update the row in the active database:

```
update t1 set c = getdate()
go
(1 row affected)
```

5. View the trace output in the Replication Server log file, `$SYBASE/REP-15_5/install/repservername.log`:

```
T. 2009/10/28 22:09:08. (138): Command(s) to
'wingak1505i.pubs2':
T. 2009/10/28 22:09:08.
(138):'begin transaction
[0a] update dbo.t1 set
c='20091028 22:09:07:703'
where a=1 and b='first row'
and c='20091028
22:05:53:843' '
```

Note: To turn off the tracing function in Replication Server, use:

```
trace "off", dsi, dsi_buf_dump
go
```

Replication Definitions for Improving Performance

A replication definition describes the source table to Replication Server, specifying the columns you want to copy. It may also describe attributes of the destination table. Destination tables that match the specified characteristics can subscribe to the replication definition.

Replication Server does not require replication definitions to maintain a standby database, although using replication definitions can improve performance when you are replicating into a standby database. In addition, **create replication definitions** on tables that contain approximate numeric datatypes so that the where clause used to construct the SQL statement applied to the standby database does not include these columns. The range and storage precision of approximate numeric datatypes (real, float) is machine-dependent, and may result in the wrong rows being modified, or the correct row not being found.

When you specify that you want to use a replication definition for replicating into a standby database:

- Replication Server optimizes updates and deletes by using the primary key defined in the replication definition to generate the **where** clause.
- You can specify whether Replication Server uses the replication definition's replicate minimal columns setting for replicating into the standby database. This setting indicates whether updates replace the values for all columns or only the columns with changed values.

See the *Replication Server Administration Guide Volume 1 > Manage Replicated Tables* for detailed information on replication definitions.

Creating a Replication Definition

Create a replication definition to describe a replicated object. You can create replication definitions for databases, functions, or tables.

1. Use **isql** to log in to Replication Server.
2. Create a replication definition for the warm standby for table "t1":

```
create replication definition t1_ws_repdef
with primary at pubs2a.pubs2s
with all tables named t1
(a integer, b char(10), c datetime)
primary key (a)
send standby replication definition columns
replicate SQLDML
go
Replication definition 't1_ws_repdef' is created.
```

Note: To replicate using SQL statement replication, which is available only in Adaptive Server version 15.0.3 and later, use the replicate SQLDML clause.

3. Update the row in table “t1”:

```
update t1 set c = getdate()
go
```

4. Compare the trace output with the output that was generated before the replication definition was created:

```
T. 2009/10/28 22:10:43. (138): Command(s) to
'wingak1505i.pubs2':
T. 2009/10/28 22:10:43. (138): 'begin transaction [0a]
update dbo.t1 set a=1, b='first row', c='20091028
22:10:42:383' where a=1 '
```

The **where** clause in the update statement, now contains only column “a” because the replication definition specified that column “a”, uniquely identifies the row.

See also

- *Tracing the Replication Server Transactions to Target Databases* on page 48

Using SQL Statement Replication for Warm Standby

SQL statement replication complements log-based replication and addresses performance degradation caused by batch jobs.

Prerequisites

You can perform SQL statement replication only with Adaptive Server 15.0.3 and later.

Task

1. In step 4 of the task, “Marking the active database for replication,” the threshold for the database is set to 10. Therefore, SQL statement replication is used only when there are more than 10 rows. Insert 10 rows in to the table “t1”:

```
insert into t1 values (2,'first row',getdate())
insert into t1 values (3,'first row',getdate())
insert into t1 values (4,'first row',getdate())
insert into t1 values (5,'first row',getdate())
insert into t1 values (6,'first row',getdate())
insert into t1 values (7,'first row',getdate())
insert into t1 values (8,'first row',getdate())
insert into t1 values (9,'first row',getdate())
insert into t1 values (10,'first row',getdate())
insert into t1 values (11,'first row',getdate())
```

2. Update a number of rows less than or equal to the threshold value.

```
update t1 set b = 'no SQL' where a < 3
go
(2 rows affected)
T. 2009/10/28 22:18:55. (138): Command(s) to
'wingak1505i.pubs2':
T. 2009/10/28 22:18:55. (138):
```

```
'begin transaction [0a]
update dbo.t1 set a=1, b='no SQL',
c='20091028 22:10:42:383'
where a=1 [0a] update dbo.t1
set a=2, b='no SQL', c='20091028
22:12:24:093' where a=2 '
```

The trace statements show the individual SQL updates to each row.

3. Update all the rows:

```
update t1 set b = 'yes SQL'
go
(11rows affected)
```

The trace output shows the SQL statement, not the individual statements for each row.

```
T. 2009/10/28 22:23:35. (138): Command(s) to
'wingak1505i.pubs2':
T. 2009/10/28 22:23:35. (138): 'begin transaction
[0a] update dbo.t1 set b = 'yes SQL' '
```

Note: To turn off the tracing function in Replication Server, use:

```
trace "off",dsi,dsi_buf_dump
go
```

See also

- *Marking the Active Database for Replication* on page 36

Materialization and Resynchronization

Materialization is copying data specified by a subscription from a primary or source database or table, to a replicate or target database or table. Resynchronization makes all of the data in the primary database or table and the replicate database or table identical. You can resynchronize a database, a table, or individual rows. The materialization process causes resynchronization.

The materialization method depends upon the amount of data to be transmitted, the portion of the data that resides at the replicate site, the time available for the process, and whether the primary database or table can have any activity against it.

The provided scenarios make several assumptions and considerations:

- Marking tables with text or image columns can take a long time. In Adaptive Server 15.0 ESD #1 and later, the **use_index** option can significantly speed up marking these columns. When text in a table is marked for replication using one method, those text pages are not remarked if a different method is added. For example, if a database is marked for replication, marking a table containing text columns in that database does not require all the text pages to be re-marked for replication.
- The database materialization and synchronization options discussed in the examples use the Adaptive Server database dump and load method. See the Adaptive Server manuals for any requirements for using these routines.
- When you dump a database, the secondary truncation point and the RepAgent for the database are included in the dump. Users and permissions are copied; but logins and roles are not.
- Adding and dropping a subscription requires the Data Server Interface (DSI) to be active or awaiting command to the replicate database.
- These scenarios assume that the connections from the Replication Server to the database exists and that replication definitions are already defined.

