SYBASE®

# ASE Replicator User's Guide

**Adaptive Server® Enterprise**

**12.5.1**

DOCUMENT ID: DC35431-01-1251-02

LAST REVISED: September 2003

Copyright © 2002-2003 by Sybase, Inc. All rights reserved.

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

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

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

Sybase, the Sybase logo, AccelaTrade, ADA Workbench, Adaptable Windowing Environment, Adaptive Component Architecture, Adaptive Server, Adaptive Server Anywhere, Adaptive Server Enterprise, Adaptive Server Enterprise Monitor, Adaptive Server Enterprise Replication, Adaptive Server Everywhere, Adaptive Server IQ, Adaptive Warehouse, Anywhere Studio, Application Manager, AppModeler, APT Workbench, APT-Build, APT-Edit, APT-Execute, APT-FORMS, APT-Translator, APT-Library, AvantGo, AvantGo Application Alerts, AvantGo Mobile Delivery, AvantGo Mobile Document Viewer, AvantGo Mobile Inspection, AvantGo Mobile Marketing Channel, AvantGo Mobile Pharma, AvantGo Mobile Sales, AvantGo Pylon, AvantGo Pylon Application Server, AvantGo Pylon Conduit, AvantGo Pylon PIM Server, AvantGo Pylon Pro, Backup Server, BizTracker, ClearConnect, Client-Library, Client Services, Convoy/DM, Copernicus, Data Pipeline, Data Workbench, DataArchitect, Database Analyzer, DataExpress, DataServer, DataWindow, DB-Library, dbQueue, Developers Workbench, Direct Connect Anywhere, DirectConnect, Distribution Director, e-ADK, E-Anywhere, e-Biz Integrator, E-Whatever, EC Gateway, ECMAP, ECRTP, eFulfillment Accelerator, Embedded SQL, EMS, Enterprise Application Studio, Enterprise Client/Server, Enterprise Connect, Enterprise Data Studio, Enterprise Manager, Enterprise SQL Server Manager, Enterprise Work Architecture, Enterprise Work Designer, Enterprise Work Modeler, eProcurement Accelerator, EWA, Financial Fusion, Financial Fusion Server, Gateway Manager, GlobalFIX, ImpactNow, Industry Warehouse Studio, InfoMaker, Information Anywhere, Information Everywhere, InformationConnect, InternetBuilder, iScript, Jaguar CTS, jConnect for JDBC, Mail Anywhere Studio, MainframeConnect, Maintenance Express, Manage Anywhere Studio, M-Business Channel, M-Business Network, M-Business Server, MDI Access Server, MDI Database Gateway, media.splash, MetaWorks, My AvantGo, My AvantGo Media Channel, My AvantGo Mobile Marketing, MySupport, Net-Gateway, Net-Library, New Era of Networks, ObjectConnect, ObjectCycle, OmniConnect, OmniSQL Access Module, OmniSQL Toolkit, Open Biz, Open Client, Open ClientConnect, Open Client/Server, Open Client/Server Interfaces, Open Gateway, Open Server, Open ServerConnect, Open Solutions, Optima++, PB-Gen, PC APT Execute, PC Net Library, PocketBuilder, Pocket PowerBuilder, Power++, power.stop, PowerAMC, PowerBuilder, PowerBuilder Foundation Class Library, PowerDesigner, PowerDimensions, PowerDynamo, PowerJ, PowerScript, PowerSite, PowerSocket, Powersoft, PowerStage, PowerStudio, PowerTips, Powersoft Portfolio, Powersoft Professional, PowerWare Desktop, PowerWare Enterprise, ProcessAnalyst, Rapport, Report Workbench, Report-Execute, Replication Agent, Replication Driver, Replication Server, Replication Server Manager, Replication Toolkit, Resource Manager, RW-DisplayLib, S-Designor, SDF, Secure SQL Server, Secure SQL Toolset, Security Guardian, SKILS, smart.partners, smart.parts, smart.script, SQL Advantage, SQL Anywhere, SQL Anywhere Studio, SQL Code Checker, SQL Debug, SQL Edit, SQL Edit/TPU, SQL Everywhere, SQL Modeler, SQL Remote, SQL Server, SQL Server Manager, SQL SMART, SQL Toolset, SQL Server/CFT, SQL Server/DBM, SQL Server SNMP SubAgent, SQL Station, SQLJ, STEP, SupportNow, S.W.I.F.T. Message Format Libraries, Sybase Central, Sybase Client/Server Interfaces, Sybase Financial Server, Sybase Gateways, Sybase MPP, Sybase SQL Desktop, Sybase SQL Lifecycle, Sybase SQL Workgroup, Sybase User Workbench, SybaseWare, Syber Financial, SyberAssist, SyBooks, System 10, System 11, System XI (logo), SystemTools, Tabular Data Stream, TradeForce, Transact-SQL, Translation Toolkit, UltraLite.NET, UNIBOM, Unilib, Uninull, Unisep, Unistring, URK Runtime Kit for UniCode, Viewer, Visual Components, VisualSpeller, VisualWriter, VQL, WarehouseArchitect, Warehouse Control Center, Warehouse Studio, Warehouse WORKS, Watcom, Watcom SQL, Watcom SQL Server, Web Deployment Kit, Web.PB, Web.SQL, WebSights, WebViewer, WorkGroup SQL Server, XA-Library, XA-Server and XP Server are trademarks of Sybase, Inc. 03/03

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

This product includes code licensed from RSA Security, Inc. Some portions licensed from IBM are available at http://oss.software.ibm.com/icu4j/.

All other company and product names used herein may be trademarks or registered trademarks of their respective companies.

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

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

# Contents

**About This Book** ..... **vii**

**CHAPTER 1** **Introduction to ASE Replicator** ..... **1**
- Understanding replication concepts ..... 1
  - Table replication ..... 2
  - Stored procedure replication ..... 2
  - Transaction replication ..... 3
  - Guaranteed delivery ..... 3
- Understanding ASE Replicator ..... 4
  - ASE Replicator components ..... 4
  - ASE Replicator processing ..... 7
  - ASE Replicator objects ..... 9
  - ASE Replicator limitations ..... 13

**CHAPTER 2** **Setting Up and Starting ASE Replicator** ..... **15**
- Setting up ASE Replicator ..... 15
  - Configuring the primary Adaptive Server ..... 16
  - Configuring the replicate servers and databases ..... 22
  - Initializing the ASE Replicator process ..... 26
- Starting ASE Replicator ..... 29
  - Using the aserep script ..... 30
  - Using the RUN script ..... 32
- Setting up a replication system ..... 33

**CHAPTER 3** **Administering ASE Replicator** ..... **35**
- Administering the ASE Replicator process ..... 36
  - Configuring ASE Replicator ..... 37
  - Shutting down the ASE Replicator process ..... 38
  - Suspending replication ..... 38
  - Resuming replication ..... 39
- Monitoring ASE Replicator ..... 40
- Managing database connections ..... 41
  - Creating a database connection ..... 41

Deleting a database connection ..... 43
Configuring a database connection ..... 43
Suspending and resuming database connections ..... 44
Getting information about database connections ..... 46
Getting connection configuration information ..... 48
Managing publications and subscriptions ..... 49
Creating publications and subscriptions ..... 50
Deleting publications and subscriptions ..... 51
Suspending and resuming subscriptions ..... 51
Getting information about publications and subscriptions ..... 52
Managing primary and replicate articles ..... 54
Creating primary articles ..... 54
Creating replicate articles ..... 55
Deleting primary articles ..... 57
Deleting replicate articles ..... 58
Materializing and validating replicate articles ..... 59
Getting information about primary articles ..... 60
Getting information about replicate articles ..... 65
ASE Replicator configuration parameters ..... 69
General configuration parameters ..... 69
Connection configuration parameters ..... 78

**CHAPTER 4 ASE Replicator Procedures ..... 83**
sp_addprimaryart ..... 86
sp_addprimaryconn ..... 89
sp_addpub ..... 90
sp_addreplicateart ..... 91
sp_addreplicateconn ..... 95
sp_addsub ..... 96
sp_configprimaryconn ..... 97
sp_configrep ..... 99
sp_configreplicateconn ..... 101
sp_dropprimaryart ..... 103
sp_dropprimaryconn ..... 104
sp_droppub ..... 104
sp_dropreplicateart ..... 105
sp_dropreplicateconn ..... 106
sp_dropsub ..... 107
sp_helpconn ..... 108
sp_helplastcommit ..... 109
sp_helplocator ..... 110
sp_helpprimaryart ..... 111
sp_helpprimaryconn ..... 114
sp_helppub ..... 118

sp_helprep ..... 120
sp_helpreplicateart ..... 125
sp_helpreplicateconn ..... 129
sp_helpsub ..... 133
sp_materializesub ..... 136
sp_resumeprimaryconn ..... 138
sp_resumerep ..... 138
sp_resumereplicateconn ..... 139
sp_resumesub ..... 140
sp_shutdownrep ..... 141
sp_suspendprimaryconn ..... 141
sp_suspendrep ..... 142
sp_suspendreplicateconn ..... 143
sp_suspendsub ..... 144
sp_tracerep ..... 144
sp_validatesub ..... 147

**CHAPTER 5 Troubleshooting ASE Replicator ..... 149**
Problems that prevent replication from starting ..... 149
Primary server configuration problems ..... 149
Remote server configuration problems ..... 150
ASE Replicator configuration problems ..... 151
User login or permission problems ..... 151
Problems that cause ongoing replication to fail ..... 153
Schema changes in primary or replicate tables ..... 153
Datatypes, constraints, and column properties ..... 154

APPENDIX A **Distribution Database Schema ..... 161**
Distribution Database base objects ..... 161
Distribution Database datatypes ..... 161
Base tables in the Distribution Database ..... 162
Base procedures in the Distribution Database ..... 172
Distribution Database shadow tables ..... 172
Shadow tables for primary tables without large objects ..... 173
Shadow tables for primary tables with large objects ..... 174
Shadow tables for primary procedures ..... 175
Distribution Database replicate objects ..... 176
Distribution procedures ..... 176
Replicate article proxy tables ..... 177

**Glossary ..... 179**

**Index ..... 181**

# About This Book

The ASE Replicator *User's Guide* describes how to use the ASE Replicator feature of Sybase® Adaptive Server® Enterprise to implement basic replication from a primary Adaptive Server to one or more remote Adaptive Servers.

**Audience**

This book is intended for System Administrators and Database Administrators who want to implement simple Adaptive Server replication with ASE Replicator.

**How to use this book**

This book contains the following chapters:

Chapter 1, "Introduction to ASE Replicator," provides an introduction to replication system concepts and an overview of the ASE Replicator. This chapter describes the major ASE Replicator components and explains how they work.

Chapter 2, "Setting Up and Starting ASE Replicator," describes the initial setup and configuration procedure for ASE Replicator. The setup procedures in this chapter must be performed after installing the software, and before replication can begin.

Chapter 3, "Administering ASE Replicator," describes administrative operations, including managing and monitoring ASE Replicator and the replication system.

Chapter 4, "ASE Replicator Procedures," describes the ASE Replicator command procedures in detail, including syntax, options, usage, and examples.

Chapter 5, "Troubleshooting ASE Replicator," describes basic troubleshooting and recovery procedures for ASE Replicator.

Appendix A, "Distribution Database Schema," describes the schema of the Distribution Database.

**Related documents**

The Sybase® Adaptive Server® Enterprise documentation set consists of the following:

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

A more recent version of the release bulletin may be available on the World Wide Web. To check for critical product or document information that was added after the release of the product CD, use the Sybase Technical Library.

- The *Installation Guide* for your platform – describes installation, upgrade, and configuration procedures for all Adaptive Server and related Sybase products.
- *What's New in Adaptive Server Enterprise?* – describes the new features in Adaptive Server version 12.5.1, the system changes added to support those features, and the changes that may affect your existing applications.
- *ASE Replicator User's Guide* – describes how to use the ASE Replicator feature of Adaptive Server to implement basic replication from a primary server to one or more remote Adaptive Servers.
- *Component Integration Services User's Guide* – explains how to use the Adaptive Server Component Integration Services feature to connect remote Sybase and non-Sybase databases.
- *Configuring Adaptive Server Enterprise* for your platform – provides instructions for performing specific configuration tasks for Adaptive Server.
- *EJB Server User's Guide* – explains how to use EJB Server to deploy and execute Enterprise JavaBeans in Adaptive Server.
- *Error Messages and Troubleshooting Guide* – explains how to resolve frequently occurring error messages and describes solutions to system problems frequently encountered by users.
- *Full-Text Search Specialty Data Store User's Guide* – describes how to use the Full-Text Search feature with Verity to search Adaptive Server Enterprise data.
- *Glossary* – defines technical terms used in the Adaptive Server documentation.
- *Historical Server User's Guide* – describes how to use Historical Server to obtain performance information for SQL Server® and Adaptive Server.
- *Java in Adaptive Server Enterprise* – describes how to install and use Java classes as data types, functions, and stored procedures in the Adaptive Server database.

- *Job Scheduler User's Guide* – provides instructions on how to install and configure, and create and schedule jobs on a local or remote Adaptive Server using the command line or a graphical user interface (GUI).
- *Monitor Client Library Programmer's Guide* – describes how to write Monitor Client Library applications that access Adaptive Server performance data.
- *Monitor Server User's Guide* – describes how to use Monitor Server to obtain performance statistics from SQL Server and Adaptive Server.
- *Performance and Tuning Guide* – is a series of four books that explains how to tune Adaptive Server for maximum performance:
  - *Basics* – the basics for understanding and investigating performance questions in Adaptive Server.
  - *Locking* – describes how the various locking schemas can be used for improving performance in Adaptive Server.
  - *Optimizer and Abstract Plans* – describes how the optimizer processes queries and how abstract plans can be used to change some of the optimizer plans.
  - *Monitoring and Analyzing* – explains how statistics are obtained and used for monitoring and optimizing performance.
- *Quick Reference Guide* – provides a comprehensive listing of the names and syntax for commands, functions, system procedures, extended system procedures, datatypes, and utilities in a pocket-sized book.
- *Reference Manual* – is a series of four books that contains the following detailed Transact-SQL® information:
  - *Building Blocks* – Transact-SQL datatypes, functions, global variables, expressions, identifiers and wildcards, and reserved words.
  - *Commands* – Transact-SQL commands.
  - *Procedures* – Transact-SQL system procedures, catalog stored procedures, system extended stored procedures, and `dbcc` stored procedures.
  - *Tables* – Transact-SQL system tables and `dbcc` tables.

- *System Administration Guide* – provides in-depth information about administering servers and databases. This manual includes instructions and guidelines for managing physical resources, security, user and system databases, and specifying character conversion, international language, and sort order settings.
- *System Tables Diagram* – illustrates system tables and their entity relationships in a poster format. Available only in print version.
- *Transact-SQL User's Guide* – documents Transact-SQL, Sybase's enhanced version of the relational database language. This manual serves as a textbook for beginning users of the database management system. This manual also contains descriptions of the pubs2 and pubs3 sample databases.
- *Using Adaptive Server Distributed Transaction Management Features* – explains how to configure, use, and troubleshoot Adaptive Server DTM features in distributed transaction processing environments.
- *Using Sybase Failover in a High Availability System* – provides instructions for using Sybase's Failover to configure an Adaptive Server as a companion server in a high availability system.
- *Utility Guide* – documents the Adaptive Server utility programs, such as isql and bcp, which are executed at the operating system level.
- *Web Services User's Guide* – explains how to configure, use, and troubleshoot Web Services for Adaptive Server.
- *XA Interface Integration Guide for CICS, Encina, and TUXEDO* – provides instructions for using the Sybase DTM XA interface with X/Open XA transaction managers.
- *XML Services in Adaptive Server Enterprise* – describes the Sybase native XML processor and the Sybase Java-based XML support, introduces XML in the database, and documents the query and mapping functions that comprise XML Services.

**Other sources of information**

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

- The Getting Started CD contains release bulletins and installation guides in PDF format, and may also contain other documents or updated information not included on the Technical Library CD. It is included with your software. To read or print documents on the Getting Started CD you need Adobe Acrobat Reader (downloadable at no charge from the Adobe Web site, using a link provided on the CD).

- The Technical Library CD contains product manuals and is included with your software. The DynaText reader (included on the Technical Library CD) allows you to access technical information about your product in an easy-to-use format.

  Refer to the *Technical Library Installation Guide* in your documentation package for instructions on installing and starting the Technical Library.

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

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

**Sybase certifications on the Web**

Technical documentation at the Sybase Web site is updated frequently.

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

1 Point your Web browser to Technical Documents at http://www.sybase.com/support/techdocs/.

2 Select Products from the navigation bar on the left.

3 Select a product name from the product list and click Go.

4 Select the Certification Report filter, specify a time frame, and click Go.

5 Click a Certification Report title to display the report.

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

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

1 Point your Web browser to Technical Documents at http://www.sybase.com/support/techdocs/.

2 Click MySybase and create a MySybase profile.

**Sybase EBFs and software updates**

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

1 Point your Web browser to the Sybase Support Page at http://www.sybase.com/support.

2 Select EBFs/Updates. Enter user name and password information, if prompted (for existing Web accounts) or create a new account (a free service).

3 Select a product.

4 Specify a time frame and click Go.

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

## Conventions

The following style conventions are used in this manual:

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

  ```
  this font
  ```

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

  ```
  this font
  ```

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

  */usr/u/sybase*

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

  sqlupgrade

Table 1 shows the conventions for syntax statements that appear in this manual:

***Table 1: Syntax statement conventions***

| Key | Definition |
|---|---|
| command | Command names, command option names, configuration parameter names, and other keywords are in this font in body text. |
| *variable* | Variables, options, or words that stand for values that you fill in, are in *this font* in body text. |
| { } | Curly braces indicate that you choose at least one of the enclosed options. Do not include braces in your option. |
| [ ] | Brackets mean choosing one or more of the enclosed options is optional. Do not include brackets in your option. |
| ( ) | Parentheses are to be typed as part of the command. |
| \| | The vertical bar means you may select only one of the options shown. |

| Key | Definition |
|---|---|
| , | The comma means you may choose as many of the options shown as you like, separating your choices with commas to be typed as part of the command. |

In this manual, most of the examples are in lowercase. However, you can disregard case when typing Transact-SQL keywords. For example, SELECT, Select, and select are the same.

Adaptive Server's sensitivity to the case of database objects, such as table names, depends on the sort order installed on Adaptive Server. You can change case sensitivity for single-byte character sets by reconfiguring the Adaptive Server sort order. See the *System Administration Guide* for more information.

**If you need help**

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

CHAPTER 1

# Introduction to ASE Replicator

This chapter provides an introduction to replication concepts and an overview of ASE Replicator.


| Topic | Page |
|---|---|
| Understanding replication concepts | 1 |
| Understanding ASE Replicator | 4 |


## Understanding replication concepts

A replication system maintains consistent, synchronized data in separate databases. It does that mainly by recording the data-changing operations in one database (called the **primary database**), and sending those operations to another database (called the **replicate database**). Data-changing operations thus captured and sent are called **replicated transactions**.

As shown in Figure 1-1, the primary database **publishes** replicated transactions, and the replicate database **subscribes** to replicated transactions.

***Figure 1-1: Simple replication scenario***

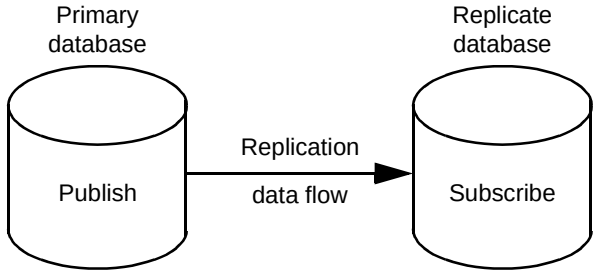
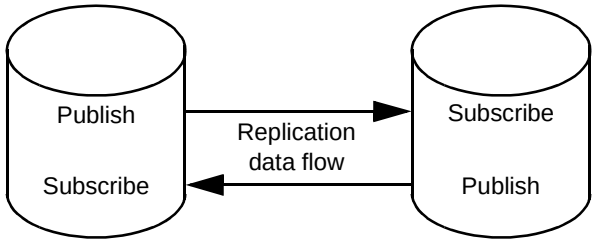



Figure 1-2 illustrates **bidirectional replication**, in which a single database acts as both a primary database and a replicate database. Bidirectional replication places a special requirement on a replication system. The system must be able to filter out replicated transactions that

were received from another database to prevent circular replication back to the original primary database.

***Figure 1-2: Bidirectional replication scenario***




## Table replication

Replicated transactions are published by table. When data-changing operations affect the contents of a published table in the primary database, they are recorded for subsequent distribution to a replicate database.

A replicate database can be a subset of a primary database, with some, but not all of the tables in the primary database. Therefore, not all of the tables in a primary database have to be published.

To receive replicated transactions, the replicate database must subscribe to a published table in the primary database, and it must identify the subscribing replicate database table. Replicated transactions from the primary database are distributed to subscribing tables in the replicate database.

## Stored procedure replication

In addition to replicating data-changing operations, another way to maintain consistent, synchronized data is to replicate the invocation of stored procedures that change data. Replicating a stored procedure invocation can sometimes be more efficient than replicating the individual data-changing operations that the procedure produces.

When a stored procedure is published, the replication system must identify the procedure and record the input parameter values that are specified when the procedure is invoked. The system must then distribute that procedure invocation to any subscribing replicate database.

Stored procedure replication places a special requirement on a replication system. When a published procedure generates a data-changing operation on a published table, the replication system must be able to recognize the operation generated by the published procedure, and replicate only the procedure invocation and not the data-changing operation produced by it.

## Transaction replication

Transaction replication ensures database integrity and transactional consistency between the databases. All data-changing operations that are replicated are considered to be "transactions," even though they might not correspond to an actual transaction in the primary database.

For example, if an actual transaction changes both published tables and unpublished tables in the primary database, only the data-changing operations on published tables are replicated. Operations on unpublished tables are not replicated, but transactional consistency is maintained if the replicate database contains only tables that correspond to published tables in the primary database.

Even though a replicated "transaction" is really just a set of data-changing operations, those operations are grouped in an atomic collection, and each collection represents the results of a committed transaction in the primary database. Only committed transaction operations should be replicated; transaction operations that are rolled back should not be replicated.

Stored procedure invocations are considered part of a transaction, just like data-changing operations on a table. The procedure invocations are not necessarily transactions in themselves.

## Guaranteed delivery

In a replication system, guaranteed delivery means that all data-changing operations or procedure invocations published by a primary database are guaranteed to be received by the subscribing replicate database, regardless of any hardware, software, or network problems that might interfere with replication.

The main mechanism used to provide guaranteed delivery is a **stable queue**, which records the replicated transactions in a nonvolatile form (on disk), until the subscribing replicate database confirms that it received them.

# Understanding ASE Replicator

ASE Replicator is a feature of Adaptive Server Enterprise that provides basic replication from an Adaptive Server primary database to one or more Adaptive Server replicate databases.

ASE Replicator provides the following replication system functionality:

- Manages replication system objects and database objects with a simple publish-and-subscribe model
- Replicates data-changing operations on tables and invocations of stored procedures
- Supports bidirectional replication, filtering out replicated transactions to prevent circular replication
- Maintains database integrity and transactional consistency at multiple sites
- Provides guaranteed delivery of replicated transactions

ASE Replicator uses Component Integration Services (CIS) to handle operation (DML) and RPC distribution to replicate databases.

For more information about CIS, see the *Component Integration Services User's Guide*.

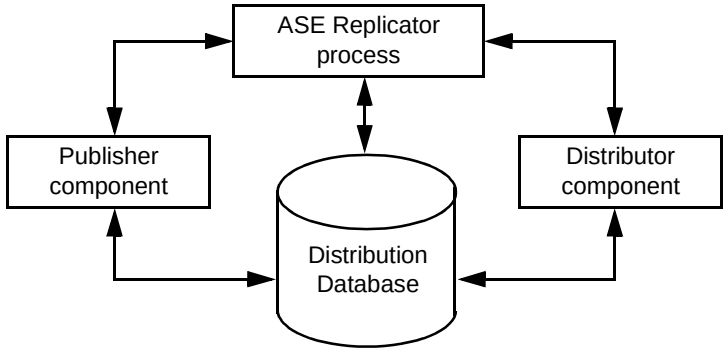
## ASE Replicator components

ASE Replicator consists of the following components:

- ASE Replicator process
- Distribution Database
- Publisher
- Distributor

The ASE Replicator components are tightly integrated (they cannot be installed or accessed separately), and they work together to provide complete replication system functionality. Figure 1-3 shows interactions between the ASE Replicator components.

***Figure 1-3: ASE Replicator components***




## ASE Replicator process

The ASE Replicator process is an external application that connects to and interacts with Adaptive Server, and it receives all its instructions from the Adaptive Server to which it connects. The ASE Replicator process manages and coordinates all the other ASE Replicator components and all replication system processing.

ASE Replicator can support multiple primary databases, replicate databases, and replicate database servers, but each instance of ASE Replicator (the ASE Replicator process) can support only one primary Adaptive Server.

Even though you can set up ASE Replicator on two Adaptive Servers to support bidirectional replication, you cannot coordinate the operations of multiple instances of ASE Replicator, nor can they share data or metadata.

## Distribution Database

The Distribution Database is a user database in Adaptive Server. It stores the metadata needed to support ASE Replicator, and it resides on the same Adaptive Server as the primary database. You create the Distribution Database when you set up the primary Adaptive Server to work with ASE Replicator.

The Distribution Database contains the following objects:

- Stable queue – consists of the ASE Replicator transaction log table, and one shadow table for each published table or stored procedure in the primary database:
  - Transaction log table – stores metadata from the primary database's Adaptive Server transaction log for all replicated transactions.

    - Shadow tables – store the data associated with transaction operations on tables and procedure invocations in the primary database.
- Distribution procedures, for both tables and stored procedures:
    - Table distribution procedures – stored procedures that read the shadow tables and apply replicated transactions to the CIS proxy (replicate) tables.
    - Stored procedure distribution procedures – read the shadow tables and execute stored procedures in the replicate database as RPCs.
- CIS proxy tables – enable ASE Replicator to use CIS to send replicated transactions to replicate tables in remote replicate databases. Each replicate table in a replicate database is represented by a CIS proxy table in the Distribution Database.
- Metadata tables – store all the metadata that ASE Replicator uses to manage and control the replication process.

ASE Replicator maintains and controls the Distribution Database.

---

**Note** In general, there is no need for direct maintenance or administration of the Distribution Database. However, the System Administrator must accommodate the Distribution Database when tuning Adaptive Server performance and allocating resources.

---

For more information about the Distribution Database, see Appendix A, “Distribution Database Schema.”

## Publisher component

The ASE Replicator Publisher component interacts with the primary database and the Distribution Database to:

- Read the primary database’s Adaptive Server transaction log to acquire primary transaction data and metadata for published tables and published stored procedures
- Build transaction operation metadata to be stored in the Distribution Database transaction log table
- Write the transaction operation metadata to the transaction log table, and write the transaction data to the shadow table (or tables)

- Update the locator value that identifies the last successfully published transaction in the primary database's transaction log, and manage the primary database's log truncation point

### Distributor component

The ASE Replicator Distributor component interacts with the Distribution Database to:

- Read the Distribution Database transaction log table to find transaction operations to replicate
- Combine operations to form complete transactions to send to the replicate database
- Execute the distribution procedure associated with each replicate table and stored procedure affected by the transaction

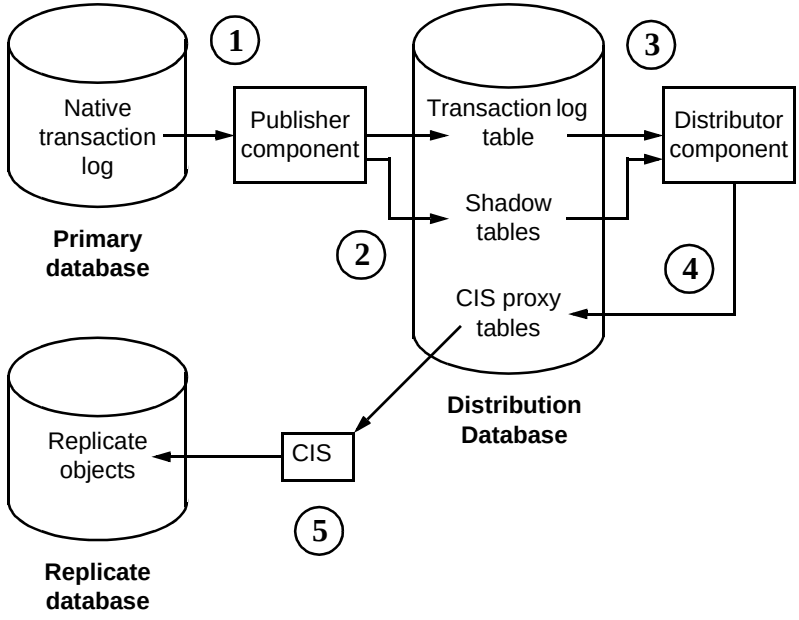
When executed by the Distributor component, distribution procedures read the shadow tables to build the transaction operations to be replicated, then apply those operations to the CIS proxy (replicate) tables in the Distribution Database.

## ASE Replicator processing

The diagram in Figure 1-4 shows how the ASE Replicator components work together to replicate transaction operations from a primary database to a replicate database.

**Note** The process is slightly different for stored procedure replication.

***Figure 1-4: ASE Replicator transaction replication***




1 The Publisher component reads the primary database's native Adaptive Server transaction log and builds transaction metadata and operation records that describe the primary transaction.

2 The Publisher component writes the transaction metadata in the transaction log table and the transaction operation data in the shadow table (or tables) in the Distribution Database.

3 The Distributor component reads the metadata in the transaction log table, determines the transaction to be applied to a replicate object, then executes the distribution procedure associated with the replicate object.

4 Distribution procedures read the transaction operation data in the shadow tables, then apply the replicated transaction operations to the CIS proxy tables in the Distribution Database.

5 CIS propagates the replicated transaction operations in the proxy tables to the replicate tables in the replicate database.

In stored procedure replication, the distribution procedures issue remote procedure calls (RPCs) to the replicate database, instead of applying operations to a CIS proxy table.

## ASE Replicator objects

ASE Replicator objects are the metadata entities that define the relationships between primary and replicate databases, and the objects within those databases.

ASE Replicator objects include:

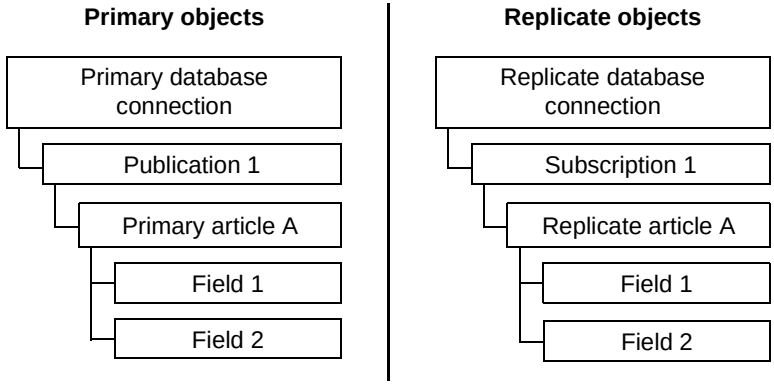
- Database connections – identify a specific database on a specific data server. There are two types of database connections: primary and replicate.
- Publications and subscriptions – act as containers to organize primary or replicate articles in a database. Publications organize the primary articles in a primary database; subscriptions organize the replicate articles in a replicate database.
- Articles – identify the individual database objects (tables or stored procedures) involved in replication. Primary articles identify the published objects in the primary database. Replicate articles subscribe to primary articles, and identify the subscribing objects in the replicate database.
- Fields – identify the objects within an article, that is, the columns in a table or the input parameters of a stored procedure. You can use fields to publish a subset of a primary object, and subscribe to a subset of a primary article.

ASE Replicator objects are stored in the Distribution Database.

Object hierarchy

Figure 1-5 illustrates the hierarchy of ASE Replicator objects.

***Figure 1-5: ASE Replicator object hierarchy***




A primary database connection contains publications, which in turn contain primary articles, and those in turn contain fields.

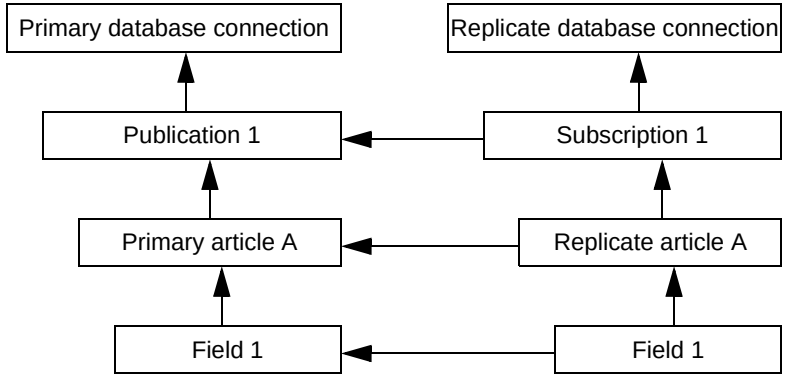
A replicate database connection contains subscriptions, which in turn contain replicate articles, and those in turn contain fields.

Object dependency

There is also a dependency between primary and replicate objects, as shown in Figure 1-6.

For primary objects, all dependencies are hierarchical. For replicate objects, dependencies are *both* hierarchical (dependent on the next-higher-level replicate object), and horizontal (dependent on the same-level primary object).

***Figure 1-6: ASE Replicator object dependency***




## Database connections

For ASE Replicator to recognize a database, you must define a connection to that database. Before you create publications or subscriptions, and primary or replicate articles, you must first define database connections.

