

Adaptive Server[®] Anywhere Database Administration Guide

Part number: 38123-01-0900-01

Last modified: June 2003

Copyright © 1989-2003 Sybase, Inc. Portions copyright © 2001-2003 iAnywhere Solutions, Inc. All rights reserved.

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 iAnywhere Solutions, Inc. iAnywhere Solutions, Inc. is a subsiduary of Sybase, Inc.

Sybase, 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, AnswerBase, Anywhere Studio, Application Manager, AppModeler, APT Workbench, APT-Build, APT-Edit, APT-Execute, APT-Library, APT-Translator, ASEP, 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, BayCam, Bit-Wise, BizTracker, Certified PowerBuilder Developer, Certified SYBASE Professional, Certified SYBASE Professional (logo), ClearConnect, Client Services, Client-Library, CodeBank, Column Design, ComponentPack, Connection Manager, Convoy/DM, Copernicus, CSP, Data Pipeline, Data Workbench, DataArchitect, Database Analyzer, DataExpress, DataServer, DataWindow, DB-Library, dbQueue, Developers Workbench, Direct Connect Anywhere, DirectConnect, Distribution Director, Dynamic Mobility Model, Dynamo, e-ADK, E-Anywhere, e-Biz Integrator, E-Whatever, EC Gateway, ECMAP, ECRTP, eFulfillment Accelerator, Electronic Case Management, Embedded SQL, EMS, Enterprise Application Studio, Enterprise Client/Server, Enterprise Connect, Enterprise Data Studio, Enterprise Manager, Enterprise Portal (logo), Enterprise SQL Server Manager, Enterprise Work Architecture, Enterprise Work Designer, Enterprise Work Modeler, eProcurement Accelerator, eremote, Everything Works Better When Everything Works Together, EWA, Financial Fusion, Financial Fusion (and design), Financial Fusion Server, Formula One, Fusion Powered e-Finance, Fusion Powered Financial Destinations, Fusion Powered STP, Gateway Manager, GeoPoint, GlobalFIX, iAnywhere, iAnywhere Solutions, ImpactNow, Industry Warehouse Studio, InfoMaker, Information Anywhere, Information Everywhere, InformationConnect, InstaHelp, InternetBuilder, iremote, iScript, Jaguar CTS, jConnect for JDBC, KnowledgeBase, Logical Memory Manager, M-Business Channel, M-Business Network, M-Business Server, Mail Anywhere Studio, MainframeConnect, Maintenance Express, Manage Anywhere Studio, MAP, MDI Access Server, MDI Database Gateway, media.splash, Message Anywhere Server, MetaWorks, MethodSet, ML Query, MobiCATS, My AvantGo, My AvantGo Media Channel, My AvantGo Mobile Marketing, MySupport, Net-Gateway, Net-Library, New Era of Networks, Next Generation Learning, Next Generation Learning Studio, O DEVICE, OASiS, OASiS (logo), ObjectConnect, ObjectCycle, OmniConnect, OmniSQL Access Module, OmniSQL Toolkit, Open Biz, Open Business Interchange, Open Client, Open Client/Server, Open Client/Server Interfaces, Open ClientConnect, Open Gateway, Open Server, Open ServerConnect, Open Solutions, Optima++, Partnerships that Work, PB-Gen, PC APT Execute, PC DB-Net, PC Net Library, PhysicalArchitect, Pocket PowerBuilder, PocketBuilder, Power Through Knowledge, Power++, power.stop, PowerAMC, PowerBuilder, PowerBuilder Foundation Class Library, PowerDesigner, PowerDimensions, PowerDynamo, Powering the New Economy, PowerS, PowerScript, PowerSite, PowerSocket, Powersoft, Powersoft Portfolio, Powersoft Professional, PowerStage, PowerStudio, PowerTips, PowerWare Desktop, PowerWare Enterprise, ProcessAnalyst, QAnywhere, Rapport, Relational Beans, RepConnector, Replication Agent, Replication Driver, Replication Server, Replication Server Manager, Replication Toolkit, Report Workbench, Report-Execute, Resource Manager, RW-DisplayLib, RW-Library, S.W.I.F.T. Message Format Libraries, SAFE, SAFE/PRO, 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 Server SNMP SubAgent, SQL Server/CFT, SQL Server/DBM, SQL SMART, SQL Station, SQL Toolset, SQLJ, Stage III Engineering, Startup.Com, STEP, SupportNow, Sybase Central, Sybase Client/Server Interfaces, Sybase Development Framework, Sybase Financial Server, Sybase Gateways, Sybase Learning Connection, Sybase MPP, Sybase SOL Desktop, Sybase SOL Lifecycle, Sybase SOL Workgroup, Sybase Synergy Program, Sybase User Workbench, Sybase Virtual Server Architecture, SybaseWare, Syber Financial, SyberAssist, SybMD, SyBooks, System 10, System 11, System XI (logo), SystemTools, Tabular Data Stream, The Enterprise Client/Server Company, The Extensible Software Platform, The Future Is Wide Open, The Learning Connection, The Model For Client/Server Solutions, The Online Information Center, The Power of One, TradeForce, Transact-SQL, Translation Toolkit, Turning Imagination Into Reality, UltraLite, UltraLite.NET, UNIBOM, Unilib, Uninull, Unisep, Unistring, URK Runtime Kit for UniCode, Versacore, Viewer, VisualWriter, VQL, Warehouse Control Center, Warehouse Studio, Warehouse WORKS, WarehouseArchitect, 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. or its subsidiaries.

All other trademarks are property of their respective owners.

Contents

About This Manual	ix
SQL Anywhere Studio documentation	. x
Documentation conventions	. xiii
The Adaptive Server Anywhere sample database	. xv
Finding out more and providing feedback	. xvi
I Starting and Connecting to Your Database	1
1 Running the Database Server	3
	. 4
Starting the server	. 8
Some common options	. 10
Stopping the database server	. 18
Starting and stopping databases	. 20
Running the server outside the current session	. 21
Troubleshooting server startup	. 34
2 Connecting to a Database	37
	. 38
Connecting from Sybase Central or Interactive SQL	. 42
Simple connection examples	45
Working with ODBC data sources	. 53
Connecting from desktop applications to a Windows CE database.	. 65
Connecting to a database using OLE DB	. 68
Connection parameters	. 70
Troubleshooting connections	. 75
Using integrated logins	. 85
3 Client/Server Communications	93
Supported network protocols	94
Using the TCP/IP protocol	. 95
Using the SPX protocol	100
	101
Adjusting communication compression settings to improve perfor	. 101
mance	102
Troublesbooting network communications	102
	. 105

	4	Adaptive Server Anywhere as an Open Server 109 Open Clients, Open Servers, and TDS 110 Setting up Adaptive Server Anywhere as an Open Server 112 Configuring Open Servers 114 Characteristics of Open Client and jConnect connections 120
	5	The Database Server 123 The database server 124
	6	Connection and Communication Parameters 173 Connection parameters 174 Network communications parameters 201
	7	Using the Built-in Web Server219About web services220Quick start221Creating web services222Starting a database server that listens for web requests225How URLs are interpreted227Procedures that handle web requests229Working with variables232Automatic character-set conversion234Errors235
II	W	orking with Database Files 237
	8	File Locations and Installation Settings 239 Installation directory structure 240 How Adaptive Server Anywhere locates files 242 Environment variables 244 Registry and INI files 249
	9	File Locations and Installation Settings239Installation directory structure240How Adaptive Server Anywhere locates files242Environment variables244Registry and INI files249Working with Database Files253Overview of database files254Using additional dbspaces256Working with write files260Using the utility database262

	Schedule and event internals	. 278 . 280
	11 International Languages and Character Sets Introduction to international languages and character sets Understanding character sets in software Understanding locales Understanding collations Understanding collations Understanding character set translation Understanding character set translation International language and character set tasks	 285 286 289 301 307 317 320 325
	12 Backup and Data Recovery Introduction to backup and recovery Understanding backups Designing backup procedures Configuring your database for data protection Backup and recovery internals Backup and recovery tasks	 337 338 343 346 355 359 367
III	Permissions, Security, and Replication	387
III	Permissions, Security, and Replication 13 Managing User IDs and Permissions Database permissions overview Setting user and group options Managing individual user IDs and permissions Managing connected users Managing groups Database object names and prefixes Using views and procedures for extra security Changing Ownership on Nested Objects How user permissions are assessed Managing the resources connections use Users and permissions in the system tables	387 389 390 393 405 406 413 415 418 420 421 422

V	Utilities, Options, and Properties	2	153
	15 Database Administration Utilities		455
	Administration utilities overview		457
	The Backup utility		458
	The Collation utility		462
	The File Hiding utility		466
	The Compression utility		468
	The Console utility		471
	The Data Source utility		472
	The Erase utility		478
	The Histogram utility		481
	The Information utility		483
	The Initialization utility		485
	The Interactive SQL utility		492
	The Language utility		497
	The License utility		499
	The Log Transfer Manager		501
	The Log Translation utility		508
	The Ping utility		514
	The Rebuild utility		517
	The Server Location utility		518
	The Service Creation utility		519
	The Spawn utility		523
	The Stop utility		525
	The Transaction Log utility		527
	The Uncompression utility		531
	The Unload utility		533
	The Upgrade utility		542
	The Validation utility		547
	The Write File utility		551
	16 Database Options		555
	Introduction to database options	•	556
	Database options	•	561
	Compatibility options	•	566
	Replication options	•	570
	Interactive SQL options	•	571
	Alphabetical list of options	•	574
	17 Database Performance and Connection Properties		627
	Database renormance and connection rioperties		620
		·	647
		•	04/

I١

18 Physical Limitations	673
Size and number limitations	
Index	677

About This Manual

SubjectThis book covers material related to running, managing, and configuring
databases. It describes database connections, the database server, database
files, security, backup procedures, and replication with the Replication
Server, as well as administration utilities and options.AudienceThis manual is for all users of Adaptive Server Anywhere. It is to be used in
conjunction with other manuals in the documentation set.