See the *Replication Server Administration Guide Volume 1 > Manage Subscriptions* for complete descriptions of various materializations methods.

Scenario 1

The primary and the replicate database can be refreshed from a different database source while existing replication definitions and subscriptions continue to be used.

If the source database has never participated in replication, temporarily add the database to a Replication Server so that it has all the tables and stored procedures needed for replication, before making a copy of it.

Materialization and Resynchronization

This scenario uses a third database (for example, a production database) to populate the source and target database environment (for example, a test database). You would use this scenario when you want to refresh a test system from a copy of a production database.

Before you begin

In the current primary, validate that the objects that have been marked for replication. If the database has `text` and `image` columns and these columns will be replicated, decide whether to mark the database for replication prior to the copy or after the primary database has been established.

1. Check whether the database has been marked for replication:

```
use pri
go
sp_reptostandby pri
go
The replication status for database 'pri' is 'ALL'.
The replication mode for database 'pri' is 'off'.
(return status = 0)
```

Note: In this scenario, the database has been marked for database replication and has not been marked for SQL statement replication.

2. If the database has not been marked for replication, check to see whether the tables are marked for replication:

```
use pri
go
sp_setreplicate
go
Name          Type
----          -
t1             user table
t2             user table
rs_marker     stored procedure
(3 rows affected)
(return status = 0)
```

Note: Any table or stored procedure beginning with "rs_" is created by Replication Server. Therefore, in this scenario, t1 and t2 are application tables, which were marked for replication and **rs_marker** is a stored procedure created by adding the database to the replication system.

3. Get the current generation number of the primary database.

```
use pri
go
dbcc gettrunc
go
secondary trunc page secondary trunc state dbrepstat
-----
2669                1                173
generation id database id database name ltl version
```

```
-----
0              7              pri              720
-----
```

4. At the RSSD, obtain the maintenance users for the connections:

```
use PRS2_RSSD
go
rs_helpuser
go
```

```
User Name          Permission(s) Name
-----
```

```
PRS2_id_user       no grants
sa                 sa
PRS2_ra            connect source
PRS2_rsi           connect source
PRS2_RSSD_prim    source, primary subscr
```

```
Maintenance Users
```

```
User name          Destination DS.DB
-----
```

```
PRS2_RSSD_maint   sunak1505x.PRS2_RSSD
pri_maint         sunak1505x.pri
pri_maint         sunak1505x.rep
(return status = 0)
```

In this example, “pri_maint” is the maintenance user for both the connection to the primary database, sunak1505x.pri. and replicate database, sunak1505x.rep.

Begin the resynchronization

You can stop all user activity to the source database; however, it is not required.

When a new database is brought online, Adaptive Server creates a database that is transactionally consistent at the time of the dump.

1. Stop all user activity of the primary database including the existing RepAgent:

```
sp_stop_rep_agent pri
go
The Replication Agent thread for database 'pri' is
being stopped.
(return status = 0)
```

2. Suspend the connection to the primary and replicate database:

```
isql -Usa -Psa_pass -SPRS2
suspend connection to sunak1505x.pri
go
Connection to 'sunak1505x.pri' is suspended
suspend connection to sunak1505x.rep
go
Connection to 'sunak1505x.rep' is suspended.
```

3. Dump the database of the source database:

```
dump database diffprim to
'/c11014900/sybase1520x/diffprim.dmp'
go
```

Materialization and Resynchronization

4. Load the source database dump to the primary database:

```
load database pri from
'/c11014900/sybase1520x/diffprim.dmp'
go
```

5. Bring the database online:

```
online database pri
go
```

6. Add the maintenance user of the primary connection to the primary database and grant the appropriate privileges. If the maintenance user already exists on this server, synchronize the “suid” of the maintenance user and any other user that will be logging in to the primary database.

```
use pri
go
sp_adduser pri_maint
go
New user added.
(return status = 0)
grant execute on rs_get_lastcommit to pri_maint
go
grant all on rs_lastcommit to pri_maint
go
```

Note: If the maintenance user is new to this server, add the maintenance user login to this server, and grant the replication role:

```
sp_addlogin <maintenance user>, <maintenance userpassword>,
<database>
go
sp_role "grant", replication_role, <maintenance
user>
go
```

If the password to the maintenance user is unknown, set a new password in Adaptive Server. In the Replication Server, change the password of the maintenance user to match with what was assigned in the Adaptive Server:

```
alter connection to <dataserver>.<database>
set password to <new password>
go
```

7. If the source database dump came from a primary database that had a Replication Agent, remove the secondary truncation point and the existing Replication Agent from the new primary database:

```
use pri
go
dbcc settrunc (ltm,ignore)
go
secondary trunc page secondary trunc state dbrepstat
-----
2667                0                166
generation id database id database name ltl version
```

```
-----
0                7                pri                720
-----
```

Note: A secondary truncation state of 0 indicates that the secondary truncation point is inactive.

```
sp_config_rep_agent pri,'disable'
go
Replication Agent disabled for database 'pri'. The
secondary truncation point
in the database is no longer active.
(return status = 0)
```

8. Enable the Replication Agent at the primary database:

```
sp_config_rep_agent pri,
'enable','PRS2','PRS2_ra','PRS2_ra_ps'
go
Replication Agent enabled for database 'pri'. The
Replication Agent thread needs
to be started using sp_start_rep_agent.
(return status = 0)
```

Note: Any Replication Server user with “connect source” privilege can be used. If the password of the Replication Server is unknown, it can be reset using the alter user Replication Server command.

9. Modify the settings in the RSSD and the primary database to have the Replication Agent start at the end of the transaction log:

```
use PRS2_RSSD
go
rs_zeroltm sunak1505x,pri
go
Locator has been reset to zero.
(return status = 0)
isql -Usa -P -Ssunak1505x
use pri
go
dbcc settrunc (ltm,valid)
go
secondary trunc page secondary trunc state dbrepstat
-----
2670                1                167
generation id database id database name ltl version
-----
0                7                pri                720
```

10. Increase the generation number of the new primary database by 1 in case the log pages of the new database are numerically less than the log pages of the previous copy of the database:

```
dbcc settrunc (ltm,gen_id,1)
go
secondary trunc page secondary trunc state dbrepstat
-----
2670                1                167
generation id database id database name ltl version
```

```
-----  
0                7                pri                720
```

Note: The previous value of the generation id was obtained in step 3 of “Before you begin” section. If text or image columns need to be marked for replication, mark the tables and columns for replication here.