Each database connection can be defined as either a **primary connection** or a **replicate connection**. The connection type depends on the role of the database in the replication system. A database with only one role requires only one connection. A database that serves as both a primary and a replicate in a bidirectional replication system requires both primary and replicate database connections.

ASE Replicator identifies each database connection by the unique combination of connection type (primary or replicate) and connection name (data server name and database name).

Maintenance User

ASE Replicator records a **Maintenance User** name for each database connection.

In a replicate database, ASE Replicator uses the Maintenance User name to apply the replicated transactions and procedure invocations.

In a primary database, ASE Replicator uses the Maintenance User name to filter out any transactions that were replicated from another primary database (in a bidirectional replication system). In a database that serves as both a primary database and a replicate database, replicate transactions applied by the Maintenance User must be distinguished from primary transactions to prevent infinite, circular replication.

---

**Note** ASE Replicator supports replication from a primary data server to any number of replicate data servers, but each replicate server can act as a replicate for only one primary server. ASE Replicator does *not* support replication from more than one primary server to a single replicate server.

---

## Publications and subscriptions

Publications and subscriptions are always defined within the context of a database connection; therefore, each publication or subscription is associated with a specific database connection. Publications are defined in primary database connections; subscriptions are defined in replicate database connections. Each database connection can contain many publications or subscriptions.

Publications and subscriptions allow you to organize primary and replicate articles. Before you can create an article, you must first create a publication or subscription to contain the article. Each publication or subscription can contain many articles.

Each subscription refers to (subscribes to) a specific publication. A publication can be subscribed to by any number of subscriptions, in any number of replicate database connections. Each replicate article within in a subscription subscribes to a primary article in the publication to which that subscription refers.

## Articles

Articles identify the database objects involved with replication. Primary articles identify the source of replicated transactions, that is, objects in the primary database. Replicate articles subscribe to primary articles, and identify the destinations of replicated transactions, which are objects in the replicate database.

Primary articles

You create primary articles to identify the objects in the primary database (tables or stored procedures) for which you want to publish transactions. Each object in the primary database can be identified by only one primary article, so there is a one-to-one relationship between a primary database object and a primary article.

Primary articles that publish tables can identify a subset of the columns in the table to be published. Primary articles that publish stored procedures can identify a subset of the input parameters to be published.

You must create each primary article in an existing publication, but after a primary article is created, you can add it to any number of publications. A primary article must belong to at least one publication, and it may belong to more than one publication.

Replicate articles

You create replicate articles to identify the objects in a replicate database that you want to receive the published transactions. Each object in the replicate database can be identified by only one replicate article.

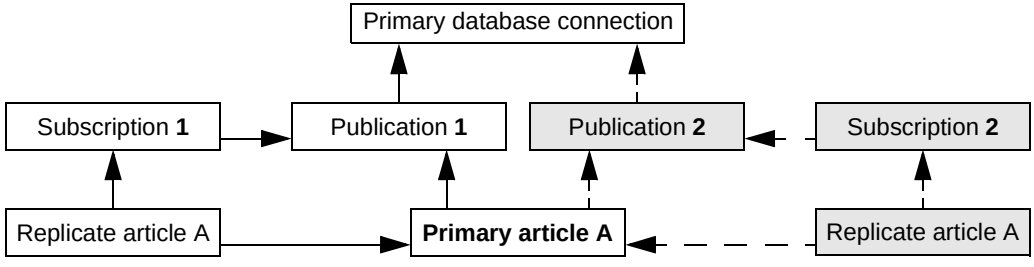
Each replicate article identifies both a primary article that is the source of the published data, and a replicate object in the replicate database that is the destination of the published data.

Replicate articles can subscribe to a subset of the columns or parameters that are published in a primary article. Replicate articles can also use a where clause to further select the operations to which they subscribe.

You must create each replicate article in an existing subscription, and that subscription must subscribe to a publication that contains the primary article identified by the replicate article.

Figure 1-7 illustrates the relationship of multiple replicate articles to a single primary article that belongs to more than one publication. Although the replicate articles belong to different subscriptions, which subscribe to different publications, the replicate articles both subscribe to the same primary article.

***Figure 1-7: Primary article in more than one publication***



### Fields

Fields are the elements within an article. A field represents the smallest database object that can be identified for replication. In tables, fields identify columns. In stored procedures, fields identify input parameters.

When you create a primary article, you can either publish all fields, or you can specify a subset of the fields for publication in that article. Likewise, when you create a replicate article, you can either subscribe to all published fields in the primary article, or you can specify a subset of the published fields to be subscribed to.

By specifying fields in both primary and replicate articles, you have two levels of selection to determine the data replicated.

## ASE Replicator limitations

Although ASE Replicator provides a basic replication facility for Adaptive Server Enterprise, it is not a comprehensive replication solution.

There are some applications for which ASE Replicator is neither intended nor well suited. These include:

- Warm standby systems
- High-volume replication systems
- Scalable, enterprise-wide distribution
- Complex and customizable subscription resolution
- Customizable, user-defined datatype translation
- Support for heterogeneous (for example, DB2 to Adaptive Server) replication

To support such high-performance replication requirements, Sybase provides the Replication Server™ product and Replication Server Options™ product sets.

CHAPTER 2

# Setting Up and Starting ASE Replicator

This chapter describes how to set up and start ASE Replicator. It also provides a brief overview of how to set up a replication system with ASE Replicator.


| Topic | Page |
|---|---|
| Setting up ASE Replicator | 15 |
| Starting ASE Replicator | 29 |
| Setting up a replication system | 33 |


## Setting up ASE Replicator

There are three major tasks required to set up ASE Replicator:

- "Configuring the primary Adaptive Server" on page 16
- "Configuring the replicate servers and databases" on page 22
- "Initializing the ASE Replicator process" on page 26

**Note** Each Adaptive Server can have only one instance of the ASE Replicator process.

### Before you begin

Before you can perform the following procedures to set up ASE Replicator, you must install Adaptive Server Enterprise 12.5.1 using the instructions in the *Installation Guide* for your platform.

**Note** You cannot install or enable the ASE Replicator on an Adaptive Server version earlier than 12.5.0.1.

## Configuring the primary Adaptive Server

To set up ASE Replicator, you must configure the primary database server—the Adaptive Server on which the Distribution Database and the primary databases will reside. This includes:

- Enabling and configuring CIS
- Setting up the ASE Replicator system user
- Defining a local name and remote alias for the primary Adaptive Server
- Defining a remote server name for the ASE Replicator process
- Configuring the tempdb database
- Creating the Distribution Database

You must have either a System Administrator or System Security Officer user role in the primary Adaptive Server to perform these procedures.

### Enabling and configuring CIS

If you are setting up ASE Replicator for an established Adaptive Server with existing databases, verify that the Adaptive Server configuration meets the following requirements:

- Component Integration Services (CIS) is enabled.
- CIS RPC handling is enabled.
- CIS maximum remote connections is set to 20 or more.

If your Adaptive Server currently meets all these configuration requirements, you can skip this section and go to “Setting up the ASE Replicator system user” on page 17.

In Adaptive Server 12.5 and later, CIS is enabled by default. In earlier versions of Adaptive Server, CIS is *not* enabled by default.

If CIS is *not* already enabled for the primary Adaptive Server, you must enable it with sp_config, then restart Adaptive Server so the static parameter enable cis takes effect.

You need not restart the Adaptive Server if CIS is already enabled.

---

**Note** Use sp_config to find out if CIS is already enabled for the primary Adaptive Server.

---

❖ **To enable and configure CIS**

1 Log in to the primary Adaptive Server with a System Administrator user role so you can change the Adaptive Server configuration.

2 Enable CIS:

```
use master
sp_configure "enable cis", 1
```

3 Set the default method for RPC handling to use CIS access methods:

```
sp_configure "cis rpc handling", 1
```

4 Set the maximum number of concurrent connections that can be made to remote servers by CIS to at least 20:

```
sp_configure "max cis remote connections", 20
```

**Note** The actual number of CIS remote server connections that your Adaptive Server needs depends on the environment. Sybase recommends at least 20 CIS remote server connections for use with ASE Replicator.

5 Restart Adaptive Server to have the static parameter enable cis take effect.

**Note** You need not restart the Adaptive Server if CIS was already enabled.

## Setting up the ASE Replicator system user

You must create an Adaptive Server user login for ASE Replicator. ASE Replicator uses this login to access the primary database's transaction log and the Distribution Database. The ASE Replicator system user login must have the Replication role.

❖ **To set up the ASE Replicator system user login**

1 Log in to the primary Adaptive Server with a System Administrator or System Security Officer user role.

2 Create the ASE Replicator system user login ID:

```
use master
sp_addlogin rep_login, passwd
```

where *rep_login* is the user login ID of the ASE Replicator, and *passwd* is the password for the ASE Replicator system user.

3 Assign the Replication role to the ASE Replicator system user login ID:

```
grant role replication_role to rep_login
```

where *rep_login* is the user login ID of the ASE Replicator system user.

4 Add the ASE Replicator system user to each primary database that will be a source of data:

```
use pdb
sp_adduser rep_login, rep_user
```

where *pdb* is the name of the primary database, *rep_login* is the user login ID of the ASE Replicator system user, and *rep_user* is the user name of the ASE Replicator system user in the primary database.

**Note** The user name (*rep_user*) is optional. If you want the user name in the database to be the same as the user login ID (*rep_login*), you need not specify the user name in sp_adduser.

5 In each primary database, grant create procedure permission to the ASE Replicator system user:

```
grant create procedure to rep_user
```

where *rep_user* is the user name of the ASE Replicator system user.

6 In each primary database, grant select permission to the ASE Replicator system user on each table that you want to publish:

```
grant select on table_name to rep_user
```

where *table_name* is the name of a table in the primary database, and *rep_user* is the user name of the ASE Replicator system user.

7 In each primary database, grant execute permission to the ASE Replicator system user on each stored procedure that you want to publish:

```
grant execute on proc_name to rep_user
```

where *proc_name* is the name of a stored procedure in the primary database, and *rep_user* is the user name of the ASE Replicator system user.

**Note** You must perform steps 4 through 7 for each primary database.

## Defining a local name and remote alias for the primary Adaptive Server

The primary Adaptive Server must have a local server name defined for itself, and a remote alias name "local" defined.

If a local server name is *not* already defined for the primary Adaptive Server, you must define a local server name, then restart Adaptive Server so the new entry in the sysservers table takes effect.

You need not restart the Adaptive Server if a local server name is already defined for the primary Adaptive Server.

---

**Note** Use sp_helpserver to find out if a local server name is already defined for the primary Adaptive Server.

---

❖ **To define a local name and remote alias for the primary Adaptive Server**

1 Log in to the primary Adaptive Server with a System Security Officer user role.

2 Define the local server name for the primary Adaptive Server:

```
use master
sp_addserver ds_name, local
```

where *ds_name* is the name of the primary Adaptive Server.

3 Define a server named "local" as a remote alias for the primary Adaptive Server:

```
sp_addserver local, ASEnterprise, ds_name
```

where *ds_name* is the name of the primary Adaptive Server.

4 Restart Adaptive Server to get the new local server name entry in the sysservers table to take effect.

---

**Note** You need not restart the Adaptive Server if a local server name was already defined for the primary Adaptive Server.

---

## Defining a remote server name for the ASE Replicator process

You must define a remote server name for the ASE Replicator process so the primary Adaptive Server can communicate with it.

❖ **To define a remote server name for ASE Replicator**

1 Log in to the primary Adaptive Server with a System Security Officer user role.

2 Define a remote server name for the ASE Replicator process:

```
use master
sp_addserver ASE_Rep, sql_server
```

where *ASE_Rep* is the name of the ASE Replicator server.

3 Add an entry for the ASE Replicator server name, host name, and port number to the interfaces file.

## Configuring the tempdb database

ASE Replicator uses a temporary table created by a stored procedure to manage its log truncation process. This requires that you set the ddl in tran option in the tempdb database.

❖ **To configure the tempdb database**

1 Log in to the primary Adaptive Server with a System Administrator user role.

2 Set the ddl in tran option for the tempdb database:

```
use master
sp_dboption tempdb, "ddl in tran", true
```

3 Run the checkpoint command in the tempdb database.

## Creating the Distribution Database

To complete the primary Adaptive Server configuration for ASE Replicator, you must create the Distribution Database. ASE Replicator uses the Distribution Database to maintain its stable queue and metadata objects.

❖ **To create the Distribution Database**

1 Log in to the primary Adaptive Server with a System Administrator user role.

2 Create database devices for the Distribution Database and its log, using the disk init command to initialize database devices. For example:

```
use master
disk init name = "DDB_dev",
physname = "/devices/ddb_dev.dat",
size = "100M"
dsync = true
```

where *DDB_dev* is the database device name of the Distribution Database device, and *ddb_dev.dat* is the name of the operating system file mapped to the database device name.

**Note** Sybase recommends that you create separate database devices for the Distribution Database and its log.

See the *System Administration Guide* for more information on creating database devices and using the disk init command.

3 Create the Distribution Database, using the create database command. For example:

```
create database DDB_name on DDB_dev = "100M"
log on DDBlog_dev = "100M"
```

where *DDB_name* is the name of the Distribution Database, *DDB_dev* is the database device name of the Distribution Database device, and *DDBlog_dev* is the database device name of the Distribution Database log device.

**Note** After you create the Distribution Database, dump the master database to facilitate recovery if the master database is damaged.

4 Add the ASE Replicator system user to the Distribution Database:

```
use DDB_name
sp_adduser rep_login, rep_user
```

where *DDB_name* is the name of the Distribution Database, *rep_login* is the user login ID of the ASE Replicator system user, and *rep_user* is the user name of the ASE Replicator system user in the Distribution Database.

5 Grant create table and create procedure permissions to the ASE Replicator system user in the Distribution Database:

```
grant create table, create procedure to rep_user
```

where *rep_user* is the user name of the ASE Replicator system user.

6 Set the following database options for the Distribution Database:

- Turn off the ddl in tran option:

```
use master
sp_dboption DDB_name, "ddl in tran", false
```

  where *DDB_name* is the name of the Distribution Database.

- Turn on the select into/bulkcopy/pllsort option:

```
sp_dboption DDB_name, "select
into/bulkcopy/pllsort", true
```

  where *DDB_name* is the name of the Distribution Database.

7 Run the checkpoint command in the Distribution Database.

## Configuring the replicate servers and databases

To allow ASE Replicator to replicate transactions to a remote server, configure the replicate data servers and databases by:

- Identifying the remote server
- Setting up a separate Maintenance User login (optional)
- Granting permissions in the replicate database

You must have a System Security Officer user role in the primary Adaptive Server, and either a System Administrator or System Security Officer user role in the remote server to perform these procedures.

**Note** An Adaptive Server on which a replicate database resides must support the CIS feature.

### Identifying the remote server

You must define a remote server name in the primary Adaptive Server for the remote (replicate database) server.

❖ **To identify the remote server**

1 Log in to the primary Adaptive Server with a System Security Officer user role.

2 Define the remote server name for the replicate database server:

```
use master
sp_addserver lname, ASEnterprise, pname
```

where *lname* is the name of the replicate database server as known to the primary Adaptive Server, and *pname* is the replicate database server's name in the interfaces file (if different from *lname*).

For more information on using sp_addserver to define remote servers, see the *Reference Manual*.

3 Add an entry for the replicate database server to the interfaces file on the primary Adaptive Server host.

---

**Note** You must perform steps 2 and 3 for each remote server you want to act as a replicate database server.

---

## Setting up a separate a Maintenance User login

In the replicate database, all replicated transactions are applied by the Maintenance User. By default, the Maintenance User login is the ASE Replicator system user login at the primary Adaptive Server.

Setting up a different Maintenance User login is optional:

- If you want ASE Replicator to use the same login (the ASE Replicator system user login) at the remote server, skip the following procedure, and continue with "Granting permissions in the replicate database" on page 24.
- If you want ASE Replicator to use a different login at the remote server (that is, different from the ASE Replicator system user login at the primary Adaptive Server), use the following procedure to add an external login for the ASE Replicator system user.

### ❖ To set up a different Maintenance User login

1 Log in to the primary Adaptive Server with a System Security Officer user role.

2 Create an external login to map the ASE Replicator system user login to a different login at the replicate database server:

```
use master
sp_addexternlogin server, rep_user, externname,
externpw
```

where *server* is the name of the replicate database server, *rep_user* is the login name of the ASE Replicator system user on the primary Adaptive Server, *externname* is the name of the login account on the replicate (remote) database server, and *externpw* is the password for the login account.

For more information on using sp_addexternlogin to create external logins, see the *Reference Manual.*

---

**Note** If you create an external login account, make sure the user ID associated with that external login is added to the remote server and to each replicate database, and grant the appropriate permissions in each replicate database to that user login name.

---

## Granting permissions in the replicate database

You must add the ASE Replicator system user (or Maintenance User) to the remote server, and you must grant permissions to that user in each replicate database.

❖ **To grant ASE Replicator permissions in the replicate database**

1 Log in to the replicate (remote) database server with either a System Administrator or System Security Officer user role.

2 Create the ASE Replicator system user (or Maintenance User) login ID in the replicate database server:

```
use master
sp_addlogin rep_login, passwd
```

where *rep_login* is the user login ID of the ASE Replicator system user (or Maintenance User), and *passwd* is the password for that user login.

3 Add the ASE Replicator system user (or Maintenance User) to each replicate database:

```
use rdb
sp_adduser rep_login, rep_user
```

where *rdb* is the name of the replicate database, *rep_login* is the user login ID of the ASE Replicator system user (or Maintenance User), and *rep_user* is the user name of the ASE Replicator system user (or Maintenance User) in the replicate database.

**Note** The user name (*rep_user*) is optional. If you want the user name in the database to be the same as the user login ID (*rep_login*), you need not specify the user name in sp_adduser.

4 Grant select permission on the syspartitions table to the ASE Replicator system user (or Maintenance User) in each replicate database:

```
grant select on syspartitions to rep_user
```

where *rep_user* is the user name of the ASE Replicator system user (or Maintenance User).

5 Grant create table permission to the ASE Replicator system user (or Maintenance User) in each replicate database:

```
grant create table to rep_user
```

where *rep_user* is the user name of the ASE Replicator system user (or Maintenance User).

6 If there are existing replicate tables and stored procedures in a replicate database, you must grant all object access permissions on each replicate object to the ASE Replicator system user (or Maintenance User).

- In each replicate database, grant all object access permissions to the ASE Replicator system user (or Maintenance User) on each replicate table:

  ```
  grant all on table_name to rep_user
  ```

  where *table_name* is the name of a table in the replicate database, and *rep_user* is the user name of the ASE Replicator system user (or Maintenance User).

- In each replicate database, grant all object access permissions to the ASE Replicator system user (or Maintenance User) on each replicate stored procedure:

  ```
  grant all on proc_name to rep_user
  ```

  where *proc_name* is the name of a stored procedure in the replicate database, and *rep_user* is the user name of the ASE Replicator system user (or Maintenance User).

**Note** You must perform steps 3 through 6 for each replicate database.

## Initializing the ASE Replicator process

The final task in setting up ASE Replicator is initializing the ASE Replicator process by:

- Running the aserep script the first time
- Setting up the sp_helpddb system procedure

You must have a System Administrator user role in the primary Adaptive Server to perform these procedures.

---

**Note** Before you begin this task, you must complete all of the setup tasks described in both of these sections:

- "Configuring the primary Adaptive Server" on page 16
- "Configuring the replicate servers and databases" on page 22

---

### Running the aserep script the first time

Use aserep to start and initialize ASE Replicator. When you run this script the first time, with a unique set of parameters, it:

- Creates the ASE Replicator instance subdirectories in the directory where ASE Replicator is installed
- Creates ASE Replicator system tables and procedures in the Distribution Database
- Creates a RUN script that you can use as a shortcut to start ASE Replicator
- Creates the sp_helpddb.sql script
- Starts the ASE Replicator process on the primary Adaptive Server host

The aserep script is provided as a shell script (*.sh*) for UNIX operating systems and a batch file (*.bat*) for Windows NT and Windows 2000 operating systems.

---

**Note** On UNIX platforms, *aserep.sh* checks for the *$SYBASE* environment variable. If the *$SYBASE* environment variable is not set, *aserep.sh* assumes it is executing in the *$SYBASE/RPL-12_5/bin* directory and it sets the *$SYBASE* variable to *'pwd'/../..* so that it can set the other paths it needs to define.

---

Command line parameters that you provide the first time you run aserep define the configuration of the ASE Replicator process.

❖ **To run the aserep script the first time**

1 Log in to the operating system on the primary Adaptive Server host.

2 Make sure that the *$SYBASE* environment variable is defined.

If it is not, set the current directory to the Sybase installation directory, and source *SYBASE.csh* or *SYBASE.sh* (UNIX), or execute *SYBASE.bat* (Windows NT or Windows 2000).

3 Set the current directory to the *$SYBASE/RPL-12_5/bin* directory:

```
cd $SYBASE/RPL-12_5/bin
```

4 Run the aserep script and specify all of the following command line parameters:

- -m *ASE_host*
  where *ASE_host* is the name of the host machine where Adaptive Server is installed.
- -a *ASE_port*
  where *ASE_port* is the port number used to connect to the Adaptive Server.
- -s *my_ASERep*
  where *my_ASERep* is the server name of the ASE Replicator process. This name must be unique on the ASE Replicator host machine.
- -r *my_ASERep_port*
  where *my_ASERep_port* is the port number ASE Replicator uses to listen for incoming connections. This port number must be unique on the ASE Replicator host machine.
- -d *DDB_name*
  where *DDB_name* is the name of the Distribution Database you created.
- -u *rep_user*
  where *rep_user* is the ASE Replicator system user login you created.
- -p *passwd*
  where *passwd* is the password for the ASE Replicator system user login.

After you run the aserep script, the ASE Replicator process starts and displays the Sybase copyright and disclosure statements in the operating system window.

If the process starts successfully, the operating system prompt does not return in that window until you shut down the ASE Replicator process.

If an error message appears shortly after the copyright and disclosure statements, and the operating system prompt returns, then the ASE Replicator process failed to start successfully.

---

**Note** If the ASE Replicator process does *not* start successfully after you run the aserep script, verify that:

- You entered all of the command line parameters correctly, and
- You completed all of the other setup procedures in this chapter.

---

## Setting up the sp_helpddb system procedure

When you run aserep the first time, it creates another script file named *sp_helpddb.sql* in the *$SYBASE/RPL-12_5/my_ASERep/scripts* directory, where *my_ASERep* is the server name of the ASE Replicator process that you specified on the aserep command line.

When executed, the *sp_helpddb.sql* script creates a system procedure named sp_helpddb in the sybsystemprocs database. sp_helpddb returns the name of the Distribution Database. Sybase Central needs that procedure to manage ASE Replicator.

❖ **To set up the sp_helpddb system procedure**

1 Log in to the primary Adaptive Server with a System Administrator user role.

2 Run the sp_helpddb.sql script:

```
isql -SASE_server -Usa -Ppwd <sp_helpddb.sql
```

where *ASE_server* is the server name of the primary Adaptive Server, and *pwd* is the sa user password.

3 Add the ASE Replicator system user to the sybsystemprocs database, and grant the ASE Replicator system user permission to execute the sp_helpddb procedure:

```
use sybsystemprocs
sp_adduser rep_user
grant execute on sp_helpddb to rep_user
```

where *rep_user* is the user name of the ASE Replicator system user.

After you complete the procedure to initialize the ASE Replicator process, ASE Replicator is up and running, and the primary Adaptive Server is configured to work with ASE Replicator.

# Starting ASE Replicator

If the ASE Replicator process is shut down, you must execute a script from the operating system prompt on the Adaptive Server host machine to start ASE Replicator. There are two scripts you can use to start ASE Replicator:

- aserep – the setup and start-up script provided with ASE Replicator.
- RUN_*my_ASERep* – where *my_ASERep* is the ASE Replicator instance name you specified when you ran the aserep script to set up and initialize ASE Replicator.

The RUN_*my_ASERep* script is created by aserep when you set up and initialize the ASE Replicator. See "Running the aserep script the first time" on page 26 for more information.

You may encounter one of the following minor problems when you start ASE Replicator:

- ASE Replicator listener not started
- Orphaned connection in CIS cache

Neither of these problems requires a corrective action.

### ASE Replicator listener not started

When you start the ASE Replicator process, it may take a brief period of time for the process to begin listening for incoming commands. If you invoke an ASE Replicator procedure before the process starts listening for commands, Adaptive Server returns a CIS connection error. In that event, all you need to do is wait a few seconds for the ASE Replicator listener to start, then invoke the command again.

There are two ways to determine when the ASE Replicator process is ready to receive commands:

- Continue executing command procedures until one returns success.
- Monitor the *system.log* file at start-up and look for the following message:

```
Enabling the listener on the maintenance port:
<my_aserep>, <portnum>
```

Orphaned connection in CIS cache

CIS caches database connections within a client session, and it does not refresh the cache when a client session ends. When the ASE Replicator process shuts down, the CIS database connections are orphaned because they are associated with a defunct client session.

The first time you invoke an ASE Replicator procedure after stopping and restarting, CIS returns an error and clears the orphaned connection from its cache. All you need to do is invoke the ASE Replicator procedure again. At that time, CIS creates a new connection, and the procedure executes normally.

## Using the aserep script

When you run aserep and specify a new instance name, it sets up and initializes the ASE Replicator process, as described in "Running the aserep script the first time" on page 26.

The aserep script is provided as a shell script (*.sh*) for UNIX operating systems and a batch file (*.bat*) for Windows NT and Windows 2000 operating systems.

Syntax

aserep -m *host* -a *ase_port* -d *ddb_name* -u *rep_user* -p *pwd* -r *rep_port*
[ -c *char_set* ] [ -s *server* ] [ -admin ] [ -trace ] [-v] [-h]

Parameters

-m *host*
The name of the host machine on which the primary Adaptive Server resides.

-a *ase_port*
The Adaptive Server client socket port number to which ASE Replicator will connect.

-d *ddb_name*
The name of the Distribution Database.

-u *rep_user*
User login name of the ASE Replicator system user.

-p *pwd*
Password for the ASE Replicator system user login name.

-r *rep_port*
The ASE Replicator client socket port number.

-c *char_set*
The character set to use when ASE Replicator connects to the Adaptive Server. This parameter is optional. If not specified, ASE Replicator uses the Adaptive Server's default character set.

-s *server*
The server (instance) name of the ASE Replicator process. This parameter is optional. If not specified, the default instance name aserep is used. If an existing instance has a name other than the default, you must specify the instance name when you invoke aserep to start that ASE Replicator instance.

---

**Note** If you specify a new instance name, aserep creates a new ASE Replicator instance with the name you specify, and configures it as you specify with the other parameters.

---

-admin
Starts the ASE Replicator instance with all database connections and subscriptions suspended. This parameter is optional.

-trace
Starts ASE Replicator with most trace flags enabled. This parameter is optional.

-v
Returns ASE Replicator software version information. This parameter is optional.

-h
Returns a command usage message. This parameter is optional.

Example

```
aserep -m boulder -a 4100 -d DDB_boulder -u Bob
-p p3g5s -r 10001
```

This command starts up the ASE Replicator on the host machine named boulder, with Adaptive Server client socket port number 4100, with the Distribution Database named DDB_boulder, with ASE Replicator system user name Bob, with password p3g5s, with ASE Replicator client socket port number 10001, and with the default character set and the default ASE Replicator instance name.

Usage

- Each time you invoke aserep with a new instance name, it sets up and initializes a new instance of the ASE Replicator process.
- When you invoke the aserep script to start an existing ASE Replicator instance, you must specify all the required command line parameters, and the instance name if it is not the default name.

**Note** If you do not specify *all* of the required parameters when you invoke the aserep script, the ASE Replicator process may start up and shut down immediately with an error.

## Using the RUN script

When you run the aserep script and specify a new instance name, it sets up and initializes the ASE Replicator process, and creates a RUN script that you can use as a shortcut to start ASE Replicator. The RUN script invokes aserep with all the required parameters (except -u and -p) to start ASE Replicator.

**Note** After you set up and initialize ASE Replicator with the aserep script, Sybase recommends that you use the RUN script thereafter to start ASE Replicator.

You must specify the ASE Replicator system user name and password when you invoke the RUN script. You can specify the following optional aserep parameters when you invoke the RUN script:

- -admin
- -trace
- -v

The RUN script is named RUN_*my_ASERep*, where *my_ASERep* is the instance name of the ASE Replicator process you specified when you ran aserep to create the instance.

The RUN script is located in the *$SYBASE/RPL-12_5/my_ASERep* instance directory.

❖ **To start ASE Replicator with the RUN script**

1 Log in to the operating system of the Adaptive Server host machine.

2 Set the current directory to the ASE Replicator instance directory:

```
cd $SYBASE/RPL-12_5/my_ASERep
```

where *my_ASERep* is the instance name of the ASE Replicator.

3 Execute the RUN script on the operating system command line:

```
RUN_my_ASERep -urep_user -ppwd
```

where *my_ASERep* is the instance name of the ASE Replicator process, *rep_user* is the ASE Replicator system user login, and *pwd* is the password for the ASE Replicator system user login.

# Setting up a replication system

Setting up a replication system with ASE Replicator involves the following tasks:

1 Create database connections to identify all of the primary and replicate databases.

2 Create publications for the primary database connections.

3 Create primary articles in the publications to identify the tables and stored procedures in each primary database that you want to publish.

4 Create subscriptions for the replicate database connections.

5 Create replicate articles in the subscriptions to identify the published articles (primary articles) that you want the replicate database objects (tables and stored procedures) to subscribe to.

6 Materialize or validate each replicate article to synchronize the replicate database object with the primary database object.

7 Resume all database connections and subscriptions to start replication.

All of these tasks, and other ASE Replicator administrative tasks, are described in Chapter 3, “Administering ASE Replicator.”

Details of the ASE Replicator command procedures are described in Chapter 4, “ASE Replicator Procedures.”

---

**Note** Before you can set up a replication system with ASE Replicator, you must complete all of the following procedures:

- “Setting up ASE Replicator” on page 15
- “Configuring the replicate servers and databases” on page 22
- “Starting ASE Replicator” on page 29

---

CHAPTER 3

# Administering ASE Replicator

This chapter describes the tasks and procedures you use to administer ASE Replicator and the replication system. This chapter also provides a list of ASE Replicator configuration parameters and describes each parameter in detail.


| Topic | Page |
|---|---|
| Administering the ASE Replicator process | 36 |
| Monitoring ASE Replicator | 40 |
| Managing database connections | 41 |
| Managing publications and subscriptions | 49 |
| Managing primary and replicate articles | 54 |
| ASE Replicator configuration parameters | 69 |


## Using Sybase Central

You can accomplish many of the ASE Replicator administration tasks described in this chapter using Sybase Central Java Edition, a graphical user interface (GUI) system administration tool that comes with Adaptive Server.

Some of the tasks you can perform with Sybase Central are:

- Creating primary and replicate database connections
- Creating and managing publications and subscriptions
- Creating primary and replicate articles
- Suspending and resuming connections and subscriptions
- Suspending and shutting down the ASE Replicator process
- Monitoring ASE Replicator system activity and performance

Sybase Central provides wizards that guide you through creating ASE Replicator objects, including primary and replicate database connections, publications and subscriptions, and primary and replicate articles.

In the Sybase Central window, ASE Replicator appears as a folder under the Adaptive Server icon in the left pane. ASE Replicator objects appear as icons in the ASE Replicator folder.

Refer to the Sybase Central online help for more information about using Sybase Central to administer ASE Replicator.

### Using a query tool

You can perform all ASE Replicator administration and maintenance tasks with a SQL query tool, such as isql.

To invoke ASE Replicator command procedures, you must log in to the primary Adaptive Server with a user name that has permissions for ASE Replicator. Usually, this is the ASE Replicator system user login that you specified when you set up the ASE Replicator process.

**Note** You can execute ASE Replicator command procedures only in the Distribution Database. Therefore, when you log in to Adaptive Server to administer ASE Replicator, you must either open the Distribution Database with the use command, or qualify each ASE Replicator procedure name with the Distribution Database name.

# Administering the ASE Replicator process

Table 3-1 lists the ASE Replicator command procedures that you can use to administer the ASE Replicator process.

***Table 3-1: ASE Replicator administration procedures***

| Procedure | Description |
|---|---|
| sp_configrep | Views or changes ASE Replicator configuration |
| sp_resumerep | Resumes all ASE Replicator operations and objects that are suspended |
| sp_shutdownrep | Shuts down the ASE Replicator process |
| sp_suspendrep | Suspends replication by shutting down all ASE Replicator operations and suspending objects, but does not shut down the ASE Replicator process |

**Note** See “Starting ASE Replicator” on page 29 for information about starting the ASE Replicator process.

## Configuring ASE Replicator

Configuration parameters allow you to adjust or "tune" the performance of ASE Replicator. See "ASE Replicator configuration parameters" on page 69 for information about specific parameters.

Some configuration parameters (called connection configuration parameters) affect the behavior of ASE Replicator database connections. See "Configuring a database connection" on page 43 for more information.

You can use sp_configrep to:

- Change the value of a configuration parameter
- Find the current value of a configuration parameter
- Get information about configuration parameters

❖ **To change the value of an ASE Replicator configuration parameter**

- Use sp_configrep with the name of the configuration parameter, and the value you want to set:

  ```
  sp_configrep param, "value"
  ```

  where *param* is the name of the ASE Replicator configuration parameter, and *value* is the value you want to set.

  **Note** Numeric values are treated as strings, and they must be enclosed in quotes.

❖ **To find the current value of an ASE Replicator configuration parameter**

- Use sp_configrep with the name of the configuration parameter:

  ```
  sp_configrep param
  ```

  where *param* is the name of the ASE Replicator configuration parameter you want to find the value of.