SQL Anywhere Studio documentation

This book is part of the SQL Anywhere documentation set. This section describes the books in the documentation set and how you can use them.

The SQL Anywhere Studio documentation

The SQL Anywhere Studio documentation is available in a variety of forms: in an online form that combines all books in one large help file; as separate PDF files for each book; and as printed books that you can purchase. The documentation consists of the following books:

- Introducing SQL Anywhere Studio This book provides an overview of the SQL Anywhere Studio database management and synchronization technologies. It includes tutorials to introduce you to each of the pieces that make up SQL Anywhere Studio.
- What's New in SQL Anywhere Studio This book is for users of previous versions of the software. It lists new features in this and previous releases of the product and describes upgrade procedures.
- ◆ Adaptive Server Anywhere Getting Started This book is for people new to relational databases or new to Adaptive Server Anywhere. It provides a quick start to using the Adaptive Server Anywhere database-management system and introductory material on designing, building, and working with databases.
- Adaptive Server Anywhere Database Administration Guide This book covers material related to running, managing, and configuring databases and database servers.
- ◆ Adaptive Server Anywhere SQL User's Guide This book describes how to design and create databases; how to import, export, and modify data; how to retrieve data; and how to build stored procedures and triggers.
- ◆ Adaptive Server Anywhere SQL Reference Manual This book provides a complete reference for the SQL language used by Adaptive Server Anywhere. It also describes the Adaptive Server Anywhere system tables and procedures.
- ◆ Adaptive Server Anywhere Programming Guide This book describes how to build and deploy database applications using the C, C++, and Java programming languages. Users of tools such as Visual Basic and PowerBuilder can use the programming interfaces provided by those tools. It also describes the Adaptive Server Anywhere ADO.NET data provider.

- ♦ Adaptive Server Anywhere Error Messages This book provides a complete listing of Adaptive Server Anywhere error messages together with diagnostic information.
- ◆ SQL Anywhere Studio Security Guide This book provides information about security features in Adaptive Server Anywhere databases. Adaptive Server Anywhere 7.0 was awarded a TCSEC (Trusted Computer System Evaluation Criteria) C2 security rating from the U.S. Government. This book may be of interest to those who wish to run the current version of Adaptive Server Anywhere in a manner equivalent to the C2-certified environment.
- MobiLink Synchronization User's Guide This book describes how to use the MobiLink data synchronization system for mobile computing, which enables sharing of data between a single Oracle, Sybase, Microsoft or IBM database and many Adaptive Server Anywhere or UltraLite databases.
- ♦ MobiLink Synchronization Reference This book is a reference guide to MobiLink command line options, synchronization scripts, SQL statements, stored procedures, utilities, system tables, and error messages.
- ◆ iAnywhere Solutions ODBC Drivers This book describes how to set up ODBC drivers to access consolidated databases other than Adaptive Server Anywhere from the MobiLink synchronization server and from Adaptive Server Anywhere remote data access.
- ◆ SQL Remote User's Guide This book describes all aspects of the SQL Remote data replication system for mobile computing, which enables sharing of data between a single Adaptive Server Anywhere or Adaptive Server Enterprise database and many Adaptive Server Anywhere databases using an indirect link such as e-mail or file transfer.
- SQL Anywhere Studio Help This book includes the context-sensitive help for Sybase Central, Interactive SQL, and other graphical tools. It is not included in the printed documentation set.
- ♦ UltraLite Database User's Guide This book is intended for all UltraLite developers. It introduces the UltraLite database system and provides information common to all UltraLite programming interfaces.
- ◆ UltraLite Interface Guides A separate book is provided for each UltraLite programming interface. Some of these interfaces are provided as UltraLite components for rapid application development, and others are provided as static interfaces for C, C++, and Java development.

In addition to this documentation set, PowerDesigner and InfoMaker include their own online documentation.

Documentation formats SQL Anywhere Studio provides documentation in the following formats:

◆ Online documentation The online documentation contains the complete SQL Anywhere Studio documentation, including both the books and the context-sensitive help for SQL Anywhere tools. The online documentation is updated with each maintenance release of the product, and is the most complete and up-to-date source of documentation.

To access the online documentation on Windows operating systems, choose Start > Programs > SQL Anywhere 9 > Online Books. You can navigate the online documentation using the HTML Help table of contents, index, and search facility in the left pane, as well as using the links and menus in the right pane.

To access the online documentation on UNIX operating systems, see the HTML documentation under your SQL Anywhere installation.

• **Printable books** The SQL Anywhere books are provided as a set of PDF files, viewable with Adobe Acrobat Reader.

The PDF files are available on the CD ROM in the *pdf_docs* directory. You can choose to install them when running the setup program.

◆ Printed books The complete set of books is available from Sybase sales or from eShop, the Sybase online store. You can access eShop by clicking How to Buy ➤ eShop at http://www.ianywhere.com.

Documentation conventions

This section lists the typographic and graphical conventions used in this documentation.

Syntax conventions The following conventions are used in the SQL syntax descriptions:

• **Keywords** All SQL keywords appear in upper case, like the words ALTER TABLE in the following example:

ALTER TABLE [owner.]table-name

• **Placeholders** Items that must be replaced with appropriate identifiers or expressions are shown like the words *owner* and *table-name* in the following example:

ALTER TABLE [owner.]table-name

• **Repeating items** Lists of repeating items are shown with an element of the list followed by an ellipsis (three dots), like *column-constraint* in the following example:

ADD column-definition [column-constraint, ...]

One or more list elements are allowed. In this example, if more than one is specified, they must be separated by commas.

• **Optional portions** Optional portions of a statement are enclosed by square brackets.

RELEASE SAVEPOINT [savepoint-name]

These square brackets indicate that the *savepoint-name* is optional. The square brackets should not be typed.

• **Options** When none or only one of a list of items can be chosen, vertical bars separate the items and the list is enclosed in square brackets.

[ASC | DESC]

For example, you can choose one of ASC, DESC, or neither. The square brackets should not be typed.

• Alternatives When precisely one of the options must be chosen, the alternatives are enclosed in curly braces and a bar is used to separate the options.

$[\text{ QUOTES} \{ \text{ ON} \mid \text{OFF} \}]$

If the QUOTES option is used, one of ON or OFF must be provided. The brackets and braces should not be typed.

Graphic icons

The following icons are used in this documentation.

♦ A client application.



• A database server, such as Sybase Adaptive Server Anywhere.



• A database. In some high-level diagrams, the icon may be used to represent both the database and the database server that manages it.



 Replication or synchronization middleware. These assist in sharing data among databases. Examples are the MobiLink Synchronization Server and the SQL Remote Message Agent.



• A programming interface.



The Adaptive Server Anywhere sample database

Many of the examples throughout the documentation use the Adaptive Server Anywhere sample database.

The sample database is held in a file named *asademo.db*, and is located in your SQL Anywhere directory.

The sample database represents a small company. It contains internal information about the company (employees, departments, and finances) as well as product information and sales information (sales orders, customers, and contacts). All information in the database is fictional.

The following figure shows the tables in the sample database and how they relate to each other.



Finding out more and providing feedback

We would like to receive your opinions, suggestions, and feedback on this documentation.

You can provide feedback on this documentation and on the software through newsgroups set up to discuss SQL Anywhere technologies. These newsgroups can be found on the *forums.sybase.com* news server.

The newsgroups include the following:

- sybase.public.sqlanywhere.general.
- sybase.public.sqlanywhere.linux.
- sybase.public.sqlanywhere.mobilink.
- sybase.public.sqlanywhere.product_futures_discussion.
- sybase.public.sqlanywhere.replication.
- sybase.public.sqlanywhere.ultralite.

Newsgroup disclaimer

iAnywhere Solutions has no obligation to provide solutions, information or ideas on its newsgroups, nor is iAnywhere Solutions obliged to provide anything other than a systems operator to monitor the service and insure its operation and availability.

iAnywhere Solutions Technical Advisors as well as other staff assist on the newsgroup service when they have time available. They offer their help on a volunteer basis and may not be available on a regular basis to provide solutions and information. Their ability to help is based on their workload.

Part I

STARTING AND CONNECTING TO YOUR DATABASE

This part describes how to start the Adaptive Server Anywhere database server, and how to connect to your database from a client application.

CHAPTER 1

Running the Database Server

About this chapter

Contents

This chapter describes how to start and stop the Adaptive Server Anywhere database server, and the options open to you on startup under different operating systems.

page
4
8
10
18
20
21
34

Introduction

Adaptive Server Anywhere provides two versions of the database server:

◆ The personal database server This executable does not support client/server communications across a network. Although the personal database server is provided for single-user, same-machine use—for example, as an embedded database engine—it is also useful for development work.

On Windows operating systems except Windows CE, the name of the personal server executable is *dbeng9.exe*. On UNIX operating systems its name is *dbeng9*.

• The network database server Intended for multi-user use, this executable supports client/server communications across a network.

On Windows operating systems, including Windows CE, the name of the network server executable is *dbsrv9.exe*. On Novell NetWare the name is *dbsrv9.nlm*, and on UNIX operating systems it is *dbsrv9*.

- Server differences The request-processing engine is identical in the two servers. Each supports exactly the same SQL, and exactly the same database features. The main differences include:
 - ♦ Network protocol support Only the network server supports communications across a network.
 - Number of connections The personal server has a limit of ten simultaneous connections. The limit for the network server depends on your license.
 - Number of CPUs The personal database server uses a maximum of two CPUs for request processing. The network database server has no set limit.
 - ◆ **Default number of internal threads** You can configure the number of requests the server can process at one time using the -gn option. The network database server has a default of 20 threads and no set limit, while the personal database server has a default and limit of 10 threads.