11. Load the replicate database with the source database dump:

```
load database rep from  
'/c11014900/sybase1520x/diffprim.dmp'  
go
```

12. Bring the replicate database online:

```
online database rep  
go
```

13. Add the maintenance user of the replicate connection, to the replicate database and grant all appropriate privileges using the information gathered prior to the resynch.

```
use rep  
go  
sp_adduser pri_maint  
go  
New user added.  
(return status = 0)  
grant execute on rs_get_lastcommit to pri_maint  
go  
grant all on rs_lastcommit to pri_maint  
go
```

Note: If the maintenance user is new to this server, add the maintenance user login to this server, and grant the replication role:

```
sp_addlogin <maintenance user>, <maintenance userpassword>,  
<database>  
go  
sp_role "grant", replication_role, <maintenance  
user>  
go
```

If the password to the maintenance user is unknown, set a new password in Adaptive Server. In the Replication Server, change the password of the maintenance user to match with what was assigned in the Adaptive Server:

```
alter connection to <dataserver>.<database>  
set password to <new password>  
go
```

14. If the source database had a RepAgent, release the secondary truncation point and remove the RepAgent:

```
use rep  
go  
dbcc settrunc (ltm,ignore)  
go  
secondary trunc page secondary trunc state dbrepstat  
-----
```

generation id	database id	database name	l1l version
2667	0		166
0	7	pri	720

Note: A secondary truncation state of 0 indicates that the secondary truncation point is inactive.

```
sp_config_rep_agent rep,'disable'
go
Replication Agent disabled for database 'rep'. The
secondary truncation point
in the database is no longer active.
(return status = 0)
```

- 15.** Truncate the rs_lastcommit table at the replicate database. Enter:

```
truncate table rs_lastcommit
go
```

- 16.** Grant appropriate permissions to the replicate tables so that the maintenance user can apply commands to the replicate database:

```
grant all on t1 to pri_maint
go
grant all on t2 to pri_maint
go
```

- 17.** Purge the inbound queue of the primary connection and the outbound queue of the replicate connection on the Replication Server to eliminate any in-process work from the old replication setup:

```
isql -Usa -Psa_pass -SPRS2
sysadmin hibernate_on
go
The Replication Server has now entered hibernation
mode.
sysadmin sqm_purge_queue, 104,1
go
sysadmin sqm_purge_queue, 106,0
go
sysadmin hibernate_off
go
The Replication Server has now finished hibernation
mode.
```

- 18.** Resume connection to the primary and the replicate database:

```
resume connection to sunak1505x.pri
go
Connection to 'sunak1505x.pri' is resumed.
resume connection to sunak1505x.rep
go
Connection to 'sunak1505x.rep' is resumed.
```

- 19.** Start the Replication Agent of the primary database:

```
sp_start_rep_agent pri
go
Replication Agent thread is started for database
```

Materialization and Resynchronization

```
'pri'.  
(return status = 0)
```

20. On the Replication Server, ensure that both the RepAgent and the DSI threads were successfully started

```
admin who  
go  
Spid      Name          State          Info  
-----  
139      DSI EXEC      Awaiting Command 104 (1) sunak1505x.pri  
135      DSI           Awaiting Message 104 sunak1505x.pri  
129      DIST         Awaiting Wakeup 104 sunak1505x.pri  
130      SQT          Awaiting Wakeup 104:1 DIST sunak1505x.pri  
123      SQM          Awaiting Message 104:1 sunak1505x.pri  
56       SQM          Awaiting Message 104:0 sunak1505x.pri  
142      REP AGENT    Awaiting Command sunak1505x.pri  
143      NRM          Awaiting Command sunak1505x.pri  
145      DSI EXEC     Awaiting Command 106 (1) sunak1505x.rep  
144      DSI          Awaiting Message 106 sunak1505x.rep  
124      SQM          Awaiting Message 106:0 sunak1505x.rep
```

21. Validate that replication is available and working.
22. Allow users on the primary database.

Scenario 2

You can materialize a replicate database from a primary database where all user activity has stopped at the primary database.

Use this materialization method when replication is broken between the primary database and the replicate database for a significant period of time and the queues are filling up, which needs to be purged. Refresh the replicate database with a copy from the current primary database. This can be used with either table replication or database replication. This scenario assumes that user activity is stopped at the primary while the database dump is taken.

Before you begin

- At the RSSD, obtain the maintenance users for the connections:

```
use PRS2_RSSD  
go  
rs_helpuser  
go  
  
User Name      Permission(s) Name  
-----  
PRS2_id_user   no grants  
sa             sa  
PRS2_ra        connect source  
PRS2_rsi       connect source  
PRS2_RSSD_prim source, primary subscr  
  
Maintenance Users
```

```

User name          Destination DS.DB
-----
PRS2_RSSD_maint   sunak1505x.PRS2_RSSD
pri_maint         sunak1505x.pri
pri_maint         sunak1505x.rep
(return status = 0)

```

In this example, “pri_maint” is the maintenance user for both the connection to the primary database, sunak1505x.pri. and replicate database, sunak1505x.rep.

Begin the resynchronization

1. If it is not already suspended, suspend the connection to the replicate database:

```

isql -Usa -Psa_pass -SPRS2
suspend connection to sunak1505x.rep
go
Connection to 'sunak1505x.rep' is suspended.

```

2. In the primary database, release the secondary truncation point—any log records not previously replicated are already reflected in the data.

```

isql -Usa -P -SSunak1505x
use pri
go
sp_stop_rep_agent pri
go
The Replication Agent thread for database 'pri' is
being stopped.
(return status = 0)
dbcc settrunc (ltm,ignore)
go
secondary trunc page secondary trunc state dbrepstat
-----
2669                0                172
generation id database id database name ltl version
-----
0                    7                pri        720

```

Note: When secondary truncation state is 0, the secondary truncation point is inactive on the primary database.