❖ **To get information about all ASE Replicator configuration parameters**

- Use sp_configrep with no parameter:

  ```
  sp_configrep
  ```

## Shutting down the ASE Replicator process

You can terminate the ASE Replicator process using sp_shutdownrep. The sp_shutdownrep command procedure offers two options:

- Graceful shutdown
- Immediate shutdown

In a graceful shutdown, the ASE Replicator components first complete their work on current transactions in the primary database, then empty the Distribution Database queue before terminating. After all ASE Replicator components have terminated, the ASE Replicator process terminates. A graceful shutdown can take a while to complete, depending on how many transaction operations are in the Distribution Database queue.

In an immediate shutdown, all ASE Replicator components terminate immediately, regardless of their current operations or condition, and the ASE Replicator process terminates. An immediate shutdown occurs almost instantly.

❖ **To shut down the ASE Replicator process gracefully**

- Use sp_shutdownrep with no parameter:

```
sp_shutdownrep
```

❖ **To shut down the ASE Replicator process immediately**

- Use sp_shutdownrep with the immediate keyword:

```
sp_shutdownrep immediate
```

## Suspending replication

Suspending replication allows you to stop replication system operation for maintenance or troubleshooting, without terminating the ASE Replicator process.

When you suspend replication, the ASE Replicator Publisher and Distributor components complete their work on current transactions in the primary database, then empty the Distribution Database stable queue before terminating.

**Note** Suspending replication also suspends all database connections and all subscriptions.

❖ **To suspend replication**

- Use sp_suspendrep:

```
sp_suspendrep
```

Suspending replication may take a while to complete, usually a few seconds.

After you suspend replication, the ASE Replicator process component is essentially in an "admin" state, in which no replication takes place, but you can execute ASE Replicator procedures to perform administrative tasks.

If a maintenance procedure affects only one database object, or one database, you can suspend just the individual subscription or database connection involved. See "Suspending and resuming subscriptions" on page 51 or "Suspending and resuming database connections" on page 44 for more information.

## Resuming replication

After replication is suspended, you must use sp_resumerep to restart replication system operation.

When you resume replication:

- The ASE Replicator process resumes all database connections and subscriptions.
- The ASE Replicator Publisher component begins scanning the primary database transaction log and recording transactions in the Distribution Database.
- The ASE Replicator Distributor component begins distributing transactions to the subscribing replicate databases.

❖ **To resume replication when it is suspended**

- Use sp_resumerep:

```
sp_resumerep
```

# Monitoring ASE Replicator

Table 3-2 lists the ASE Replicator command procedures that you can use to monitor and get information about ASE Replicator components and objects.

***Table 3-2: ASE Replicator help procedures***

| Procedure | Description |
|---|---|
| sp_helpconn | Returns information about database connections |
| sp_helplastcommit | Returns the timestamp and locator value of the most recent transaction committed in the replicate database |
| sp_helplocator | Returns fields in the specified locator string |
| sp_helpprimaryart | Returns information about primary articles |
| sp_helpprimaryconn | Returns information about primary database connections |
| sp_helppub | Returns information about publications |
| sp_helprep | Returns statistics or status information for the replication system, subscriptions, and database connections |
| sp_helpreplicateart | Returns information about replicate articles |
| sp_helpreplicateconn | Returns information about replicate database connections |
| sp_helpsub | Returns information about subscriptions |

For information about database connections, see “Getting information about database connections” on page 46 and “Getting connection configuration information” on page 48.

For information about publications and subscriptions, see “Getting information about publications and subscriptions” on page 52.

For information about articles, see “Getting information about primary articles” on page 60 and “Getting information about replicate articles” on page 65.

# Managing database connections

Database connections are identified by the form *ds.db*, where:

- *ds* is the name of the data server on which the database resides.
- *db* is the name of the database.

Table 3-3 lists the ASE Replicator command procedures that you can use to manage database connections.

***Table 3-3: ASE Replicator connection management procedures***

| Procedure | Description |
|---|---|
| sp_addprimaryconn | Defines a new primary database connection |
| sp_addreplicateconn | Defines a new replicate database connection |
| sp_configprimaryconn | Sets or returns information about primary connection configuration parameters |
| sp_configreplicateconn | Sets or returns information about replicate connection configuration parameters |
| sp_dropprimaryconn | Deletes an existing primary database connection |
| sp_dropreplicateconn | Deletes an existing replicate database connection |
| sp_helpconn | Returns information about database connections |
| sp_helpprimaryconn | Returns information about primary database connections |
| sp_helprep | Returns statistics or status information for the replication system, connections, and subscriptions |
| sp_helpreplicateconn | Returns information about replicate database connections |
| sp_resumeprimaryconn | Resumes suspended primary database connections |
| sp_resumereplicateconn | Resumes suspended replicate database connections |
| sp_suspendprimaryconn | Suspends primary database connections |
| sp_suspendreplicateconn | Suspends replicate database connections |

## Creating a database connection

You create primary database connections and replicate database connections separately for each database. If a database will serve as both a primary database and a replicate database in bidirectional replication, you must create both primary and replicate database connections to that database.

You must create database connections before you create any other ASE Replicator objects, such as publications, subscriptions, and articles.

## Creating a primary database connection

If you do not specify a Maintenance User name when you create a primary connection, the ASE Replicator system user is the Maintenance User for the primary connection.

❖ **To create a primary database connection with the default Maintenance User**

- Use sp_addprimaryconn with the following syntax:

  ```
  sp_addprimaryconn "conn_name"
  ```

  where *conn_name* is the connection name in the form *ds.db*.

You have the option to specify a different Maintenance User name when you create a primary connection (for example, if the primary database will also act as a replicate database in bidirectional replication).

❖ **To specify a different Maintenance User name for a primary connection**

- Use sp_addprimaryconn with the following syntax:

  ```
  sp_addprimaryconn "conn_name", maint_user
  ```

  where *conn_name* is the connection name in the form *ds.db*, and *maint_user* is the Maintenance User name for the primary database.

## Creating a replicate database connection

If you do not specify a Maintenance User name when you create a replicate connection, the ASE Replicator system user is the Maintenance User for the replicate connection.

❖ **To create a replicate database connection with the default Maintenance User**

- Use sp_addreplicateconn with the following syntax:

  ```
  sp_addreplicateconn "conn_name"
  ```

  where *conn_name* is the connection name in the form *ds.db*.

You have the option to specify a different Maintenance User name when you create a replicate connection.

❖ **To specify a different Maintenance User name for a replicate connection**

- Use sp_addreplicateconn with the following syntax:

  ```
  sp_addreplicateconn "conn_name", maint_user,
  maint_pw
  ```

where *conn_name* is the connection name in the form *ds.db*, *maint_user* is the Maintenance User name for the replicate database, and *maint_pw* is the password for the Maintenance User.

**Note** If you specify a Maintenance User name that does not already exist as an external login on the primary Adaptive Server, ASE Replicator creates an external login for the Maintenance User name you specify.

## Deleting a database connection

Before you can delete an ASE Replicator database connection, you must delete all publications or subscriptions associated with that database connection.

You delete primary database connections and replicate database connections separately for each database.

❖ **To delete a primary database connection**

- Use sp_dropprimaryconn with the following syntax:

  ```
  sp_dropprimaryconn "conn_name"
  ```

  where *conn_name* is the connection name in the form *ds.db*.

❖ **To delete a replicate database connection**

- Use sp_dropreplicateconn with the following syntax:

  ```
  sp_dropreplicateconn "conn_name"
  ```

  where *conn_name* is the connection name in the form *ds.db*.

## Configuring a database connection

You can set or change certain configuration parameters for each database connection.

For more information about connection configuration parameters for primary and replicate databases, see “Connection configuration parameters” on page 78.

❖ **To configure a primary database connection**

- Use sp_configprimaryconn with the following syntax:

  ```
  sp_configprimaryconn "conn_name", param, value
  ```

  where *conn_name* is the connection name in the form *ds.db*, *param* is the name of the configuration parameter, and *value* is the value of the configuration parameter.

❖ **To configure a replicate database connection**

- Use sp_configreplicateconn with the following syntax:

  ```
  sp_configreplicateconn "conn_name", param, value
  ```

  where *conn_name* is the connection name in the form *ds.db*, *param* is the name of the configuration parameter, and *value* is the value of the configuration parameter.

## Suspending and resuming database connections

You can suspend and resume database connections for maintenance or troubleshooting. For example, you must suspend a database connection before you can perform the following ASE Replicator maintenance tasks:

- Adding or deleting publications or subscriptions
- Adding or deleting primary articles

Suspending a primary database connection stops published transactions from being sent to the Distribution Database stable queue. Suspending a replicate database connection stops queued transactions in the Distribution Database from being sent to the replicate database.

### Suspending database connections

You can suspend either a specific (primary or replicate) database connection, or all primary or all replicate database connections.

---

**Note** When you suspend a replicate database connection, ASE Replicator suspends all of the subscriptions in that connection.

---

❖ **To suspend a specific primary database connection**

- Use sp_suspendprimaryconn with the following syntax to specify a database connection:

  ```
  sp_suspendprimaryconn "conn_name"
  ```

  where *conn_name* is the connection name in the form *ds.db*.

❖ **To suspend *all* primary database connections**

- Use sp_suspendprimaryconn without specifying a database connection:

  ```
  sp_suspendprimaryconn
  ```

❖ **To suspend a specific replicate database connection**

- Use sp_suspendreplicateconn with the following syntax to specify a database connection:

  ```
  sp_suspendreplicateconn "conn_name"
  ```

  where *conn_name* is the connection name in the form *ds.db*.

❖ **To suspend *all* replicate database connections**

- Use sp_suspendreplicateconn without specifying a database connection:

  ```
  sp_suspendreplicateconn
  ```

---

**Note** After a database connection is suspended, you must resume the connection to continue replication.

---

## Resuming database connections

You can resume either a specific (primary or replicate) database connection, or all primary or all replicate database connections.

---

**Note** When you resume a replicate database connection, you also resume all subscriptions in that connection.

---

❖ **To resume a specific primary database connection**

- Use sp_resumeprimaryconn with the following syntax to specify a database connection:

  ```
  sp_resumeprimaryconn "conn_name"
  ```

  where *conn_name* is the connection name in the form *ds.db*.

❖ **To resume *all* primary database connections**

- Use sp_resumeprimaryconn without specifying a database connection:

```
sp_resumeprimaryconn
```

❖ **To resume a specific replicate database connection**

- Use sp_resumereplicateconn with the following syntax to specify a database connection:

```
sp_resumereplicateconn "conn_name"
```

  where *conn_name* is the connection name in the form *ds.db*.

❖ **To resume *all* replicate database connections**

- Use sp_resumereplicateconn without specifying a database connection:

```
sp_resumereplicateconn
```

## Getting information about database connections

ASE Replicator provides two types of information about database connections:

- Metadata and status
- Statistics

The sp_helpprimaryconn and sp_helpreplicateconn procedures return metadata, status, and statistics information about database connections.

### Getting metadata and status information for connections

Connection metadata and status information includes:

- Database server name and database name (*ds.db*)
- Maintenance User name
- Name of the last commit proxy table (replicate connections only)
- Restart locator value
- Connection status and status description

❖ **To get metadata and status information about a specific primary database connection**

- Use sp_helpprimaryconn with the info keyword, and specify a database connection:

  ```
  sp_helpprimaryconn info, "conn_name"
  ```

  where *conn_name* is the connection name in the form *ds.db*.

❖ **To get metadata and status information about *all* primary database connections**

- Use sp_helpprimaryconn without specifying a keyword or database connection:

  ```
  sp_helpprimaryconn
  ```

❖ **To get metadata and status information about a specific replicate database connection**

- Use sp_helpreplicateconn with the info keyword, and specify a database connection:

  ```
  sp_helpreplicateconn info, "conn_name"
  ```

  where *conn_name* is the connection name in the form *ds.db*.

❖ **To get metadata and status information about *all* replicate database connections**

- Use sp_helpreplicateconn without specifying a keyword or database connection:

  ```
  sp_helpreplicateconn
  ```

## Getting statistics information for connections

Connection statistics information includes:

- Statistic timestamp – time the statistic was generated.
- Start timestamp – time the connection was last started.
- Number of operations read since the connection was last started (primary connections only).
- Number of active subscriptions (replicate connections only).

❖ **To get statistics information about a specific primary database connection**

- Use sp_helpprimaryconn with the stats keyword, and specify a database connection:

  ```
  sp_helpprimaryconn stats, "conn_name"
  ```

  where *conn_name* is the connection name in the form *ds.db*.

❖ **To get statistics information about *all* primary database connections**

- Use sp_helpprimaryconn with the stats keyword, without specifying a database connection:

  ```
  sp_helpprimaryconn stats
  ```

❖ **To get statistics information about a specific replicate database connection**

- Use sp_helpreplicateconn with the stats keyword, and specify a database connection:

  ```
  sp_helpreplicateconn stats, "conn_name"
  ```

  where *conn_name* is the connection name in the form *ds.db*.

❖ **To get statistics information about *all* replicate database connections**

- Use sp_helpreplicateconn with the stats keyword, without specifying a database connection:

  ```
  sp_helpreplicateconn stats
  ```

## Getting connection configuration information

The sp_configprimaryconn and sp_configreplicateconn procedures can return information about the configuration of primary and replicate database connections.

See “Connection configuration parameters” on page 78 for information about database connection configuration parameters.

❖ **To get configuration information about a primary database connection**

- Use sp_configprimaryconn with the following syntax to specify a database connection:

  ```
  sp_configprimaryconn "conn_name"
  ```

  where *conn_name* is the connection name in the form *ds.db*.

❖ **To get information about a specific configuration parameter for a primary database connection**

- Use sp_configprimaryconn with the following syntax to specify the database connection and the parameter name:

```
sp_configprimaryconn "conn_name", param
```

where *conn_name* is the connection name in the form *ds.db*, and *param* is the configuration parameter name.

❖ **To get configuration information about a replicate database connection**

- Use sp_configreplicateconn with the following syntax to specify a database connection:

```
sp_configreplicateconn "conn_name"
```

where *conn_name* is the connection name in the form *ds.db*.

❖ **To get information about a specific configuration parameter for a replicate database connection**

- Use sp_configreplicateconn with the following syntax to specify the database connection and the parameter name:

```
sp_configreplicateconn "conn_name", param
```

where *conn_name* is the connection name in the form *ds.db*, and *param* is the configuration parameter name.

# Managing publications and subscriptions

Publications and subscriptions are always defined within the context of a database connection. Each publication or subscription is associated with a specific database connection.

Table 3-4 lists the ASE Replicator procedures you can use to manage publications and subscriptions.

***Table 3-4: ASE Replicator publication and subscription procedures***

| Procedure | Description |
|---|---|
| sp_addpub | Creates a new publication |
| sp_addsub | Creates a new subscription for a publication |
| sp_droppub | Deletes an existing publication |
| sp_dropsub | Deletes an existing subscription |

| Procedure | Description |
|---|---|
| sp_helppub | Returns information about publications |
| sp_helpsub | Returns information about subscriptions |
| sp_helprep | Returns statistics or status information for the replication system, connections, and subscriptions |
| sp_materializesub | Materializes and validates a subscription |
| sp_resumesub | Resumes subscriptions |
| sp_suspendsub | Suspends a specified subscription |
| sp_validatesub | Validates a subscription |

## Creating publications and subscriptions

Publications and subscriptions allow you to organize primary and replicate articles in a database. Before you can create an article, you must first create a publication or subscription.

**Note** You must suspend the database connection before you create a publication or subscription. See "Suspending and resuming database connections" on page 44 for more information.

❖ **To create a publication**

- Use sp_addpub with the following syntax:

  ```
  sp_addpub pub_name, "conn_name"
  ```

  where *pub_name* is the name of the new publication, and *conn_name* is the primary database connection name in the form *ds.db*.

❖ **To create a subscription**

- Use sp_addsub with the following syntax:

  ```
  sp_addsub sub_name, pub_name, "conn_name"
  ```

  where *sub_name* is the name of the new subscription, *pub_name* is the name of the publication that the new subscription subscribes to, and *conn_name* is the replicate database connection name in the form *ds.db*.

Publications are associated with a specific primary database and subscriptions are associated with a specific replicate database. Each database can have more than one publication or subscription associated with it.

## Deleting publications and subscriptions

Before you can delete either a publication or a subscription, you must first delete all the articles in the publication or subscription.

Before you can delete a publication, you must first delete all subscriptions that subscribe to that publication.

---

**Note** You must suspend the database connection before you delete a publication or subscription. See "Suspending and resuming database connections" on page 44 for more information.

---

❖ **To delete a publication**

- Use sp_droppub with the following syntax:

  ```
  sp_droppub pub_name
  ```

  where *pub_name* is the name of the publication.

❖ **To delete a subscription**

- Use sp_dropsub with the following syntax:

  ```
  sp_dropsub sub_name
  ```

  where *sub_name* is the name of the subscription.

## Suspending and resuming subscriptions

You can suspend and resume subscriptions for maintenance or troubleshooting. For example, you must suspend a subscription before you can delete a replicate article in that subscription.

Suspending a subscription stops queued transactions in the transaction log table from being sent to the replicate tables for that subscription.

❖ **To suspend a subscription**

- Use sp_suspendsub with the following syntax:

  ```
  sp_suspendsub sub_name
  ```

  where *sub_name* is the name of the subscription.

**Note** After a subscription is suspended, you must resume the subscription to continue replication.

You can resume all subscriptions or a specified subscription.

❖ **To resume a specified subscription**

- Use sp_resumesub with the following syntax to specify the subscription:

  ```
  sp_resumesub sub_name
  ```

  where *sub_name* is the name of the subscription.

❖ **To resume *all* suspended subscriptions**

- Use sp_resumesub without specifying a subscription:

  ```
  sp_resumesub
  ```

## Getting information about publications and subscriptions

You can get metadata information about publications and subscriptions, and status and statistics information about subscriptions with the following procedures:

- sp_helppub – returns metadata information about publications
- sp_helpsub – returns metadata and status information about subscriptions

### Getting metadata and status information for publications and subscriptions

Metadata and status information about publications and subscriptions includes:

- Database server name and database name (*ds.db*)
- Publication or subscription name
- Publication name for subscription (subscriptions only)
- Subscription status and status description (subscriptions only)

❖ **To get metadata information about a publication**

- Use sp_helppub with the info keyword, and specify the publication name:

  ```
  sp_helppub info, pub_name
  ```

  where *pub_name* is the name of the publication.

❖ **To get metadata information about *all* publications**

- Use sp_helppub without specifying a publication:

```
sp_helppub
```

❖ **To get metadata and status information about a subscription**

- Use sp_helpsub with the info keyword, and specify the subscription name:

```
sp_helpsub info, sub_name
```

where *sub_name* is the name of the subscription.

❖ **To get metadata information about *all* subscriptions**

- Use sp_helpsub without specifying a subscription:

```
sp_helpsub
```

## Getting statistics information for subscriptions

Statistics information for subscriptions includes:

- Statistic timestamp – time the statistic was generated.
- Start timestamp – time the subscription was last started (resumed).
- Number of transactions sucessfully replicated to articles in the subscription.
- Number of transactions for articles in the subscription that were skipped because of errors.
- Average number of operations in transactions replicated.
- Average number of replicate articles involved with each transaction replicated.
- Locator value of the last committed transaction for the subscription.

❖ **To get statistics information for a specific subscription**

- Use sp_helpsub with the stats keyword, and specify the subscription:

```
sp_helpsub stats, sub_name
```

where *sub_name* is the name of the subscription.

❖ **To get statistics information for *all* subscriptions**

- Use sp_helpsub with the stats keyword, without specifying a subscription:

```
sp_helpsub stats
```

# Managing primary and replicate articles

Articles identify the database objects affected by replicated transactions. Primary articles identify the source of replicated transactions, which are objects in the primary database. Replicate articles subscribe to primary articles and identify the destinations of replicated transactions, which are objects in the replicate database.

Table 3-5 lists the ASE Replicator procedures you can use to manage primary and replicate articles.

***Table 3-5: ASE Replicator primary and replicate article procedures***

| Procedure | Description |
|---|---|
| sp_addprimaryart | Creates a new primary article |
| sp_addreplicateart | Creates a new replicate article |
| sp_dropprimaryart | Deletes an existing primary article |
| sp_dropreplicateart | Deletes an existing replicate article |
| sp_helpprimaryart | Returns information about primary articles |
| sp_helppub | Returns information about publications |
| sp_helpreplicateart | Returns information about replicate articles |
| sp_helpsub | Returns information about subscriptions |

## Creating primary articles

Before you can create a primary article, you must create a primary database connection, and create at least one publication in that connection.

Each primary article identifies a primary object (table or stored procedure) that is published for replication. Therefore, the name of a primary article is the name of the primary object it publishes.

When you create a primary article, you have two options for selecting the fields (table columns or stored procedure parameters) to be published in the primary article. You can either publish all the fields, or specify individual fields to be published in the primary article.

---

**Note** You must suspend the database connection before you create a primary article. See “Suspending and resuming database connections” on page 44 for more information.

---

❖ **To create a primary article and publish specified fields**

- Use sp_addprimaryart with the following syntax to specify the publication name, the name of the primary article (primary object), and the numbers of the fields to be published:

  ```
  sp_addprimaryart pub_name, pri_art, "fields"
  ```

  where *pub_name* is the name of the publication, *pri_art* is the name of the primary article and primary object, and *fields* is a numeric list of the fields to be published.

❖ **To create a primary article and publish *all* fields**

- Use sp_addprimaryart with the following syntax to specify only the publication name and the name of the primary article (primary object):

  ```
  sp_addprimaryart pub_name, pri_art
  ```

  where *pub_name* is the name of the publication, and *pri_art* is the name of the primary article and primary object.

After you have created a primary article, you can add the primary article to additional publications with the sp_addprimaryart procedure.

❖ **To add an existing primary article to a publication**

- Use sp_addprimaryart with the following syntax to specify the publication name and the name of the existing primary article:

  ```
  sp_addprimaryart pub_name, pri_art
  ```

  where *pub_name* is the name of the publication, and *pri_art* is the name of the existing primary article.

You can add an existing primary article to as many publications as you want.

## Creating replicate articles

Before you can create a replicate article, you must create a replicate database connection, and create at least one subscription in that connection.

Each replicate article identifies a replicate object (table or stored procedure) that subscribes to a primary article (published primary object). Therefore, the name of a replicate article is the same as the name of the replicate object it identifies. Typically, a primary object and a replicate object have the same name, but they can have different names.

**Note** If the replicate table does not exist in the replicate database before you create a replicate article, ASE Replicator creates the replicate table in the replicate database, using the name of the primary article.

When you create a replicate article, you have two options for selecting the published fields (table columns or stored procedure parameters) to be subscribed to by the replicate article. You can either subscribe to all the published fields, or specify individual published fields to be subscribed to by the replicate article.

**Note** You must suspend the subscription before you create a replicate article. See "Suspending and resuming subscriptions" on page 51 for more information.

❖ **To create a replicate article and subscribe to specified published fields**

- Use sp_addreplicateart with the following syntax to specify the subscription name, the name of the primary article (primary object) that the replicate article subscribes to, the name of the replicate article (replicate object in the replicate database), and the numbers of the published fields (columns or parameters) to be subscribed to:

  ```
  sp_addreplicateart sub_name, pri_art, rep_art,
  "fields"
  ```

  where *sub_name* is the name of the subscription, *pri_art* is the name of the primary article that the replicate article subscribes to, *rep_art* is the name of the replicate article, and *fields* is a numeric list of the published fields to subscribe to.

In addition to selecting from the published fields, you can specify a where clause to select the data to be replicated based on the value or values in the published fields.

❖ **To create a replicate article with a *where* clause**

- Use sp_addreplicateart with the following syntax to specify the subscription name, the name of the primary article (primary object) that the replicate article subscribes to, the name of the replicate article (replicate object in the replicate database), the numbers of the published fields to be subscribed to (optionally), and the where clause:

  ```
  sp_addreplicateart sub_name, pri_art, rep_art,
  "fields", "where_clause"
  ```

where *sub_name* is the name of the subscription, *pri_art* is the name of the primary article that the replicate article subscribes to, *rep_art* is the name of the replicate article, *fields* is a numeric list of the published fields to subscribe to, and *where_clause* is the where clause that selects the data to be replicated from the specified published fields.

❖ **To create a replicate article and subscribe to *all* published fields**

- Use sp_addreplicateart with the following syntax to specify only the subscription name and the name of the primary article (primary object) that the replicate article subscribes to:

  ```
  sp_addreplicateart sub_name, pri_art
  ```

  where *sub_name* is the name of the subscription, and *pri_art* is the name of the primary article that the replicate article subscribes to.

If the name of the replicate object is not the same as the name of the primary object, then the name of the replicate article cannot be the same as the name of the primary article.

The sp_addreplicateart procedure allows you to specify a different name for the replicate article (and replicate object) when you create a replicate article.

❖ **To create a replicate article with a different name from the primary article**

- Use sp_addreplicateart with the following syntax to specify the subscription name, the name of the primary article (primary object) that the replicate article subscribes to, and the name of the replicate article (replicate object):

  ```
  sp_addreplicateart sub_name, pri_art, rep_art
  ```

  where *sub_name* is the name of the subscription, *pri_art* is the name of the primary article that the replicate article subscribes to, and *rep_art* is the name of the replicate article.

## Deleting primary articles

Before you can delete a primary article from a publication, you must first:

- Delete all replicate articles that subscribe to that primary article in that publication
- Suspend the primary database connection that contains the publication that the primary article resides in

Because a primary article can reside in more than one publication, you can delete a primary article from one publication without deleting it from other publications. To be removed from the primary database, a primary article must be deleted from all publications it resides in.

**Note** You must suspend the database connection before you delete a primary article. See “Suspending and resuming database connections” on page 44 for more information.

❖ **To delete a primary article**

- Use sp_dropprimaryart with the following syntax to specify the publication name and the name of the primary article:

  ```
  sp_dropprimaryart pub_name, pri_art
  ```

  where *pub_name* is the name of the publication, and *pri_art* is the name of the primary article.

The sp_dropprimaryart procedure gives you the option of deleting all primary articles in a publication.

❖ **To delete *all* primary articles in a publication**

- Use sp_dropprimaryart with the following syntax to specify only the publication name:

  ```
  sp_dropprimaryart pub_name
  ```

  where *pub_name* is the name of the publication you want to delete all primary articles from.

## Deleting replicate articles

Before you can delete a replicate article from a subscription, you must first suspend the subscription. After the replicate article is deleted, you can resume the subscription.

See “Suspending and resuming subscriptions” on page 51 for more information.

**Note** If the replicate table was created by ASE Replicator when the replicate article was created, ASE Replicator deletes the replicate table in the replicate database when you delete the replicate article.

❖ **To delete a replicate article**

- Use sp_dropreplicateart with the following syntax to specify the subscription name, and the name of the replicate article:

  ```
  sp_dropreplicateart sub_name, rep_art
  ```

  where *sub_name* is the name of the subscription, and *rep_art* is the name of the replicate article.

The sp_dropreplicateart procedure gives you the option of deleting all replicate articles in a subscription.

❖ **To delete *all* replicate articles in a subscription**

- Use sp_dropreplicateart with the following syntax to specify only the subscription name:

  ```
  sp_dropreplicateart sub_name
  ```

  where *sub_name* is the name of the subscription you want to delete all replicate articles from.

## Materializing and validating replicate articles

After you create a replicate article in a subscription, you must either materialize or validate the replicate article before you can start replication to the replicate object identified in the replicate article.

---

**Note** You must suspend the subscription before you materialize or validate a replicate article. See "Suspending and resuming subscriptions" on page 51 for more information.

---

Materializing a replicate article for a table copies data from the primary object identified by the primary article to which the replicate article subscribes. Data is copied using the insert into ... select from command, and it is based on the subscribed fields and the where clause specified in the replicate article (if applicable).

❖ **To materialize a specific replicate article in a subscription**

- Use sp_materializesub with the following syntax to specify the subscription and the name of the replicate article:

  ```
  sp_materializesub sub_name, rep_art
  ```

where *sub_name* is the name of the subscription, and *rep_art* is the name of the replicate article.

❖ **To materialize *all* replicate articles in a subscription**

- Use sp_materializesub with the following syntax to specify the subscription:

  ```
  sp_materializesub sub_name
  ```

  where *sub_name* is the name of the subscription.

If the replicate object identified in a replicate article already contains data synchronized with the primary object in the primary database, you can validate the replicate article instead of materializing it.

In the case of a replicate article for a stored procedure, the article needs only to be validated, and not materialized.

❖ **To validate a specific replicate article in a subscription**

- Use sp_validatesub with the following syntax to specify the subscription and the name of the replicate article:

  ```
  sp_validatesub sub_name, rep_art
  ```

  where *sub_name* is the name of the subscription, and *rep_art* is the name of the replicate article.

❖ **To validate all replicate articles in a subscription**

- Use sp_validatesub with the following syntax to specify the subscription:

  ```
  sp_validatesub sub_name
  ```

  where *sub_name* is the name of the subscription.

## Getting information about primary articles

There are several types of information you can get about primary articles:

- Metadata information
- Primary article published field information
- Information about all primary articles in a specified publication or in a specified primary database
- Information about all primary articles with no subscribers in a specified publication or in a specified primary database

- Information about primary articles with no subscribing replicate articles in a specified subscription
- Information about all publications that contain a specified primary article
- Information about all unpublished primary objects in a primary database

The sp_helpprimaryart procedure returns primary article metadata information, primary article field information, and information about which publications contain a specified primary article.

## Getting metadata information for primary articles

Metadata information for primary articles includes:

- Database server name and database name of the primary database
- Publication name
- Owner of the primary object
- Name of the primary object
- Stored procedure group number (stored procedures only)
- Type of the primary object (table or stored procedure)
- Number of replicate articles that subscribe to the primary article
- Shadow table name

❖ **To get metadata information about a specific primary article**

- Use sp_helpprimaryart with the info keyword, and specify the primary article:

  ```
  sp_helpprimaryart info, pri_art
  ```

  where *pri_art* is the name of the primary article.

The sp_helpprimaryart procedure allows you to qualify a primary article by publication or by primary database.

❖ **To get metadata information about a specific primary article in a specific publication**

- Use sp_helpprimaryart with the info keyword, and specify a primary article name and a publication:

  ```
  sp_helpprimaryart info, pri_art, pub=pub_name
  ```

where *pri_art* is the name of the primary article, and *pub_name* is the name of a publication.

❖ **To get metadata information about a specific primary article in a specific primary database**

- Use sp_helpprimaryart with the info keyword, and specify a primary article name and a primary database connection:

  ```
  sp_helpprimaryart info, pri_art, conn="conn_name"
  ```

  where *pri_art* is the name of the primary article, and *conn_name* is the name of a primary database connection.

❖ **To get metadata information about *all* primary articles**

- Use sp_helpprimaryart with no keyword:

  ```
  sp_helpprimaryart
  ```

## Getting information about published fields in primary articles

Information about published fields in primary articles includes:

- Database server name and database name of the primary database
- Owner of the primary object
- Name of the primary object
- Stored procedure group number (stored procedures only)
- Type of the primary object (table or stored procedure)
- Field identifier (ordinal position)
- Field name (column name or parameter name)
- Datatype of the field
- Precision of the datatype (precision of numeric datatypes, or length of string or binary datatypes)
- Scale of the datatype (numeric datatypes only)

You must qualify a primary article by publication or by primary database when you request published field information.

❖ **To get information about published fields in a specific primary article in a specific publication**

- Use sp_helpprimaryart with the fields keyword, and specify a primary article name and a publication:

  ```
  sp_helpprimaryart fields, pri_art, pub=pub_name
  ```

  where *pri_art* is the name of the primary article, and *pub_name* is the name of a publication.

❖ **To get information about published fields in a specific primary article in a specific primary database**

- Use sp_helpprimaryart with the fields keyword, and specify a primary article name and a primary database connection:

  ```
  sp_helpprimaryart fields, pri_art, conn="conn_name"
  ```

  where *pri_art* is the name of the primary article, and *conn_name* is the name of a primary database connection.

## Getting information about primary articles in publications

You can get the following information about primary articles in publications:

- All publications in all primary databases that contain a primary article with a specified name
- All publications in a specific primary database that contain a primary article with a specified name
- All primary articles in a specified publication or in a specified primary database

The sp_helpprimaryart procedure returns information about publications that contain a specified primary article.

❖ **To get information about publications that contain a specific primary article**

- Use sp_helpprimaryart with the pubs keyword, and specify a primary article:

  ```
  sp_helpprimaryart pubs, pri_art
  ```

  where *pri_art* is the name of the primary article.

❖ **To get information about publications in a specific primary database that contain a specific primary article**

- Use sp_helpprimaryart with the pubs keyword, and specify a primary article name and a primary database connection:

  ```
  sp_helpprimaryart pubs, pri_art, conn="conn_name"
  ```

  where *pri_art* is the name of the primary article, and *conn_name* is the name of a primary database connection.

The sp_helppub procedure returns information about primary articles in a specified publication.

❖ **To get information about *all* primary articles in a specified publication**

- Use sp_helppub with the arts keyword, and specify a publication:

  ```
  sp_helppub arts, pub_name
  ```

  where *pub_name* is the name of a publication.

The sp_helpprimaryconn procedure returns information about primary articles and unpublished primary objects in a primary database.

❖ **To get information about *all* primary articles in *all* publications in a specific primary database**

- Use sp_helpprimaryconn with the arts keyword, and specify a primary database connection:

  ```
  sp_helpprimaryconn arts, conn="conn_name"
  ```

  where *conn_name* is the name of a primary database connection.

## Getting information about unpublished objects

You can use the sp_helpprimaryconn procedure to get a list of all unpublished objects in a primary database. Unpublished objects in a primary database are objects for which no primary articles exist.

❖ **To get information about *all* unpublished primary objects in a specific primary database**

- Use sp_helpprimaryconn with the unpub keyword, and specify a primary database connection:

  ```
  sp_helpprimaryconn unpub, conn="conn_name"
  ```

  where *conn_name* is the name of a primary database connection.