For more information about database server options, see "The database server" on page 124.

• **Startup defaults** To reflect their use as a personal server and a server for many users, the startup defaults are slightly different for each.

First steps

	You can start a personal server running on a single database very simply. For example, you can start both a personal server and a database called <i>test.db</i> by typing the following command in the directory where <i>test.db</i> is located: dbeng9 test
Where to specify commands	You can specify commands in several ways, depending on your operating system. For example, you can:
	• type the command at the command prompt.
	• place the command in a shortcut or desktop icon.
	• run the command in a batch file.
	• include the command as a StartLine (START) connection parameter in a connection string.
	For more information, see "StartLine connection parameter [START]" on page 199.
	There are slight variations in the basic command from platform to platform. These are described in the following section.
	Tou can also start a personal server using a database file name in a connection string. For more information, see "Connecting to an embedded database" on page 47.
Start the database ser	ver
	The way you start the database server varies slightly depending on the operating system you use. This section describes how to specify commands for the simple case of running a single database with default settings, on each supported operating system.
Notes	• These commands start the personal server (dbeng9). To start a network server, simply replace dbeng9 with dbsrv9 .
	• If the database file is in the starting directory for the command, you do not need to specify <i>path</i> .
	• If you do not specify a file extension in <i>database-file</i> , the extension <i>.db</i> is assumed.

To start the personal database server using default options (Windows)

- 1. Open a command prompt.
- 2. Type the following command:

start dbeng9 path\database-file

If you omit the database file, the Server startup options dialog appears, where you can locate a database file using a Browse button.

To start the personal database server using default options (UNIX)

- 1. Open a command prompt.
- 2. Type the following command:

dbeng9 path/database-file

To start the personal database server using default options (Net-Ware)

1. The database server for NetWare is a NetWare Loadable Module (NLM) (*dbsrv9.nlm*). A NLM is a program that you can run on your NetWare server. Load a database server on your NetWare server as follows:

load dbsrv9.nlm path\database-file

The database file must be on a NetWare volume. A typical filename is of the form *DB:\database\sales.db*.

You can load the server from a client machine using the Novell remote console utility. See your Novell documentation for details.

You can put the command into your Novell *autoexec.ncf* file so that Adaptive Server Anywhere loads automatically each time you start the NetWare server.

There is no personal server for Novell NetWare, just a network server.

What else is there to it?

Although you can start a personal server in the simple way described above, there are many other aspects to running a database server in a production environment. For example,

 You can choose from many options to specify such features as how much memory to use as cache, how many CPUs to use (on multi-processor machines), and the network protocols to use (network server only). The options are one of the major ways of tuning Adaptive Server Anywhere behavior and performance.

- You can run the server as a Windows **service**. This allows it to continue running even when you log off the machine.
- You can start the personal server from an application and shut it down when the application has finished with it. This is typical when using the database server an **embedded database**.

The remainder of this chapter describes these options in more detail.

Starting the server

The general form for the server command is as follows:

```
executable [
server-options ] [
database-file [ database-options ], . . . ]
```

If you supply no options and no database file, then on Windows operating systems a dialog appears, allowing you to use a Browse button to locate your database file.

The elements of the database server command include the following:

• **Executable** This can be either the personal server or the network server.

For the file names on different operating systems, see "Introduction" on page 4.

In this chapter, unless discussing network-specific options, we use the personal server in sample commands. The network server takes a very similar set of options.

- **Server options** These options control the behavior of the database server for all running databases.
- **Database file** You can specify zero, one, or more database file names. Each of these databases starts and remains available for applications.

Caution

The database file and the transaction log file must be located on the same physical machine as the database server. Database files and transaction log files located on a network drive can lead to poor performance and data corruption.

• **Database options** For each database file you start, you can provide database options that control certain aspects of its behavior.

In this section, we look at some of the more important and commonly-used options.

For full reference information on each of these options, see "The database server" on page 124.

In examples throughout this chapter where there are several options, we display them on separate lines for clarity, as they could be written in a configuration file. If you type them directly at the command prompt, you must type them all on one line.

Case sensitivity Database and server options are generally case sensitive. You should type all options in lower case.

Listing available options

To list the database server options

- 1. Open a command prompt.
- 2. Type the following command:

dbeng9 -?

Starting the database server on Windows CE

The database server supplied for Windows CE is the network database server (*dbsrv9.exe*). The network server supports communications over a TCP/IP network link.

The usual client/server arrangement has the database server running on a machine with more power and resources than the client applications. Clearly, this is not the case with Windows CE; instead, the less powerful machine is running the database server.

The advantage to supplying a network server on Windows CE is that you can run database applications on your desktop computer to carry out tasks on your Windows CE database. For example:

- You can use Sybase Central on your desktop PC to manage your database.
- You can use Interactive SQL on your desktop to load and unload data, and carry out queries.
- You can use InfoMaker to produce reports.

The Windows CE database server does not start the TCP/IP network link unless it is explicitly requested:

dbsrv9 -x tcpip ...

On Pocket PCs, attempting to start a second Adaptive Server Anywhere server while a first server is already running brings the first server to the foreground. This is standard behavior for Pocket PC applications. Because of this, two servers cannot be run at the same time on a Pocket PC. As an alternative to multiple servers, one server can run more than one database if necessary. Note that -ga all is required to autostart a database on a running server.

Some common options

This section describes some of the most common options, and points out when you may wish to use them. They are:

- Using configuration files
- Naming the server and the databases
- Performance
- Permissions
- Maximum page size
- Special modes
- Threading
- Network communications (network server only)

Using configuration files

If you use an extensive set of options, you can store them in a configuration file and invoke that file in a server command. The configuration file can contain options on several lines. For example, the following configuration file starts the personal database server and the sample database. It sets a cache of 10 Mb, and names this instance of the personal server **Elora**. Lines with # as the first comment in the line are treated as comments.

```
# Configuration file for server Elora
-n Elora
-c 10M
path\asademo.db
```

In the example, *path* is the name of your SQL Anywhere directory. On UNIX, you would use a forward slash instead of the backslash in the file path.

If you name the file *sample.cfg*, you could use these options as follows:

dbeng9 @sample.cfg

For more information, see "@filename server option" on page 131.

Naming the server and the databases

You can use the -n as a server option (to name the server) or as a database option (to name the database).

The server and database names are among the connection parameters that client applications may use when connecting to a database. The server name appears on the desktop icon and in the title bar of the server window.

Naming databases You may want to provide a meaningful database name for users of client applications. The database is identified by that name until it is stopped.

If you don't provide a database name, the default name is the root of the database file name (the file name without the *.db* extension). For example, in the following command the first database is named **asademo**, and the second **sample**.

dbeng9 asademo.db sample.db

You can name databases by supplying a -n option following the database file. For example, the following command starts the sample database and names it **MyDB**:

dbeng9 asademo.db -n MyDB

Naming the server Providing a database server name helps avoid conflicts with other server names on your network. It also provides a meaningful name for users of client applications. The server keeps its name for its lifetime (until it is shut down). If you don't provide a server name, the server is given the name of the first database started.

You can name the server by supplying a -n option before the first database file. For example, the following command starts a server on the **asademo** database and gives the server the name **Cambridge**:

dbeng9 -n Cambridge asademo.db

If you supply a server name, you can start a database server with no database started. The following command starts a server named **Galt** with no database started:

dbeng9 -n Galt

For more information about starting databases on a running server, see "Starting and stopping databases" on page 20.

Case sensitivity Server names and database names are case insensitive as long as the character set is single-byte.

For more information, see "Connection strings and character sets" on page 318.

Controlling performance and memory from the command line

Several options can have a major impact on database server performance, including:

◆ Cache size The -c option controls the amount of memory that Adaptive Server Anywhere uses as a cache. This can be a major factor in affecting performance.

Generally speaking, the more memory made available to the database server, the faster it performs. The cache holds information that may be required more than once. Accessing information in cache is many times faster than accessing it from disk. The default initial cache size is computed based on the amount of physical memory, the operating system, and the size of the database files. On Windows operating systems, and UNIX the database server takes additional cache when the available cache is exhausted.

For more information about performance tuning, see "Monitoring and Improving Performance" [*ASA SQL User's Guide*, page 153].

For information on controlling cache size, see "Cache size" on page 129.

 Number of processors If you are running on a multi-processor machine, you can set the number of processors with the -gt option.

For more information, see "-gt server option" on page 150 and "Controlling threading from the command line" on page 14.

♦ Other performance-related options There are several options available for tuning network performance, including -gb (database process priority), and -u (buffered disk I/O).

For more information about startup options, see "The database server" on page 124.

Controlling permissions from the command line

Some options control the permissions required to carry out certain global operations, including permissions to start and stop databases, load and unload data, and create and delete database files.

For more information, see "Running the database server in a secure fashion" [*SQL Anywhere Studio Security Guide*, page 14].

Setting a maximum page size

The database server cache is arranged in pages-fixed-size areas of memory.

Since the server uses a single cache for its lifetime (until it is shut down), all pages must have the same size.

A database file is also arranged in pages, with a size that is specified on the command line. Every database page must fit into a cache page. By default, the server page size is the same as the largest page size of the databases on the command line. Once the server starts, you cannot start a database with a larger page size than the server.

To allow databases with larger page sizes to be started after startup, you can force the server to start with a specified page size using the -gp option. If you use larger page sizes, remember to increase your cache size. A cache of the same size will accommodate only a fraction of the number of the larger pages, leaving less flexibility in arranging the space.

The following command starts a server that reserves an 8 Mb cache and can accommodate databases of page sizes up to 4096 bytes.

dbsrv9 -gp 4096 -c 8M -n myserver

Running in special modes

You can run Adaptive Server Anywhere in special modes for particular purposes.

♦ Read-only You can run databases in read-only mode by supplying the -r option.