3. Dump the primary database:

```

dump database pri to '/c11014900/sybase1520x/ASE-
15_0/bin/pri.dmp'
go

```

4. Load the replicate database:

```

load database rep from '/c11014900/sybase1520x/ASE-
15_0/bin/pri.dmp'
go

```

5. Purge queues of any existing work from the primary database (inbound queue) to the replicate database (outbound queue):

```

isql -Usa -Psa_pass -SPRS2
sysadmin hibernate_on
go

```

Materialization and Resynchronization

```
The Replication Server has now entered hibernation
mode.
sysadmin sqm_purge_queue, 104,1
go
sysadmin sqm_purge_queue, 106,0
go
sysadmin hibernate_off
go
The Replication Server has now finished hibernation
mode.
```

6. Bring the replicate database online:

```
online database rep
go
```

7. To the replicate database, add the maintenance user of the replicate connection, and grant the appropriate privileges.

```
use rep
go
sp_adduser pri_maint
go
New user added.
(return status = 0)
grant execute on rs_get_lastcommit to pri_maint
go
grant all on rs_lastcommit to pri_maint
go
```

Note: If the maintenance user is new to this server, add the maintenance user login to this server, and grant the replication role:

```
sp_addlogin <maintenance user>, <maintenance userpassword>,
<database>
go
sp_role "grant", replication_role, <maintenance
user>
go
```

If the password to the maintenance user is unknown, set a new password in Adaptive Server. In the Replication Server, change the password of the maintenance user to match with what was assigned in the Adaptive Server:

```
alter connection to <dataserver>.<database>
set password to <new password>
go
```

8. Resume the connection to the replicate database:

```
resume connection to sunak1505x.rep
go
Connection to 'sunak1505x.rep' is resumed.
```

9. On the Replication Server, confirm that the connection to the replicate database has successfully resumed:

```
admin who
go
Spid      Name                State                Info
```

```

-----
103   DSI EXEC   Awaiting Command   104(1) sunak1505x.pri
100   DSI       Awaiting Message   104 sunak1505x.pri
95    DIST      Awaiting Wakeup    104 sunak1505x.pri
96    SQT       Awaiting Wakeup    104:1 DISTsunak1505x.pri
92    SQM       Awaiting Message   104:1 sunak1505x.pri
56    SQM       Awaiting Message   104:0 sunak1505x.pri
      REP AGENT Down              sunak1505x.pri
      NRM       Down              sunak1505x.pri
107   DSI EXEC   Awaiting Command   106(1) sunak1505x.rep
106   DSI       Awaiting Message   106 sunak1505x.rep
91    SQM       Awaiting Message   106:0 sunak1505x.rep

```

Note: If the connection does not successfully resume, look at the Replication Server log to see what errors occurred, correct the errors, and resume the connection.

- 10.** Add the replication definition and subscription to the Replication Server, if needed. In this example, we are using a database replication definition and a database subscription. Enter:

```

create database replication definition pri_db_repdef
with primary at sunak1505x.pri
replicate DDL
go
Database replication definition 'pri_db_repdef'
for sunak1505x.pri is created.
create subscription rep_db_repdef
for database replication definition pri_db_repdef
with primary at sunak1505x.pri
with replicate at sunak1505x.rep
without materialization
subscribe to truncate table
go

Subscription 'rep_db_repdef' is in the process of
being created

```

- 11.** Reset the start point on the primary database to the end of the primary database log.
- a.** Zero the `rs_locator` entry for this connection in the RSSD:

```

use PRS2_RSSD
go
rs_zeroltm sunak1505x,pri
go
Locator has been reset to zero.
(return status = 0)

```

- b.** Enable the secondary truncation point in the primary database:

```

isql -Usa -P -SSunak1505x
use pri
go
dbcc settrunc (ltm,valid)
go
secondary trunc page secondary trunc state dbrepstat
-----
2669                               1                173
generation id database id database name ltl version

```

```
-----
0              7              pri      720
```

12. Start the Replication Agent on the primary database:

```
use pri
go
sp_start_rep_agent pri
go
Replication Agent thread is started for database
'pri'.
(return status = 0)
```

13. Validate that the Replication Server connections are ready to replicate:

```
admin who
go
Spid      Name      State      Info
-----
103      DSI EXEC  Awaiting  Command  104(1) sunak1505x.pri
100      DSI      Awaiting  Message  104 sunak1505x.pri
95       DIST     Awaiting  Wakeup   104 sunak1505x.pri
96       SQT      Awaiting  Wakeup   104:1 DIST sunak1505x.pri
92       SQM      Awaiting  Message  104:1 sunak1505x.pri
56       SQM      Awaiting  Message  104:0 sunak1505x.pri
108      REP AGENT Awaiting  Command  sunak1505x.pri
109      NRM      Awaiting  Command  sunak1505x.pri
107      DSI EXEC  Awaiting  Command  106(1) sunak1505x.rep
106      DSI      Awaiting  Message  106 sunak1505x.rep
91       SQM      Awaiting  Message  106:0 sunak1505x.rep
```

14. Validate that data can be replicated from the primary to the replicate.:

Note: If Replication Agent goes down, look in the ASE error log for messages. If the DSI is suspended, look in the Replication Server error log or the ASE error log for messages. To start the Replication Agent on the primary database, use **sp_start_rep_agent <dbname>** command. To resume the DSI at the Replication Server, use the resume connection to **<dataserver>.<database>** command. To move past and not apply transactions in the outbound queue, use the skip transaction clause of the resume connection command.

15. Allow users on the primary database.

Scenario 3

You can materialize a replicate database from a primary database where user activity cannot be stopped and using MSA replication.

It is assumed that activity cannot be stopped at the primary database while the database dump is in progress. This scenario uses the primary database to populate the replicate database and uses database replication definition and subscription.

If the primary database is replicating to multiple replicate databases, the complete process including defining the subscription, dumping the primary database, and loading the replicate database must be completed for each replicate database, defining the subscription for the next replicate database.