### Getting information about primary articles with no subscribers

There are three commands that return information about primary articles for which there are no subscribers:

- sp_helpprimaryconn – Returns information about primary articles in a specified primary database.
- sp_helppub – Returns information about primary articles in a specified publication.
- sp_helpsub – Returns information about primary articles with no subscribing replicate articles in a specified subscription.

❖ **To get information about *all* primary articles in a specific primary database for which there are no subscribers**

- Use sp_helpprimaryconn with the unsub keyword, and specify a primary database connection:

  ```
  sp_helpprimaryconn unsub, conn="conn_name"
  ```

  where *conn_name* is the name of a primary database connection.

❖ **To get information about *all* primary articles in a specific publication for which there are no subscribers**

- Use sp_helppub with the unsub keyword, and specify a publication:

  ```
  sp_helppub unsub, pub_name
  ```

  where *pub_name* is the name of a publication.

❖ **To get information about primary articles with no subscribing replicate articles in a specific subscription**

- Use sp_helpsub with the unsub keyword, and specify a subscription:

  ```
  sp_helpsub unsub, sub_name
  ```

  where *sub_name* is the name of a subscription.

## Getting information about replicate articles

There are several types of information you can get about replicate articles:

- Metadata information
- Replicate article field information

- Information about all replicate articles in a specified subscription or in a specified replicate database

The sp_helpreplicateart procedure returns replicate article metadata information, replicate article field information, and information about which subscriptions contain a specified replicate article.

## Getting replicate article metadata information

Metadata information for replicate articles includes:

- Database server name and database name of the replicate database
- Subscription name
- Owner of the replicate object
- Name of the replicate object
- Stored procedure group number (stored procedures only)
- Type of the replicate object (table or stored procedure)
- Proxy table name
- Distribution procedure name
- Validation status
- Where clause (if specified when the replicate article was created)
- Publication name (identified in the subscription)
- Owner of the primary article that the replicate article subscribes to
- Name of the primary article that the replicate article subscribes to
- Primary stored procedure group number (stored procedures only)

❖ **To get metadata information about a specific replicate article**

- Use sp_helpreplicateart with the info keyword, and specify a replicate article:

  ```
  sp_helpreplicateart info, rep_art
  ```

  where *rep_art* is the name of the replicate article.

The sp_helpreplicateart procedure allows you to qualify a replicate article by subscription or by replicate database.

❖ **To get metadata information about a specific replicate article in a specific subscription**

- Use sp_helpreplicateart with the info keyword, and specify a replicate article name and a subscription:

```
sp_helpreplicateart info, rep_art, sub=sub_name
```

where *rep_art* is the name of the replicate article, and *sub_name* is the name of a subscription.

❖ **To get metadata information about a specific replicate article in a specific replicate database**

- Use sp_helpreplicateart with the info keyword, and specify a replicate article name and a replicate database connection:

```
sp_helpreplicateart info, rep_art, conn="conn_name"
```

where *rep_art* is the name of the replicate article, and *conn_name* is the name of a replicate database connection.

❖ **To get metadata information about *all* replicate articles**

- Use sp_helpreplicateart with no keyword, and without specifying a replicate article:

```
sp_helpreplicateart
```

## Getting information about fields in replicate articles

Information about fields in replicate articles includes:

- Database server name and database name of the replicate database
- Subscription name
- Owner of the replicate object
- Name of the replicate object
- Replicate stored procedure group number (stored procedures only)
- Type of the replicate object (table or stored procedure)
- Replicate field identifier (ordinal position)
- Replicate field name (column name or parameter name)
- Datatype of the field
- Precision of the datatype (precision of numeric datatypes, or length of string or binary datatypes)

- Scale of the datatype (numeric datatypes only)
- Publication name
- Owner of the primary article
- Name of the primary article
- Primary stored procedure group number (stored procedures only)
- Primary field identifier (ordinal position)
- Primary field name (column name or parameter name)

You must qualify a replicate article by subscription or by replicate database when you request field information.

❖ **To get information about fields in a specific replicate article in a specific subscription**

- Use sp_helpreplicateart with the fields keyword, and specify a replicate article name and a subscription:

  ```
  sp_helpreplicateart fields, rep_art, sub=sub_name
  ```

  where *rep_art* is the name of the replicate article, and *sub_name* is the name of a subscription.

❖ **To get information about fields in a specific replicate article in a specific replicate database**

- Use sp_helpreplicateart with the fields keyword, and specify a replicate article name and a replicate database connection:

  ```
  sp_helpreplicateart fields, rep_art,
  conn="conn_name"
  ```

  where *rep_art* is the name of the replicate article, and *conn_name* is the name of a replicate database connection.

## Getting information about replicate articles in subscriptions

You can get the following information about replicate articles in subscriptions:

- All replicate articles in all subscriptions in a specified replicate database
- All replicate articles in a specified subscription

The sp_helpreplicateconn procedure returns information about replicate articles in a replicate database.

❖ **To get information about *all* replicate articles in *all* subscriptions in a specific replicate database**

- Use sp_helpreplicateconn with the arts keyword, and specify a replicate database connection:

  ```
  sp_helpreplicateconn arts, conn="conn_name"
  ```

  where *conn_name* is the name of a replicate database connection.

The sp_helpsub procedure returns information about replicate articles in a specified subscription.

❖ **To get information about *all* replicate articles in a specified subscription**

- Use sp_helpsub with the arts keyword, and specify a subscription:

  ```
  sp_helpsub arts, sub_name
  ```

  where *sub_name* is the name of a subscription.

# ASE Replicator configuration parameters

You can configure the function and behavior of ASE Replicator by setting or changing the values of configuration parameters. There are two types of configuration parameters for ASE Replicator:

- General configuration parameters – affect the function and behavior of the ASE Replicator process.
- Connection configuration parameters – affect the function and behavior of an individual ASE Replicator database connection.

## General configuration parameters

This section lists all general configuration parameters for ASE Replicator. To change these configuration parameters, use the sp_configrep procedure.

---

**Note** Some configuration parameters (indicated by an asterisk in Table 3-6) cannot be changed with sp_configrep. These parameters must be specified on the aserep command line when the ASE Replicator process is started.

---

Table 3-6 gives a brief description of each configuration parameter.

***Table 3-6: ASE Replicator configuration parameters***

| Parameter | Description |
|---|---|
| admin_port * | ASE Replicator client socket port number |
| ase_charset * | Adaptive Server default character set |
| ase_host * | Name of the host machine on which Adaptive Server resides |
| ase_port * | Adaptive Server client socket port number |
| batch_size | Number of commands to batch |
| batch_timeout | Timeout limit for command batching |
| ddb_name * | Name of the Distribution Database |
| log_directory | Directory for system log files |
| log_trace_verbose | Enable/disable verbose trace message content |
| log_wrap | Number of 1k blocks before wrapping log files |
| monitor_delay | Status monitor ping interval in seconds |
| queue_size | Maximum number of log operations kept in an internal queue |
| scan_sleep_increment | Number of seconds sleep time increases between empty log scans |
| scan_sleep_max | Maximum number of seconds between log scans |
| stat_trunc_interval | Number of days after which statistics are deleted from repository |
| stat_write_timeout | Frequency statistics are written to repository |
| status_monitoring | Enable/disable status monitoring |
| truncate_numops | Minimum number of replicated operations in stable queue before truncation occurs |

The following sections describe each configuration parameter in detail.

## admin_port

| Summary information | |
|---|---|
| Default value | 10000 |
| Range of values | 1 to 65535 |
| Status | Static |
| Display level | N/A |
| Required role | ASE Replicator system user |

admin_port identifies the client socket port number on which the ASE Replicator process listens for commands.

**Note** admin_port cannot be changed with sp_configrep. You must specify the ASE Replicator client socket port on the aserep command line when the ASE Replicator process is started.

## ase_charset

| Summary information | |
|---|---|
| Default value | |
| Range of values | N/A |
| Status | Static |
| Display level | N/A |
| Required role | ASE Replicator system user |

ase_charset identifies the character set to be used on the ASE Replicator connection to the Adaptive Server.

**Note** ase_charset cannot be changed with sp_configrep. You can specify a character set on the aserep command line when the ASE Replicator process is started.

## ase_host

| Summary information | |
|---|---|
| Default value | |
| Range of values | N/A |
| Status | Static |
| Display level | N/A |
| Required role | ASE Replicator system user |

ase_host identifies the network name of the Adaptive Server host machine.

**Note** ase_host cannot be changed with sp_configrep. You must specify the Adaptive Server host machine name on the aserep command line when the ASE Replicator process is started.

## ase_port

| Summary information | |
|---|---|
| Default value | 1111 |
| Range of values | 1 to 65535 |
| Status | Static |
| Display level | N/A |
| Required role | ASE Replicator system user |

ase_port identifies the client socket port number on which the ASE Replicator process communicates with the Adaptive Server.

**Note** ase_port cannot be changed with sp_configrep. You must specify the Adaptive Server client socket port on the aserep command line when the ASE Replicator process is started.

## batch_size

| Summary information | |
|---|---|
| Default value | 100 |
| Range of values | 0 to 1000 |
| Status | Dynamic |
| Display level | N/A |
| Required role | ASE Replicator system user |

batch_size specifies the maximum number of operations to be sent to the Distribution Database transaction log in one batch.

The Publisher component reads operations in the native Adaptive Server transaction log and puts them in a batch to send to the Distribution Database. You can adjust the value of batch_size to tune performance of the primary (Publisher) side of ASE Replicator.

## batch_timeout

| Summary information | |
|---|---|
| Default value | 5000 |
| Range of values | 0 to 30000 |

| Summary information | |
|---|---|
| Status | Dynamic |
| Display level | N/A |
| Required role | ASE Replicator system user |

batch_timeout specifies the number of milliseconds to wait to collect the batch_size number of commands before sending the batch to the Distribution Database transaction log. This parameter works with batch_size to tune performance.

## ddb_name

| Summary information | |
|---|---|
| Default value | |
| Range of values | N/A |
| Status | Static |
| Display level | N/A |
| Required role | ASE Replicator system user |

ddb_name identifies the name of the Distribution Database.

**Note** ddb_name cannot be changed with sp_configrep. You must specify the Distribution Database name on the aserep command line when the ASE Replicator process is started.

## log_directory

| Summary information | |
|---|---|
| Default value | /software/sybase125/RPL-12_5/*my_ASERep* |
| Range of values | N/A |
| Status | Dynamic |
| Display level | N/A |
| Required role | ASE Replicator system user |

log_directory identifies the directory in which ASE Replicator saves system log files. The value of log_directory is the full path of the log directory.

**Note** The value *my_ASERep* shown in the default value is the name of the ASE Replicator instance you created when you initialized the ASE Replicator process.

## log_trace_verbose

| Summary information | |
|---|---|
| Default value | true |
| Range of values | false, true |
| Status | Dynamic |
| Display level | N/A |
| Required role | ASE Replicator system user |

log_trace_verbose specifies whether verbose trace message content is enabled. If true, ASE Replicator provides additional detailed information that identifies the component generating the trace message.

## log_wrap

| Summary information | |
|---|---|
| Default value | 10000 |
| Range of values | 500 to 2097151 |
| Status | Dynamic |
| Display level | N/A |
| Required role | ASE Replicator system user |

log_wrap specifies the number of 1K blocks written to the ASE Replicator system log file before wrapping.

ASE Replicator maintains one log file and overwrites it each time the log file wraps. A small log_wrap value reduces the disk space used by the log file, but it may keep the log file too small to contain enough history to diagnose a problem.

### monitor_delay

| Summary information | |
|---|---|
| Default value | 60 |
| Range of values | 0 to 10080 |
| Status | Dynamic |
| Display level | N/A |
| Required role | ASE Replicator system user |

monitor_delay specifies the status monitor ping interval used by Sybase Central, in seconds. A value of 0 specifies no status monitoring. A very small non-zero value can adversely affect overall ASE Replicator performance.

This parameter is used only by Sybase Central.

### queue_size

| Summary information | |
|---|---|
| Default value | 1000 |
| Range of values | 1 to 2147483647 |
| Status | Dynamic |
| Display level | N/A |
| Required role | ASE Replicator system user |

queue_size specifies the maximum number of log operations kept in the ASE Replicator internal, in-memory queues. Larger queue_size values allow more data to be stored in memory, potentially improving performance at the expense of more memory usage.

### scan_sleep_increment

| Summary information | |
|---|---|
| Default value | 5 |
| Range of values | 0 to 60 |
| Status | Dynamic |
| Display level | N/A |
| Required role | ASE Replicator system user |

scan_sleep_increment specifies the number of seconds that scan sleep time increases between consecutive empty log scans.

Scan sleep time is the time that the Publisher component waits to start a transaction log scan after a log scan returns no data. Each subsequent time that a log scan returns no data, the scan sleep time increases by the amount specified in scan_sleep_increment, up to the amount of time specified by scan_sleep_max.

When a log scan returns data, the scan sleep time is set to zero.

## scan_sleep_max

| Summary information | |
|---|---|
| Default value | 60 |
| Range of values | 0 to 60 |
| Status | Dymanic |
| Display level | N/A |
| Required role | ASE Replicator system user |

scan_sleep_max specifies the maximum scan sleep time that the Publisher component waits to start a transaction log scan after a log scan returns no data.

## stat_trunc_interval

| Summary information | |
|---|---|
| Default value | 1 |
| Range of values | 0 to 365 |
| Status | Dynamic |
| Display level | N/A |
| Required role | ASE Replicator system user |

stat_trunc_interval specifies the number of days that statistics information is stored in the Distribution Database statistics table before it is deleted. A value of 0 specifies no statistics recording.

## stat_write_timeout

| Summary information | |
|---|---|
| Default value | 0 |
| Range of values | >= 0 |
| Status | Dynamic |

| Summary information | |
|---|---|
| Display level | N/A |
| Required role | ASE Replicator system user |

stat_write_timeout specifies the frequency (in minutes) at which statistics information is written to the Distribution Database statistics table.

### status_monitoring

| Summary information | |
|---|---|
| Default value | false |
| Range of values | false, true |
| Status | Dynamic |
| Display level | N/A |
| Required role | ASE Replicator system user |

status_monitoring turns ASE Replicator status monitoring on and off in Sybase Central. This parameter is used only by Sybase Central.

### truncate_numops

| Summary information | |
|---|---|
| Default value | 1000 |
| Range of values | 0 to 2147483647 |
| Status | Dynamic |
| Display level | N/A |
| Required role | ASE Replicator system user |

truncate_numops specifies the minimum number of replicated operations that must be in the Distribution Database stable queue before truncation occurs. A value of 0 specifies no truncation.

Larger truncate_numops values keep more data in the stable queue, taking up more space, even though the operations have been successfully replicated. If the truncate_numops value is too small, truncation occurs more often and it may cause lock contention with ASE Replicator components that read from and write to the stable queue.

## Connection configuration parameters

This section lists all the connection configuration parameters for ASE Replicator. To change these configuration parameters, use sp_configprimaryconn or sp_configreplicateconn.

Table 3-7 gives a brief description of each ASE Replicator connection configuration parameter.

***Table 3-7: ASE Replicator connection configuration parameters***

| Parameter | Description |
|---|---|
| gen_id | Database generation ID (first two bytes in the connection's locator value) |
| lti_version | Log scan protocol version number |
| mode | Scan mode for the database log |
| numrecs | Maximum number of records returned by each log scan |
| queue_size | Maximum number of log operations kept in an internal queue |
| scan_sleep_increment | Number of seconds sleep time increases between empty log scans |
| scan_sleep_max | Maximum number of seconds between log scans |
| timeout | Number of seconds to block, if the end of the log is reached before the maximum number of records are read |

Primary database connections use all these connection configuration parameters. Replicate database connections use only gen_id and queue_size.

The following sections describe each connection configuration parameter in detail.

### gen_id

| Summary information | |
|---|---|
| Default value | 0 |
| Range of values | 0 to 32767 |
| Status | Dynamic |
| Display level | N/A |
| Required role | ASE Replicator system user |

gen_id stores the database generation ID, which is the first two bytes in the database connection's locator value. This parameter is used for both primary and replicate database connections.

### lti_version

| Summary information | |
|---|---|
| Default value | 400 |
| Range of values | N/A |
| Status | Static |
| Display level | N/A |
| Required role | ASE Replicator system user |

lti_version specifies the log scanning protocol version number. This parameter is used for primary database connections only.

### mode

| Summary information | |
|---|---|
| Default value | block |
| Range of values | block, poll |
| Status | Dynamic |
| Display level | N/A |
| Required role | ASE Replicator system user |

mode specifies the log scanning mode for the primary database log. This parameter is used for primary database connections only.

### numrecs

| Summary information | |
|---|---|
| Default value | 1000 |
| Range of values | 0 to 2147483647 |
| Status | Dynamic |
| Display level | N/A |
| Required role | ASE Replicator system user |

numrecs specifies the maximum number of log records returned in each log scan. This parameter is used for primary database connections only.

## queue_size

| Summary information | |
|---|---|
| Default value | 1000 |
| Range of values | 1 to 2147483647 |
| Status | Dynamic |
| Display level | N/A |
| Required role | ASE Replicator system user |

queue_size specifies the maximum number of log operations kept in an internal, in-memory queue for the database connection. This parameter is used for both primary and replicate database connections.

Larger queue_size values allow more data to be stored in memory, potentially improving performance at the expense of more memory usage.

## scan_sleep_increment

| Summary information | |
|---|---|
| Default value | 5 |
| Range of values | 0 to 60 |
| Status | Dynamic |
| Display level | N/A |
| Required role | ASE Replicator system user |

scan_sleep_increment specifies the number of seconds that scan sleep time increases between consecutive empty log scans. This parameter is used for primary database connections only.

Scan sleep time is the time that the Publisher component waits to start a transaction log scan after a log scan returns no data. Each subsequent time that a log scan returns no data, the scan sleep time increases by the amount specified in scan_sleep_increment, up to the amount of time specified by scan_sleep_max.

When a log scan returns data, the scan sleep time is set to zero.

### scan_sleep_max

| Summary information | |
|---|---|
| Default value | 60 |
| Range of values | 0 to 60 |
| Status | Dynamic |
| Display level | N/A |
| Required role | ASE Replicator system user |

scan_sleep_max specifies the maximum number of seconds that the Publisher component waits to start a transaction log scan after a log scan returns no data. This parameter is used for primary database connections only.

### timeout

| Summary information | |
|---|---|
| Default value | 15 |
| Range of values | 0 to 2147483647 |
| Status | Dynamic |
| Display level | N/A |
| Required role | ASE Replicator system user |

timeout specifies the number of seconds for the log scan to block if the end of the log is reached before the maximum number of records to be returned from a log scan are read. This parameter takes effect only if the value of the mode parameter is block.

This parameter is used for primary database connections only.

CHAPTER 4

# ASE Replicator Procedures

This chapter describes the ASE Replicator command procedures that you use to perform ASE Replicator administration and maintenance tasks.

ASE Replicator procedures are created by the aserep script when you initialize the ASE Replicator process. ASE Replicator procedures are located in the Distribution Database, and they are owned by the ASE Replicator system user.

### Executing ASE Replicator procedures

ASE Replicator procedures can be run only in the Distribution Database. Therefore, when you log in to the Adaptive Server to administer ASE Replicator, you must either open the Distribution Database with the use command, or qualify each procedure name with the Distribution Database name.

All ASE Replicator procedures report a return status. The following message indicates that the procedure executed successfully:

```
return status = 0
```

Return value examples in this book do not include the return status.

### Entering parameter values

If a parameter value for an ASE Replicator procedure contains punctuation or embedded blanks, or is a reserved word, you must enclose it in single or double quotes. If the parameter is an object name qualified by a database name or owner name, enclose the entire name in single or double quotes. For example:

```
"owner.table"
```

If a procedure has multiple optional parameters, instead of supplying all the parameters, you can supply parameters in this form:

```
@parametername = value
```

The parameter names in the syntax statements match the parameter names defined by the procedures.

For example, the syntax for sp_addreplicateart is:

sp_addreplicateart *sub_name* [, *pri_art* [, *rep_art* [, *field_nums* [, *where_clause*]]]]

To use sp_addreplicateart to create a replicate article in the subscription subdoc, for the primary article table1, with no replicate article name specified, and subscribing to published fields 2, 3, and 4 in the primary article, you can invoke the procedure as:

```
sp_addreplicateart subdoc, table1, @field_nums="2-4"
```

**Note** Unlike Adaptive Server system procedures, you *cannot* use "null" as a placeholder for ASE Replicator procedures. If you attempt to do so, Adaptive Server returns an error.

If you specify more parameters than the number expected by the procedure, the extra parameters are ignored.

List of procedures

Table 4-1 lists all ASE Replicator procedures along with a brief description.

***Table 4-1: ASE Replicator procedures***

| Procedure name | Description | Page |
|---|---|---|
| sp_addprimaryart | Creates a new primary article in a publication | 86 |
| sp_addprimaryconn | Defines a new primary database connection | 89 |
| sp_addpub | Creates a new publication | 90 |
| sp_addreplicateart | Creates a new replicate article in a subscription | 91 |
| sp_addreplicateconn | Defines a new replicate database connection | 95 |
| sp_addsub | Creates a new subscription for a publication | 96 |
| sp_configprimaryconn | Sets or returns information about primary connection configuration parameters | 97 |
| sp_configrep | Sets or returns information about ASE Replicator configuration parameters | 99 |
| sp_configreplicateconn | Sets or returns information about replicate connection configuration parameters | 101 |
| sp_dropprimaryart | Deletes an existing primary article from a publication | 103 |
| sp_dropprimaryconn | Deletes an existing primary database connection | 104 |
| sp_droppub | Deletes an existing publication | 104 |
| sp_dropreplicateart | Deletes an existing replicate article from a subscription | 105 |
| sp_dropreplicateconn | Deletes an existing replicate database connection | 106 |
| sp_dropsub | Deletes an existing subscription | 107 |
| sp_helpconn | Returns information about database connections | 108 |
| sp_helplastcommit | Returns timestamp and locator value of the most recent transaction committed in the replicate database | 109 |
| sp_helplocator | Returns fields in the specified locator string | 110 |
| sp_helpprimaryart | Returns information about primary articles | 111 |
| sp_helpprimaryconn | Returns information about primary database connections | 114 |
| sp_helppub | Returns information about publications | 118 |

| Procedure name | Description | Page |
|---|---|---|
| sp_helprep | Returns statistics or status information for overall replication system, connections, and subscriptions | 120 |
| sp_helpreplicateart | Returns information about replicate articles | 125 |
| sp_helpreplicateconn | Returns information about replicate database connections | 129 |
| sp_helpsub | Returns information about subscriptions | 133 |
| sp_materializesub | Materializes and validates a subscription | 136 |
| sp_resumeprimaryconn | Resumes primary database connections | 138 |
| sp_resumerep | Resumes replication processes that are suspended or quiesced | 138 |
| sp_resumereplicateconn | Resumes replicate database connections and subscriptions | 139 |
| sp_resumesub | Resumes subscriptions | 140 |
| sp_shutdownrep | Shuts down ASE Replicator | 141 |
| sp_suspendprimaryconn | Suspends primary database connections | 141 |
| sp_suspendrep | Suspends all ASE Replicator components, connections, and subscriptions, but does not shut down ASE Replicator | 142 |
| sp_suspendreplicateconn | Suspends replicate database connections and subscriptions | 143 |
| sp_suspendsub | Suspends a specified subscription | 144 |
| sp_tracerep | Turns trace flags on or off, lists trace flags and their status | 144 |
| sp_validatesub | Validates a subscription | 147 |

The rest of this chapter describes each ASE Replicator procedure in detail.

# sp_addprimaryart

Description — Creates a new primary article (or articles) in a specified publication.

Syntax — sp_addprimaryart *pub_name* [, *pri_art* [, *field_nums*]]

Parameters

*pub_name*
is the name of the publication to which the new article is added.

*pri_art*
is the name of a primary object. Primary object names can be specified in the form *owner.name*.

*field_nums*
is one or more numbers that identify the fields to be published. Numbers can be entered with separating commas, or with dashes to indicate inclusive ranges.

Examples

**Example 1**

```
sp_addprimaryart pubdoc, table1
```

Creates a new primary article in the publication pubdoc for the primary object table1, with all fields in the primary object published, or adds an existing primary article named table1 to the publication pubdoc.

**Example 2**

```
sp_addprimaryart pubdoc, table1, "2-6, 8"
```

Creates a new primary article in the publication pubdoc for the primary object table1, publishing only fields (columns) 2, 3, 4, 5, 6, and 8 from the primary object.

Usage

- Before you invoke sp_addprimaryart to create a primary article, suspend the database connection using sp_suspendprimaryconn. After the primary article is created, you can resume the database connection with sp_resumeprimaryconn.
- To publish a primary object, you must first create a publication with sp_addpub, and then use sp_addprimaryart to create a primary article for the primary object. Creating a primary article publishes the primary object for replication. Only existing user tables and user stored procedures can be published in primary articles.
- When a primary article is first created, you can specify the fields (columns or parameters) of the primary object for publication. After a primary article is created, it can be added to other publications, however, field selection is no longer available. If you want to publish a different set of fields for an existing primary article, first delete the primary article from all publications it belongs to, then re-create the primary article with a different set of fields.
- When you publish a table, you must publish at least one field (column).

- When you publish a stored procedure, you can choose to publish none of the input parameters of the procedure. To specify none of the input parameters for publication, use the number 0 (zero) for the *field_nums* parameter in sp_addprimaryart. For example:

  ```
  sp_addprimaryart pubdoc, proc1, "0"
  ```

- If you need to change a published primary table or stored procedure in a way that affects its entry in the sysobjects table, you must first delete the primary article from all publications it belongs to, alter the table or stored procedure, then re-create the primary article.

  See the Adaptive Server *Reference Manual* for more information on object changes that affect the sysobjects table.

- If the primary object (*pri_art*) is a table, the fields available for publication are the columns in the table. If the primary object (*pri_art*) is a stored procedure, the fields available for publication are the stored procedure's input parameters (if any).

- If the name of a field (column or parameter) in a primary object conflicts with the name of a shadow table column, ASE Replicator returns an error message indicating that the primary object cannot be published because of a field name conflict. In that event, you must change the name of the field in the primary object if you want to publish the object.

  See "Distribution Database shadow tables" on page 172 for more information about shadow table column names.

- ASE Replicator creates a table named rl_lastcommit in the replicate database to keep track of transactions committed there. If you create a primary database connection to a replicate database (to implement bidirectional replication, for example), you cannot publish the rl_lastcommit table.

  **Note** ASE Replicator does not support replicating transactions to or from the rl_lastcommit table.

- When sp_addprimaryart is invoked with no primary object (*pri_art*) specified, all user tables and user procedures in the primary database are published.

  **Note** System tables and system procedures are *not* published by using the sp_addprimaryart procedure with no primary object specified.

- When sp_addprimaryart is invoked with a primary object (*pri_art*) specified, all fields in the specified primary object are published.
- When sp_addprimaryart is invoked with a primary object (*pri_art*) specified and field numbers (*field_nums*) specified, only the specified fields in the primary object are published.
- When the requested action occurs successfully, no results are returned.
- When an error occurs, an error message is returned.

See also

sp_addprimaryconn, sp_addpub, sp_addreplicateart, sp_helpprimaryart, sp_helpprimaryconn, sp_helppub

# sp_addprimaryconn

Description

Defines a new primary database connection.

Syntax

sp_addprimaryconn *conn_name* [, *maint_user*]

Parameters

*conn_name*

is the name of a primary database connection. Connection names must be specified in the form *ds.db*, where:

- *ds* is the name of the data server on which the primary database resides.
- *db* is the name of the primary database.

*maint_user*

is the Maintenance User name for the primary database.

Examples

```
sp_addprimaryconn "boulder.doc", fred
```

Defines a primary database connection to the database doc on data server boulder, where the Maintenance User name is fred.

Usage

- sp_addprimaryconn creates only a primary database connection to the database specified. If the primary database will also act as a replicate database in bidirectional replication, use sp_addreplicateconn to create a replicate database connection to the same database.
- The default Maintenance User for primary database connections is the ASE Replicator system user.
- When the requested action occurs successfully, no results are returned.
- When an error occurs, an error message is returned.

See also

sp_addprimaryart, sp_addpub, sp_addreplicateconn, sp_configprimaryconn, sp_helpprimaryart, sp_helpprimaryconn, sp_helppub

# sp_addpub

Description

Creates a new publication for a specified primary database connection.

Syntax

sp_addpub *pub_name*, *conn_name*

Parameters

*pub_name*
is the name of the new publication.

*conn_name*
is the name of a primary database connection. Connection names must be specified in the form *ds.db*, where:

- *ds* is the name of the data server on which the primary database resides.
- *db* is the name of the primary database.

Examples

```
sp_addpub pubdoc, "boulder.doc"
```

Creates a new publication named pubdoc for the primary connection to the database doc on the data server boulder.

Usage

- Before you invoke sp_addpub, you must suspend the database connection using sp_suspendprimaryconn. After the publication is created, you can resume the database connection with sp_resumeprimaryconn.
- The single quote (or apostrophe) character is not allowed in a publication name. For example, the publication name pub's is not allowed.
- To publish a primary object, after you create a publication, you must create at least one primary article in that publication with sp_addprimaryart.
- When the requested action occurs successfully, no results are returned.
- When an error occurs, an error message is returned.

See also

sp_addprimaryart, sp_addprimaryconn, sp_addsub, sp_helpprimaryart, sp_helpprimaryconn, sp_helppub

# sp_addreplicateart

Description

Creates a new replicate article (or articles) in a specified subscription.

Syntax

sp_addreplicateart *sub_name* [, *pri_art* [, *rep_art*
[, *field_nums*
[, *where_clause*]]]]

Parameters

*sub_name*
is the name of the subscription to which the new article is added.

*pri_art*
is the name of a primary article. Primary article names can be specified in the form *owner.name*.

*rep_art*
is the name of a replicate article, if different from the primary article. This parameter can be used when the name of the replicate object is different from the name of the primary object, which is always the same as the name of the primary article. Replicate article names can be specified in the form *owner.name*.

*field_nums*
is one or more numbers that identify the published fields to be subscribed to.

*where_clause*
is a SQL-type where clause that further selects the published data to be received by the replicate object identified by the replicate article. The SQL keyword where is optional.

Examples

**Example 1**

```
sp_addreplicateart subdoc, table1, @field_nums="2-4"
```

Creates a new replicate article in the subscription subdoc, for the primary article table1, subscribing only to published fields 2, 3, and 4 in the primary article.

**Example 2**

```
sp_addreplicateart subdoc, table1, @where_clause="where
style = 'round' or
style = 'square'"
```

Creates a new replicate article in the subscription subdoc, for the primary article table1, subscribing to all published fields in the primary article, and replicating only rows in which the value of style is either round or square.

**Example 3**

```
sp_addreplicateart subdoc, table1, reptable1, "1,3-4",
"where style = 'round' or style = 'square'"
```

Creates a new replicate article in the subscription subdoc, for the primary article table1, with the replicate object named reptable1, subscribing only to published fields 1, 3, and 4 in the primary article, and replicating only rows in which the value of style is either round or square.

Usage

- Before you invoke sp_addreplicateart to create a replicate article, you must suspend the subscription using sp_suspendsub. After the replicate article is created, you can resume the subscription using sp_resumesub.
- When a replicate article for a table is created, a proxy table is created in the Distribution Database. If the replicate table does not exist in the replicate database, it is created at the replicate database with default attributes and with columns based on the published fields selected for subscription.

  **Note** If the replicate table is created by ASE Replicator when the replicate article is created, ASE Replicator deletes the replicate table in the replicate database when you delete the replicate article.

- When ASE Replicator creates a replicate table in the replicate database, the owner of the table is either:
  - The ASE Replicator system user, if no separate login is specified for the Maintenance User, or

  - The Maintenance User login that the ASE Replicator system user was mapped to with sp_addexternlogin when the replicate database was set up or when the replicate database connection was created.

- When a replicate article for a stored procedure is created, a distribution procedure that calls the replicate stored procedure is created in the Distribution Database.

**Note** When a replicate article for a stored procedure is created, the replicate procedure must already exist in the replicate database.

- The replicate article (*rep_art*) you specify must identify a replicate object of the same type (table or stored procedure) as the primary article (*pri_art*). When the replicate object is a table, the published fields available for subscription are columns in the primary table. When the replicate object is a stored procedure, the published fields available for subscription are the primary stored procedure's input parameters.
- When a replicate article subscribes to a table, it must subscribe to at least one field (column).
- When a replicate article subscribes to a procedure, you can choose to subscribe to none of the published fields. To specify that none of the published fields should be subscribed to, use the number 0 (zero) for the *field_nums* parameter in sp_addreplicateart. For example:

```
sp_addreplicateart subdoc, proc1, repproc1, "0"
```

- If you specify a where clause for a replicate procedure article, you must reference the primary fields (parameter names) without using the at sign (@). For example:

```
sp_addreplicateart subdoc, proc1, repproc1, "1-3",
"where param1 = 'round' or param1 = 'square'"
```

- The published fields that a replicate article subscribes to are mapped to the *first available* fields (columns or parameters) in the replicate object. For example, if a replicate article subscribes to columns 1, 3, and 4 in the primary article, those fields are mapped to columns 1, 2, and 3 in the replicate table.
- The primary article (*pri_art*) you specify must exist in the publication to which the specified subscription (*sub_name*) subscribes.
- ASE Replicator creates a table named rl_lastcommit in the replicate database to keep track of transactions committed there. If you create a primary database connection to a replicate database (to implement

bidirectional replication, for example), you cannot publish the rl_lastcommit table.