For more information, see "-r server option" on page 157.

◆ Bulk load This is useful when loading large quantities of data into a database using the Interactive SQL INPUT command. Do not use the -b option if you are using LOAD TABLE to bulk load data.

For more information, see "-b server option" on page 133, and "Importing and Exporting Data" [ASA SQL User's Guide, page 521].

◆ Starting without a transaction log Use the -f database option for recovery—either to force the database server to start after the transaction log has been lost, or to force the database server to start using a transaction log it would otherwise not find. Note that -f is a database option, not a server option.

Once the recovery is complete, you should stop your server and restart without the -f option.

For more information, see "The database server" on page 124.

Controlling threading from the command line

Adaptive Server Anywhere can use multiple operating system threads to handle requests. This allows different requests to run simultaneously on separate CPUs. Each request runs on a single thread and no request is executed concurrently on multiple CPUs.

For a list of options that control threading, see "Controlling threading behavior" on page 15.

Threading in Adaptive Server Anywhere

To understand how threading support works, you need to understand **requests** and **tasks**.

RequestsSuppose Adaptive Server Anywhere is being used concurrently by two
users, or by two connections from a single application. Each connection
submits a query (or other SQL statement) to Adaptive Server Anywhere.
Each of these SQL statements is a separate request to the server.

TasksA task picks up a request and handles that request until it is complete. On
Windows NT/2000/XP, tasks are lightweight threads called fibers; on all
other platforms, tasks are threads. The number of tasks determines the
number of requests that Adaptive Server Anywhere can handle concurrently.

Tasks on Windows 95/98/Me, NetWare, and UNIX

On Windows 95/98/Me, NetWare, and UNIX machines, each task is an operating system thread. When Adaptive Server Anywhere receives a request, an operating system thread picks up the request. The operating system thread runs until the request is complete.

If the server receives a request while the first request is running, the second request is assigned to a different thread. If a thread is blocked while completing a task, it does not pick up another request; instead, it waits until it can complete the current request.

Tasks on Windows NT/XP/2000

On Windows NT/2000/XP, tasks are lightweight threads called fibers. When Adaptive Server Anywhere receives a request, a fiber picks up the request and runs until the request is complete. If the database server receives another request while the first request is running, the second request is assigned to a different fiber. If a fiber blocks while processing a request, it yields control to another fiber until it can proceed: it does not pick up another request.

Threads host fibers, and the fibers are scheduled co-operatively. A fiber must explicitly yield control to another fiber when it is waiting to process a request, for example, while waiting for an I/O operation to complete. If a fiber blocks and does not yield control, it blocks the thread that is hosting it and prevents other fibers from running on that thread. If more than one thread is hosting fibers, only the thread that is hosting the waiting fiber is blocked: other threads are still free to run fibers.

You can set the number of threads used to run fibers with the -gx option. Unless you are using Java or Remote Data Access, you should need only one thread per CPU.

Setting the number of threads when using Java or Remote Data Access blocks, that fiber may not yield control to another fiber, which in turn blocks the thread. For this reason, the number of operating system threads assigned to the database server defaults to one more thread than the number of CPUs on the machine. This ensures that there is at least one thread available to host fibers if the thread being used by Java or Remote Data Access is blocked. Specifying a higher number of threads than the default by using the -gx option has a minimal impact on performance.

Controlling threading behavior

There are four database server options that control threading behavior. Not all of these options are required on every platform.

Number of tasks The -gn option controls the number of tasks used to process requests. Effectively, this is the number of requests that can be handled concurrently. Each request uses a task. When there are more requests than there are tasks, any outstanding requests must wait until a currently-running task completes. By default, there are 20 tasks for the network database server and 10 tasks for the personal database server. There is no benefit to setting a number of tasks that is greater than the maximum number of server connections.

For more information, see "-gn server option" on page 148.

 Stack size per internal execution thread You can set the stack size per internal execution thread in the server using the -gss option. The -gss option allows you to lower the memory usage of the database server in environments with limited memory. This option has no effect on Windows operating systems.

For more information, see "-gss server option" on page 149.

 Number of processors If you have more than one processor, you can control how many processors the threads exploit by specifying the -gt option. By default, all processors available on the machine are used.

- For more information, see "-gt server option" on page 150.
- Number of threads assigned to the database server process On Windows NT/2000/XP, the -gx option controls the number of threads that are dedicated to hosting fibers to service requests. By default, this is set to one more than the number of CPUs on the machine. On UNIX, NetWare, and Windows 95/98/Me where each task is its own thread, this option is not needed.

For more information, see "-gx server option" on page 150.

Selecting communications protocols

	Any communication between a client application and a database server requires a communications protocol. Adaptive Server Anywhere supports a set of communications protocols for communications across networks and for same-machine communications.
	By default, the database server starts up all available protocols. You can limit the protocols available to a database server using the -x option. On the client side, many of the same options can be controlled using the CommLinks (LINKS) connection parameter.
	For more information on running the server using these options, see "Supported network protocols" on page 94.
Available protocols for	The personal database server (dbeng9.exe) supports the following protocols:
the personal server	• Shared memory This protocol is for same-machine communications, and always remains available. It is available on all platforms.
	◆ TCP/IP This protocol is for same-machine communications only, from TDS clients, Open Client, or the jConnect JDBC driver. You must run TCP/IP if you wish to connect from Open Client or jConnect.
	For more information on TDS clients, see "Adaptive Server Anywhere as an Open Server" on page 109.
	◆ Named Pipes This protocol is provided on Windows NT only. Named Pipes is for same machine communications for applications that wish to run under a certified security environment.
Available protocols for	The network database server (<i>dbsrv9.exe</i>) supports the following protocols:
the network server	• Shared memory This protocol is for same-machine communications, and always remains available. It is available on all platforms.
	• SPX This protocol is supported on all platforms except for UNIX.
	• TCP/IP This protocol is supported on all platforms.

♦ Named Pipes This protocol is supported on Windows NT only. Named Pipes is for same machine communications for applications that wish to run under a certified security environment.

Specifying protocols You can instruct a server to use only some of the available network protocols when starting up, using the -x option. The following command starts the asademo database using the TCP/IP and SPX protocols:

```
dbsrv9 -x "tcpip,spx" path\asademo.db
```

Although not strictly required in this example, the quotes are necessary if there are spaces in any of the arguments to -x.

You can add additional parameters to tune the behavior of the server for each protocol. For example, the following command (typed all on one line) instructs the server to use two network cards, one with a specified port number.

```
dbsrv9 -x "tcpip{MyIP=192.75.209.12:2367,192.75.209.32}" path\ asademo.db
```

For more information about available network communications parameters that can serve as part of the -x option, see "Network communications parameters" on page 201.

Stopping the database server

You can stop the database server by:

- Clicking Shutdown on the database server window.
- Using the *dbstop* utility.

The *dbstop* utility is particularly useful in batch files, or for stopping a server on another machine. It requires a connection string in its command.

For more information on *dbstop* options, see "Stopping a database server using the dbstop command-line utility" on page 525.

- Letting it shut down automatically by default when the application disconnects. (This only works if the server is a personal server started by an application connection string.)
- Pressing Q when the server display window has the focus on UNIX or NetWare machines.

Examples

To stop a server using the dbstop utility

1. Start a server. For example, the following command executed from the Adaptive Server Anywhere installation directory starts a server named Ottawa using the sample database:

dbsrv9 -n Ottawa asademo.db

2. Stop the server using *dbstop*:

dbstop -c "eng=Ottawa;uid=DBA;pwd=SQL"

Who can stop the server?

When you start a server, you can use the -gk option to set the level of permissions required for users to stop the server with *dbstop*. The default level of permissions required is **DBA**, but you can also set the value to **all** or **none**. (Interactively, of course, anybody at the machine can click Shutdown on the server window.)

Shutting down operating system sessions

If you close an operating system session where a database server is running, or if you use an operating system command to stop the database server, the server shuts down, but not cleanly. The next time the database loads, recovery is required, and happens automatically.

For more information about recovery, see "Backup and Data Recovery" on page 337.

It is better to stop the database server explicitly before closing the operating system session. On NetWare, however, shutting down the NetWare server machine properly does stop the database server cleanly.

Examples of commands that will not stop a server cleanly include:

- Stopping the process in the Windows Task Manager.
- Using a UNIX slay or kill command.

Starting and stopping databases

A database server can have more than one database loaded at a time. You can start databases and start the server at the same time, as follows:

dbeng9 asademo sample

	Caution The database file must be on the same machine as the database server. Managing a database file that is located on a network drive can lead to file corruption.
Starting a database on a running server	You can also start databases after starting a server in one of the following ways:
	• Connect to a database using a DatabaseFile (DBF) connection parameter while connected to a server. The DatabaseFile (DBF) connection parameter specifies a database file for a new connection. The database file is started on the current server.
	For more information, see "Connecting to an embedded database" on page 47, or "DatabaseFile connection parameter [DBF]" on page 183.
	• Use the START DATABASE statement, or choose Start Database from the File menu in Sybase Central when you have a server selected.
	For more information, see "START DATABASE statement" [ASA SQL Reference, page 566].
Limitations	• The server holds database information in memory using pages of a fixed size. Once a server has been started, you cannot start a database that has a larger page size than the server.
	• The -gd server option decides the permissions required to start databases.
Stopping a database	You can stop a database by:
	• Disconnecting from a database started by a connection string. Unless you explicitly set the AutoStop (ASTOP) connection parameter to <i>NO</i> , this happens automatically.
	For more information, see "AutoStop connection parameter [ASTOP]" on page 177.
	• Using the STOP DATABASE statement from Interactive SQL or Embedded SQL.
	For more information, see "STOP DATABASE statement" [ASA SQL Reference, page 575].
Running the server outside the current session

When you log on to a computer using a user ID and a password, you establish a **session**. When you start a database server, or any other application, it runs within that session. When you log off the computer, all applications associated with the session terminate.