Before you begin

- At the RSSD, obtain the maintenance users for the connections:

```

use PRS2_RSSD
go
rs_helpuser
go
User Name          Permission(s) Name
-----
PRS2_id_user       no grants
sa                 sa
PRS2_ra            connect source
PRS2_rsi           connect source
PRS2_RSSD_prim    source, primary subscr

User name          Destination DS.DB
-----
PRS2_RSSD_maint   sunak1505x.PRS2_RSSD
pri_maint         sunak1505x.pri
pri_maint         sunak1505x.rep
(return status = 0)

```

In this example, “pri_maint” is the maintenance user for both the connection to the primary database, sunak1505x.pri. and replicate database, sunak1505x.rep.

Begin the resynchronization

- If the database subscription exists, drop the subscription:

```

isql -Usa -P -Ssunak1505x
use PRS2_RSSD
go
rs_helpdbsub
go
DBSub.Name ReplicatedDS.DB ReplicateRS Status at RRS DBRep.Def.Name
-----
rep_db_sub sunak1505x          PRS2          Valid          pri_db_rep

PrimaryDS.DB PrimaryRS Status at PRS Method Trunc.Table Creation
Date
-----
sunak1505x          PRS2          Valid Use Dump Marker Yes Dec 9 2009
3:38PM

```

Note: The connection to the replicate database must be “Awaiting Command” prior to dropping the subscription.

```

isql -Usa -Psa_pass -SPRS2
drop subscription rep_db_sub
for database replication definition pri_db_rep
with primary at sunak1505x.pri
with replicate at sunak1505x.rep
without purge
go

```

Materialization and Resynchronization

```
Subscription 'rep_db_sub' is in the process of being
dropped.
```

2. Validate that the primary database is marked, enter:

```
isql -Usa -P -Ssunak1505x
use pri
go
sp_reptostandby pri
go
The replication status for database 'pri' is 'ALL'.
The replication mode for database 'pri' is 'off'.
(return status = 0)
```

Note: In this scenario, the database “pri” is marked to replicate both DML and DDL and is not marked for SQL statement replication.

3. Validate that the RepAgent on the primary database is running and the connection to the replicate database exists and is not suspended:

```
isql -Usa -Psa_pass -SPRS2
admin who
go
Spid Name          State              Info
-----
62  DSI EXEC         Awaiting Command  104(1) sunak1505x.pri
57  DSI              Awaiting Message  104 sunak1505x.pri
59  DIST            Awaiting Wakeup   104 sunak1505x.pri
60  SQT             Awaiting Wakeup   104:1 DIST sunak1505x.pri
58  SQM             Awaiting Message  104:1 sunak1505x.pri
56  SQM             Awaiting Message  104:0 sunak1505x.pri
61  REP AGENT       Awaiting Command   sunak1505x.pri
63  NRM             Awaiting Command   sunak1505x.pri
68  DSI EXEC        Awaiting Command  105(1) sunak1505x.rep
67  DSI             Awaiting Message  105 sunak1505x.rep
66  SQM             Awaiting Message  105:0 sunak1505x.rep
```

4. Define the subscription with the option that we are going to use a dump to synchronize the data:

```
define subscription rep_db_sub
for database replication definition pri_db_rep
with primary at sunak1505x.pri
with replicate at sunak1505x.rep
subscribe to truncate table
use dump marker
go
Subscription 'rep_db_sub' is in the process of being
defined.
```

The connection to the replicate database is still not suspended.

```
admin who
go
Spid Name          State              Info
-----
68  DSI EXEC        Awaiting Command  105(1) sunak1505x.rep
67  DSI             Awaiting Message  105 sunak1505x.rep
66  SQM             Awaiting Message  105:0 sunak1505x.rep
```

The above **admin who** command displays only the connections that are affected and not the complete list of connections.

5. Dump the primary database:

```
dump database pri to 'pri.dmp'
go
```

The connection to the replicate database is now suspended.

```
admin who
go
Spid Name          State          Info
-----
68 DSI EXEC Suspended      105(1) sunak1505x.rep
67 DSI          Suspended      105 sunak1505x.rep
66 SQM          Awaiting Message 105:0 sunak1505x.rep
```

6. Load the replicate database:

```
load database rep from '/c11014900/sybase1520x/ASE-
15_0/bin/pri.dmp'
go
```

7. Bring the replicate database online:

```
online database rep
go
```

8. Add the maintenance user of the replicate connection to the replicate database and grant the appropriate privileges before the resynchronization.

```
use rep
go
sp_adduser pri_maint
go
New user added.
(return status = 0)
grant execute on rs_get_lastcommit to pri_maint
go
grant all on rs_lastcommit to pri_maint
go
```

Note: If the maintenance user is new to this server, add the maintenance user login to this server, and grant the replication role:

```
sp_addlogin <maintenance user>, <maintenance userpassword>,
<database>
go
sp_role "grant", replication_role, <maintenance
user>
go
```

If the password to the maintenance user is unknown, set a new password in Adaptive Server. In the Replication Server, change the password of the maintenance user to match with what was assigned in the Adaptive Server:

```
alter connection to <dataserver>.<database>
set password to <new password>
go
```

Materialization and Resynchronization

9. Truncate the `rs_lastcommit` table at the replicate database:

```
use rep
go
truncate table rs_lastcommit
go
```

10. Release the secondary truncation point from the replicate database:

```
dbcc settrunc (ltm,ignore)
go
secondary trunc page secondary trunc state dbrepstat
-----
2668                0                172
generation id database id database name ltl version
-----
0                    8                rep        720
```

11. Resume connection to the replicate database. If the DSI stays suspended, look at the Replication Server logs for any errors encountered while resuming the DSI:

```
resume connection to sunak1505x.rep
go
Connection to 'sunak1505x.rep' is resumed
```

Scenario 4

You can materialize a primary database from a replicate database where user activity can be stopped and the replicate database is a duplicate of the primary database.

Before you begin

Use this materialization method if a problem occurs to your primary database and the replicate database, which is the duplicate of your primary database, is old data. In the current primary database, validate the objects that have been marked for replication. If the database has text and image columns and these columns will be replicated, decide whether to mark the database for replication prior to the copy or after replication from the primary database has been established.

1. Check whether the database has been marked for replication:

```
use pri
go
sp_reptostandby pri
go
The replication status for database 'pri' is 'ALL'.
The replication mode for database 'pri' is 'off'.
(return status = 0)
```

Note: In this scenario, the database has been marked for database replication and has not been marked for SQL statement replication.