**Note** ASE Replicator does not support replicating transactions to or from the rl_lastcommit table.

- When sp_addreplicateart is invoked with no primary article (*pri_art*) specified, a replicate article subscribing to all published fields is created for every primary article in the publication.
- When sp_addreplicateart is invoked with only a primary article (*pri_art*) specified, a replicate article subscribing to all published fields in the specified primary article is created.
- When sp_addreplicateart is invoked with both a primary article (*pri_art*) and a replicate article (*rep_art*) specified, a replicate article is created with the *rep_art* name specified.
- When sp_addreplicateart is invoked with a primary article (*pri_art*) specified, and field numbers (*field_nums*) specified, a replicate article subscribing to the specified published fields in the specified primary article is created.
- When sp_addreplicateart is invoked with a primary article (*pri_art*) specified, and a where clause (*where_clause*) is specified, a replicate article subscribing to the selected data specified by the where clause is created.

**Note** If you specify a where clause for a replicate procedure article, you must reference the primary fields (input parameters) without the at sign (@).

- You can combine field numbers (*field_nums*) and a where clause (*where_clause*) to select both a subset of the published fields and selected data from the primary article.
- When the requested action occurs successfully, no results are returned.
- When an error occurs, an error message is returned.

See also sp_addprimaryart, sp_addreplicateconn, sp_addsub, sp_helpreplicateart, sp_helpreplicateconn, sp_helpsub

# sp_addreplicateconn

Description

Defines a new replicate database connection.

Syntax

sp_addreplicateconn *conn_name* [, *maint_user* [, *maint_pw*]]

Parameters

*conn_name*
is the name of a replicate database connection. Connection names must be specified in the form *ds*.*db*, where:

- *ds* is the name of the data server on which the replicate database resides.
- *db* is the name of the replicate database.

*maint_user*
is the Maintenance User login for the replicate database.

*maint_pw*
is the Maintenance User password for the replicate database.

Examples

```
sp_addreplicateconn "boulder.doc", fred, P8g3n
```

Defines a replicate database connection to the database doc on data server boulder, where the Maintenance User login is fred and the Maintenance User password is P8g3n.

Usage

- sp_addreplicateconn creates only a replicate database connection. If the replicate database will also act as a primary database in bidirectional replication, use sp_addprimaryconn to create a primary database connection to that database.
- ASE Replicator creates a table named rl_lastcommit in the replicate database to keep track of transactions committed there. If you create a primary database connection to a replicate database (to implement bidirectional replication, for example), you cannot publish the rl_lastcommit table.

  **Note** ASE Replicator does not support replicating transactions to or from the rl_lastcommit table.

- The default Maintenance User for replicate database connections is the ASE Replicator system user login at the primary Adaptive Server.
- The Maintenance User login must exist in the replicate data server and replicate database identified in the connection name. sp_addreplicateconn returns an error if you attempt to create a connection with the default Maintenance User when either of the following conditions exist:

  - The ASE Replicator system user login is not a valid login on the replicate data server.
  - The ASE Replicator system user login is not a valid login in the replicate database.
- To use a Maintenance User login other than the default ASE Replicator system user login, you can either:
  - Specify a Maintenance User login and password when you invoke sp_addreplicateconn, or
  - Map the ASE Replicator system user login to a different login (valid on the replicate data server) using sp_addexternlogin in the primary Adaptive Server.
- When the requested action occurs successfully, no results are returned.
- When an error occurs, an error message is returned.

See also

sp_addprimaryconn, sp_addreplicateart, sp_addsub, sp_configreplicateconn, sp_helpreplicateart, sp_helpreplicateconn, sp_helpsub

# sp_addsub

Description

Creates a new subscription for a specified publication.

Syntax

sp_addsub *sub_name*, *pub_name*, *conn_name*

Parameters

*sub_name*
is the name of the new subscription.

*pub_name*
is the name of the publication to which the new subscription subscribes.

*conn_name*
is the name of a replicate database connection. Connection names must be specified in the form *ds.db*, where:

- *ds* is the name of the data server on which the replicate database resides.
- *db* is the name of the replicate database.

Examples

```
sp_addsub subdoc, pubdoc, "boulder.doc"
```

Creates a new subscription named subdoc to the publication pubdoc, for the replicate connection to the database doc on the data server boulder.

Usage

- Before you invoke sp_addsub, you must suspend the database connection using sp_suspendreplicateconn. After the subscription is created, you can resume the database connection with sp_resumereplicateconn.
- The single quote (or apostrophe) character is not allowed in a subscription name. For example, the subscription name sub's is not allowed.
- To subscribe to a primary object, after you create a subscription, you must create at least one replicate article in that subscription with sp_addreplicateart.
- When the requested action occurs successfully, no results are returned.
- When an error occurs, an error message is returned.

See also

sp_addpub, sp_addreplicateart, sp_addreplicateconn, sp_helpreplicateart, sp_helpreplicateconn, sp_helpsub

# sp_configprimaryconn

Description

Sets or returns information about primary connection configuration parameters.

Syntax

sp_configprimaryconn *conn_name* [, *param* [, *value*]]

Parameters

*conn_name*

is the name of a primary database connection. Connection names must be specified in the form *ds.db*, where:

- *ds* is the name of the data server on which the primary database resides.
- *db* is the name of the primary database.

*param*

is the name of an ASE Replicator connection configuration parameter.

Table 4-2 lists ASE Replicator primary connection configuration parameters:

***Table 4-2: Primary connection configuration parameters***

| Parameter | Description |
|---|---|
| gen_id | Database generation ID (first two bytes in the connection's locator value) |
| lti_version | Log scan protocol version number |
| mode | Scan mode for the primary database log |
| numrecs | Maximum number of records returned by each log scan |
| queue_size | Maximum number of log operations kept in an internal queue |
| scan_sleep_increment | Number of seconds sleep time increases between empty log scans |
| scan_sleep_max | Maximum number of seconds between log scans |
| timeout | Number of seconds to block if the end of the log is reached before the maximum number of records (numrecs) are read |

See "Connection configuration parameters" on page 78 for more detailed information about these connection configuration parameters.

*value*

is the value to which the configuration parameter (*param*) is set. Numeric values are treated as strings and must be enclosed in quotes.

Examples

```
sp_configprimaryconn "boulder.doc", timeout, "30"
```

Sets the value of the timeout configuration parameter to 30 for the primary connection to the database doc on the data server boulder.

Usage

- When listing information about connection configuration parameters, the following result set is returned:

***Table 4-3: ASE Replicator configuration parameter information***

| Column | Datatype | Description |
|---|---|---|
| parameter_name | varchar(128) | Name of the configuration parameter |
| default_value | varchar(255) | Default value of the parameter |
| legal_values | varchar(255) | Legal values of the parameter |
| description | varchar(255) | Description of the parameter |

- When sp_configprimaryconn is invoked with a connection name (*conn_name*) specified, but no parameter (*param*), it returns the values of all connection configuration parameters for the specified connection.

- When sp_configprimaryconn is invoked with a connection name (*conn_name*) and a parameter (*param*) specified, but no value (*value*), it returns the value of the specified parameter for the specified connection.
- When sp_configprimaryconn is invoked with a connection name (*conn_name*), a parameter (*param*), and a value (*value*) specified, it sets the specified parameter to the specified value for the specified connection.
- When the requested action occurs successfully, no results are returned.
- When an error occurs, an error message is returned.

See also sp_addprimaryconn, sp_dropprimaryconn, sp_helpprimaryconn

# sp_configrep

Description Sets or returns information about ASE Replicator configuration parameters.

Syntax sp_configrep [*param* [, *value*]]

Parameters *param*

is the name of an ASE Replicator configuration parameter.

Table 4-4 lists all ASE Replicator configuration parameters.

Some configuration parameters (indicated by an asterisk) cannot be changed with the sp_configrep procedure. These parameters must be specified on the aserep command line when the ASE Replicator process is started.

***Table 4-4: ASE Replicator configuration parameters***

| Parameter | Description |
|---|---|
| admin_port * | ASE Replicator client socket port number |
| ase_charset * | Adaptive Server default character set |
| ase_host * | Name of the host machine where Adaptive Server resides |
| ase_port * | Adaptive Server client socket port number |
| batch_size | Number of commands to batch |
| batch_timeout | Timeout limit for command batching |
| ddb_name * | Name of the Distribution Database |
| log_directory | Directory for system log files |
| log_trace_verbose | Enable or disable verbose trace message content |
| log_wrap | Number of 1K blocks before wrapping log files |
| monitor_delay | Sybase Central monitor ping interval, in seconds |

| Parameter | Description |
|---|---|
| queue_size | Maximum number of log operations kept in an internal queue |
| scan_sleep_increment | Number of seconds sleep time increases between empty log scans |
| scan_sleep_max | Maximum number of seconds between log scans |
| stat_trunc_interval | Number of days after which statistics are deleted from repository |
| stat_write_timeout | Frequency statistics are written to repository |
| status_monitoring | Enable or disable Sybase Central status monitoring |
| truncate_numops | Minimum number of replicated operations in stable queue before truncation occurs |

See "ASE Replicator configuration parameters" on page 69 for more detailed information about these configuration parameters.

*value*
is the value to which the configuration parameter (*param*) is set.

**Note** Numeric values are treated as strings and must be enclosed in quotes.

Examples

```
sp_configrep log_wrap, "10"
```

Sets the ASE Replicator log_wrap configuration parameter to the value 10.

Usage

- When listing information about configuration parameters, the following result set is returned:

***Table 4-5: ASE Replicator configuration parameter information***

| Column | Datatype | Description |
|---|---|---|
| parameter_name | varchar(128) | Name of the configuration parameter |
| default_value | varchar(255) | Default value of the parameter |
| legal_values | varchar(255) | Legal values of the parameter |
| description | varchar(255) | Description of the parameter |

- When sp_configrep is invoked with no parameter (*param*) specified, it returns a list of all ASE Replicator configuration parameters, with information for each parameter.
- When sp_configrep is invoked with a parameter (*param*) specified, but no value (*value*), it returns information for the specified parameter.

- When sp_configrep is invoked with both a parameter (*param*) and value (*value*) specified, it sets the specified parameter to the specified value.
- When you change the value of a dynamic parameter, the change occurs immediately after you invoke sp_configrep. To change the value of a static parameter, you must shut down and restart the ASE Replicator process after you set the value.
- When the requested action occurs successfully, no results are returned.
- When an error occurs, an error message is returned.

See also sp_configprimaryconn, sp_configreplicateconn, sp_helprep

# sp_configreplicateconn

Description Sets or returns information about replicate connection configuration parameters.

Syntax sp_configreplicateconn *conn_name* [, *param* [, *value*]]

Parameters *conn_name*

is the name of a replicate database connection. Connection names must be specified in the form *ds.db*, where:

- *ds* is the name of the data server on which the replicate database resides.
- *db* is the name of the replicate database.

*param*

is the name of an ASE Replicator connection configuration parameter.

Table 4-6 lists the ASE Replicator replicate connection configuration parameters:

***Table 4-6: Replicate connection configuration parameters***

| Parameter | Description |
|---|---|
| gen_id | Database generation ID (first two bytes in the connection's locator value) |
| queue_size | Maximum number of log operations kept in an internal queue |

See "Connection configuration parameters" on page 78 for more detailed information about these connection configuration parameters.

*value*
is the value to which the configuration parameter (*param*) is set. Numeric values are treated as strings and must be enclosed in quotes.

Examples

```
sp_configreplicateconn "boulder.doc", queue_size, "100"
```

Sets the value of the queue_size configuration parameter to 100 for the replicate connection to the database doc on the data server boulder.

Usage

- When listing information about connection configuration parameters, the following result set is returned:

***Table 4-7: ASE Replicator configuration parameter information***

| Column | Datatype | Description |
|---|---|---|
| parameter_name | varchar(128) | Name of the configuration parameter |
| default_value | varchar(255) | Default value of the parameter |
| legal_values | varchar(255) | Legal values of the parameter |
| description | varchar(255) | Description of the parameter |

- When sp_configreplicateconn is invoked with a connection name (*conn_name*) specified, but no parameter (*param*), it returns the values of all connection configuration parameters for the specified connection.
- When sp_configreplicateconn is invoked with a connection name (*conn_name*) and a parameter (*param*) specified, but no value (*value*), it returns the value of the specified parameter for the specified connection.
- When sp_configreplicateconn is invoked with a connection name (*conn_name*), a parameter (*param*), and a value (*value*) specified, it sets the specified parameter to the specified value for the specified connection.
- When the requested action occurs successfully, no results are returned.
- When an error occurs, an error message is returned.

See also

sp_addreplicateconn, sp_dropreplicateconn, sp_helpreplicateconn

# sp_dropprimaryart

Description

Deletes an existing primary article (or articles) from a specified publication.

Syntax

sp_dropprimaryart *pub_name* [, *pri_art*]

Parameters

*pub_name*
is the name of the publication to which the existing article belongs.

*pri_art*
is the name of a primary article. Primary article names can be specified in the form *owner.name*.

Examples

```
sp_dropprimaryart pubdoc, table1
```

Deletes the primary article table1 in the publication pubdoc.

Usage

- Before you invoke sp_dropprimaryart, you must suspend the database connection using sp_suspendprimaryconn. After the primary article is deleted, you can resume the database connection with sp_resumeprimaryconn.
- When sp_dropprimaryart is invoked with only a publication name (*pub_name*) specified, all primary articles in the specified publication are dropped.
- When sp_dropprimaryart is invoked with both a publication name (*pub_name*) and primary article name (*pri_art*) specified, the specified article in the specified publication is dropped.
- If you attempt to drop a primary article that is subscribed to by a replicate article in an associated subscription, sp_dropprimaryart returns an error. You must drop all replicate articles from the associated subscriptions that subscribe to a primary article before you drop the primary article.
- When the requested action occurs successfully, no results are returned.
- When an error occurs, an error message is returned.

See also

sp_addprimaryart, sp_droppub, sp_dropreplicateart, sp_helpprimaryart, sp_helpprimaryconn, sp_helppub

# sp_dropprimaryconn

Description
Deletes an existing primary database connection definition.

Syntax
sp_dropprimaryconn *conn_name*

Parameters
*conn_name*
is the name of a primary database connection. Connection names must be specified in the form *ds.db*, where:

- *ds* is the name of the data server on which the primary database resides.
- *db* is the name of the primary database.

Examples

```
sp_dropprimaryconn "boulder.doc"
```

Deletes the primary database connection definition for the database doc on the data server boulder.

Usage
- If a primary connection has any publication defined for it, you must drop the publication before you drop the primary connection.
- If you attempt to drop a primary connection that has a publication defined, sp_dropprimaryconn returns an error.
- When the requested action occurs successfully, no results are returned.
- When an error occurs, an error message is returned.

See also
sp_addprimaryconn, sp_droppub, sp_dropreplicateconn, sp_helpprimaryart, sp_helpprimaryconn, sp_helppub

# sp_droppub

Description
Deletes an existing publication.

Syntax
sp_droppub *pub_name*

Parameters
*pub_name*
is the name of the publication to delete.

Examples

```
sp_droppub pubdoc
```

Deletes the publication pubdoc.

Usage

- Before you invoke sp_droppub, you must suspend the database connection using sp_suspendprimaryconn. After the publication is deleted, you can resume the database connection with sp_resumeprimaryconn.
- If a publication has any primary article defined for it, you must drop the primary article before you drop the publication.
- If you attempt to drop a publication that has a primary article defined, sp_droppub returns an error.
- When the requested action occurs successfully, no results are returned.
- When an error occurs, an error message is returned.

See also

sp_addpub, sp_dropsub, sp_helpprimaryart, sp_helpprimaryconn, sp_helppub

# sp_dropreplicateart

Description

Deletes an existing replicate article (or articles) from a specified subscription.

**Note** If ASE Replicator creates a replicate table when a replicate article is created, ASE Replicator deletes the replicate table in the replicate database when you delete that replicate article.

Syntax

sp_dropreplicateart *sub_name* [, *rep_art*]

Parameters

*sub_name*
is the name of the subscription to which the existing article belongs.

*rep_art*
is the name of a replicate article. Replicate article names can be specified in the form *owner.name*.

Examples

```
sp_dropreplicateart subdoc, reptable1
```

Deletes the replicate article reptable1 in the subscription subdoc.

Usage

- Before you invoke sp_dropreplicateart, you must suspend the subscription using sp_suspendsub. After the replicate article is deleted, you can resume the subscription using sp_resumesub.

- When sp_dropreplicateart is invoked with only a subscription name (*sub_name*) specified, all replicate articles in the specified subscription are dropped.
- When sp_dropreplicateart is invoked with both a subscription name (*sub_name*) and replicate article name (*rep_art*) specified, the specified replicate article in the specified subscription is dropped.
- If any replicate articles subscribe to a primary article, you must drop the subscribing replicate article before you can drop the primary article from the associated publication.
- When the requested action occurs successfully, no results are returned.
- When an error occurs, an error message is returned.

See also

sp_addreplicateart, sp_dropprimaryart, sp_dropsub, sp_helpreplicateart, sp_helpreplicateconn, sp_helpsub

# sp_dropreplicateconn

Description

Deletes an existing replicate database connection definition.

Syntax

sp_dropreplicateconn *conn_name*

Parameters

*conn_name*
is the name of a replicate database connection. Connection names must be specified in the form *ds.db*, where:

- *ds* is the name of the data server on which the replicate database resides.
- *db* is the name of the replicate database.

Examples

```
sp_dropreplicateconn "boulder.doc"
```

Deletes the replicate database connection definition for the database doc on the data server boulder.

Usage

- If a replicate connection has any subscription defined for it, you must drop the subscription before you drop the replicate connection.
- If you attempt to drop a replicate connection that has a subscription defined, sp_dropreplicateconn returns an error.
- When the requested action occurs successfully, no results are returned.

- When an error occurs, an error message is returned.

See also

sp_addreplicateconn, sp_dropprimaryart, sp_dropsub, sp_helpreplicateart, sp_helpreplicateconn, sp_helpsub

# sp_dropsub

Description

Deletes an existing subscription.

Syntax

sp_dropsub *sub_name*

Parameters

*sub_name*
is the name of the subscription to delete.

Examples

```
sp_dropsub subdoc
```

Deletes the subscription subdoc.

Usage

- Before you invoke sp_dropsub, you must suspend the database connection using sp_suspendreplicateconn. After the subscription is deleted, you can resume the database connection with sp_resumereplicateconn.
- If a subscription has any replicate article defined for it, you must drop the replicate article before you drop the subscription.
- If you attempt to drop a subscription that has a replicate article defined, sp_dropsub returns an error.
- When the requested action occurs successfully, no results are returned.
- When an error occurs, an error message is returned.

See also

sp_addsub, sp_dropreplicateart, sp_dropreplicateconn, sp_helpreplicateart, sp_helpreplicateconn, sp_helpsub

# sp_helpconn

Description

Returns information about database connections.

Syntax

sp_helpconn [*conn_name*]

Parameters

*conn_name*

is the name of a database connection. Connection names must be specified in the form *ds.db*, where:

- *ds* is the name of the data server on which the database resides.
- *db* is the name of the database.

Examples

```
sp_helpconn "boulder.doc"
```

Returns information about any primary and replicate database connections for the database doc on the data server boulder.

Usage

- When listing information about connections, the following result set is returned:

***Table 4-8: ASE Replicator connection information***

| Column | Datatype | Description |
|---|---|---|
| conn_type | varchar(3) | Type of connection:<br>PRI – primary<br>REP – replicate |
| ds | sysname | If *conn_type* is PRI, then *ds* is the name of the local primary data server.<br>If *conn_type* is REP, then *ds* is the name of the replicate data server. |
| db | sysname | If *conn_type* is PRI, then *db* is the name of the local primary database.<br>If *conn_type* is REP, then *db* is the name of the replicate database. |
| maint_user | sysname | Maintenance User name in the database identified in *db*. |
| lastcmt_proxy | sysname | Lastcommit proxy table name. |
| restart_locator | varchar(64) | If *conn_type* is PRI, then *restart_locator* identifies the restart position in the database log.<br>If *conn_type* is REP, then *restart_locator* identifies the restart position in the stable queue. |
| timestamp | datetime | Timestamp of the restart locator. |

| Column | Datatype | Description |
|---|---|---|
| status | smallint | Status of the connection:<br>1 – up<br>0 – down<br>-1 – error |
| status_desc | varchar(255) | Description of the connection status. |

- When sp_helpconn is invoked with no keyword, it returns information about all primary and replicate connections.
- When an error occurs, an error message is returned.

See also

sp_helpprimaryconn, sp_helpreplicateconn

# sp_helplastcommit

Description

Returns information about the most recent transaction committed in the replicate database for the article specified.

Syntax

sp_helplastcommit *rep_art*, {conn=*conn_name*|sub=*sub_name*}

Parameters

*rep_art*
is name of a replicate article. Replicate article names can be specified in the form *owner.name*.

conn=*conn_name*
is a label identifying a replicate database connection. The connection name (*conn_name*) must be specified in the form *ds.db*, where:

- *ds* is the name of the data server on which the database resides.
- *db* is the name of the database.

sub=*sub_name*
is a label identifying a subscription.

Examples

```
sp_helplastcommit table1, "sub=subdoc"
```

Returns the timestamp and locator value of the most recent transaction committed for the replicate article table1 in the subscription subdoc. Also returns the validation point for the replicate article table1.

Usage

- When listing information from the rl_lastcommit table, the following result set is returned:

***Table 4-9: ASE Replicator rl_lastcommit information***

| Column | Datatype | Description |
|---|---|---|
| timestamp | datetime | Timestamp of the locator value |
| locator | varchar(64) | Locator value |
| validation_pt | varchar(64) | Validation point for the specified replicate article |

- sp_helplastcommit is for use primarily during troubleshooting procedures.
- You can use sp_helplocator to return the fields in locator values returned by sp_helplastcommit.
- When an error occurs, an error message is returned.

See also sp_helplocator, sp_helprep

# sp_helplocator

Description Returns fields in a locator string.

Syntax sp_helplocator *value*

Parameters *value*
is the value of the locator string that identifies a transaction in the stable queue.

Examples

```
sp_helplocator
"000000000003ad0000022d4d000b00022d4d00061e00350000000
00000000000"
```

Returns the fields in the locator string with the value specified, as follows:

```
item           hex_value     value
Generation ID    0000          0
txid_page        00022d4d      142669
txid_row         0006          6
log_ts_high      0000          0
log_ts_low       0003ad00      240896
opid_page        00022d4d      142669
opid_row         000b          11
op_code          1e            30
conn_id          0035          53
```

Usage

- When listing fields in a locator string, the following result set is returned:

***Table 4-10: ASE Replicator locator field information***

| Column | Datatype | Description |
|---|---|---|
| item | varchar(20) | Name of the field |
| hex_value | varchar(10) | Hexadecimal value of the field |
| value | varchar(10) | Decimal value of the field |

- sp_helplocator is for use primarily in troubleshooting procedures.
- You can use sp_helplocator to return the fields in locator values returned by sp_helplastcommit.
- When an error occurs, an error message is returned.

See also

sp_helplastcommit, sp_helprep

# sp_helpprimaryart

Description

Returns information about primary articles.

Syntax

sp_helpprimaryart [info]

sp_helpprimaryart {info|pubs}, *pri_art*
[, {conn=*conn_name*|pub=*pub_name*}]

sp_helpprimaryart fields, *pri_art*, {conn=*conn_name*|pub=*pub_name*}

Parameters

info
is a keyword that requests information about all primary articles in all primary databases, or information about a specified primary article in a specified primary database or publication.

fields
is a keyword that requests field information for a specified primary article in a specified primary database or publication.

pubs
is a keyword that requests information about all publications that the specified primary article belongs to.

*pri_art*
is the name of a primary article. Primary article names can be specified in the form *owner.name*.

conn=*conn_name*
is a label identifying a primary database connection. The connection name (*conn_name*) must be specified in the form *ds.db*, where:

- *ds* is the name of the data server on which the database resides.
- *db* is the name of the database.

pub=*pub_name*
is a label identifying a publication.

Examples

**Example 1**

```
sp_helpprimaryart
```

Returns information about all primary articles in all primary database connections.

---

**Note** Using sp_helpprimaryart with no keyword is functionally identical to specifying only the info keyword.

---

**Example 2**

```
sp_helpprimaryart info, "joe.table1",
"conn=boulder.doc"
```

Returns information about the primary article table1 with owner joe in the primary database doc on the data server boulder.

**Example 3**

```
sp_helpprimaryart fields, "joe.table1",
"conn=boulder.doc"
```

Returns field information about the primary article table1 with owner joe in the primary database doc on the data server boulder.

**Example 4**

```
sp_helpprimaryart pubs, "joe.table1",
"conn=boulder.doc"
```

Returns all publications that own the primary article table1 with owner joe in the primary database doc on the data server boulder.

Usage

- When listing information about primary articles, the following result set is returned:

***Table 4-11: ASE Replicator primary article information***

| Column | Datatype | Description |
|---|---|---|
| ds | sysname | Name of the local primary data server. |
| db | sysname | Name of the local primary database. |
| pub_name | sysname | Publication name. |
| owner | sysname | Owner of primary article. |
| part_name | sysname | Name of primary article. |
| proc_num | smallint | Stored procedure group number. If only one procedure exists, then *proc_num*=1. Tables are *proc_num*=0. |
| art_type | varchar(9) | Type of article:<br>table<br>procedure |
| numsubscribers | int | Number of replicate articles that subscribe to this primary article. |
| shadow_table | sysname | Name of the shadow table for this primary article. |

- When listing information about primary article fields, the following result set is returned:

***Table 4-12: ASE Replicator primary article field information***

| Column | Datatype | Description |
|---|---|---|
| ds | sysname | Name of the local primary data server. |
| db | sysname | Name of the local primary database. |
| owner | sysname | Owner of primary article. |
| part_name | sysname | Name of primary article. |
| proc_num | smallint | Stored procedure group number. If only one procedure exists, then *proc_num*=1. Tables are *proc_num*=0. |
| art_type | varchar(9) | Type of primary object:<br>table<br>procedure |
| field_id | smallint | Field identifier, ordinal position. |
| field_name | sysname | Name of the published field (column or parameter name). |
| datatype | sysname | Datatype of the field. |
| precision | int | Length of string or binary field, or precision of a numeric field, if applicable. Otherwise, 0. |
| scale | smallint | Scale of numeric field, if applicable. Otherwise, 0. |

- When sp_helpprimaryart is invoked with no keyword, it returns information about all primary articles in all primary database connections.
- When you invoke sp_helpprimaryart with the fields keyword and an article name (*pri_art*), you must specify either a connection name (conn=*conn_name*) or a publication name (pub=*pub_name*).
- To get information about replicate articles, use sp_helpreplicateart.
- To get information about publications or subscriptions, use sp_helppub or sp_helpsub.
- To get information about database connections, use sp_helpconn, sp_helpprimaryconn, or sp_helpreplicateconn.
- When an error occurs, an error message is returned.

See also sp_helpprimaryconn, sp_helppub, sp_helpreplicateart

# sp_helpprimaryconn

Description Returns information and statistics for primary database connections.

Syntax sp_helpprimaryconn [info]

sp_helpprimaryconn {info|stats} [, *conn_name*]

sp_helpprimaryconn {pubs|empty|unpub|arts|unsub|params}, *conn_name*

Parameters

info
is a keyword that requests information about primary database connections.

stats
is a keyword that requests statistics for primary database connections.

pubs
is a keyword that requests information about all publications in the specified primary database connection.

empty
is a keyword that requests information about all empty publications (publications with no articles) in the specified primary database connection.

unpub
is a keyword that requests information about all unpublished objects in the database identified by the specified primary database connection.

arts
is a keyword that requests information about all primary articles in all publications in the specified primary database connection.

unsub
is a keyword that requests information about all primary articles in the specified primary database connection, for which there are no subscribers.

params
is a keyword that requests information about all configuration parameters for the specified primary database connection.

*conn_name*
is the name of a primary database connection. Connection names must be specified in the form *ds.db*, where:

- *ds* is the name of the data server on which the primary database resides.
- *db* is the name of the primary database.

Examples

**Example 1**

```
sp_helpprimaryconn
```

Returns information about all primary database connections.

---

**Note** Using sp_helpprimaryconn with no keyword is functionally identical to specifying only the info keyword.

---

**Example 2**

```
sp_helpprimaryconn info, "boulder.doc"
```

Returns information about the primary connection to the database doc on the data server boulder.

**Example 3**

```
sp_helpprimaryconn stats, "boulder.doc"
```

Returns statistics information about the primary connection to the database doc on the data server boulder.

Usage

- When listing information about primary connections, the following result set is returned:

***Table 4-13: ASE Replicator primary connection information***

| Column | Datatype | Description |
|---|---|---|
| conn_type | varchar(3) | Type of connection: PRI – primary |

| Column | Datatype | Description |
|---|---|---|
| ds | sysname | Name of the local primary data server |
| db | sysname | Name of the local primary database |
| maint_user | sysname | Maintenance User name in the database identified in *db* |
| lastcmt_proxy | sysname | Lastcommit proxy table name |
| restart_locator | varchar(64) | Identifies restart position in the database log |
| timestamp | datetime | Timestamp of the restart locator |
| status | smallint | Status of the connection:<br>1 – up<br>0 – down<br>-1 – error |
| status_desc | varchar(255) | Description of the connection status |

- When listing statistics information about primary connections, the following result set is returned:

***Table 4-14: ASE Replicator primary connection statistics***

| Column | Datatype | Description |
|---|---|---|
| type | varchar(3) | Type of connection: PRI – primary |
| name | sysname | Name of the local primary data server and database |
| tstamp | datetime | Timestamp the statistic was generated |
| statistic | varchar(128) | Name of the statistic |
| value | varchar(255) | Value of the statistic |

- When listing configuration parameter information about primary connections, the following result set is returned:

***Table 4-15: ASE Replicator primary connection configuration***

| Column | Datatype | Description |
|---|---|---|
| parameter_name | varchar(128) | Name of the configuration parameter |
| default_value | varchar(255) | Default value of the parameter |
| legal_values | varchar(255) | Legal values of the parameter |
| description | varchar(255) | Description of the parameter |

- When listing information about publications in primary connections, the following result set is returned:

***Table 4-16: ASE Replicator primary connection publications***

| Column | Datatype | Description |
|---|---|---|
| ds | sysname | Name of the local primary data server |
| db | sysname | Name of the local primary database |
| pub_name | sysname | Publication name |

- When listing information about unpublished objects in a primary database, the following result set is returned:

***Table 4-17: ASE Replicator primary database unpublished objects***

| Column | Datatype | Description |
|---|---|---|
| ds | sysname | Name of the local primary data server |
| db | sysname | Name of the local primary database |
| obj_owner | sysname | Owner of primary object |
| obj_name | sysname | Name of primary object |
| obj_type | varchar(9) | Type of primary object:<br>table<br>procedure |

- When listing information about primary articles in a primary database connection, the following result set is returned:

***Table 4-18: ASE Replicator primary articles***

| Column | Datatype | Description |
|---|---|---|
| ds | sysname | Name of the local primary data server. |
| db | sysname | Name of the local primary database. |
| pub_name | sysname | Publication name. |
| owner | sysname | Owner of primary article. |
| part_name | sysname | Name of primary article. |
| proc_num | smallint | Stored procedure group number. If only one procedure exists, then *proc_num*=1. Tables are *proc_num*=0. |
| art_type | varchar(9) | Type of article:<br>table<br>procedure |
| numsubscribers | int | Number of replicate articles that subscribe to this primary article. |
| shadow_table | sysname | Name of the shadow table for this primary article. |

- When sp_helpprimaryconn is invoked with no keyword, it returns information about all primary database connections.
- When you invoke sp_helpprimaryconn and specify the info or stats keyword, the primary database connection name (*conn_name*) is optional.
- When you invoke sp_helpprimaryconn and specify the pubs, empty, unpub, arts, unsub, or params keyword, the primary database connection name (*conn_name*) is required.
- To get information about primary and replicate connections, use sp_helpconn.
- To get information about replicate connections only, use sp_helpreplicateconn.
- When an error occurs, an error message is returned.

See also sp_helpprimaryart, sp_helppub, sp_helpreplicateconn

# sp_helppub

Description Returns information about publications.

Syntax sp_helppub [info|empty]

sp_helppub {info|arts|unsub}, *pub_name*

Parameters info
is a keyword that requests information for all publications or for the specified publication.

empty
is a keyword that requests information about all empty publications (publications with no articles) in all primary database connections.

arts
is a keyword that requests information about all primary articles in the specified publication.

unsub
is a keyword that requests information about all primary articles in the specified publication, for which there are no subscribers.

*pub_name*
is the name of a publication.

Examples

**Example 1**

```
sp_helppub
```

Returns information about all publications for all primary database connections.

**Note** Using sp_helppub with no keyword is functionally identical to specifying only the info keyword.

**Example 2**

```
sp_helppub info, pubdoc
```

Returns information about the publication pubdoc.

**Example 3**

```
sp_helppub unsub, pubdoc
```

Returns information about all primary articles for which there are no subscribers in the publication pubdoc.

Usage

- When listing information about publications, the following result set is returned:

***Table 4-19: ASE Replicator publication information***

| Column | Datatype | Description |
|---|---|---|
| ds | sysname | Name of the local primary data server |
| db | sysname | Name of the local primary database |
| pub_name | sysname | Publication name |

- When listing information about primary articles in publications, the following result set is returned:

***Table 4-20: ASE Replicator publication articles***

| Column | Datatype | Description |
|---|---|---|
| ds | sysname | Name of the local primary data server. |
| db | sysname | Name of the local primary database. |
| pub_name | sysname | Publication name. |
| owner | sysname | Owner of primary article. |
| part_name | sysname | Name of primary article. |
| proc_num | smallint | Stored procedure group number. If only one procedure exists, then *proc_num*=1. Tables are *proc_num*=0. |

| Column | Datatype | Description |
|---|---|---|
| art_type | varchar(9) | Type of article:<br>table<br>procedure |
| numsubscribers | int | Number of replicate articles that subscribe to this primary article. |
| shadow_table | sysname | Name of the shadow table for this primary article. |

- When sp_helppub is invoked with no keyword, it returns information about all publications for all primary database connections.
- When you invoke sp_helppub and specify the info keyword, the publication name (*pub_name*) is optional.
- When you invoke sp_helppub and specify the empty keyword, you cannot specify a publication name (*pub_name*).
- When you invoke sp_helppub and specify the arts or unsub keyword, the publication name (*pub_name*) is required.
- When an error occurs, an error message is returned.