It is common to require database servers to be available all the time. To make this easier, you can run Adaptive Server Anywhere for Windows NT/2000/XP and for UNIX in such a way that, when you log off the computer, the database server remains running. The way you do this depends on your operating system.

♦ UNIX daemon You can run the UNIX database server as a daemon using the -ud option, enabling the database server to run in the background, and to continue running after you log off.

For more information, see "Running the UNIX database server as a daemon" on page 21.

• Windows service You can run the Windows database server as a service. This has many convenient properties for running high availability servers.

For more information, see "Understanding Windows services" on page 23.

Running the UNIX database server as a daemon

To run the UNIX database server in the background, and to enable it to run independently of the current session, you run it as a **daemon**.

Do not use '&' to run the database server in the background

If you use the UNIX & (ampersand) command to run the database server in the background, it will not work—the server will hang. You must instead run the database server as a daemon.

As well, attempting to start a server in the background from within a program using the typical fork()-exec() sequence will not work.

You can run the UNIX database server as a daemon in one of the following ways:

1. Use the -ud option when starting the database server. For example:

dbsrv9 -ud asademo

2. Use the *dbspawn* tool to start the database server, for example:

dbspawn dbsrv9 asademo

One advantage of using *dbspawn* is that the *dbspawn* process does not terminate until it has confirmed that the daemon has started and is ready to accept requests. If for any reason the daemon fails to start, the exit code for *dbspawn* will be non-zero.

When you start the daemon directly using the -ud option, the *dbeng9* and *dbsrv9* commands create the daemon process and return immediately (exiting and allowing the next command to be executed) before the daemon initializes itself or attempts to open any of the databases specified in the command.

Using *dbspawn* can be useful when writing a script or other automated process that needs to start a server as a daemon prior to running one or more applications that will user the server because you want to ensure that the daemon is running before starting the applications. The following is an example of how to test this using a csh script.

This example uses an sh script to test whether the daemon is running before starting the applications.

3. Spawn a daemon from within a C program, for example:

```
if( fork() == 0 ) {
   /* child process = start server daemon */
   execl( "/opt/sybase/SYBSsa9/bin/dbsrv9", "dbsrv9", "-ud",
        "asademo" );
   exit(1);
}
/* parent process */
....
```

Note that the -ud option is used.

For more information, see "-ud server option" on page 162 and "The Spawn utility" on page 523.

Understanding Windows services

Although you can run the database server like any other Windows NT/2000/XP program rather than as a service, there are limitations to running it as a standard program, particularly in multi-user environments. When you start a program, it runs under your Windows NT/2000/XP login Limitations of running as session, which means that if you log off the computer, the program a standard executable terminates. Only one person logs onto Windows NT/2000/XP (on any one computer) at one time. This restricts the use of the computer if you wish to keep a program running much of the time, as is commonly the case with database servers. You must stay logged onto the computer running the database server for the database server to keep running. This can also present a security risk as the Windows NT/2000/XP computer must be left in a logged on state. Advantages of services Installing an application as a Windows service enables it to run even when you log off. When you start a service, it logs on using a special system account called LocalSystem (or using another account that you specify). Since the service is not tied to the user ID of the person starting it, the service remains open even when the person who started it logs off. You can also configure a service to start automatically when the Windows computer starts, before a user logs on. Managing services Sybase Central provides a more convenient and comprehensive way of managing Adaptive Server Anywhere services than the Windows services manager.

Programs that can be run as Windows services

You can run the following programs as services:

• Network Database Server (*dbsrv9.exe*)

- Personal Database Server (*dbeng9.exe*)
- SQL Remote Message Agent (*dbremote.exe*)
- The MobiLink Synchronization server (*dbmlsrv9.exe*)
- A sample application

Not all these applications are supplied in all editions of Adaptive Server Anywhere.

Managing services

You can carry out the following service management tasks from the command line, or on the Services tab in Sybase Central:

- Add, edit, and remove services.
- Start, stop, and pause services.
- Modify the parameters governing a service.
- Add databases to a service so you can run several databases at one time.

The service icons in Sybase Central display the current state of each service using a traffic light icon (running, paused, or stopped).



Adding a service

This section describes how to set up services using Sybase Central and the Service Creation utility.

To add a new service (Sybase Central)

- 1. In the left pane, select the Adaptive Server Anywhere plug-in.
- 2. In the right pane, click the Services tab.
- 3. Click the Add Service button in the toolbar.
- 4. Follow the instructions in the wizard.

To add a new service (Command line)

- 1. Open a command prompt.
- 2. Execute the Service Creation utility using the -w option.

For example, to create a personal server service called myserv, which starts the specified engine with the specified parameters type the following. The engine runs as the LocalSystem user, enter the following command, all on one line:

```
dbsvc -as -w myserv
"C:\Program Files\Sybase\SQL Anywhere 9\win32\dbeng9.exe"
-n william -c 8m "C:\Program Files\Sybase\SQL Anywhere 9\
        sample.db"
```

For more information about the Service Creation utility and options, see "The Service Creation utility" on page 519.

- Service names must be unique within the first eight characters.
- If you choose to start a service automatically, it starts whenever the computer starts Windows. If you choose to start the service manually, you need to start the service from Sybase Central each time. You may want to select Disabled if you are setting up a service for future use.
- Type options for the executable, without the executable name itself, in the window. For example, if you want a network server to run using the sample database with a cache size of 20 Mb and the name **myserver**, you would type the following in the Parameters box of the Service Creation wizard in Sybase Central:

```
-c 20M
-n myserver c:\Program Files\Sybase\SQL Anywhere 9\
asademo.db
```

Line breaks are optional.

For information on valid options, see the description of each program in "Database Administration Utilities" on page 455.

Notes

• Choose the account under which the service will run: the special LocalSystem account or another user ID.

For more information about this choice, see "Setting the account options" on page 28.

 If you want the service to be accessible from the Windows desktop, check Allow Service to Interact with Desktop. If this option is cleared, no icon or window appears on the desktop.

For more information on the configuration options, see "Configuring services" on page 26.

Removing a service

Removing a service removes the server name from the list of services. Removing a service does not remove any software from your hard disk.

If you wish to re-install a service you previously removed, you need to re-type the options.

To remove a service (Sybase Central)

1. In the left pane, select the Adaptive Server Anywhere plug-in.

In the right pane, click the Services tab.

2. In the right pane, right-click the icon of the service you want to remove and choose Delete from the popup menu.

To remove a service (Command line)

- 1. Open a command prompt.
- 2. Execute the Service Creation utility using the -d option.

For example, to delete the service called myserv, without prompting for confirmation, type the following command:

dbsvc -y -d myserv

For more information about the Service Creation utility and options, see "The Service Creation utility" on page 519.

Configuring services

A service runs a database server or other application with a set of options.

For a full description of the options for each of the administration utilities, see "Database Administration Utilities" on page 455.

In addition to the options, services accept other parameters that specify the account under which the service runs and the conditions under which it starts.

To change the parameters for a service

- 1. In the left pane, select the Adaptive Server Anywhere plug-in.
- 2. In the right pane, right-click the service you want to change and choose Properties from the popup menu.
- 3. Alter the parameters as needed on the tabs of the Service property sheet.
- 4. Click OK when finished.

Changes to a service configuration take effect the next time someone starts the service. The Startup option is applied the next time Windows is started.

Setting the startup option

The following options govern startup behavior for Adaptive Server Anywhere services. You can set them on the General tab of the Service property sheet.

- Automatic If you choose the Automatic setting, the service starts whenever the Windows operating system starts. This setting is appropriate for database servers and other applications running all the time.
- Manual If you choose the Manual setting, the service starts only when a user with Administrator permissions starts it. For information about Administrator permissions, see your Windows documentation.
- Disabled If you choose the Disabled setting, the service will not start.

Specifying options

The Configuration tab of the Service property sheet provides a Parameters text box for specifying options for a service. Do not type the name of the program executable in this box.

Examples
 To start a network server service running two databases, with a cache size of 20 Mb, and with a name of my_server, you would type the following in the Parameters box:

-c 20M -n my_server c:\Program Files\Sybase\SQL Anywhere 9\db_1.db c:\Program Files\Sybase\SQL Anywhere 9\db_2.db • To start a SQL Remote Message Agent service, connecting to the sample database as user ID DBA, you would type the following:

```
-c "uid=DBA;pwd=SQL;dbn=asademo"
```

The following figure illustrates a sample property sheet.

ServiceTest2 Service	e Properties	×
General Configuration	Account Dependencies Polling	1
This service will run the	following executable:	
File name: C:\Progr	am Files/Subace/SOL Anywhere 91	wip32\dbcrv9.eve
nienalie. Jerrogi	ann nestavoasetadr wuxwuere str	Carrow Carrows
		Browse
Parameters:		
	OK Cancel	Apply Help

The options for a service are the same as those for the executable. For a full description of the options for each program, see "The Database Server" on page 123.

Setting the account options

You can choose under which account the service runs. Most services run under the special LocalSystem account, which is the default option for services. You can set the service to log on under another account by opening the Account tab on the Service property sheet, and typing the account information.

If you choose to run the service under an account other than LocalSystem, that account must have the "Log On As A Service" privilege. This can be

granted from the Windows User Manager application under Advanced Privileges.

When an icon appearsWhether or not an icon for the service appears on the taskbar or desktopon the taskbardepends on the account you select, and whether Allow Service to Interactwith Desktop is selected, as follows:

- If a service runs under LocalSystem, and Allow Service to Interact with Desktop is selected on the Service property sheet, an icon appears on the desktop of every user logged in to Windows on the computer running the service. Consequently, any user can open the application window and stop the program running as a service.
- If a service runs under LocalSystem, and Allow Service to Interact with Desktop is cleared on the Service property sheet, no icon appears on the desktop for any user. Only users with permissions to change the state of services can stop the service.
- If a service runs under another account, no icon appears on the desktop. Only users with permissions to change the state of services can stop the service.