2. If the database has not been marked for replication, check to see whether the tables are marked for replication:

```
use pri
go
```

```

sp_setreplicate
go
Name          Type
-----
t1            user table
t2            user table
rs_marker     stored procedure
(3 rows affected)
(return status = 0)

```

Note: Any table or stored procedure beginning with "rs_" is created by Replication Server. Therefore, in this scenario, t1 and t2 are application tables, which were marked for replication and **rs_marker** is a stored procedure created by adding the database to the replication system.

3. Get the current generation number of the primary database.

```

use pri
go
dbcc gettrunc
go
secondary trunc page secondary trunc state dbrepstat
-----
2669                                1                173
generation id database id database name ltl version
-----
0                                7                pri                720

```

4. At the RSSD, obtain the maintenance users for the connections:

```

use PRS2_RSSD
go
rs_helpuser
go

```

User Name	Permission(s)	Name
PRS2_id_user	no grants	
sa	sa	
PRS2_ra	connect source	
PRS2_rsi	connect source	
PRS2_RSSD_prim	source, primary	subscr

```

Maintenance Users
User name          Destination DS.DB
-----
PRS2_RSSD_maint   sunak1505x.PRS2_RSSD
pri_maint         sunak1505x.pri
pri_maint         sunak1505x.rep
(return status = 0)

```

In this example, "pri_maint" is the maintenance user for both the connection to the primary database, sunak1505x.pri. and replicate database, sunak1505x.rep.

Begin the resynchronization

You need not stop user activity at the source database; because when the new database goes online, Adaptive Server creates a database that is transactionally consistent at the time of the dump.

1. Stop all user activity of the replicate database including the existing DSI connection:

```
isql -Usa -Psa_pass -SPRS2
suspend connection to sunak1505.rep
go
Connection to 'sunak1505x.rep' is suspended.
```

2. Stop all user activity of the primary database including the existing Replication Agent and DSI:

```
sp_stop_rep_agent pri
go
The Replication Agent thread for database 'pri' is
being stopped.
(return status = 0)
isql -Usa -Psa_pass -SPRS2
suspend connection to sunak1505x.pri
go
Connection to 'sunak1505x.pri' is suspended.
```

3. Validate that the Replication Server is not actively using either the primary or the replicate database connection:

```
admin who
go
Spid      Name      State                Info
-----
          DSI EXEC   Suspended            104(1) sunak1505x.pri
          DSI      Suspended            104 sunak1505x.pri
129      DIST      Awaiting Wakeup     104 sunak1505x.pri
130      SQT       Awaiting Wakeup     104:1 DIST sunak1505x.pri
123      SQM       Awaiting Message    104:1 sunak1505x.pri
56       SQM       Awaiting Message    104:0 sunak1505x.pri
          REP AGENT Down                 sunak1505x.pri
          NRM      Down                 sunak1505x.pri
          DSI EXEC   Suspended            106(1) sunak1505x.rep
          DSI      Suspended            106 sunak1505x.rep
124      SQM       Awaiting Message    106:0 sunak1505x.rep
```

4. Dump the replicate database:

```
dump database diffprim to
'/c11014900/sybase1520x/rep.dmp'
go
```

5. Load the replicate database dump to the primary database:

```
load database pri from
'/c11014900/sybase1520x/rep.dmp'
go
```

6. Bring the primary database online:

```
online database pri
go
```

7. Add the maintenance user of the primary connection to the primary database and grant the appropriate privileges. If the maintenance user already exists on this server, synchronize the “suid” of the maintenance user and any other user that will be logging in to the primary database.

```
use pri
go
sp_adduser pri_maint
go
New user added.
(return status = 0)
grant execute on rs_get_lastcommit to pri_maint
go
grant all on rs_lastcommit to pri_maint
go
```

Note: If the maintenance user is new to this server, add the maintenance user login to this server, and grant the replication role:

```
sp_addlogin <maintenance user>, <maintenance userpassword>,
<database>
go
sp_role "grant", replication_role, <maintenance
user>
go
```

If the password to the maintenance user is unknown, set a new password in Adaptive Server. In the Replication Server, change the password of the maintenance user to match with what was assigned in the Adaptive Server:

```
alter connection to <dataserver>.<database>
set password to <new password>
go
```

8. If the replicate database dump had a Replication Agent, remove the secondary truncation point and the existing RepAgent from the new primary database:

```
use pri
go
dbcc settrunc (ltn,ignore)
go
secondary trunc page secondary trunc state dbrepstat
-----
2667                0                166
generation id database id database name ltl version
-----
0                    7                pri        720
```

Note: A secondary truncation state of 0 indicates that the secondary truncation point is inactive.

```
sp_config_rep_agent pri,'disable'
go
Replication Agent disabled for database 'pri'. The
```

Materialization and Resynchronization

```
secondary truncation point
in the database is no longer active.
(return status = 0)
```

9. Enable the RepAgent at the primary database:

```
sp_config_rep_agent pri,
'enable','PRS2','PRS2_ra','PRS2_ra_ps'
go
Replication Agent enabled for database 'pri'. The
Replication Agent thread
needs to be started using sp_start_rep_agent.
(return status = 0)
```

Note: Any Replication Server user with **connect source** privilege can be used. If the password of the Replication Server is unknown, it can be reset using the alter user Replication Server command.

10. Ensure that either the database or the tables on the updated primary database are marked for replication.

11. Modify the settings in the RSSD and the primary database to have the Replication Agent start at the end of the transaction log:

```
use PRS2_RSSD
go
rs_zeroltm sunak1505x,pri
go
Locator has been reset to zero.
(return status = 0)
isql -Usa -P -Ssunak1505x
use pri
go
dbcc settrunc (ltm,valid)
go
secondary trunc page secondary trunc state dbrepstat
-----
2670                      1                      167
generation id database id database name ltl version
-----
0                          7                      pri                      720
```

12. Increase the generation number by 1 in case the log pages of the new database are numerically less than the log pages of the previous copy of the database:

```
dbcc settrunc (ltm,gen_id,1)
go
secondary trunc page secondary trunc state dbrepstat
-----
2670                      1                      167
generation id database id database name ltl version
-----
0                          7                      pri                      720
```

Note: The generation number was obtained prior to beginning the resynchronization.