See also

sp_helpprimaryart, sp_helpprimaryconn, sp_helpsub

# sp_helprep

Description

Returns statistics or status information about ASE Replicator, ASE Replicator objects, or ASE Replicator components.

Syntax

```
sp_helprep [stats [, { reset|pri_conns|rep_conns|subs|
        {pri_conn|rep_conn}=conn_name|
        sub=sub_name }]]

sp_helprep [status [, {conns|subs|
        {pri_conn|rep_conn}=conn_name|
        sub=sub_name }]]

sp_helprep [version]
```

Parameters

stats
is a keyword that requests statistics information about ASE Replicator.

status
is a keyword that requests status information about ASE Replicator. This is the default value.

version
is a keyword that requests the ASE Replicator version string.

reset
is a keyword that resets *all* statistics counters to zero.

pri_conns
is a keyword that requests statistics or status information for all primary database connections.

rep_conns
is a keyword that requests statistics or status information for all replicate database connections.

subs
is a keyword that requests statistics or status information for all subscriptions.

pri_conn=
is a label identifying a primary database connection to request statistics or status information for that connection.

rep_conn=
is a label identifying a replicate database connection to request statistics or status information for that connection.

*conn_name*
is a connection name specified in the form *ds.db*, where:

- *ds* is the name of the data server on which the database resides.
- *db* is the name of the database.

sub=*sub_name*
is a label identifying a subscription to request statistics or status information for that subscription.

Examples

**Example 1**

```
sp_helprep
```

Returns current status of overall replication system.

---

**Note** Using sp_helprep with no keyword is functionally identical to specifying only the status keyword.

---

**Example 2**

```
sp_helprep stats, pri_conns
```

Returns current statistics for all primary database connections.

**Example 3**

```
sp_helprep stats, "pri_conn=boulder.doc"
```

Returns current statistics for the primary database connection to the database doc on data server boulder.

**Example 4**

```
sp_helprep stats, "sub=subdoc"
```

Returns current statistics for the subscription subdoc.

**Example 5**

```
sp_helprep status, "sub=subdoc"
```

Returns current status of the subscription subdoc.

**Example 6**

```
sp_helprep version
```

Returns the ASE Replicator version string.

Usage

- When listing statistics, the following result set is returned:

***Table 4-21: ASE Replicator statistics information***

| Column | Datatype | Description |
|---|---|---|
| type | varchar(3) | Type of entity: SYS – system |
| name | sysname | Name of the ASE Replicator instance |
| tstamp | datetime | Timestamp the statistic was generated |
| statistic | varchar(128) | Name of the statistic |
| value | varchar(255) | Value of the statistic |

- When listing status of the overall replication system, the following result set is returned:

***Table 4-22: ASE Replicator system status information***

| Column | Datatype | Description |
|---|---|---|
| type | varchar(32) | Type of object:<br>PRI – status of a primary connection<br>REP – status of a replicate connection<br>SUB – status of a subscription |

| Column | Datatype | Description |
|---|---|---|
| name | varchar(128) | Name of object:<br>if *type* = PRI or REP, *name* = ds.db<br>if *type* = SUB, *name* = subname |
| status | smallint | Status of object:<br>1 – up<br>0 – down<br>-1 – error |
| status_desc | varchar(255) | Description of status |

- When listing status of connections, the following result set is returned:

***Table 4-23: ASE Replicator connection status information***

| Column | Datatype | Description |
|---|---|---|
| conn_type | varchar(3) | Type of connection:<br>PRI – primary<br>REP – replicate |
| ds | sysname | If *conn_type* is PRI, then *ds* is the name of the local primary data server.<br>If *conn_type* is REP, then *ds* is the name of the replicate data server. |
| db | sysname | If *conn_type* is PRI, then *db* is the name of the local primary database.<br>If *conn_type* is REP, then *db* is the name of the replicate database. |
| maint_user | sysname | Maintenance User name in the database identified in *db*. |
| lastcmt_proxy | sysname | Lastcommit proxy table name. |
| restart_locator | varchar(64) | If *conn_type* is PRI, then *restart_locator* identifies the restart position in the database log.<br>If *conn_type* is REP, then *restart_locator* identifies the restart position in the stable queue. |
| timestamp | datetime | Timestamp of the restart locator. |
| status | smallint | Status of the connection:<br>1 – up<br>0 – down<br>-1 – error |
| status_desc | varchar(255) | Description of the connection status. |

- When listing status of subscriptions, the following result set is returned:

***Table 4-24: ASE Replicator subscription status information***

| Column | Datatype | Description |
|---|---|---|
| ds | sysname | Name of the replicate data server |
| db | sysname | Name of the replicate database |
| sub_name | sysname | Name of the subscription |
| status | smallint | Status of the subscription:<br>1 – up<br>0 – down<br>-1 – error |
| status_desc | varchar(255) | Status description |
| pub_name | sysname | Name of the associated publication |

- When you invoke sp_helprep with only the stats keyword specified, it returns statistics for the overall replication system.
  - When you invoke sp_helprep with the stats keyword and the pri_conns keyword, it returns statistics for all primary database connections.
  - When you invoke sp_helprep with the stats keyword and the rep_conns keyword, it returns statistics for all replicate database connections.
  - When you invoke sp_helprep with the stats keyword and the subs keyword, it returns statistics for all subscriptions on all replicate database connections.
  - When you invoke sp_helprep with the stats keyword and specify a connection (pri_conn=*conn_name* or rep_conn=*conn_name*), it returns statistics for the specified connection.
  - When you invoke sp_helprep with the stats keyword and specify a subscription (sub=*sub_name*), it returns statistics for the specified subscription.
- When you invoke sp_helprep with no keyword specified, it returns status of the overall replication system. Using sp_helprep with no keyword is functionally identical to specifying only the status keyword.
  - When you invoke sp_helprep with the status keyword and the pri_conns keyword, it returns status of all primary database connections.
  - When you invoke sp_helprep with the status keyword and the rep_conns keyword, it returns status of all replicate database connections.

- When you invoke sp_helprep with the status keyword and the subs keyword, it returns status of all subscriptions on all replicate database connections.
- When you invoke sp_helprep with the status keyword and specify a connection (pri_conn=*conn_name* or rep_conn=*conn_name*), it returns status of the specified connection.
- When you invoke sp_helprep with the status keyword and specify a subscription (sub=*sub_name*), it returns status of the specified subscription.

- When an error occurs, an error message is returned.

See also

sp_helpconn, sp_helpprimaryart, sp_helpprimaryconn, sp_helppub, sp_helpreplicateart, sp_helpreplicateconn, sp_helpsub

# sp_helpreplicateart

Description

Returns information about replicate articles.

Syntax

sp_helpreplicateart [info]

sp_helpreplicateart info, *rep_art* [, {conn=*conn_name*|sub=*sub_name*}]

sp_helpreplicateart fields, *rep_art*, {conn=*conn_name*|sub=*sub_name*}

Parameters

info
: is a keyword that requests information about all replicate articles in all replicate databases, or information about a specified replicate article in a specified replicate database or subscription.

fields
: is a keyword that requests field information for a specified replicate article in a specified replicate database or subscription.

*rep_art*
: is the name of a replicate article. Replicate article names can be specified in the form *owner.name*.

conn=*conn_name*
: is a label identifying a replicate database connection. The connection name (*conn_name*) must be specified in the form *ds.db*, where:

  - *ds* is the name of the data server on which the database resides.
  - *db* is the name of the database.

sub=*sub_name*
is a label identifying a subscription.

Examples

**Example 1**

```
sp_helpreplicateart
```

Returns information about all replicate articles in all replicate database connections.

**Note** Using sp_helpreplicateart with no keyword is functionally identical to specifying only the info keyword.

**Example 2**

```
sp_helpreplicateart @rep_art=table1
```

Returns information about all replicate articles named table1 in all replicate databases. This is functionally identical to Example 3.

**Example 3**

```
sp_helpreplicateart fields, table1, "conn=boulder.doc"
```

Returns field information about the replicate article table1 in the replicate database doc on the data server boulder.

**Example 4**

```
sp_helpreplicateart fields, "joe.table1", "sub=subdoc"
```

Returns field information about the replicate article table1 with owner joe in the subscription subdoc.

Usage

- When listing information about replicate articles, the following result set is returned:

***Table 4-25: ASE Replicator replicate article information***

| Column | Datatype | Description |
|---|---|---|
| ds | sysname | Name of the replicate data server. |
| db | sysname | Name of the replicate database. |
| sub_name | sysname | Subscription name. |
| rart_owner | sysname | Owner of replicate article. |
| rart_name | sysname | Name of replicate article. |
| rart_proc_num | smallint | Stored procedure group number for replicate article. If only one procedure exists, then *proc_num*=1. Tables are *proc_num*=0. |

| Column | Datatype | Description |
|---|---|---|
| art_type | varchar(9) | Type of replicate article:<br>table<br>procedure |
| proxy_table | sysname | If replicate article is a table, name of the proxy table. Otherwise, null. |
| dist_proc | sysname | Name of distribution stored procedure for the replicate article. |
| is_validated | varchar(64) | Validation flag:<br>true<br>false |
| where_clause | varchar(255) or varchar(1837) | Condition for selecting a subset of data rows to replicate. (Datatype size depends on Adaptive Server version.) |
| pub_name | sysname | Publication name. |
| part_owner | sysname | Owner of primary article. |
| part_name | sysname | Name of primary article. |
| part_proc_num | smallint | Stored procedure group number for primary article. If only one procedure exists, then *proc_num*=1. Tables are *proc_num*=0. |

- When listing information about replicate article fields, the following result set is returned:

***Table 4-26: ASE Replicator replicate article field information***

| Column | Datatype | Description |
|---|---|---|
| ds | sysname | Name of the replicate data server. |
| db | sysname | Name of the replicate database. |
| sub_name | sysname | Subscription name. |
| rart_owner | sysname | Owner of replicate article. |
| rart_name | sysname | Name of replicate article. |
| rart_proc_num | smallint | Stored procedure group number for replicate article. If only one procedure exists, then *proc_num*=1. Tables are *proc_num*=0. |
| art_type | varchar(9) | Type of replicate article:<br>table<br>procedure |
| rart_field_id | smallint | Replicate article field identifier, ordinal position. |
| rart_field_name | sysname | Name of the replicate article field (column or parameter name). |

| Column | Datatype | Description |
|---|---|---|
| datatype | sysname | Datatype of the field. |
| precision | int | Length of string or binary field, or precision of a numeric field, if applicable. Otherwise, 0. |
| scale | smallint | Scale of numeric field, if applicable. Otherwise, 0. |
| pub_name | sysname | Publication name. |
| part_owner | sysname | Owner of primary article. |
| part_name | sysname | Name of primary article. |
| part_proc_num | smallint | Stored procedure group number for primary article. If only one procedure exists, then *proc_num*=1. Tables are *proc_num*=0. |
| part_field_id | smallint | Primary article field identifier, ordinal position. |
| part_field_name | sysname | Name of the primary article field (column or parameter name). |

- When sp_helpreplicateart is invoked with no keyword, it returns information about all replicate articles in all replicate database connections.
- When you invoke sp_helpreplicateart with the fields keyword and an article name (*rep_art*), you must specify either a connection name (conn=*conn_name*) or a subscription name (sub=*sub_name*).
- To get information about primary articles, use sp_helpprimaryart.
- To get information about publications or subscriptions, use sp_helppub or sp_helpsub.
- To get information about database connections, use sp_helpconn, sp_helpprimaryconn, or sp_helpreplicateconn.
- When an error occurs, an error message is returned.

See also

sp_helpprimaryart, sp_helpreplicateconn, sp_helpsub

# sp_helpreplicateconn

Description

Returns information and statistics for replicate database connections.

Syntax

sp_helpreplicateconn [info]

sp_helpreplicateconn {info|stats} [, *conn_name*]

sp_helpreplicateconn {subs|empty|arts|params}, *conn_name*

Parameters

info
: is a keyword that requests information for replicate database connections.

stats
: is a keyword that requests statistics information for replicate database connections.

subs
: is a keyword that requests information about all subscriptions in the specified replicate database connection.

empty
: is a keyword that requests information about all empty subscriptions (subscriptions with no articles) in the specified replicate database connection.

arts
: is a keyword that requests information about all replicate articles in all subscriptions in the specified replicate database connection.

params
: is a keyword that requests information about all configuration parameters for the specified replicate database connection.

*conn_name*
: is the name of a replicate database connection. The connection name (*conn_name*) must be specified in the form *ds.db*, where:

- *ds* is the name of the data server on which the database resides.
- *db* is the name of the database.

Examples

**Example 1**

```
sp_helpreplicateconn
```

Returns information about all replicate database connections.

---

**Note** Using sp_helpreplicateconn with no keyword is functionally identical to specifying only the info keyword.

---

**Example 2**

```
sp_helpreplicateconn info, "boulder.doc"
```

Returns information about the replicate connection to the database doc on the data server boulder.

**Example 3**

```
sp_helpreplicateconn stats, "boulder.doc"
```

Returns statistics information about the replicate connection to the database doc on the data server boulder.

**Example 4**

```
sp_helpreplicateconn subs, "boulder.doc"
```

Returns information about all subscriptions in the replicate connection to the database doc on the data server boulder.

Usage

- When listing information about replicate connections, the following result set is returned:

***Table 4-27: ASE Replicator replicate connection information***

| Column | Datatype | Description |
|---|---|---|
| conn_type | varchar(3) | Type of connection: REP – replicate |
| ds | sysname | Name of the replicate data server |
| db | sysname | Name of the replicate database |
| maint_user | sysname | Maintenance User name in the database identified in *db* |
| lastcmt_proxy | sysname | Lastcommit proxy table name |
| restart_locator | varchar(64) | Identifies restart position in the stable queue |
| timestamp | datetime | Timestamp of the restart locator |
| status | smallint | Status of the connection:<br>1 – up<br>0 – down<br>-1 – error |
| status_desc | varchar(255) | Description of the connection status. |

- When listing statistics information about replicate connections, the following result set is returned:

***Table 4-28: ASE Replicator replicate connection statistics***

| Column | Datatype | Description |
|---|---|---|
| type | varchar(3) | Type of connection: REP – replicate |

| Column | Datatype | Description |
|---|---|---|
| name | sysname | Name of the remote replicate data server and database |
| tstamp | datetime | Timestamp the statistic was generated |
| statistic | varchar(128) | Name of the statistic |
| value | varchar(255) | Value of the statistic |

- When listing parameter information about replicate connections, the following result set is returned:

***Table 4-29: ASE Replicator replicate connection configuration***

| Column | Datatype | Description |
|---|---|---|
| parameter_name | varchar(128) | Name of the configuration parameter |
| default_value | varchar(255) | Default value of the parameter |
| legal_values | varchar(255) | Legal values of the parameter |
| description | varchar(255) | Description of the parameter |

- When listing information about subscriptions in replicate connections, the following result set is returned:

***Table 4-30: ASE Replicator replicate connection subscriptions***

| Column | Datatype | Description |
|---|---|---|
| ds | sysname | Name of the replicate data server |
| db | sysname | Name of the replicate database |
| sub_name | sysname | Subscription name |
| status | smallint | Status of the subscription |
| status_desc | varchar(255) | Description of the subscription status |
| pub_name | sysname | Publication name |

- When listing information about replicate articles in a replicate database connection, the following result set is returned:

***Table 4-31: ASE Replicator replicate articles***

| Column | Datatype | Description |
|---|---|---|
| ds | sysname | Name of the replicate data server. |
| db | sysname | Name of the replicate database. |
| sub_name | sysname | Subscription name. |
| rart_owner | sysname | Owner of replicate article. |
| rart_name | sysname | Name of replicate article. |

| Column | Datatype | Description |
|---|---|---|
| rart_proc_num | smallint | Stored procedure group number for replicate article. If only one procedure exists, then *proc_num*=1. Tables are *proc_num*=0. |
| art_type | varchar(9) | Type of replicate article:<br>table<br>procedure |
| proxy_table | sysname | If replicate article is a table, name of the proxy table. Otherwise, null. |
| dist_proc | sysname | Name of distribution stored procedure for the replicate article. |
| valid_pt | varchar(64) | Validation point (locator) that identifies where transactions for the replicate article begin in the database log. |
| where_clause | varchar(255) or varchar(1837) | Condition for selecting a subset of data rows to replicate. (Datatype size depends on Adaptive Server version.) |
| pub_name | sysname | Publication name. |
| part_owner | sysname | Owner of primary article. |
| part_name | sysname | Name of primary article. |
| part_proc_num | smallint | Stored procedure group number for primary article. If only one procedure exists, then *proc_num*=1. Tables are *proc_num*=0. |

- When sp_helpreplicateconn is invoked with no keyword, it returns information about all replicate connections.
- When you invoke sp_helpreplicateconn with no connection name (*conn_name*) specified, it returns information about all replicate database connections.
- When you invoke sp_helpreplicateconn and specify the info or stats keyword, the replicate database connection name (*conn_name*) is optional.
- When you invoke sp_helpreplicateconn and specify the subs, empty, arts, or params keyword, the replicate database connection name (*conn_name*) is required.
- To get information about all primary and replicate connections, use sp_helpconn.
- To get information about primary connections only, use sp_helpprimaryconn.
- When an error occurs, an error message is returned.

See also sp_helpprimaryconn, sp_helpreplicateart, sp_helpsub

# sp_helpsub

Description Returns information and statistics for subscriptions.

Syntax sp_helpsub [info|stats|empty]

sp_helpsub {info|stats|arts|unsub}, *sub_name*

Parameters

info
is a keyword that requests information for all subscriptions or for the specified subscription.

stats
is a keyword that requests statistics information for all subscriptions or for the specified subscription.

empty
is a keyword that requests information about all empty subscriptions (subscriptions with no articles) in all replicate database connections.

arts
is a keyword that requests information about all replicate articles in the specified subscription.

unsub
is a keyword that requests information about all primary articles in the corresponding publication for which there are no subscribers in the specified subscription.

*sub_name*
is the name of a subscription.

Examples **Example 1**

```
sp_helpsub
```

Returns information about all subscriptions in all replicate database connections.

---

**Note** Using sp_helpsub with no keyword is functionally identical to specifying only the info keyword.

---

**Example 2**

```
sp_helpsub info, subdoc
```

Returns information about the subscription subdoc.

**Example 3**

```
sp_helpsub stats, subdoc
```

Returns statistics information for the subscription subdoc.

Usage

- When listing information about subscriptions, the following result set is returned:

***Table 4-32: ASE Replicator subscription information***

| Column | Datatype | Description |
|---|---|---|
| ds | sysname | Name of the replicate data server |
| db | sysname | Name of the replicate database |
| sub_name | sysname | Subscription name |
| status | smallint | Status of the subscription |
| status_desc | varchar(255) | Description of the subscription status |
| pub_name | sysname | Publication name |

- When listing statistics information about subscriptions, the following result set is returned:

***Table 4-33: ASE Replicator subscription statistics***

| Column | Datatype | Description |
|---|---|---|
| type | varchar(3) | Type of entity: SUB – subscription |
| name | sysname | Name of the subscription |
| tstamp | datetime | Timestamp the statistic was generated |
| statistic | varchar(128) | Name of the statistic |
| value | varchar(255) | Value of the statistic |

- When listing information about replicate articles in subscriptions, the following result set is returned:

***Table 4-34: ASE Replicator subscription replicate articles***

| Column | Datatype | Description |
|---|---|---|
| ds | sysname | Name of the replicate data server. |
| db | sysname | Name of the replicate database. |
| sub_name | sysname | Subscription name. |
| rart_owner | sysname | Owner of replicate article. |
| rart_name | sysname | Name of replicate article. |

| Column | Datatype | Description |
|---|---|---|
| rart_proc_num | smallint | Stored procedure group number for replicate article. If only one procedure exists, then *proc_num*=1. Tables are *proc_num*=0. |
| art_type | varchar(9) | Type of replicate article:<br>table<br>procedure |
| proxy_table | sysname | If replicate article is a table, name of the proxy table. Otherwise, null. |
| dist_proc | sysname | Name of distribution stored procedure for the replicate article. |
| is_validated | varchar(64) | Validation flag:<br>true<br>false |
| where_clause | varchar(255) or varchar(1837) | Condition for selecting a subset of data rows to replicate. (Datatype size depends on Adaptive Server version.) |
| pub_name | sysname | Publication name. |
| part_owner | sysname | Owner of primary article. |
| part_name | sysname | Name of primary article. |
| part_proc_num | smallint | Stored procedure group number for primary article. If only one procedure exists, then *proc_num*=1. Tables are *proc_num*=0. |

- When listing information about primary articles not subscribed to by the specified subscription, the following result set is returned:

***Table 4-35: ASE Replicator subscription primary articles***

| Column | Datatype | Description |
|---|---|---|
| ds | sysname | Name of the local primary data server. |
| db | sysname | Name of the local primary database. |
| pub_name | sysname | Publication name. |
| owner | sysname | Owner of primary article. |
| part_name | sysname | Name of primary article. |
| proc_num | smallint | Stored procedure group number. If only one procedure exists, then *proc_num*=1. Tables are *proc_num*=0. |
| art_type | varchar(9) | Type of article:<br>table<br>procedure |

| Column | Datatype | Description |
|---|---|---|
| numsubscribers | int | Number of replicate articles that subscribe to this primary article. |
| shadow_table | sysname | Name of the shadow table for this primary article. |

- When sp_helpsub is invoked with no keyword, it returns information about all subscriptions for all primary database connections.
- When you invoke sp_helpsub and specify the info or stats keyword, the subscription name (*sub_name*) is optional.
- When you invoke sp_helpsub and specify the empty keyword, you cannot specify a subscription name (*sub_name*).
- When you invoke sp_helpsub and specify the arts or unsub keyword, the subscription name (*sub_name*) is required.
- When an error occurs, an error message is returned.

See also

sp_helppub, sp_helpreplicateart, sp_helpreplicateconn

# sp_materializesub

Description

Materializes and validates all replicate articles in a specified subscription, or a specified replicate article in a specified subscription.

Syntax

sp_materializesub *sub_name* [, *rep_art*]

Parameters

*sub_name*
is the name of the subscription containing replicate articles to materialize and validate.

*rep_art*
is the name of a replicate article to materialize and validate. Replicate article names can be specified in the form *owner.name*.

Examples

```
sp_materializesub subdoc, repdoc
```

Materializes and validates the replicate article repdoc in the subscription subdoc.

Usage

- Before you invoke sp_materializesub, you must suspend the subscription using sp_suspendsub. After the replicate article (or articles) is materialized and validated, you can resume the subscription using sp_resumesub.
- When a replicate article for a table is materialized, data in the primary object identified by the primary article to which the replicate article subscribes is copied to the replicate table using an insert into ... select from command, based on the subscribed fields and the where clause specified in the replicate article (if applicable).
- When a replicate article is validated, the Publisher component places a marker in the Adaptive Server transaction log, indicating the point at which transaction distribution for that replicate article should begin. Any transactions to which the replicate article subscribes that occur prior to the marker location in the transaction log are not distributed to the replicate object identified by the replicate article.
- If you invoke sp_materializesub to materialize a replicate article for a stored procedure, the subscription for that article is validated only, and not materialized. Replicate articles for stored procedures need not be materialized.
- When you invoke sp_materializesub and specify only a subscription (*sub_name*), all replicate articles in the specified subscription are materialized (tables only) and validated (both tables and stored procedures).
- When you invoke sp_materializesub and specify a subscription (*sub_name*) and a replicate article (*rep_art*), the specified replicate article in the specified subscription is materialized (table only) and validated (either table or stored procedure).
- As an alternative to sp_materializesub, you can validate a replicate article for a stored procedure using sp_validatesub.
- When the requested action occurs successfully, no results are returned.
- When an error occurs, an error message is returned.

See also

sp_addreplicateart, sp_validatesub

# sp_resumeprimaryconn

Description — Resumes all primary database connections or a specified primary database connection.

Syntax — sp_resumeprimaryconn [*conn_name*]

Parameters — *conn_name*
is the name of a primary database connection. The connection name (*conn_name*) must be specified in the form *ds.db*, where:

- *ds* is the name of the data server on which the primary database resides.
- *db* is the name of the primary database.

Examples

```
sp_resumeprimaryconn "boulder.doc"
```

Resumes the primary connection to the database doc on the data server boulder.

Usage
- When you invoke sp_resumeprimaryconn without specifying a connection name (*conn_name*), it resumes all primary database connections.
- When you invoke sp_resumeprimaryconn and specify a connection name (*conn_name*), it resumes the primary connection to the specified database.
- When the requested action occurs successfully, no results are returned.
- When an error occurs, an error message is returned.

See also — sp_helpprimaryconn, sp_suspendprimaryconn

# sp_resumerep

Description — Resumes all ASE Replicator components, database connections, and subscriptions that are suspended.

Syntax — sp_resumerep

Examples

```
sp_resumerep
```

Resumes all ASE Replicator replication processes.

Usage
- Resuming replication starts all the ASE Replicator replication system processing.

- When you invoke sp_resumerep, ASE Replicator components perform the following actions:
  - The Publisher component evaluates the last saved locator value and starts scanning the primary database's native Adaptive Server transaction log at the point indicated by the locator value.
  - The Distributor component evaluates the last saved locator value and starts scanning the stable queue at the point indicated by the locator value.
- To suspend all ASE Replicator components, database connections, and subscriptions, use sp_suspendrep.
- When the requested action occurs successfully, no results are returned.
- When an error occurs, an error message is returned.

See also sp_suspendrep

# sp_resumereplicateconn

Description Resumes all replicate database connections or a specified replicate database connection.

Syntax sp_resumereplicateconn [*conn_name*]

Parameters *conn_name*
is the name of a replicate database connection. The connection name (*conn_name*) must be specified in the form *ds.db*, where:
- *ds* is the name of the data server on which the replicate database resides.
- *db* is the name of the replicate database.

Examples

```
sp_resumereplicateconn "boulder.doc"
```

Resumes the replicate connection to the database doc on the data server boulder.

Usage
- When you resume a replicate database connection with the sp_resumereplicateconn procedure, it resumes all subscriptions associated with that replicate database connection.

- When you invoke sp_resumereplicateconn without specifying a connection name (*conn_name*), it resumes all replicate database connections.
- When you invoke sp_resumereplicateconn and specify a connection name (*conn_name*), it resumes the replicate connection to the specified database.
- When the requested action occurs successfully, no results are returned.
- When an error occurs, an error message is returned.

See also sp_helpreplicateconn, sp_suspendreplicateconn

# sp_resumesub

Description Resumes all subscriptions or a specified subscription.

Syntax sp_resumesub [*sub_name*]

Parameters *sub_name*
is the name of the subscription to resume.

Examples

```
sp_resumesub subdoc
```

Resumes the subscription subdoc.

Usage
- When a subscription is resumed, the Distributor component evaluates the last saved locator value for that subscription, starts scanning the stable queue at the point indicated by the locator value, and starts sending processed transactions to the proxy tables for the replicate site identified by the replicate connection associated with the subscription.
- When sp_resumesub is invoked with no subscription (*sub_name*) specified, it resumes all subscriptions for all replicate database connections.
- When sp_resumesub is invoked with a subscription (*sub_name*) specified, it resumes the specified subscription.
- When the requested action occurs successfully, no results are returned.
- When an error occurs, an error message is returned.

See also sp_helpsub, sp_suspendsub

# sp_shutdownrep

Description — Shuts down the ASE Replicator process.

Syntax — sp_shutdownrep [immediate]

Parameters — immediate
is the keyword for shutting down the ASE Replicator process immediately.

Examples

```
sp_shutdownrep immediate
```

Shuts down the ASE Replicator process immediately, without quiescing the system or flushing any queues.

Usage

- When you invoke sp_shutdownrep with no keyword, ASE Replicator shuts down gracefully by first emptying its internal queues and completing any current but uncommitted transactions, then suspending all database connections and subscriptions.
- When you invoke sp_shutdownrep with the immediate keyword, ASE Replicator rolls back any work in progress and then shuts down.
- When the requested action occurs successfully, no results are returned.
- When an error occurs, an error message is returned.

See also — sp_suspendrep

# sp_suspendprimaryconn

Description — Suspends all primary database connections or a specified primary database connection.

Syntax — sp_suspendprimaryconn [*conn_name*]

Parameters — *conn_name*
is the name of a primary database connection. The connection name (*conn_name*) must be specified in the form *ds.db*, where:

- *ds* is the name of the data server on which the primary database resides.
- *db* is the name of the primary database.

Examples

```
sp_suspendprimaryconn "boulder.doc"
```

Suspends the primary connection to the database doc on the data server boulder.

Usage

- When you invoke sp_suspendprimaryconn with no connection name (*conn_name*) specified, it suspends all primary connections.
- When you invoke sp_suspendprimaryconn and specify a connection name (*conn_name*), it suspends the primary connection to the specified database.
- When the requested action occurs successfully, no results are returned.
- When an error occurs, an error message is returned.

See also sp_helpprimaryconn, sp_resumeprimaryconn

# sp_suspendrep

Description Suspends all ASE Replicator processing (log reading and distribution), database connections, and subscriptions, but does not shut down the ASE Replicator process or its components.

Syntax sp_suspendrep

Examples

```
sp_suspendrep
```

Suspends all ASE Replicator replication processing.

Usage

- Suspending the replication system rolls back any current operations and stops all replication system processing immediately.
- When you invoke sp_suspendrep, ASE Replicator components perform the following actions:
  - The Publisher component stops scanning the primary database's native ASE transaction log, discards all transactions in its internal queue, and rolls back any incomplete transactions it is processing in the stable queue.
  - The Distributor component rolls back any of its transactions being processed in the replicate database and stops scanning the stable queue.

- To resume replication operations after suspending, use sp_resumerep.
- When the requested action occurs successfully, no results are returned.
- When an error occurs, an error message is returned.

See also sp_resumerep

# sp_suspendreplicateconn

Description Suspends all replicate database connections or a specified replicate database connection.

Syntax sp_suspendreplicateconn [*conn_name*]

Parameters *conn_name*
is the name of a replicate database connection. The connection name (*conn_name*) must be specified in the form *ds.db*, where:

- *ds* is the name of the data server on which the replicate database resides.
- *db* is the name of the replicate database.

Examples

```
sp_suspendreplicateconn "boulder.doc"
```

Suspends the replicate connection to the database doc on the data server boulder.

Usage
- When you suspend a replicate database connection with sp_suspendreplicateconn, it suspends all subscriptions associated with that replicate database connection.
- When you invoke sp_suspendreplicateconn with no connection name (*conn_name*) specified, it suspends all replicate connections.
- When you invoke sp_suspendreplicateconn and specify a connection name (*conn_name*), it suspends the replicate connection to the specified database.
- When the requested action occurs successfully, no results are returned.
- When an error occurs, an error message is returned.

See also sp_helpreplicateconn, sp_resumereplicateconn

# sp_suspendsub

Description — Suspends a specified subscription.

Syntax — sp_suspendsub *sub_name*

Parameters — *sub_name*
is the name of the subscription to suspend.

Examples

```
sp_suspendsub subdoc
```

Suspends the subscription subdoc.

Usage

- When a subscription is suspended, all distribution of data to the replicate objects identified in the subscription stops, and the Distributor component records a locator value for that subscription, which it uses to determine where to start scanning in the stable queue when the subscription is resumed.
- When sp_suspendsub is invoked, it suspends the specified subscription (*sub_name*).
- When the requested action occurs successfully, no results are returned.
- When an error occurs, an error message is returned.

See also — sp_helpsub, sp_resumesub

# sp_tracerep

Description — Turns ASE Replicator trace flags on or off, lists all ASE Replicator trace flags and their status.

Syntax — sp_tracerep [*flag*|all_flags [, {true|false}]]

Parameters — *flag*
is the name of an ASE Replicator trace flag.

Table 4-36 lists all the ASE Replicator trace flags:

***Table 4-36: ASE Replicator trace flags***

| Flag | Type | Description |
|---|---|---|
| BMGRTRACE | TRACE | Bean Management trace events. |
| CONNTRC | TRACE | Traces ASE Replicator connection management. |

| Flag | Type | Description |
|---|---|---|
| DDBLOGTRC | TRACE | Trace Distribution Database transaction log operations. |
| DDBLRTRC | TRACE | Trace Distribution Database transaction log reader operations. |
| DISTTRC | TRACE | Trace distribution activities. |
| ERROR | SYSTEM | Serious error. Manual intervention may be required for recovery. |
| FATAL | SYSTEM | Critical error. The system cannot function without manual intervention. |
| INFORMATION | SYSTEM | Important information. No action required. |
| LATRC | TRACE | Trace DDBAdmin events. |
| LATRCSQL | TRACE | Trace DDBAdmin SQL execution. |
| LOBTRACE | TRACE | Trace LOB Replication activities. |
| LRTRACE | TRACE | Trace transaction log reader operations. |
| LWTRACE | TRACE | Trace distribution log write operations. |
| RACONTRC | TRACE | Traces connection and query execution. |
| RACONTRCSQL | TRACE | Traces SQL statements to be executed. |
| RPLTRC | TRACE | Provides general ASE Replicator trace information. |
| STATTRC | TRACE | Trace statistics operation. |
| SUBPROCTRC | TRACE | Trace subscription processor operations. |
| WARNING | SYSTEM | The system has suffered a minor problem. Functionality is not affected or problem is recoverable. |

all_flags
: is a keyword that indicates that all trace flags should be turned on or off.

true
: is a keyword that indicates that the trace flag should be turned on.

false
: is a keyword that indicates that the trace flag should be turned off.