Changing the executable file

To change the program executable file associated with a service, click the Configuration tab on the Service property sheet and type the new path and file name in the File Name text box.

If you move an executable file to a new directory, you must modify this entry.

Adding new databases to a service

Each network server or personal server can run more than one database. If you wish to run more than one database at a time, we recommend that you do so by attaching new databases to your existing service, rather than by creating new services.

To add a new database to a service

- 1. In the left pane, select the Adaptive Server Anywhere plug-in.
- 2. In the right pane, click the Services tab.
- 3. Right-click the service and choose Properties from the popup menu.
- 4. Click the Configuration tab.
- 5. Add the path and filename of the new database to the end of the list of options in the Parameters box.

6. Click OK to save the changes.

The new database starts the next time the service starts.

Databases can be started on running servers by client applications, such as Interactive SQL.

For more information about how to start a database on a server from Interactive SQL, see "START DATABASE statement" [*ASA SQL Reference*, page 566].

For more information about how to implement this function in an Embedded SQL application, see "db_start_database function" [ASA *Programming Guide*, page 216].

Starting a database from an application does not attach it to the service. If the service is stopped and restarted, the additional database will not be started automatically.

Setting the service polling frequency

Sybase Central can poll at specified intervals to check the state (started, stopped, or paused) of each service, and update the icons to display the current state. By default, polling is off. If you leave it off, you must click Refresh to see changes to the state.

* To set the Sybase Central polling frequency

- 1. In the left pane, click the Adaptive Server Anywhere plug-in.
- 2. In the right pane, click the Services tab.
- 3. Right-click the service and choose Properties from the popup menu.
- 4. Click the Polling tab.
- 5. Select Enable Polling.
- 6. Set the polling frequency.

The frequency applies to all services, not just the one selected. The value you set in this window remains in effect for subsequent sessions, until you change it.

7. Click OK.

Starting, stopping, and pausing services

To start, stop, or pause a service

- 1. In the left pane, select the Adaptive Server Anywhere plug-in.
- 2. Right-click the service and choose Start, Stop, or Pause from the popup menu.

To resume a paused service, right-click the service and choose Continue from the popup menu.

If you start a service, it keeps running until you stop it. Closing Sybase Central or logging off does not stop the service.

Stopping a service closes all connections to the database and stops the database server. For other applications, the program closes down.

Pausing a service prevents any further action being taken by the application. It does not shut the application down or (in the case of server services) close any client connections to the database. Most users do not need to pause their services.

The Windows Service Manager

You can use Sybase Central to carry out all the service management for Adaptive Server Anywhere. Although you can use the Windows NT Service Manager in the Control Panel for some tasks, you cannot install or configure an Adaptive Server Anywhere service from the Windows NT Service Manager.

If you open the Windows Service Manager (from the Windows Control Panel), a list of services appears. The names of the Adaptive Server Anywhere services are formed from the Service Name you provided when installing the service, prefixed by Adaptive Server Anywhere. All the installed services appear together in the list.

Running more than one service

This section describes some topics specific to running more than one service at a time.

Service dependencies

In some circumstances you may wish to run more than one executable as a service, and these executables may depend on each other. For example, you may wish to run a server and a SQL Remote Message Agent or Log Transfer Manager to assist in replication.

In cases such as these, the services must start in the proper order. If a

SQL Remote Message Agent service starts up before the server has started, it fails because it cannot find the server.

You can prevent these problems by using **service groups**, which you manage from Sybase Central.

Service groups overview

You can assign each service on your system to be a member of a service group. By default, each service belongs to a group, as listed in the following table.

Service	Default group
Network server	ASANYServer
Personal server	ASANYEngine
SQL Remote Message Agent	ASANYRemote
MobiLink Synchronization Server	ASANYMobiLink
Replication Agent	ASANYLTM

Before you can configure your services to ensure that they start in the correct order, you must check that your service is a member of an appropriate group. You can check which group a service belongs to, and change this group, from Sybase Central.

To check and change which group a service belongs to

- 1. In the left pane, select the Adaptive Server Anywhere plug-in.
- 2. In the right pane, click the Services tab.
- 3. Right-click the service and choose Properties from the popup menu.
- 4. Click the Dependencies tab. The top text box displays the name of the group the service belongs to.
- 5. Click Change to display a list of available groups on your system.
- 6. Select one of the groups, or type a name for a new group.
- 7. Click OK to assign the service to that group.

Managing service dependencies

With Sybase Central you can specify **dependencies** for a service. For example:

- You can ensure that at least one member of each of a list of service groups has started before the current service.
- You can ensure that any number of services start before the current service. For example, you may want to ensure that a particular network server has started before a SQL Remote Message Agent that is to run against that server starts.

* To add a service or group to a list of dependencies

- 1. In the left pane, select the Adaptive Server Anywhere plug-in.
- 2. Right-click the service and choose Properties from the popup menu.
- 3. Click the Dependencies tab.
- 4. Click Add Services or Add Service Groups to add a service or group to the list of dependencies.
- 5. Select one of the services or groups from the list.
- 6. Click OK to add the service or group to the list of dependencies.

Troubleshooting server startup

This section describes some common problems that may occur when starting the database server.

Ensure that your transaction log file is valid

The server won't start if the existing transaction log is invalid. For example, during development you may replace a database file with a new version, without deleting the transaction log at the same time. This causes the transaction log file to be different than the database, and results in an invalid transaction log file.

Ensure that you have sufficient disk space for your temporary file

Adaptive Server Anywhere uses a temporary file to store information while running. This file is stored in the directory pointed to by the TMP or TEMP environment variable, typically *c*:*temp*.

If you do not have sufficient disk space available to the temporary directory, you will have problems starting the server.

Ensure that network communication software is running

Appropriate network communication software must be installed and running before you run the database server. If you are running reliable network software with just one network installed, this should be straightforward. If you experience problems, if you are running non-standard software, or if you are running multiple networks, you may want to read the full discussion of network communication issues in "Client/Server Communications" on page 93.

You should confirm that other software requiring network communications is working properly before running the database server.

If you are running under the TCP/IP protocol, you may want to confirm that ping and telnet are working properly. The ping and telnet applications are provided with many TCP/IP protocol stacks.

Debugging network communications startup problems

If you are having problems establishing a connection across a network, you can use debugging options at both the client and the server to diagnose problems. On the server, you use the -z option. The startup information appears on the database server window: you can use the -o option to log the

results to an output file.

Make sure you're using the right asasrv.ini file

If you are having problems establishing a connection to the correct server across a network, try deleting the *asasrv.ini* file. This file contains server information, including server name, protocol and address. It is possible that the server information in this file is overriding information you specified in the connection string. Deleting this file causes Adaptive Server Anywhere to create a new *asasrv.ini* file containing the information you specify in the connection string. The *asasrv.ini* file should be located in your Adaptive Server Anywhere executable directory (the same directory as the ODBC/DBLib DLL).

If you continue to experience problems establishing a connection, you should also delete any copy of *asasrv.ini* located in any of the following places:

- the win32 or win64 subdirectory of your SQL Anywhere installation directory (you can find this in the directory listed in the HKEY_LOCAL_MACHINE\SOFTWARE\Sybase\Adaptive Server Anywhere\9.0\Location registry key)
- Windows directory
- Windows system directory
- anywhere else in your path

Create a debug logfile

You can use the LogFile connection parameter to create a debug log file. Log files can provide more detailed information about where a connection failure occurred, thereby helping you troubleshoot and correct the problem. For more information about log files, see "Logfile connection parameter [LOG]" on page 194.

CHAPTER 2

Connecting to a Database

About this chapter	This chapter describes how client applications connect to data contains information about connecting to databases from ODE and embedded SQL applications. It also describes connecting Central and Interactive SQL.	bases. It 3C, OLE DB, from Sybase
	For more information on connecting to a database from S Client applications, see "Adaptive Server Anywhere as an Ope page 109.	ybase Open en Server" on
	For more information on connecting via JDBC (if you are in Sybase Central or Interactive SQL), see "JDBC Programmi Programming Guide, page 103].	not working ng" [<i>ASA</i>
Contents	Торіс:	page
	Introduction to connections	38
	Connecting from Sybase Central or Interactive SQL	42
	Simple connection examples	45
	Working with ODBC data sources	53
	Connecting from desktop applications to a Windows CE database	65
	Connecting to a database using OLE DB	68
	Connection parameters	70
	Troubleshooting connections	75
	Using integrated logins	85

Introduction to connections

Any client application that uses a database must establish a **connection** to that database before any work can be done. The connection forms a channel through which all activity from the client application takes place. For example, your user ID determines permissions to carry out actions on the database—and the database server has your user ID because it is part of the request to establish a connection.

How connections are Eastablished To establish a connection, the client application calls functions in one of the Adaptive Server Anywhere interfaces. Adaptive Server Anywhere provides the following interfaces:

- **ODBC** ODBC connections are discussed in this chapter.
- **OLE DB** OLE DB connections are discussed in this chapter.
- Embedded SQL Embedded SQL connections are discussed in this chapter.
- Sybase Open Client Open Client connections are not discussed in this chapter.

For information on connecting from Open Client applications, see "Adaptive Server Anywhere as an Open Server" on page 109.

◆ JDBC Sybase Central and Interactive SQL have the connection logic described in this chapter built into them. Other JDBC applications cannot use the connection logic discussed in this chapter.

For more information on connecting via JDBC, see "JDBC Programming" [*ASA Programming Guide*, page 103].

The interface uses connection information included in the call from the client application, perhaps together with information held on disk in a file data source, to locate and connect to a server running the required database. The following figure is a simplified representation of the pieces involved.