13. Truncate the rs_lastcommit table at the replicate database:

```
use rep
go
```

```
truncate table rs_lastcommit
go
```

14. Purge queues of any existing work from primary to replicate:

```
isql -Usa -Psa_pass -SPRS2
sysadmin hibernate_on
go
The Replication Server has now entered hibernation mode.
```

Note: The `sysadmin sqm_purge_queue` command requires the Replication Server to be in hibernate or standalone mode. When the Replication Server is in hibernate or standalone mode, no work is performed.

```
sysadmin sqm_purge_queue, 104,1
go
sysadmin sqm_purge_queue, 106,0
go
```

Note: The queue number used in the `sysadmin sqm_purge_queue` command, is the connection number for the connection, as shown in the `admin who` command. The queue type is either 1 for inbound or 0 for outbound and is available from the `admin who` command.

```
sysadmin hibernate_off
go
The Replication Server has now finished hibernation
mode.
```

15. Turn of the hibernation mode after the queues are purged:

```
isql -Usa -Psa_pass -SPRS2
sysadmin hibernate_off
go
```

16. Resume the connection to the primary and replicate database:

```
resume connection to sunak1505x.pri
go
Connection to 'sunak1505x.pri' is resumed.
resume connection to sunak1505x.rep
go
Connection to 'sunak1505x.rep' is resumed
```

17. On the primary database, start the RepAgent:

```
sp_start_rep_agent pri
go
Replication Agent thread is started for database
'pri'.
(return status = 0)
```

18. In the Replication Server, ensure that both the RepAgent and the DSI threads have been successfully started:

```
admin who
go
Spid Name          State          Info
-----
139 DSI EXEC Awaiting Command 104(1) sunak1505x.pri
135 DSI      Awaiting Message 104 sunak1505x.pri
```

Materialization and Resynchronization

```
129 DIST      Awaiting Wakeup 104 sunak1505x.pri
130 SQT      Awaiting Wakeup 104:1 DIST sunak1505x.pri
123 SQM      Awaiting Message 104:1 sunak1505x.pri
56  SQM      Awaiting Message 104:0 sunak1505x.pri
143 REP AGENT Awaiting Command sunak1505x.pri
145 NRM      Awaiting Command sunak1505x.pri
```

19. Validate that replication is available and working.
20. Allow users on the primary database.

Scenario 5

You can materialize a table to the replicate database where the Replication Server is doing the materialization. In this scenario there are either no users using the table or the primary table can be locked while the materialization is taking place and there are few rows in the table. This scenario lets Replication Server populate the replicate table.

Ensure that the password of the “sa” user is the same for the primary ASE and the Replication Servers. You cannot use this option if these passwords are not identical.

Before you begin

The maintenance user for the replicate database must have access to insert data into the replicate table. The table has already been marked using **sp_setreptable** or **sp_setreplicate**.

Begin the resynchronization

1. Create the replication definition:

```
create replication definition t1_repdef
with primary at sunak1505x.pri
with all tables named t1
(a char(10),
b char(10))
primary key (a)
go
Replication definition 't1_repdef' is created
```

2. Create the subscription:

```
create subscription t1_sub
for t1_repdef
with replicate at sunak1505x.rep
incrementally
subscribe to truncate table
go
Subscription 't1_sub' is in the process of being
created.
```

Note: This option places a hold on the table at the primary database while the selection of the rows for materialization is taking place.

3. Check if the subscription is valid at the primary and replicate site using the **check subscription** command:

```

check subscription t1_sub
for t1_repdef
with replicate at sunak1505x.rep
go
Subscription t1_sub is VALID at the replicate.
Subscription t1_sub is VALID at the primary.

```

4. If the subscription is not valid, check the Replication Server to see if a materialization queue has been created.

```

admin who
go
Spid      Name      State      Info
-----
2         SQM      Awaiting Message 106:-2147483541
t1_repdef.t1_sub

```

If the subscription has not materialized within an acceptable time, and the materialization queue still exists, look in the Replication Server log for any error messages, correct the error, drop the materialization queue (**sysadmin drop_queue**), drop the subscription, and re-create the subscription.

Scenario 6

You can materialize a table where activity to the primary table is going on while the materialization takes place. Use this option when there is a more than a small amount of data to synchronize between the primary and replicate table. This scenario assumes that the table has already been marked using **sp_setreptable** or **sp_setreplicate**.

Begin the resynchronization

1. Create the replication definition:

```

create replication definition t1_repdef
with primary at sunak1505x.pri
with all tables named t1
(a char(10),
b char(10))
primary key (a,b)
go
Replication definition 't1_repdef' is created

```

If the replication definition was previously created with **replicate minimal columns**, alter the replication definition to **replicate all columns**.

2. Define the subscription:

```

define subscription t1_sub
for t1_repdef
with replicate at sunak1505x.rep
subscribe to truncate table
go
Subscription t1_sub is in the process of being
defined.

```

3. Validate that all connections are successful:

```
admin who
go
Spid      Name          State          Info
-----
139      DSI EXEC      Awaiting Command 104(1) sunak1505x.pri
135      DSI           Awaiting Message 104 sunak1505x.pri
129      DIST          Awaiting Wakeup 104 sunak1505x.pri
130      SQT           Awaiting Wakeup 104:1 DIST sunak1505x.pri
123      SQM           Awaiting Message 104:1 sunak1505x.pri
56       SQM           Awaiting Message 104:0 sunak1505x.pri
143      REP AGENT     Awaiting Command sunak1505x.pri
144      NRM           Awaiting Command sunak1505x.pri
156      DSI EXEC      Awaiting Command 106(1) sunak1505x.rep
155      DSI           Awaiting Message 106 sunak1505x.rep
124      SQM           Awaiting Message 106:0 sunak1505x.rep
```

4. Activate subscription with suspension:

```
activate subscription t1_sub
for t1_repdef
with replicate at sunak1505x.rep
with suspension
go
Subscription 't1_sub' is in the process of being
activated.
```

Note: The with suspension clause suspends the DSI to the target to allow the initial materialization of the replicate database. Any changes to the primary table are placed in the outbound queue of the Replication Server, waiting for the DSI to be resumed.