Examples

**Example 1**

```
sp_tracerep LATRC
```

Returns information about the ASE Replicator LATRC trace flag.

**Example 2**

```
sp_tracerep LATRC, true
```

Turns on tracing for the ASE Replicator LATRC trace flag.

**Example 3**

```
sp_tracerep all_flags, true
```

Turns on tracing for all ASE Replicator trace flags.

Usage

- When listing information about ASE Replicator trace flags, the following result set is returned:

***Table 4-37: ASE Replicator trace flag information***

| Column | Datatype | Description |
|---|---|---|
| trace_flag | varchar(32) | Name of the trace flag |
| current_value | varchar(5) | Current value of the trace flag:<br>true - flag is on<br>false - flag is off |
| trace_file | varchar(6) | Type of flag and the output file:<br>system<br>trace<br>debug |
| description | varchar(255) | Description of the trace flag |

- When sp_tracerep is invoked without specifying an ASE Replicator trace flag (*flag*), it returns information about all ASE Replicator trace flags.
- When sp_tracerep is invoked with an ASE Replicator trace flag (*flag*), it returns information about the specified trace flag.
- When sp_tracerep is invoked with an ASE Replicator trace flag (*flag*) and a keyword (true or false), it sets the specified trace flag to the value specified.
- Trace flags listed in Table 4-36 on page 144 as SYSTEM flags cannot be turned off.
- When sp_tracerep changes the specified ASE Replicator trace flag setting, no results are returned.
- When an error occurs, an error message is returned.

See also

sp_helpqueue, sp_helpreperrors

# sp_validatesub

Description

Validates a subscription.

Syntax

sp_validatesub *sub_name* [, *rep_art*]

Parameters

*sub_name*
is the name of the subscription that contains replicate articles to validate.

*rep_art*
is the name of a replicate article to validate.

Examples

```
sp_validatesub subdoc, repdoc
```

Validates the replicate article repdoc in the subscription subdoc.

Usage

- Before you invoke sp_validatesub, you must suspend the subscription using sp_suspendsub. After the replicate article is validated, you can resume the subscription using sp_resumesub.
- When a replicate article is validated, the Publisher component places a marker in the stable queue indicating the point at which transaction distribution for that replicate article should begin. Any transactions to which the replicate article subscribes that occur prior to the marker in the stable queue are not distributed to the replicate object identified in the replicate article.
- To be ready for replication, replicate articles for tables must be materialized using sp_materializesub, unless the table already contains data synchronized with the primary database. If you choose not to use sp_materializesub to materialize the data in the replicate table, you must copy the appropriate data from the primary table to the replicate table to prepare for replication.
- If you invoke sp_validatesub to validate a replicate article for a table, the subscription for that article is validated only, and not materialized.
- To be ready for replication, replicate articles for stored procedures need only be validated using sp_validatesub.
- When you invoke sp_validatesub and specify only a subscription (*sub_name*), all replicate articles in the specified subscription are validated.
- When you invoke sp_validatesub and specify a subscription (*sub_name*) and a replicate article (*rep_art*), the specified replicate article in the specified subscription is validated.

- When the requested action occurs successfully, no results are returned.
- When an error occurs, an error message is returned.

See also sp_materializesub

CHAPTER 5

# Troubleshooting ASE Replicator

This chapter describes how to solve common ASE Replicator problems.


| Topic | Page |
|---|---|
| Problems that prevent replication from starting | 149 |
| Problems that cause ongoing replication to fail | 153 |


## Problems that prevent replication from starting

Problems that prevent replication from starting are usually setup or configuration problems, such as:

- Primary server configuration problems
- Remote server configuration problems
- ASE Replicator configuration problems
- User login or permission problems

### Primary server configuration problems

The first step in setting up ASE Replicator is configuring the primary database server—the Adaptive Server on which the Distribution Database and the primary database (or databases) reside. If the primary Adaptive Server is not configured as described in Chapter 2, “Setting Up and Starting ASE Replicator,” you cannot start replication.

Some typical symptoms of primary Adaptive Server configuration problems are:

- The ASE Replicator process fails to start up, or shuts down immediately after starting.
- You cannot log in to the Adaptive Server using the ASE Replicator system user name.

- The ASE Replicator process starts and runs and you can log in to the Adaptive Server, but all ASE Replicator procedures, including sp_helprep, always return an error.
- In the Sybase Central window, the ASE Replicator icon does not appear under the primary Adaptive Server icon in the left pane.
- You cannot create ASE Replicator objects, such as database connections, publications and subscriptions, or primary and replicate articles.

❖ **To correct Adaptive Server configuration problems**

- See "Configuring the primary Adaptive Server" on page 16 and verify that all the configuration tasks described in that section are completed.

## Remote server configuration problems

Part of setting up ASE Replicator is configuring each remote server on which a replicate database resides. If a remote server is not configured as described in Chapter 2, "Setting Up and Starting ASE Replicator," you cannot start replication to that server.

---

**Note** If you add a new remote server or replicate database to an existing replication system, you must follow the procedures in Chapter 2, "Setting Up and Starting ASE Replicator," to configure that remote server or replicate database.

---

Some typical symptoms of remote server configuration problems are:

- You cannot create ASE Replicator replicate objects, such as a replicate database connection, subscription, or replicate article.
- Using the sp_addreplicateconn procedure to create a replicate database connection returns an error.
- Using the sp_addreplicateart procedure to create a replicate article returns an error.
- Using a replicate object "help" procedure, such as sp_helpreplicateconn always returns an error.

❖ **To correct remote server configuration problems**

- See "Configuring the replicate servers and databases" on page 22 and verify that all the configuration tasks described in that section are completed.

## ASE Replicator configuration problems

If ASE Replicator is not configured as described in Chapter 2, "Setting Up and Starting ASE Replicator," you cannot start replication.

Some typical symptoms of ASE Replicator configuration problems include:

- The ASE Replicator process fails to start up, or shuts down immediately after starting.

  **Note** If you do not specify *all* of the required parameters when you invoke the aserep script, the ASE Replicator process may start up and shut down immediately with an error.

- The ASE Replicator process starts and runs, but all ASE Replicator procedures, including sp_helprep, always return an error.
- In the Sybase Central window, the ASE Replicator icon does not appear under the primary Adaptive Server icon in the left pane.
- You cannot create ASE Replicator objects, such as database connections, publications and subscriptions, or primary and replicate articles.

❖ **To correct ASE Replicator configuration problems**

- See "Initializing the ASE Replicator process" on page 26 and make sure that the ASE Replicator configuration meets the requirements described in that section.

## User login or permission problems

ASE Replicator requires a user login on its host Adaptive Server (the primary database server). That login is referred to as the ASE Replicator system user login. ASE Replicator also requires a user login on each remote database server with which it connects. That login is referred to as the Maintenance User login.

ASE Replicator requires a user login added to each primary database and each replicate database that ASE Replicator connects with. That user login must have the appropriate object access permissions in each database.

To allow bidirectional replication, in which a primary database also acts as a replicate database, ASE Replicator identifies a Maintenance User for each database. ASE Replicator uses the Maintenance User login to apply replicated transactions to a replicate database. When publishing the transactions from a primary database, ASE Replicator filters out any transactions applied by the Maintenance User in that database.

If you use the same login name and password for all ASE Replicator database connections, user login and permission problems are less likely to occur, but using such a common password may not be feasible in a bidirectional replication system, and it may not meet your database security requirements.

Some typical symptoms of user login or permission problems are:

- The aserep script fails to create a valid ASE Replicator instance.
- The ASE Replicator process fails to start up, or shuts down immediately after starting.
- The ASE Replicator process starts up, but all ASE Replicator procedures, including sp_helprep, always return an error.
- You cannot create ASE Replicator objects, such as database connections, publications and subscriptions, or primary and replicate articles.

❖ **To correct user login or permission problems**

1 See Chapter 2, “Setting Up and Starting ASE Replicator,” and verify that all the configuration tasks are completed.

2 Verify that the ASE Replicator system user login (or Maintenance User login) is:

- Valid in the primary data server and in each replicate data server
- Added to each primary and replicate database
- Granted appropriate permissions in each primary and replicate database

# Problems that cause ongoing replication to fail

Problems that cause ongoing replication to fail are usually related to changes in the environment, either intentional changes such as changing the schema of a table, or unintentional changes such as failures of software, hardware, or network infrastructure.

Some problems with datatypes, constraints, and column properties might cause ongoing replication to fail, or in some cases, prevent replication from starting:

- Under some conditions, approximate numeric datatypes float, double precision, and real can cause ASE Replicator errors, and the timestamp datatype cannot be replicated at all.
- Replicating columns with the IDENTITY property can cause ASE Replicator errors.
- Datatype, primary key constraint, and null-handling inconsistencies between primary and replicate tables can cause ASE Replicator errors.

## Schema changes in primary or replicate tables

ASE Replicator does not support schema caching. Therefore, if you alter a published primary object (table or stored procedure) in a way that changes the object's entry in the sysobjects table, ASE Replicator will not be able to find transactions for the altered object in the Adaptive Server transaction log.

See the Adaptive Server *Reference Manual* for more information on object changes that affect the sysobjects table.

The following sections describe the procedures to use when you need to alter a primary or replicate object while replication is in progress.

### Altering a published primary object

Use the following procedure if you need to alter a published primary object (table or stored procedure) while replication is in progress:

❖ **To alter a published object in the primary database**

1 Suspend any subscriptions that contain a replicate article that subscribes to the primary article for the object you need to alter.

2 Delete any replicate articles that subscribe to that primary article.

3 Suspend the primary connection and delete the primary article from all publications it belongs to.

4 Alter the table or stored procedure in the primary database.

5 Create a new primary article to publish the altered object, and add that primary article to any publications necessary.

6 Resume the primary connection.

7 Create a new replicate article to subscribe to the new primary article in each subscription necessary.

8 Materialize or validate the new replicate article (or articles).

9 Resume any subscription that contains a new replicate article.

### Altering a replicate object

Use the following procedure if you need to alter a replicate object (table or stored procedure) while replication is in progress:

❖ **To alter a replicate object**

1 Suspend the subscription that contains the replicate article that identifies the replicate object you need to alter.

2 Delete the replicate article.

3 Alter the table or stored procedure in the replicate database.

4 Create a new replicate article to identify the replicate object.

5 Materialize or validate the new replicate article.

6 Resume the subscription that contains the replicate article.

## Datatypes, constraints, and column properties

ASE Replicator has certain limitations regarding datatypes, constraints, and column properties:

- Approximate numeric datatypes float, double precision, and real might cause ASE Replicator errors in the following situations:
  - Replicating an update or delete operation that changes a column with an approximate numeric datatype

    - A column with an approximate numeric datatype also has a primary key constraint

  For more information about approximate numeric datatypes, see "Replicating approximate numeric datatypes" on page 155.

- The timestamp datatype cannot be replicated.

  For more information about the timestamp datatype, see "Replicating the value of the timestamp datatype" on page 156.

- Replicating columns with the IDENTITY property can cause ASE Replicator errors.

  For more information about the IDENTITY property, see "Replicating IDENTITY columns" on page 156.

- The following inconsistencies between primary and replicate table columns might cause ASE Replicator errors:

  - Datatypes (see "Incompatible datatypes" on page 157)
  - Primary key constraints (see "Incompatible primary key constraints" on page 158)
  - Null type (see "Different null types" on page 158)

## Replicating approximate numeric datatypes

Approximate numeric datatypes include float, double precision, and real. The exact value of an approximate numeric datatype can vary from one platform to another, and this can cause replication errors.

Replicating an update or delete operation with an approximate numeric datatype causes an error if both of the following conditions are true:

- The corresponding values on the primary and replicate data servers are not identical, and
- No primary key constraint is defined for the table.

When no primary key constraint is defined, all columns in the table are the primary key, and any variation in the value of any column between the two databases causes an error.

**Note** For the same reason, errors occur if any column with a primary key constraint has an approximate numeric datatype.

To prevent replication problems with approximate numeric datatypes, you must declare a primary key constraint in the primary table, and no column identified in the primary key constraint can have an approximate numeric datatype.

## Replicating the value of the timestamp datatype

The user-defined timestamp datatype is based on the varbinary datatype, and its value is automatically generated by Adaptive Server when a row is inserted or updated. A timestamp column cannot be modified by an insert or update command, so therefore, ASE Replicator does not support replication to a timestamp column in a replicate database.

If a replicate article subscribes to a published timestamp column, and the replicate table has a corresponding timestamp column, ASE Replicator returns an error the first time it attempts to replicate a transaction to that table.

---

**Note** To replicate the actual data value from a primary timestamp column, you must create or alter the replicate table so that the corresponding column is varbinary.

---

When you invoke sp_addreplicateart and ASE Replicator creates a replicate table, the replicate column that corresponds to a published timestamp column is created with a varbinary datatype.

## Replicating IDENTITY columns

When ASE Replicator sends a transaction to a replicate table with an IDENTITY column, the data value in the primary IDENTITY column is not replicated. ASE Replicator allows the remote (replicate) Adaptive Server to update or insert the IDENTITY value in the replicate table.

Turning on the IDENTITY_INSERT or IDENTITY_UPDATE query-processing option does not affect ASE Replicator behavior.

---

**Note** To replicate the actual data value from a primary IDENTITY column, you must create or alter the replicate table so that the corresponding column does not have the IDENTITY property.

---

When you invoke sp_addreplicateart and ASE Replicator creates a replicate table, the replicate column that corresponds to a published IDENTITY column is created with the IDENTITY property.

## Incompatible datatypes

The corresponding fields (columns or parameters) of primary and replicate objects must have compatible datatypes and length. Ideally, the datatype and length of a published field should be the same as the datatype and length of the corresponding field in a replicate object.

---

**Note** If a column is identified in the replicate table's primary key constraint, errors can result if its datatype is not identical to the datatype of the corresponding primary table column.

---

When the datatype and length of a replicate object's field is not compatible with the published datatype, errors occur.

To find the cause of a datatype incompatibility, you must check the mapping between the primary article published fields and the replicate article subscribed fields.

For example, if a primary article publishes the following four fields:

- column1 – bit
- column2 – varchar(64)
- column3 – smallint
- column4 – varchar(255)

and a replicate article subscribes to published fields 2, 3, and 4, the first three columns in the replicate object should be defined with the corresponding datatypes:

- column1 – varchar(64)
- column2 – smallint
- column3 – varchar(255)

As an alternative, the first three columns in the replicate object may be defined with different, but compatible datatypes, such as:

- column1 – varchar(128)
- column2 – int
- column3 – varchar(1024)

**Note** The published fields that a replicate article subscribes to are always mapped to the *first available* fields (columns or parameters) in the replicate object.

## Incompatible primary key constraints

The most common primary key problems are:

- The replicate table's primary key constraint does not identify the same columns as the primary table's primary key.
- The replicate table has a primary key constraint, but the replicate article does not subscribe to all the columns in the primary table's primary key.
- The replicate table has a primary key constraint, but the primary table does not.
- The replicate table's primary key constraint includes a column that allows null values in the primary table.
- The datatype of a primary key column in the replicate table is different than the datatype in the corresponding primary table column.

All these problems can allow a replicated operation that attempts to create a row with non-unique data in the replicate table's primary key column (or columns).

To avoid primary key problems in a replication system, you can either remove the primary key constraint on the replicate table, or:

- Make sure that the primary table and replicate table identify the same columns for their primary key constraints.
- Make sure that the replicate article subscribes to all the primary table's primary key columns.
- Make sure that both primary and replicate tables use the same datatype (and length) for all replicated primary key columns.

## Different null types

If a column in the primary table allows null values and the corrresponding column in the replicate table does not, an error occurs when ASE Replicator attempts to insert a null value in the replicate table's column.

To avoid this problem, make sure that each pair of corresponding columns in the primary and replicate tables is defined with the same null type.

APPENDIX A

# Distribution Database Schema

This appendix describes the ASE Replicator Distribution Database schema.


| Topic | Page |
|---|---|
| Distribution Database base objects | 161 |
| Distribution Database shadow tables | 172 |
| Distribution Database replicate objects | 176 |


## Distribution Database base objects

Base objects are created in the Distribution Database when you initialize the ASE Replicator process using the aserep script.

There are two types of Distribution Database base objects:

- Tables
- Procedures

In addition to the base objects, ASE Replicator creates user-defined datatypes for use by Distribution Database objects.

### Distribution Database datatypes

ASE Replicator creates six user-defined datatypes in the Distribution Database. These datatypes are used by Distribution Database procedures and in Distribution Database tables.

Table A-1 lists the Distribution Database user-defined datatypes, along with their underlying standard Adaptive Server datatypes, and a brief description.

***Table A-1: Distribution Database datatypes***

| Distribution Database datatype | Adaptive Server datatype | Description |
|---|---|---|
| rpl_sysname | sysname | Allows sysname to be used in tables |
| rpl_boolean | bit | Provides a generic Boolean datatype |
| rpl_blob | bit | Indicates image column value change |
| rpl_clob | bit | Indicates text column value change |
| rpl_uclob | bit | Indicates unitext column value change |
| rpl_java | bit | Indicates java column value change |

The rpl_blob, rpl_clob, rpl_uclob, and rpl_java datatypes are used in shadow tables as Boolean datatypes.

## Base tables in the Distribution Database

Base tables exist before you create any ASE Replicator objects, such as database connections, publications or subscriptions, or primary or replicate articles.

Table A-2 lists all the Distribution Database base tables.

***Table A-2: Distribution Database base tables***

| Table name | Description |
|---|---|
| ddb_system | Distribution Database system table |
| connections | Connections table |
| conn_properties | Connection properties table |
| publications | Publications table |
| pri_articles | Primary articles table |
| part_fields | Primary article fields table |
| part_pub_relation | Primary articles/publications relation table |
| subscriptions | Subscriptions table |
| rep_articles | Replicate articles table |
| rart_fields | Replicate article fields table |
| tran_log | Transaction log table |
| config | Configuration table |
| trace | Trace table |
| rpl_statistics | Statistics table |

Distribution Database base tables are described in the following sections.

## Distribution Database system table

The ddb_system table keeps track of other Distribution Database base objects and housekeeping items.

Table A-3 lists the columns in the ddb_system table.

***Table A-3: Distribution Database system table***

| Column name | Datatype | Description |
|---|---|---|
| selector | varchar(64) | Identifier of the object |
| value | varchar(255) | Name or value of the object |
| type | smallint | Type of the object |

Each row in the ddb_system table identifies a single Distribution Database object or housekeeping item.

## Connections table

The connections table stores information about ASE Replicator database connections.

Table A-4 lists the columns in the connections table.

***Table A-4: Connections table***

| Column name | Datatype | Description |
|---|---|---|
| conn_id | numeric(18,0) | Connection ID. |
| conn_type | smallint | Type of connection (primary or replicate). |
| ds | rpl_sysname | Name of the database server. |
| db | rpl_sysname | Name of the database. |
| lastcommit_proxy | rpl_sysname | Name of lastcommit proxy table (replicate connections only). |
| maint_user | rpl_sysname | Maintenance User login name. |
| reset_extlogin | rpl_boolean | Indicates whether ASE Replicator created the Maintenance User as an external login. |
| log_locator | varchar(70) | If primary connection, log_locator identifies the restart position in the database log.<br>If replicate connection, log_locator identifies the restart position in the stable queue. |
| status | smallint | Status of the connection. |

| Column name | Datatype | Description |
|---|---|---|
| status_desc | varchar(255) | Status description. |

Each row in the connections table identifies a single database connection.

When you create a primary or replicate database connection, a row is added to the connections table to identify the new connection.

When you delete a primary or replicate database connection, the row identifying that connection is deleted from the connections table.

## Connection properties table

The conn_properties table stores all connection configuration parameter values for all ASE Replicator database connections.

See "Connection configuration parameters" on page 78 for information about connection configuration parameters.

Table A-5 lists the columns in the conn_properties table.

***Table A-5: Connection properties table***

| Column name | Datatype | Description |
|---|---|---|
| conn_id | numeric(18,0) | Connection ID |
| property | varchar(128) | Name of connection configuration parameter |
| value | varchar(255) | Value of connection configuration parameter |

Each row in the conn_properties table identifies the value of a single connection configuration parameter for a specific database connection.

When you create a primary or replicate database connection, a row is added to the conn_properties table for each configuration parameter for the new connection.

When you set or change the value of a connection configuration parameter, the row that identifies the value of the specified configuration parameter for the specified database connection is updated.

When you delete a primary or replicate database connection, a row is deleted from the conn_properties table for each configuration parameter for the deleted connection.

## Publications table

The publications table stores information for all ASE Replicator publications.

Table A-6 lists the columns in the publications table.

***Table A-6: Publications table***

| Column name | Datatype | Description |
|---|---|---|
| pubid | numeric(18,0) | Publication ID |
| name | varchar(128) | Name of publication |
| conn_id | numeric(18,0) | ID of the primary connection this publication belongs to |

Each row in the publications table identifies a single publication.

When you create a publication, a row is added to the publications table for the new publication.

When you delete a publication, the row that identifies that publication is deleted from the publications table.

## Primary articles table

The pri_articles table stores information about all ASE Replicator primary articles.

Table A-7 lists the columns in the pri_articles table.

***Table A-7: Primary articles table***

| Column name | Datatype | Description |
|---|---|---|
| part_id | numeric(18,0) | Primary article ID |
| part_name | rpl_sysname | Name of primary article |
| conn_id | numeric(18,0) | Connection ID |
| art_type | smallint | Type of primary article (table or procedure) |
| shadow | rpl_sysname | Name of primary article shadow table |
| rarts_count | int | Number of replicate articles subscribing to this primary article |
| owner | rpl_sysname | Owner of the primary object |
| proc_num | smallint | Stored procedure group number |

Each row in the pri_articles table identifies a single primary article.

When you create a primary article, a row is added to the pri_articles table for the new article.

When you create or delete a replicate article, the row that identifies the primary article subscribed to by that replicate article is updated in the pri_articles table to change the value of the rarts_count column.

When you delete a primary article, the row that identifies that primary article is deleted from the pri_articles table.

## Primary article fields table

The part_fields table stores information about all published fields for all primary articles.

Table A-8 lists the columns in the part_fields table.

***Table A-8: Primary article fields table***

| Column name | Datatype | Description |
|---|---|---|
| part_id | numeric(18,0) | Primary article ID |
| part_colid | int | Column ID of primary article field |
| part_colname | rpl_sysname | Column name of primary article field |
| is_identity | rpl_boolean | Indicates whether the column is an identity column |
| is_lob | rpl_boolean | Indicates whether the column is a large object datatype |
| is_null | rpl_boolean | Indicates whether the column can have a null value |
| dtype | int | Column datatype |
| prec | int | Length of string or binary field, or precision of a numeric field, if applicable |
| scale | smallint | Scale of numeric field, if applicable |

Each row in the part_fields table identifies a single published field in a primary article.

When you create a primary article, a row is added to the part_fields table for each published field in the new article.

When you delete a primary article, a row is deleted from the part_fields table for each published field in that article.

## Primary articles/publications relation table

The part_pub_relation table stores information about the relationships of primary articles to publications.

Table A-9 lists the columns in the part_pub_relation table.

***Table A-9: Primary articles/publications relation table***

| Column name | Datatype | Description |
|---|---|---|
| part_id | numeric(18,0) | Primary article ID |
| pubid | numeric(18,0) | Publication ID |

Each row in the part_pub_relation table identifies a relationship between a single primary article and a single publication.

**Note** Each primary article must be associated with a single publication when it is created. However, after it is created, a primary article can be added to any number of additional publications.

When you create a new primary article or add an existing primary article to a publication, a row is added to the part_pub_relation table for the specified article and the specified publication.

When you delete a primary article or remove a primary article from a publication, the row that identifies the relationship between the primary article and the specified publication is deleted from the part_pub_relation table.

## Subscriptions table

The subscriptions table stores information for all ASE Replicator subscriptions.

Table A-10 lists the columns in the subscriptions table.

***Table A-10: Subscriptions table***

| Column name | Datatype | Description |
|---|---|---|
| subid | numeric(18,0) | Subscription ID |
| name | varchar(128) | Name of subscription |
| pubid | numeric(18,0) | ID of the publication to which this subscription subscribes |
| conn_id | numeric(18,0) | ID of the replicate connection this subscription belongs to |
| status | smallint | Status of the subscription |
| status_desc | varchar(255) | Status description |

Each row in the subscriptions table identifies a single subscription.

When you create a subscription, a row is added to the subscriptions table for the new subscription.

When you delete a subscription, the row that identifies that subscription is deleted from the subscriptions table.

## Replicate articles table

The rep_articles table stores information about all ASE Replicator replicate articles.

Table A-11 lists the columns in the rep_articles table.

***Table A-11: Replicate articles table***

| Column name | Datatype | Description |
|---|---|---|
| rart_id | numeric(18,0) | Replicate article ID |
| part_id | numeric(18,0) | ID of primary article to which the replicate article subscribes |
| subid | numeric(18,0) | ID of subscription to which the replicate article belongs |
| rart_name | rpl_sysname | Name of replicate article |
| proc_num | smallint | Stored procedure group number |
| proxy_name | rpl_sysname | Name of the proxy table for the article |
| dist_proc | rpl_sysname | Name of the distribution procedure for the article |
| owner | rpl_sysname | Owner of the replicate object |
| valid_pt | varchar(70) | Locator value of the validation point for the article |
| where_clause | varchar(1837) | Optional where clause for the article |

Each row in the rep_articles table identifies a single replicate article.

When you create a replicate article, a row is added to the rep_articles table for the new article.

When you delete a replicate article, the row that identifies that replicate article is deleted from the rep_articles table.

## Replicate article fields table

The rart_fields table stores information about all subscribed fields for all replicate articles.

Table A-12 lists the columns in the rart_fields table.

***Table A-12: Replicate article fields table***

| Column name | Datatype | Description |
|---|---|---|
| rart_id | numeric(18,0) | Replicate article ID |
| rart_colid | smallint | Column ID of replicate article field |
| rart_colname | rpl_sysname | Column name of replicate article field |
| is_identity | rpl_boolean | Indicates whether the column is an identity column |
| is_lob | rpl_boolean | Indicates whether the column is a large object datatype |
| is_null | rpl_boolean | Indicates whether the column can have a null value |
| dtype | int | Column datatype ID |
| dtype_name | rpl_sysname | Column datatype name |
| prec | int | Length of string or binary field, or precision of a numeric field, if applicable |
| scale | smallint | Scale of numeric field, if applicable |
| proxy_colid | smallint | Column ID of proxy table column |
| proxy_colname | rpl_sysname | Column name of proxy table column |
| part_id | numeric(18,0) | ID of primary article to which the replicate article subscribes |
| part_colid | int | Column ID of primary article field |
| part_colname | rpl_sysname | Column name of primary article field |

Each row in the rart_fields table identifies a single subscribed field in a replicate article.

When you create a replicate article, a row is added to the rart_fields table for each subscribed field in the new article.

When you delete a replicate article, a row is deleted from the rart_fields table for each subscribed field in that article.

## Transaction log table

The tran_log table stores transaction operation information for all replicated transactions.

Table A-13 lists the columns in the tran_log table.

***Table A-13: Transaction log table***

| Column name | Datatype | Description |
|---|---|---|
| conn_id | numeric(18,0) | ID of the primary connection this transaction operation came from |
| txid_page | int | Transaction ID page |
| txid_row | smallint | Transaction ID row |
| log_ts_high | smallint | Log timestamp high value |
| log_ts_low | int | Log timestamp low value |
| opid_page | int | Operation ID page |
| opid_row | smallint | Operation ID row |
| op_code | tinyint | Identifies type of operation |
| op_xstat | int | Identifies operation as either update or delete |
| commit_tstamp | datetime | Timestamp of the transaction commit in the Adaptive Server log |
| username | rpl_sysname | User login that performed the operation |
| part_id | numeric(18,0) | Primary article ID |
| tlog_tstamp | datetime | Time stamp of the operation in the transaction log table |

Each row in the tran_log table identifies a single replicated transaction operation from the primary database associated with the primary connection identified by the value in the conn_id column.

The ASE Replicator Publisher component adds rows to the tran_log table when it reads new transaction operations from the native Adaptive Server transaction log.

The tran_log table is truncated by the truncate_queue procedure.

## Configuration table

The config table stores information about the configuration of the ASE Replicator process.

See “General configuration parameters” on page 69 for information about ASE Replicator configuration parameters.

Table A-14 lists the columns in the config table.

***Table A-14: Configuration table***

| Column name | Datatype | Description |
|---|---|---|
| property | varchar(128) | Name of the configuration parameter |
| value | varchar(255) | Value of the configuration parameter |

Each row in the config table identifies the value of a single configuration parameter.

When you set or change the value of a configuration parameter, the row that identifies the value of the specified configuration parameter is updated.

## Trace table

The trace table stores information about ASE Replicator trace flags.

Table A-15 lists the columns in the trace table.

***Table A-15: Trace table***

| Column name | Datatype | Description |
|---|---|---|
| flag | varchar(128) | Name of the trace flag |
| value | varchar(5) | Value of the trace flag (true or false) |

Each row in the trace table identifies the value of a single trace flag.

When you set or change the value of a trace flag, the row that identifies the value of the specified trace flag is updated.

## Statistics table

The rpl_statistics table stores information about ASE Replicator statistics.

Table A-16 lists the columns in the rpl_statistics table.

***Table A-16: Statistics table***

| Column name | Datatype | Description |
|---|---|---|
| type | varchar(3) | Type of entity the statistic applies to |
| name | varchar(128) | Name of the entity |
| tstamp | datetime | Time stamp when the statistic value was generated |
| statistic | varchar(128) | Name of the statistic |
| value | varchar(255) | Value of the statistic |

Each row in the rpl_statistics table identifies the value of a single statistic.

New rows are added to the rpl_statistics table when statistics are generated. Statistics are generated automatically at the time interval specified by the stat_write_timeout configuration parameter.

Rows are truncated from the rpl_statistics table automatically at the time interval specified by the stat_trunc_interval configuration parameter.

## Base procedures in the Distribution Database

Most of the base procedures in the Distribution Database are the command procedures described in Chapter 4, “ASE Replicator Procedures.”

In addition to the command procedures, a truncate_queue procedure is created when you initialize the ASE Replicator process. truncate_queue is invoked by the ASE Replicator Distributor component to truncate the Distribution Database transaction log table (tran_log), as specified by ASE Replicator configuration parameters.

# Distribution Database shadow tables

When you create a new primary article, ASE Replicator creates a shadow table in the Distribution Database for that article.

Shadow table names begin with the characters sh, followed by an incremented “odometer” value. For example, the name of the first shadow table created is sha.

ASE Replicator creates a unique index for each shadow table. Shadow table index names begin with the characters shidx, followed by an odometer value. For example, the name of the first shadow table index created is shidxa.

Shadow tables have several columns that point to a location in the transaction log table, and one column for each published field in the article. Each row in a shadow table identifies a single transaction operation for the primary article.

The shadow table schema depends on the type of primary object published:

- Table with no large-object columns
- Table with one or more large-object columns
- Stored procedure

This following sections describe each type of Distribution Database shadow table.

## Shadow tables for primary tables without large objects

When you create a primary article for a table with no large-object columns, ASE Replicator creates a shadow table with the columns described in Table A-17.

***Table A-17: Shadow table without large object***

| Column name | Datatype | Description |
|---|---|---|
| conn_id | numeric(18,0) | ID of the primary connection |
| txid_page | int | Identifies the transaction ID page in the transaction log table |
| txid_row | smallint | Identifies the transaction ID row in the transaction log table |
| log_ts_high | smallint | Log timestamp high value in the transaction log table |
| log_ts_low | int | Log timestamp low value in the transaction log table |
| opid_page | int | Identifies the operation ID page in the transaction log table |
| opid_row | smallint | Identifies the operation ID row in the transaction log table |
| op_code | tinyint | Identifies type of operation |
| op_xstat | int | Identifies operation as either update or delete |
| image_type_ | char(1) | Identifies operation type or procedure execution |
| col1 | Specified in primary object | Column for a published field in the primary object |
| ... | ... | ... |
| col*n* | | Additional column for each published field in the primary object |

The shadow table column for each published field has the datatype of the published field, and each row contains the data from that primary column for the transaction operation identified by that row.

## Shadow tables for primary tables with large objects

When you create a primary article for a table with one or more large-object columns, ASE Replicator creates a shadow table with the columns described in Table A-18.

***Table A-18: Shadow table with large object***

| Column name | Datatype | Description |
|---|---|---|
| conn_id | numeric(18,0) | ID of the primary connection |
| txid_page | int | Identifies the transaction ID page in the transaction log table |
| txid_row | smallint | Identifies the transaction ID row in the transaction log table |
| log_ts_high | smallint | Log timestamp high value in the transaction log table |
| log_ts_low | int | Log timestamp low value in the transaction log table |
| opid_page | int | Identifies the operation ID page in the transaction log table |
| opid_row | smallint | Identifies the operation ID row in the transaction log table |
| op_code | tinyint | Identifies type of operation |
| op_xstat | int | Identifies operation as either update or delete |
| image_type_ | char(1) | Identifies operation type or procedure execution |
| pkey | numeric(5,0) | Primary key column(s) value |
| col1 | Specified in primary object | Column for a published field in the primary object |
| col2 | rpl_clob | Column for a published large-object (text) field in the primary object |
| ... | ... | ... |
| col*n* | | Additional column for each published field in the primary object |

Except for large-object primary table columns, the shadow table column for each published field has the datatype of the published field, and each row contains the data from that primary column for the transaction operation identified by that row.