What to read

The following table identifies where you can find answers to questions.

Consider reading
"Connecting from Sybase Central or Interactive SQL" on page 42
"Simple connection examples" on page 45
"Working with ODBC data sources" on page 53
"Connection parameters" on page 70
"Troubleshooting connections" on page 75
"Client/Server Communications" on page 93
"Connection strings and character sets" on page 318

How connection parameters work

When an application connects to a database, it uses a set of **connection parameters** to define the connection. Connection parameters include information such as the server name, the database name, and a user ID.

A keyword-value pair (of the form *parameter=value*) specifies each

connection parameter. For example, you specify the password connection parameter for the default password as follows:

Password=SQL

Connection parameters are assembled into **connection strings**. In a connection string, a semicolon separates each connection parameter, as follows:

ServerName=asademo9;DatabaseName=asademo

There are a number of connection parameters which affect how an engine is autostarted. It is recommended that you use the following connection parameters instead of providing the corresponding server options within the StartLine (START) connection parameter:

- EngineName (ENG)
- DatabaseFile (DBF)
- DatabaseSwitches (DBS)
- ♦ DatabaseName (DBN)

Representing connectionThis chapter has many examples of connection strings represented in the
following form:

```
parameter1=value1
parameter2=value2
```

This is equivalent to the following connection string:

parameter1=value1;parameter2=value2

You must type a connection string on a single line with the parameter settings separated by semicolons.

Connection parameters passed as connection strings

Connection parameters are passed to the interface library as a **connection string**. This string consists of a set of parameters, separated by semicolons:

parameter1=value1;parameter2=value2;...

In general, the connection string built up by an application and passed to the interface library does not correspond directly to the way a user enters the information. Instead, a user may fill in a dialog, or the application may read connection information from an initialization file.

Many of the Adaptive Server Anywhere utilities accept a connection string as the -c option and pass the connection string on to the interface library without change. For example, the following is a typical Collation utility (*dbcollat*) command line (which should be typed all on one line):

dbcollat -c "uid=DBA;pwd=SQL;dbn=asademo" c:\temp\asademo.col

Interactive SQL connection strings

Interactive SQL processes the connection string internally. These utilities do not simply pass on the connection parameters to the interface library. Do not use Interactive SQL to test command strings from the command prompt.

Saving connection parameters in ODBC data sources

Many client applications, including application development systems, use the ODBC interface to access Adaptive Server Anywhere. When connecting to the database, ODBC applications typically use ODBC data sources. An ODBC data source is a set of connection parameters, stored in the registry or in a file.

For more information on ODBC data sources, see "Working with ODBC data sources" on page 53.

For Adaptive Server Anywhere, ODBC data sources can be used not only by ODBC applications on Windows, but also by other applications:

- Adaptive Server Anywhere client applications on UNIX can use ODBC data sources, as well as those on Windows operating systems. On UNIX, the data source is stored as a file.
- Adaptive Server Anywhere client applications using the OLE DB or embedded SQL interfaces can use ODBC data sources, as well as ODBC applications.
- Interactive SQL and Sybase Central can use ODBC data sources.

Connecting from Sybase Central or Interactive SQL

To use Sybase Central or Interactive SQL for managing your database, you must first connect to it. In the Connect dialog, you tell Sybase Central or Interactive SQL what database you want to connect to, where it is located, and how you want to connect to it.

The connecting process depends on your situation. For example, if you have a server already running on your machine and this server contains only one database, all you have to do in the Connect dialog is provide a user ID and a password. Sybase Central or Interactive SQL then knows to connect immediately to the database on the running server.

If this running server has more than one database loaded on it, if it is not yet running, or if it is running on another machine, you need to provide more detailed information in the Connect dialog so that Sybase Central or Interactive SQL knows which database to connect to.

This section describes how to access the Connect dialog in Sybase Central and Interactive SQL.

For more information about connection examples, including examples for Sybase Central and Interactive SQL, see "Simple connection examples" on page 45.

Opening the Connect dialog

A common Connect dialog is available in both Sybase Central and Interactive SQL to let you connect to a database.

When you start Sybase Central, you need to manually display this dialog. When you start Interactive SQL, the dialog automatically appears; you can also make it appear for a new connection by choosing Window \succ New Window.

To open the Connect dialog (Sybase Central)

1. In Sybase Central, choose Tools \succ Connect.

If you have more than one Sybase Central plug-in installed, choose Adaptive Server Anywhere from the displayed list.

or Click the Connect button on the main toolbar.

or Press F11.

Тір

You can make subsequent connections to a given database easier and faster using a "connection profile" [*ASA Getting Started*, page 156].

To open the Connect dialog (Interactive SQL)

1. In Interactive SQL, choose SQL ➤ Connect.

or Press F11.

Once the Connect dialog appears, you must specify the connection parameters you need to connect. For example, you can connect to the Adaptive Server Anywhere sample database by choosing ASA 9.0 Sample from the ODBC Data Source Name list and clicking OK.

Specifying a driver for your connection

	When you are working with a database, all your requests and commands go through a driver to the database itself. Sybase Central and Interactive SQL support two main JDBC drivers: jConnect and the iAnywhere JDBC driver. Both are included with Adaptive Server Anywhere.
	The iAnywhere JDBC driver is, in many cases, the preferred driver. It is the default driver for Interactive SQL and Sybase Central. Sybase jConnect is a platform-independent JDBC driver.
	When you connect to a database in the Connect dialog, you can choose which driver you want to use for the connection. This is an optional configuration; the iAnywhere JDBC driver is the preferred driver and is automatically used for all connections unless you specify otherwise.
	For more information on JDBC drivers, see "Choosing a JDBC driver" [ASA <i>Programming Guide</i> , page 104], "Using the jConnect JDBC driver" [ASA <i>Programming Guide</i> , page 110], and "Working with ODBC data sources" on page 53.
Data sources and the jConnect driver	As a general rule, the jConnect driver cannot use ODBC data sources. However, Sybase Central and Interactive SQL are special cases. When you use the jConnect driver in either of them, you can specify an ODBC data source to establish a connection. For example, you can connect to the sample database using the ASA 9.0 Sample data source, even if you are using the jConnect driver.
	This customized functionality is only available while you are working in Sybase Central or Interactive SQL. If you are constructing a JDBC application, do not try to use a data source to connect to a database.

To specify a driver for the connection

- 1. In Sybase Central, choose Tools \succ Connect to open the Connect dialog.
- 2. Configure the necessary settings on the Identification and Database tabs of the dialog.
- 3. On the Advanced tab of the dialog, select either jConnect5 or iAnywhere JDBC driver.

Working with the Connect dialog

The Connect dialog lets you define parameters for connecting to a server or database. The same dialog is used in both Sybase Central and Interactive SQL.

The Connect dialog has the following tabs:

- The Identification tab lets you identify yourself to the database and specify a data source.
- The Database tab lets you identify a server and/or database to connect to.
- The Advanced tab lets you add additional connection parameters and specify a driver for the connection.

In Sybase Central, after you connect successfully, the database name appears in the left pane of the main window, under the server that it is running on. The user ID for the connection appears after the database name.

In Interactive SQL, the connection information (including the database name, your user ID, and the database server) appears in the title bar above the SQL Statements pane.

Simple connection examples

Although the connection model for Adaptive Server Anywhere is configurable, and can become complex, in many cases connecting to a database is very simple.

Who should read this
section?This section describes some simple cases of applications connecting to an
Adaptive Server Anywhere database. This section may be all you need to get
started.

For more information about available connection parameters and their use, see "Connection parameters" on page 70.

Connecting to the sample database from Sybase Central or Interactive SQL

Many examples and exercises throughout the documentation start by connecting to the sample database from Sybase Central or Interactive SQL.

To connect to the sample database (Sybase Central)

- 1. To start Sybase Central: from the Start menu, choose Programs ➤ Sybase SQL Anywhere 9 ➤ Sybase Central.
- 2. To open the Connect dialog: from the Tools menu, choose Connect.
- 3. Select the ODBC Data Source Name option and click Browse.
- 4. Select ASA 9.0 Sample and click OK.

To connect to the sample database (Interactive SQL)

- To start Interactive SQL: from the Start menu, choose Programs ➤ Sybase
 ➤ SQL Anywhere 9 ➤ Adaptive Server Anywhere ➤ Interactive SQL.
- 2. To open the Connect dialog: from the SQL menu, choose Connect.
- 3. Select the ODBC Data Source Name option and click Browse.
- 4. Select ASA 9.0 Sample and click OK.

The database name, user ID, and server name appear in the title bar above the SQL Statements pane.

Note You do not need to type a user ID and a password for this connection because the data source already contains this information.

Connecting to a database on your own machine from Sybase Central or Interactive SQL

The simplest connection scenario is when the database you want to connect to resides on your own machine. If this is the case for you, ask yourself the following questions:

- Is the database already running on a server? If so, you can specify fewer parameters in the Connect dialog. If not, you need to identify the database file so that Sybase Central or Interactive SQL can start it for you.
- Are there multiple databases running on your machine? If so, you need to tell Sybase Central or Interactive SQL which database in particular to connect to. If there is only one, Sybase Central or Interactive SQL assumes that it is the one you want to connect to, and you don't need to specify it in the Connect dialog.

The procedures below depend on your answers to these questions.

To connect to a database on an already-running local server

- 1. Start Sybase Central or Interactive SQL and open the Connect dialog (if it doesn't appear automatically).
- 2. On the Identification tab of the dialog, type a user ID and a password.
- 3. Do one of the following:
 - If the server only contains the one database, click OK to connect to it.
 - If the server contains multiple databases, click the Database tab of the dialog and specify a database name. This is usually the database file name, without the path or extension.