5. Validate that the DSI is now suspended. Enter:

```
admin who
go
Spid      Name          State          Info
-----
139      DSI EXEC      Awaiting Command 104(1) sunak1505x.pri
135      DSI           Awaiting Message 104 sunak1505x.pri
129      DIST          Awaiting Wakeup 104 sunak1505x.pri
130      SQT           Awaiting Wakeup 104:1 DIST
sunak1505x.pri
123      SQM           Awaiting Message 104:1 sunak1505x.pri
56       SQM           Awaiting Message 104:0 sunak1505x.pri
143      REP AGENT     Awaiting Command sunak1505x.pri
144      NRM           Awaiting Command sunak1505x.pri
156      DSI EXEC      Suspended       106(1) sunak1505x.rep
155      DSI           Suspended       106 sunak1505x.rep
124      SQM           Awaiting Message 106:0 sunak1505x.rep
```

6. Copy the data from the primary table:

```
% bcp pri..t1 out 't1.bcp' -Usa -P -Ssunak1505x -c
Starting copy...
5 rows copied.
Clock Time (ms.): total = 9 Avg = 1 (555.56 rows per
sec.)
```

7. Insert the data in to the target table:

```
% bcp rep..t1 in 't1.bcp' -Usa -P -Ssunak1505x -c
Starting copy...
5 rows copied.
Clock Time (ms.): total = 30 Avg = 6 (166.67 rows
per sec.
```

8. On the Replication Server, set autocorrection on:

```
set autocorrection on
for t1_repdef
with replicate at sunak1505x.rep
go
autocorrection' is modified for replication
definition 't1_repdef' with replicate at
'sunak1505x.rep'.
```

Note: If there are multiple replication definitions from the same primary table going to the same replication table, make sure that autocorrection is set on all the replication definitions for this table.

9. Resume the connection to let the data stored in the queues be applied to the target table:

```
resume connection to sunak1505x.rep
go
Connection to 'sunak1505x.rep' is resumed.
```

Note: Because of the overhead associated with autocorrection, Sybase recommends that you do not keep autocorrection on when doing normal replication activities.

10. Send through a marker on the primary so that we know when we no longer need autocorrection. In this example, we are going to use **rs_ticket. On the replication server, validate that **rs_ticket** is available on the DSI. By default, it is available.**

```
isql -Usa -Psa_pass -SPRS2
admin config, "connection", sunak1505x, rep, "ticket"
go
Configuration          Config Value          Run Value
-----
dsi_rs_ticket_report <server default> <server
default>
Default Value  Legal Values Datatype  Status
-----
on              list: on, off string Connection/route
restart required
```

Start the **rs_ticket** marker at the primary database.

```
isql -Usa -P -Ssunak1505x
use pri
go
rs_ticket "done t1"
go
(return status = 0)
```

To check whether **rs_ticket** made it to the replicate database:

Materialization and Resynchronization

```
use rep
go
select ticket from rs_ticket_history where hl =
"done t1"
go
ticket
-----
V=2;H1=done t1;PDB(pri)=11/30/09
12:14:26.253;EXEC(143)=11/30/09 12:14:26.261;B
(143)=19705;DIST(129)=11/30/09
12:14:27.273;DSI(158)=11/30/0912:14:28
294;DSI_T=12;DSI_C=15;RRS=PRS2
```

Note: `rs_ticket` version 2, which includes all stored procedures and tables to support it, is available in Replication Server 15.1 and later. If you have Replication Servers earlier than 15.1, read the Administration Guide for limitations.

11. Turn off autocorrection when the marker is seen at the replicate database:

```
set autocorrection off
for t1_repdef
with replicate at sunak1505x.rep
go
'autocorrection' is modified for replication
definition 't1_repdef' with
replicate at 'sunak1505x.rep'.
```

12. Validate the subscription:

```
validate subscription t1_sub
for t1_repdef
with replicate at sunak1505x.rep
go
```

Index

A

- active database, marking 36
- adding
 - active database 32
 - standby database 37

B

- before you begin 5

C

- check upgrade incompatible version
 - emergency bug fix, one-off, controlled, or instrumental release 10
- checking for a valid installation
 - in Replication Server 10
- choose update installation 10
- configuration 9
- conventions
 - style 1
 - syntax 1
- creating
 - logical connection 32
 - replication definition 49
- custom installation 10

D

- default directory 10
- directories
 - default 10
 - Sybase_Install_Registry 10

E

- errors
 - while starting installer 10

F

- files
 - installation log, Replication Server 10

- si_reg.xml 10
- full installation 10

G

- GUI mode
 - installing Replication Server 10

I

- install directory path 6
- installation 9
 - check upgrade incompatible version 10
 - GUI mode 10
 - Replication Server CD or DVD, mounting 9
 - starting 10
 - typical software components 10
- installer
 - custom installation 10
 - errors in starting 10
 - full installation 10
 - typical installation 10
- instructions 5

M

- materialization 53
- materialization and resynchronization
 - scenario five 74
 - scenario four 68
 - scenario one 53
 - scenario six 75
 - scenario three 64
 - scenario two 60
- mount commands by platform 9
- mounting the Replication Server installation media
 - 9
- multisite availability 21

P

- platform
 - mount commands 9
- prerequisites 5
- primary database, marking 26

Index

R

replicating

- active database 36
- DDL in MSA 28
- DDL in warm standby 46
- DML in MSA 29
- DML in warm standby 47
- primary database 26

replication definitions 49

Replication Server

- configuring 14
- download 6
- EBF 7
- in GUI mode 10
- verifying 19

Replication Server, adding

- primary database 21
- replicate database 24

Replication Server, MSA 21

Replication Server, tracing 48

resynchronization 53

S

SAP Service Marketplace (SMP) 10

Scenario

- five 74
- four 68
- one 53
- six 75

three 64

two 60

SQL-statement replication 50

standard installation 10

standby database, initialize 40

starting

installation 10

switching

active and standby databases 43

Sybase Product Download Center (SPDC) 10

SySAM 10

T

trace function 48

tracing

transactions 48

types of installations

custom 10

full 10

standard 10

typical 10

typical installation 10

W

warm standby 31

warm standby databases

active 31

standby 31