For each large-object primary table column, the shadow table column has a user-defined Boolean datatype that identifies the type of large-object data in the primary column, and the value of that Boolean indicates whether a change

was made in the primary column data by the transaction operation identified by that row.

The user-defined Boolean datatypes are:

- rpl_blob – indicates image primary column data.
- rpl_clob – indicates text primary column data.
- rpl_uclob – indicates unitext primary column data.
- rpl_java – indicates java primary column data.

All these datatypes map to the Adaptive Server bit datatype.

## Shadow tables for primary procedures

When you create a primary article for a stored procedure, ASE Replicator creates a shadow table with the columns described in Table A-19.

***Table A-19: Shadow table for procedure***

| Column name | Datatype | Description |
|---|---|---|
| conn_id | numeric(18,0) | ID of the primary connection |
| txid_page | int | Identifies the transaction ID page in the transaction log table |
| txid_row | smallint | Identifies the transaction ID row in the transaction log table |
| log_ts_high | smallint | Log timestamp high value in the transaction log table |
| log_ts_low | int | Log timestamp low value in the transaction log table |
| opid_page | int | Identifies the operation ID page in the transaction log table |
| opid_row | smallint | Identifies the operation ID row in the transaction log table |
| op_code | tinyint | Identifies type of operation |
| op_xstat | int | Identifies operation as either update or delete |
| image_type_ | char(1) | Identifies operation type or procedure execution |
| p1 | Specified in primary object | Column for a published field in the primary object |

| Column name | Datatype | Description |
|---|---|---|
| ... | ... | ... |
| p*n* | | Additional column for each published field in the primary object |

The shadow table column for each published field has the datatype of the published procedure parameter, and each row contains the data from that parameter for the procedure invocation identified by that row.

**Note** The replicate article for a procedure may subscribe to none of the published fields in the primary article, so the shadow table for such a replicate article may contain no columns for published fields.

# Distribution Database replicate objects

When you create a replicate article, ASE Replicator creates one or more replicate objects in the Distribution Database. The replicate object (or objects) created depend on the type of object the replicate article identifies:

- Distribution procedure – created for each replicate article (either table or procedure).
- Proxy table – created only for a replicate article that identifies a table in the replicate database.

Replicate objects are used by the ASE Replicator Distributor component to replicate transaction operations to the replicate database.

This following sections describe the Distribution Database replicate objects.

## Distribution procedures

A distribution procedure is created for each replicate article. The purpose of the distribution procedure is to:

- Read the shadow table of the primary article that the replicate article subscribes to, and
- Apply the transaction operations to the replicate database, using data read from the shadow tables.

The distribution procedure for a table applies insert, update, and delete statements to the replicate article proxy table.

The distribution procedure for a stored procedure executes remote procedure calls in the replicate database.

Distribution procedure names begin with the characters dp, followed by an incremented "odometer" value. For example, the name of the first distribution procedure created is dpa.

## Replicate article proxy tables

A proxy table is created for each replicate article that identifies a table in the replicate database. The proxy table contains one column for each published field to which the replicate article subscribes.

Proxy tables allow the ASE Replicator Distributor component to apply replicated transaction operations to a local table in the Distribution Database, and use the Adaptive Server CIS feature to manage connections to remote servers and apply transaction operations to tables in remote databases.

Proxy table names begin with the characters px, followed by an incremented "odometer" value. For example, the name of the first proxy table created is pxa.

# Glossary

This glossary describes ASE Replicator terms used in this book. For a description of Adaptive Server and SQL terms, refer to the *Adaptive Server Glossary*.

**bidirectional replication** A replication scenario in which a single database acts as both a primary database and a replicate database. See also **primary database** and **replicate database**.

**database connection** An ASE Replicator object that identifies a primary or replicate database. See also **primary database** and **replicate database**.

**Distribution Database** A user database residing on the same Adaptive Server with the primary databases for ASE Replicator. The Distribution Database contains the stable queue and all the metadata needed to support replication. See also **stable queue**.

**field** The smallest database object entity that can be identified in an ASE Replicator system, either the column of a table or the input parameter of a stored procedure. See also **primary article** and **replicate article**.

**locator** A string value maintained by ASE Replicator that identifies a location in the stable queue or transaction log. See also **stable queue** and **transaction log**.

**Maintenance User** A data server login name that ASE Replicator uses to apply replicated transactions in a replicate database.

**materialization** The process of copying data specified by a replicate article from a primary database to a replicate database, thereby initializing the replicate table, and activating the replicate article so that ASE Replicator can begin replicating data to the replicate database. See also **validation** and **replicate article**.

**primary article** An ASE Replicator object that identifies a single primary database object (table or stored procedure) for which transactions will be published. See also **replicate article**.

| | |
|---|---|
| **primary database** | A database that contains published objects (tables and stored procedures), and that is a source of transactions to be replicated. See also **replicate database**. |
| **primary object** | A database object (table or stored procedure) in a primary database. See also **primary database**. |
| **publication** | A container object associated with a primary database connection, and which contains primary articles. See also **primary database**. |
| **publish** | The process of identifying and replicating transactions from a primary database. See also **subscribe**. |
| **published field** | A field identified in a primary article. See also **field**. |
| **replicate article** | An ASE Replicator object that identifies a single replicate database object (table or stored procedure) that subscribes to a primary article. See also **primary article**. |
| **replicate database** | A database that receives replicated transactions. See also **primary database**. |
| **replicate object** | A database object (table or stored procedure) in a replicate database. See also **replicate database**. |
| **stable queue** | A store-and-forward queue in which ASE Replicator records transaction operations to be replicated. Operations written into the stable queue remain there until they can be delivered to the replicate database. See also **transaction log**. |
| **subscribe** | The process of selecting published transactions and identifying the replicate objects that should receive them. See also **publish**. |
| **subscription** | A container object associated with a replicate database connection, and that points to a specific publication. See also **replicate database**. |
| **transaction log** | Generally, the log of transactions that affect the data managed by a database server. ASE Replicator creates a stable queue transaction log in the Distribution Database. See also **stable queue**. |
| **transactional consistency** | A condition in which all transactions in the primary database are applied in the replicate database in the same order that they were applied in the primary database. |
| **validation** | The process of making a replicate article ready to receive replicated transactions. Validation places a marker in the primary database transaction log to identify the location at which replication should begin for the replicate article. See also **materialization** and **replicate article**. |

# Index

## A


Adaptive Server
  adding remote servers 22–23
  character set 30, 71
  **cis rpc handling** parameter 17
  CIS, enabling 16–17
  configuration of 16–25
  configuration problems 149–150
  **enable cis** parameter 16–17
  external login 23–25, 43, 151–152
  host machine 20, 26–28, 30–32, 71
  local name 19
  **max cis remote connections** parameter 17
  port number 30, 72
  primary data server 16–22
  query processing options 156
  remote alias 19
  replicate data server 22–25
  Replication role 17–18
  **sp_addexternlogin** procedure 23
  **sp_addlogin** procedure 17, 24
  **sp_addserver** procedure 19, 20, 23
  **sp_adduser** procedure 18, 21, 24
  **sp_config** procedure 16–17
  **sp_dboption** procedure 20, 22
  **sp_helpserver** procedure 19
  **sybsystemprocs** database 28
  **sysobjects** table 88, 153–154
  **syspartitions** table 25
  **sysservers** table 19
  **tempdb** database 20
  user logins 17–18, 151–152
  version required for ASE Replicator 15
adding
  ASE Replicator remote alias 20
  ASE Replicator system user login 17–18
  database connections 41–43
  existing primary article to publication 55
  external login for Maintenance User 23–25, 43
  primary articles 54–55, 86–89
  primary database connections 89–90
  publications 50, 90–91
  remote servers 22–23
  replicate articles 55–57, 91–94
  replicate database connections 95–96
  subscriptions 50, 96–97
  user logins 17–18
  users to databases 18
**admin_port** configuration parameter 70
administering ASE Replicator 35–81
alias names of servers 19
altering database objects 153–154
approximate numeric datatypes 153, 154–156
articles, definition of 11–12
ASE Replicator
  adding database connections 41–43, 89–90, 95–96
  administering 35–81
  articles 11–12
  character set 30
  command procedures 83–148
  components of 4–8
  configuration parameters 69–77
  configuration problems 151
  configuring an instance of 26–29, 37, 99–101
  configuring database connections 43–44, 97–99, 101–102
  connection configuration 48–49
  connection configuration parameters 78–81
  creating an instance of 26–29
  creating database connections 41–43, 89–90, 95–96
  database connections 10–11, 41–49
  deleting database connections 43, 104, 106–107
  dependency of objects 10
  Distribution Database 5–6, 20–22, 161–177
  Distributor component 7, 139, 140, 142–143, 144
  dropping database connections 43, 104, 106–107
  executing command procedures 36, 83–84
  external login created by 43

flow of processing 7–8
help procedures 40
hierarchy of objects 9–10
host machine 20, 26–28, 30–32, 71
immediate shutdown 38
interfaces file entry 20
introduction to 4–13
Maintenance User 10–11, 23–25, 42–43, 92
monitoring 40, 120–125
objects 9–13
port number 30, 70
primary articles 12, 54–69
process component 5, 26–29, 36–40
publications 11, 49–53
Publisher component 6–7, 72–73, 75–76, 137, 139, 142–143, 147
remote alias 20
replicate articles 12, 54–69
replicate tables created by 56, 58, 92, 105
resuming database connections 44–46
resuming replication 39, 138
**rl_lastcommit** table 88, 93, 95
setup procedure 15–29
shutting down 38, 141
**sp_helpddb** procedure 28–29
starting 29–33
starting replication 138
start-up error 28
statistics 53, 76–77, 116, 120–125, 130
subscriptions 11, 49–53
suspending database connections 44–46
suspending replication 38–39, 142–143
system logs 73–74
system user login 17–18, 151–152
tracing 73–74, 144–146
troubleshooting 149–159
use of CIS 4
version of 121
**ase_charset** configuration parameter 71
**ase_host** configuration parameter 71
**ase_port** configuration parameter 72
**aserep** script 26–28, 29–32, 151
operating system variations 26, 30
parameters of 27–28
syntax help 31


## B


basic replication concepts 1–3
**batch_size** configuration parameter 72
**batch_timeout** configuration parameter 72
bidirectional replication
ASE Replicator support for 5
database connections 41–43
definition of 1–2
Maintenance User 10–11, 151–152


## C


character set 30, 71
**cis rpc handling** Adaptive Server parameter 17
columns
datatypes 154–158
**double precision** datatype 154–156
fields in articles 13
**float** datatype 154–156
IDENTITY property 153, 156
in shadow tables 172–176
large object 174–175
mapping to replicate tables 93
null type 153, 158–159
primary key 153, 155, 158
publishing 86–89
**real** datatype 154–156
subscribing to 91–94
**timestamp** datatype 156
**varbinary** datatype 156
command procedures, ASE Replicator
executing 36, 83–84
help procedures 40
**sp_addprimaryart** 55, 86–89
**sp_addprimaryconn** 42, 89–90
**sp_addpub** 50, 90–91
**sp_addreplicateart** 56, 91–94
**sp_addreplicateconn** 42, 95–96
**sp_addsub** 50, 96–97
**sp_configprimaryconn** 44, 48–49, 78, 97–99
**sp_configrep** 37, 69, 99–101
**sp_configreplicateconn** 44, 48–49, 78, 101–102
**sp_dropprimaryart** 58, 103
**sp_dropprimaryconn** 43, 104
**sp_droppub** 51, 104–105

**sp_dropreplicateart** 59, 105–106
**sp_dropreplicateconn** 43, 106–107
**sp_dropsub** 51, 107
**sp_helpconn** 108–109
**sp_helplastcommit** 109–110
**sp_helplocator** 110–111
**sp_helpprimaryart** 61–64, 111–114
**sp_helpprimaryconn** 46–48, 64, 114–118
**sp_helppub** 52, 64, 65, 118–120
**sp_helprep** 120–125
**sp_helpreplicateart** 66–68, 125–128
**sp_helpreplicateconn** 46–48, 69, 129–133
**sp_helpsub** 53, 65, 69, 133–136
**sp_materializesub** 59, 136–137
**sp_resumeprimaryconn** 45, 138
**sp_resumerep** 39, 138–139
**sp_resumereplicateconn** 46, 139–140
**sp_resumesub** 52, 140
**sp_shutdownrep** 38, 141
**sp_suspendprimaryconn** 45, 141–142
**sp_suspendrep** 39, 142–143
**sp_suspendreplicateconn** 45, 143
**sp_suspendsub** 51, 144
**sp_tracerep** 144–146
**sp_validatesub** 60, 147–148
Component Integration Services (CIS)
connection cache 29–30
enabling 16–17
on replicate server 22
proxy tables 6, 176–177
remote server connections 17
used by ASE Replicator 4
components of ASE Replicator 4–8
concepts, replication 1–3
**config** table 170–171
configuration parameters 37, 69–77
*See also* connection configuration parameters
**admin_port** 70
**ase_charset** 71
**ase_host** 71
**ase_port** 72
**batch_size** 72
**batch_timeout** 72
**ddb_name** 73
**log_directory** 73
**log_trace_verbose** 74
**log_wrap** 74
**monitor_delay** 75
**queue_size** 75
**scan_sleep_increment** 75
**scan_sleep_max** 76
**stat_trunc_interval** 76
**stat_write_timeout** 76
**status_monitoring** 77
**truncate_numops** 77
configuring
Adaptive Server 16–25, 149–150
ASE Replicator 151
ASE Replicator instance 26–29, 37, 99–101
database connections 43–44, 97–99, 101–102
remote servers 22–25, 150–151
replicate databases 22–25
**tempdb** database 20
**conn_properties** table 164
connection configuration parameters 78–81
*See also* configuration parameters
**gen_id** 78
**lti_version** 79
**mode** 79
**numrecs** 79
**queue_size** 80
**scan_sleep_increment** 80
**scan_sleep_max** 81
**timeout** 81
connections
adding 41–43
bidirectional replication 41
CIS cache 29–30
CIS remote server 17
configuration parameters 78–81
configuring 43–44, 97–99, 101–102
creating 41–43
deleting 43
dropping 43
getting information about 46–49, 108–109, 114–118, 129–133
introduction to 10–11
metadata in Distribution Database 164
monitoring 46–48
naming 41
resuming 44–46
resuming primary 138

resuming replicate 138, 139–140
statistics 116, 120–125
status of 120–125
suspending 44–46
suspending primary 141–143
suspending replicate 142–143
**connections** table 163–164
creating
ASE Replicator instance 26–29
ASE Replicator system user login 17–18, 151–152
database connections 41–43
Distribution Database 20–22
Maintenance User login 23–25, 151–152
primary articles 54–55, 86–89
primary database connections 89–90
publications 50, 90–91
replicate articles 55–57, 91–94
replicate database connections 95–96
replication system 33
subscriptions 50, 96–97

## D

database devices 21
database options
Distribution Database 22
**tempdb** database 20
databases
*See also* Distribution Database; primary databases; replicate databases
adding connections 41–43, 89–90, 95–96
adding users to 18
ASE Replicator connections 10–11, 41–49
configuring connections 43–44, 97–99, 101–102
creating connections 41–43, 89–90, 95–96
deleting connections 43, 104, 106–107
dropping connections 43, 104, 106–107
getting connection information 46–49, 108–109, 114–118, 129–133
log devices 21
materializing data in 59–60
monitoring connections 46–48
primary keys 154
resuming connections 44–46
suspending connections 44–46
**sybsystemprocs** 28
**tempdb** 20
datatypes 154–158
approximate numeric 153, 154–156
compatibility issues 157–158
**double precision** 153, 154–156
**float** 153, 154–156
**real** 153, 154–156
**timestamp** 153, 156
user-defined in Distribution Database 161–162
**varbinary** 156
**ddb_name** configuration parameter 73
default character set 30, 71
deleting
database connections 43
primary articles 57–58, 103
primary database connections 104
publications 51, 104–105
replicate articles 58–59, 105–106
replicate database connections 106–107
subscriptions 51, 107
dependency of ASE Replicator objects 10
Distribution Database 161–177
command procedures in 36, 83–84
**config** table 170–171
**conn_properties** table 164
**connections** table 163–164
creating the 20–22
database options 22
**ddb_name** configuration parameter 73
**ddb_system** table 163
distribution procedures 6, 176–177
introduction to 5–6
name of 20–22, 73
**part_fields** table 166
**part_pub_relation** table 166–167
permissions in 21
**pri_articles** table 165–166
proxy tables 6, 176–177
**publications** table 165
**rart_fields** table 168–169
**rep_articles** table 168
**rpl_statistics** table 76–77, 171–172
shadow tables 6, 172–176
stable queue 5–6
**subscriptions** table 167–168

**trace** table 171
**tran_log** table 5, 169–170
**truncate_queue** procedure 172
user-defined datatypes 161–162, 174–175
Distributor component, ASE Replicator 139, 140, 142–143, 144
introduction to 7
**double precision** datatype 153, 154–156
dropping
database connections 43
primary articles 57–58, 103
primary database connections 104
publications 51, 104–105
replicate articles 58–59, 105–106
replicate database connections 106–107
subscriptions 51, 107

## E

**enable cis** Adaptive Server parameter 16–17
entering parameter values in command procedures 83–84
environment variable, **$SYBASE** 26–27
errors
ASE Replicator start-up 28
**aserep** script 151
creating replicate connection 95–96
dropping primary article 103
dropping primary connection 104
dropping publication 105
dropping replicate connection 106
dropping subscription 107
in Adaptive Server configuration 149–150
in ASE Replicator configuration 151
in remote server configuration 150–151
primary article name conflict 88
troubleshooting 149–159
executing command procedures 36, 83–84
external login
created by ASE Replicator 43
for remote database 23–25
Maintenance User 151–152

## F

fields
datatype compatibility 157–158
getting information about primary 62–63, 111–114
getting information about replicate 67–68, 125–128, 131–132
in primary article 54–55, 86–89
in replicate article 55–57, 91–94
selecting by number 55, 56
selecting with **where** clause 56, 94
files
database device 21
interfaces 20, 23
flags, trace 144–146
**float** datatype 153, 154–156

## G

**gen_id** connection configuration parameter 78

## H

help procedures, ASE Replicator 40
help using **aserep** script 31
hierarchy of ASE Replicator objects 9–10
host machine, ASE Replicator 20, 26–28, 30–32, 71

## I

IDENTITY columns 153, 156
**IDENTITY_INSERT** query-processing option 156
**IDENTITY_UPDATE** query-processing option 156
instance name, ASE Replicator 31
interfaces file 20, 23
remote server entries 23
introduction to ASE Replicator 4–13
**isql** utility 36

## L

local name
  Adaptive Server 19
locator values
  in **rl_lastcommit** table 109–110
  stable queue 109–111
log devices 21
log files, ASE Replicator 73–74
**log_directory** configuration parameter 73
**log_trace_verbose** configuration parameter 74
**log_wrap** configuration parameter 74
**lti_version** connection configuration parameter 79

## M

Maintenance User
  adding external login 23–25
  creating external login 23–25
  for database connection 10–11, 42–43
  for remote server 23–25, 151–152
  permissions in replicate database 24–25
  replicate table owner 92
materializing replicate articles 59–60, 136–137
**max cis remote connections** Adaptive Server parameter 17
metadata
  articles 11–12
  ASE Replicator objects 9–13
  database connections 10–11, 46–48
  dependency of objects 10
  hierarchy of objects 9–10
  in Distribution Database 20
  primary articles 60–65, 135
  primary fields 13
  publications 52–53
  replicate articles 65–69, 134–135
  replicate fields 13, 131–132
  subscriptions 52–53
  tables in Distribution Database 6
**mode** connection configuration parameter 79
mode of transaction log scanning 79
**monitor_delay** configuration parameter 75
monitoring
  ASE Replicator process 40, 120–125
  database connections 46–48
  subscriptions 52–53

## N

names
  alias of server 19
  of Adaptive Server host machine 30
  of Adaptive Server local alias 19
  of ASE Replicator instance 31
  of ASE Replicator remote alias 20
  of ASE Replicator system user 17–18
  of data servers 41
  of database connections 41
  of database devices 21
  of databases 10, 41
  of Distribution Database 20–22, 73
  of local servers 19
  of Maintenance User 23–24
  of primary articles 55
  of publications 50
  of remote servers 19, 22–23
  of replicate articles 57
  of servers 10
  of subscriptions 50
  of user in database 18, 21, 24
**null** placeholder for parameters 84
null type, column 153, 158–159
**numrecs** connection configuration parameter 79

## O

objects, ASE Replicator 9–13
  dependency of 10
  hierarchy of 9–10
objects, database 11–13
  altering 153–154
  datatype compatibility 157–158
  primary keys 158
operating systems
  **aserep** script variations 26, 30

## P

parameters
*See also* configuration parameters; connection configuration parameters
**admin_port** 70
ASE Replicator configuration 37, 69–77
ASE Replicator connection configuration 78–81
**ase_charset** 71
**ase_host** 71
**ase_port** 72
**aserep** script 27–28
**batch_size** 72
**batch_timeout** 72
**cis rpc handling** 17
**ddb_name** 73
**enable cis** 16–17
entering in command procedures 83–84
**gen_id** connection configuration 78
**log_directory** 73
**log_trace_verbose** 74
**log_wrap** 74
**lti_version** connection configuration 79
**max cis remote connections** 17
**mode** connection configuration 79
**monitor_delay** 75
**null** placeholder 84
**numrecs** connection configuration 79
**queue_size** (general) 75
**queue_size** connection configuration 80
**scan_sleep_increment** (general) 75
**scan_sleep_increment** connection configuration 80
**scan_sleep_max** (general) 76
**scan_sleep_max** connection configuration 81
**stat_trunc_interval** 76
**stat_write_timeout** 76
**status_monitoring** 77
stored procedure input 2, 13
**timeout** connection configuration 81
**truncate_numops** 77
**part_fields** table 166
**part_pub_relation** table 166–167
permissions
ASE Replicator system user 18, 21, 25
Distribution Database 21
Maintenance User 25
on **sp_helpddb** procedure 28
primary database 18
replicate database 24–25
port number
of Adaptive Server 30, 72
of ASE Replicator 30, 70
**pri_articles** table 165–166
primary articles 54–69
adding 54–55, 86–89
adding existing to publication 55
creating 54–55, 86–89
deleting 57–58, 103
different name from replicate 57
dropping 57–58, 103
getting information about 60–65, 111–114, 117, 119, 135
introduction to 12
metadata in Distribution Database 165–167
publishing fields in 54–55, 86–89
with no subscribers 65
primary data server
configuring 16–22
primary databases
adding ASE Replicator system user 18
altering objects in 153
ASE Replicator connections 10, 41–49
configuring connections 97–99
connection configuration parameters 78–81, 97–99, 116
datatypes 154–158
default Maintenance User 90
definition of 1
deleting publications 104–105
dropping publications 104–105
getting connection information 46–49
Maintenance User 42
permissions in 18
primary keys 154
publications 90–91
publishing objects 54–55, 86–89
resuming connections 138
suspending connections 141–143
unpublished objects 64, 117
primary keys 153, 154, 158
with approximate numeric datatypes 155
procedures

*See also* command procedures
altering 153–154
ASE Replicator help 40
distribution procedures 6, 176–177
entering parameter values in 83–84
input parameters 2
replication of 2–3
**sp_configprimaryconn** 78
**sp_configreplicateconn** 78
**sp_helpprimaryart** ??–64
**sp_suspendrep** 39
**truncate_queue**, Distribution Database 172
process, ASE Replicator component 36–40
administering 36–39
configuring 37
immediate shutdown 38
initializing 26–29
introduction to 5
shutting down 38
starting 29–33
proxy tables, Distribution Database 6, 176–177
publications 11, 49–53
adding 50, 90–91
adding existing primary articles to 55
adding primary articles to 54–55, 86–89
creating 50, 90–91
deleting 51, 104–105
dropping 51, 104–105
dropping primary articles from 57–58, 103
getting information about 52–53, 116, 118–120
metadata in Distribution Database 165, 166–167
removing primary articles from 57–58, 103
**publications** table 165
Publisher component, ASE Replicator 137, 139, 142–143, 147
configuration 72–73, 75–76
introduction to 6–7

## Q

query tool, **isql** 36
query-processing options
**IDENTITY_INSERT** 156
**IDENTITY_UPDATE** 156
queue
locator value 109–111
shadow tables in Distribution Database 172–176
stable 3, 5–6, 77
truncating 77, 172
**queue_size** configuration parameter 75
**queue_size** connection configuration parameter 80

## R

**rart_fields** table 168–169
**real** datatype 153, 154–156
remote alias
Adaptive Server 19
ASE Replicator 20
remote procedure calls (RPC) 6
CIS handling 17
remote server connections (CIS) 17
remote servers
adding to Adaptive Server 22–23
configuration problems 150–151
configuring 22–25
interfaces file enrty 23
Maintenance User 23–25, 151–152
**rep_articles** table 168
replicate articles 54–69
adding 55–57, 91–94
creating 55–57, 91–94
deleting 58–59, 105–106
different name from primary 57
dropping 58–59, 105–106
getting information about 65–69, 125–128, 131–132, 134–135
introduction to 12
materializing 59–60, 136–137
metadata in Distribution Database 168–169
subscribing to published fields 55–57, 91–94
validating 59–60, 147–148
**where** clause in 56, 94
replicate databases
altering objects in 154
ASE Replicator connections 10, 41–49
CIS support requirement 22
configuring 22–25
configuring connections 101–102
connection configuration parameters 78–81, 101–

102
datatypes 154–158
default Maintenance User 95
definition of 1
deleting subscriptions 107
dropping subscriptions 107
getting connection information 46–49
Maintenance User 23–25, 42–43, 92, 151–152
materializing data in 59–60
permissions 24–25
primary keys 154
proxy tables 6
replicate table owner 92
resuming connections 138, 139–140
**rl_lastcommit** table 88, 93, 95, 109–110
subscribing to published objects 55–57, 91–94
subscriptions 96–97
suspending connections 142–143
tables created by ASE Replicator 56, 58, 92, 105
replicated transactions
definition of 1
replication
basic concepts of 1–3
bidirectional 1–2, 5, 10–11
definition of 1
introduction to 1–3
of procedures 2–3
of tables 2
of transactions 3
process of 7–8
resuming 39
suspending 38–39
troubleshooting 149–159
Replication role 17–18
Replication Server 13
replication system
monitoring 40
status of 122
replication system, setting up 33
resuming
database connections 44–46, 138–140
replication system 39
subscriptions 51–52, 138–140
**rl_lastcommit** table 88, 93, 95, 109–110
**rpl_statistics** table 171–172
**RUN** script 32–33
**RUN** script, ASE Replicator 26

## S

**scan_sleep_increment** configuration parameter 75
**scan_sleep_increment** connection configuration parameter 80
**scan_sleep_max** configuration parameter 76
**scan_sleep_max** connection configuration parameter 81
scanning mode, transaction log 79
scripts
**aserep** 26–28, 29–32
**RUN** 26, 32–33
**sp_helpddb.sql** 26, 28–29
**SYBASE** 27
server names
ASE Replicator 31
setting up
ASE Replicator 15–29
replication system 33
shadow tables, Distribution Database 6, 172–176
shutting down ASE Replicator 38, 141
**sp_addexternlogin** procedure 23
**sp_addlogin** command procedure 17
**sp_addlogin** procedure 24
**sp_addprimaryart** command procedure 55, 86–89
**sp_addprimaryconn** command procedure 42, 89–90
**sp_addpub** command procedure 50, 90–91
**sp_addreplicateart** command procedure 56, 91–94
**sp_addreplicateconn** command procedure 42, 95–96
**sp_addserver** command procedure 19, 20, 23
**sp_addsub** command procedure 50, 96–97
**sp_adduser** command procedure 18, 21
**sp_adduser** procedure 24
**sp_config** procedure 16–17
**sp_configprimaryconn** command procedure 44, 48–49, 97–99
**sp_configprimaryconn** procedure 78
**sp_configrep** command procedure 37, 69, 99–101
**sp_configreplicateconn** command procedure 44, 48–49, 101–102
**sp_configreplicateconn** procedure 78
**sp_dboption** command procedure 20, 22
**sp_dropprimaryart** command procedure 58, 103

**sp_dropprimaryconn** command procedure 43, 104
**sp_droppub** command procedure 51, 104–105
**sp_dropreplicateart** command procedure 59, 105–106
**sp_dropreplicateconn** command procedure 43, 106–107
**sp_dropsub** command procedure 51, 107
**sp_helpconn** command procedure 108–109
**sp_helpddb** command procedure 28–29
**sp_helpddb.sql** script 28–29
creating the 26
**sp_helplastcommit** command procedure 109–110
**sp_helplocator** command procedure 110–111
**sp_helpprimaryart** command procedure 61–??, 111–114
**sp_helpprimaryart** procedure ??–64
**sp_helpprimaryconn** command procedure 46–48, 64, 114–118
**info** keyword 47
**stats** keyword 48
**sp_helppub** command procedure 52, 64, 65, 118–120
**sp_helprep** command procedure 120–125
**sp_helpreplicateart** command procedure 66–68, 125–128
**sp_helpreplicateconn** command procedure 46–48, 69, 129–133
**info** keyword 47
**stats** keyword 48
**sp_helpserver** command procedure 19
**sp_helpsub** command procedure 53, 65, 69, 133–136
**sp_materializesub** command procedure 59, 136–137
**sp_resumeprimaryconn** command procedure 45, 138
**sp_resumerep** command procedure 39, 138–139
**sp_resumereplicateconn** command procedure 46, 139–140
**sp_resumesub** command procedure 52, 140
**sp_shutdownrep** command procedure 38, 141
**sp_suspendprimaryconn** command procedure 45, 141–142
**sp_suspendrep** command procedure 39, 142–143
**sp_suspendreplicateconn** command procedure 45, 143
**sp_suspendsub** command procedure 51, 144
**sp_tracerep** command procedure 144–146
**sp_validatesub** command procedure 60, 147–148
stable queue 3, 5–6
locator value 109–111
shadow tables 172–176
**tran_log** table 169–170
truncation of 77, 172
starting
ASE Replicator 29–33
database connections 45–46, 138
replication 39
subscriptions 51–52
start-up error, ASE Replicator 28
**stat_trunc_interval** configuration parameter 76
**stat_write_timeout** configuration parameter 76
static parameter, **enable cis** 16–17
statistics
database connections 46–48, 116, 120–125, 130
**rpl_statistics** table, Distribution Database 76–77
subscription 53, 133–136
status
of database connections 46–48, 120–125
of replication system 122
of subscriptions 120–125
**status_monitoring** configuration parameter 77
stopping
ASE Replicator 38
database connections 44–45, 141–143
replication 38–39
subscriptions 51–52
stopping ASE Replicator 141
stored procedure replication
definition of 2–3
distribution procedures 176–177
publishing procedures 86–89
replicate procedures 93
shadow tables 175–176
subscribing to published procedures 91–94
validating replicate articles 147–148
stored procedures
altering 153–154
input parameters 2, 13
subscription
statistics 53
subscriptions 11, 49–53
adding 50, 96–97
adding replicate articles to 55–57, 91–94
creating 50, 96–97
deleting 51, 107
dropping 51, 107
dropping replicate articles from 58–59
getting information about 52–53, 131, 133–136
materializing 59–60, 136–137

metadata in Distribution Database 167–168
monitoring 52–53
resuming 51–52, 138–140
starting 51–52
statistics 133–136
status of 120–125
stopping 51–52
suspending 51–52, 142–144
validating 59–60, 147–148
**subscriptions** table 167–168
suspending
database connections 44–46, 141–143
replication system 38–39, 142–143
subscriptions 51–52, 142–144
Sybase Central 35–36, 75, 77
**$SYBASE** environment variable 26–27
**SYBASE** script 27
**sybsystemprocs** database 28
**sysobjects** table 88, 153–154
**syspartitions** table 25
**sysservers** table 19
system logs, ASE Replicator 73–74
system table **ddb_system** 163
system user, ASE Replicator 151–152
adding login 17–18
as Maintenance User 42–43
permissions in primary database 18
permissions in replicate database 24–25
Replication role 17–18


## T


table replication
definition of 2
distribution procedures 176–177
materializing replicate articles 59–60, 136–137
shadow tables 6, 173–175
validating replicate articles 147–148
tables
altering 153–154
**config** 170–171
**conn_properties** 164
**connections** 163–164
**ddb_system** 163
in Distribution Database 162–172
**part_fields** 166
**part_pub_relation** 166–167
**pri_articles** 165–166
**publications** 165
**rart_fields** 168–169
**rep_articles** 168
replicate created by ASE Replicator 56, 58, 92, 105
replication of 2
**rl_lastcommit** 88, 93, 95, 109–110
**rpl_statistics** 171–172
**subscriptions** 167–168
**sysobjects** 88
**syspartitions** 25
**sysservers** 19
**trace** 171
**tran_log** 169–170
**tempdb** database 20
database options 20
**timeout** connection configuration parameter 81
**timestamp** datatype 153, 156
trace flags, ASE Replicator 144–146
starting with flags enabled 31
**trace** table 171
**tran_log** table 169–170
transaction log
metadata in Distribution Database 5, 169–170
scanning mode 79
transactions
replication of 3
troubleshooting 149–159
**truncate_numops** configuration parameter 77
**truncate_queue** procedure, Distribution Database 172
truncating the stable queue 77, 172


## U


unpublished objects, getting information about 64, 117
user logins
ASE Replicator system user 17–18, 151–152
external login 23–25, 151–152
Maintenance User 23–25, 42–43, 151–152
permissions 24–25, 151–152

Replication role 17–18
utilities, **isql** 36

## V

validating replicate articles 59–60, 147–148
values, entering procedure parameters 83–84
**varbinary** datatype 156
variable, **$SYBASE** environment 26–27
version
of Adaptive Server 15
of ASE Replicator 31, 121

## W

**where** clause, in replicate articles 56, 94