To start and connect to a database

- 1. Start Sybase Central or Interactive SQL and open the Connect dialog (if it doesn't appear automatically).
- 2. On the Identification tab of the dialog, type a user ID and a password.
- 3. Click the Database tab of the dialog.
- 4. Specify a file in the Database File field (including the full path, name, and extension). You can search for a file by clicking Browse.
- 5. If you want the database name for subsequent connections to be different from the file name, type a name in the Database Name field (without including a path or extension).

Tips

If the database is already loaded (started) on the server, you only need to provide a database name for a successful connection. The database file is not necessary.

You can connect using a data source (a stored set of connection parameters) for either of the above scenarios by selecting the appropriate data source option at the bottom of the Identification tab of the Connect dialog.

For information about using data sources in conjunction with the JDBC driver (jConnect), see "Specifying a driver for your connection" on page 43.

See also

- "Opening the Connect dialog" on page 42
- "Simple connection examples" on page 45

Connecting to an embedded database

	An embedded database , designed for use by a single application, runs on the same machine as the application and is largely hidden from the application user.
	When an application uses an embedded database, the personal server is generally not running when the application connects. In this case, you can start the database using the connection string, and by specifying the database file in the DatabaseFile (DBF) parameter of the connection string.
Using the DBF parameter	The DatabaseFile (DBF) parameter specifies which database file to use. The database file automatically loads onto the default server, or starts a server if none are running.
	The database unloads when there are no more connections to the database (generally when the application that started the connection disconnects). If the connection started the server, it stops once the database unloads.
	The following connection parameters show how to load the sample database as an embedded database:
	dbf=path\asademo.db uid=DBA pwd=SQL
	where <i>path</i> is the name of your Adaptive Server Anywhere installation directory.
Using the StartLine [Start] parameter	The following connection parameters show how you can customize the startup of the sample database as an embedded database. This is useful if you wish to use options, such as the cache size:

```
Start=dbeng9 -c 8M
dbf=path\asademo.db
uid=DBA
pwd=SQL
```

See also

- "Opening the Connect dialog" on page 42
- "Simple connection examples" on page 45

Connecting using a data source

You can save sets of connection parameters in a **data source**. ODBC and Embedded SQL applications use data sources. You can create data sources from the ODBC Administrator.

If you are constructing an application, you should only use data sources for ODBC applications. It is possible to specify data sources when you are using the JDBC driver (jConnect), but only within Sybase Central or Interactive SQL.

For more information, see "Specifying a driver for your connection" on page 43.

To connect from using a data source (Sybase Central or Interactive SQL)

- 1. Start Sybase Central or Interactive SQL and open the Connect dialog (if it doesn't appear automatically).
- 2. On the Identification tab, type a user ID and password.
- 3. On the lower half of the Identification tab, do one of the following:
 - Select the ODBC Data Source Name option and specify a data source name (equivalent to the DataSourceName (DSN) connection parameter, which references a data source in the registry). You can view a list of data sources by clicking Browse.
 - Select the ODBC Data Source File option and specify a data source file (equivalent to the FileDataSourceName (FILEDSN) connection parameter, which references a data source held in a file). You can search for a file by clicking Browse.

The ASA 9.0 Sample data source holds a set of connection parameters, including the database file and a **StartLine** (**START**) parameter to start the database.

See also

- "Opening the Connect dialog" on page 42
- "Simple connection examples" on page 45

Connecting to a server on a network

To connect to a database running on a network server somewhere on a local or wide area network, the client software must locate the database server. Adaptive Server Anywhere provides a network library to handle this task.

Network connections occur over a **network protocol**. Several protocols are supported, including TCP/IP and SPX.

For more information about client/server communications over a network, see "Client/Server Communications" on page 93.



Specifying the server

Adaptive Server Anywhere server names must be unique on a local domain for a given network protocol. The following connection parameters provide a simple example for connecting to a server running elsewhere on a network:

```
eng=svr_name
dbn=db_name
uid=user_id
pwd=password
CommLinks=all
```

The client library first looks for a personal server of the given name, and then looks on the network for a server of the given name.

The above example finds any server started using the default port number. However, you can start servers using other port numbers by providing more information in the **CommLinks** (**LINKS**) parameter. For information, see "CommLinks connection parameter [LINKS]" on page 179.

Specifying the protocol If several protocols are available, you can instruct the network library which ones to use to improve performance. The following parameters use only the TCP/IP protocol:

```
eng=svr_name
dbn=db_name
uid=user_id
pwd=password
CommLinks=tcpip
```

The network library searches for a server by broadcasting over the network, which can be a time-consuming process. Once the network library locates a server, the client library stores its name and network address in a file (*asasrv.ini*), and reuses this entry for subsequent connection attempts to that server using the specified protocol. Subsequent connections can be many times faster than a connection achieved by broadcast.

Many other connection parameters are available to assist Adaptive Server Anywhere in locating a server efficiently over a network. For more information see "Network communications parameters" on page 201.

To connect to a database on a network server (Sybase Central or Interactive SQL)

- 1. Start Sybase Central or Interactive SQL and open the Connect dialog (if it does not appear automatically).
- 2. On the Identification tab of the dialog, type a user ID and a password.
- 3. On the Database tab of the dialog, type the Server Name. You can search for a server by clicking Find.
- 4. Identify the database by specifying a Database Name.

Tips

You can connect using a data source (a stored set of connection parameters) by selecting the appropriate data source option at the bottom of the Identification tab of the Connect dialog.

For information about using data sources in conjunction with the JDBC driver (jConnect), see "Specifying a driver for your connection" on page 43.

By default, all network connections in Sybase Central and Interactive SQL use the TCP/IP network protocol.

See also

- "Opening the Connect dialog" on page 42
- "Simple connection examples" on page 45

Using default connection parameters

	You can leave many connection parameters unspecified, and instead use the default behavior to make a connection. Be cautious about relying on default behavior in production environments, especially if you distribute your application to customers who may install other Adaptive Server Anywhere applications on their machine.
Default database server and database	If a single personal server is running, with a single loaded database, you can connect using entirely default parameters:
	uid=user_id pwd=password
Default database server	If more than one database is loaded on a single personal server, you can leave the server as a default, but you need to specify the database you wish to connect to:
	dbn=db_name uid=user_id pwd=password
Default database	If more than one server is running, you need to specify which server you wish to connect to. If only one database is loaded on that server, you do not need to specify the database name. The following connection string connects to a named server, using the default database:
	eng=server_name uid=user_id pwd=password
No defaults	The following connection string connects to a named server, using a named database:
	eng=server_name dbn=db_name uid=user_id pwd=password
	For more information about default behavior, see "Troubleshooting connections" on page 75.

Connecting from Adaptive Server Anywhere utilities

All Adaptive Server Anywhere database utilities that communicate with the server (rather than acting directly on database files) do so using Embedded SQL. They follow the procedure outlined in "Troubleshooting connections" on page 75 when connecting to a database.

How database tools obtain connection parameter values Many of the administration utilities obtain the connection parameter values by:

1. Using values specified on the command line (if there are any). For example, the following command starts a backup of the default database on the default server using the user ID DBA and the password SQL:

dbbackup -c "uid=DBA;pwd=SQL" c:\backup

For more information about options for each database tool, see the chapter "Database Administration Utilities" on page 455.

2. Using the SQLCONNECT environment variable settings if any values are missing. Adaptive Server Anywhere does not set this variable automatically.

For more information about the SQLCONNECT environment variable, see "Environment variables" on page 244.

3. Prompting you for a user ID and password to connect to the default database on the default server, if parameters are not set in the command line, or the SQLCONNECT environment variable.

Working with ODBC data sources

	Microsoft Corporation defines the Open Database Connectivity (ODBC) interface, which is a standard interface for connecting client applications to database-management systems in the Windows 95/98/Me and Windows NT/2000/XP environments. Many client applications, including application development systems, use the ODBC interface to access a wide range of database systems.
Where data sources are held	You connect to an ODBC database using an ODBC data source. You need an ODBC data source on the client computer for each database you want to connect to.
	The ODBC data source contains a set of connection parameters. You can store sets of Adaptive Server Anywhere connection parameters as an ODBC data source, in either the system registry or as files.
	If you have a data source, your connection string can simply name the data source to use:
	• Data source Use the DataSourceName (DSN) connection parameter to reference a data source in the registry:
	DSN=my data source
	• File data source Use the FileDataSourceName (FILEDSN) connection parameter to reference a data source held in a file:
	FileDSN=mysource.dsn
	For Adaptive Server Anywhere, the use of ODBC data sources goes beyond Windows applications using the ODBC interface:
	• Adaptive Server Anywhere client applications on UNIX can use ODBC data sources, as well as those on Windows operating systems.
	 Adaptive Server Anywhere client applications using the OLE DB or embedded SQL interfaces can use ODBC data sources, as well as ODBC applications.
	• Interactive SQL and Sybase Central can use ODBC data sources.
Creating an ODBC	data source

You can create ODBC data sources on Windows 95/98/Me and Windows NT/2000/XP operating systems using the ODBC Administrator, which provides a central place for creating and managing ODBC data sources. Adaptive Server Anywhere also includes a cross-platform utility named *dbdsn* to create data sources.

Before you begin This section describes how to create an ODBC data source. Before you create a data source, you need to know which connection parameters you want to include in it.

For more information, see "Simple connection examples" on page 45, and "Connection parameters" on page 70.

ODBC Administrator On Windows 95/98/Me and Windows NT/2000/XP, you can use the Microsoft ODBC Administrator to create and edit data sources. You can work with User Data Sources, File Data Sources, and System Data Sources in this utility.

To create an ODBC data source (ODBC Administrator)

1. Start the ODBC Administrator:

In Sybase Central, choose Tools \succ Adaptive Server Anywhere $9 \succ$ Open ODBC Administrator.

or From the Windows Start menu, choose Programs ➤ Sybase ➤ SQL Anywhere 9 ➤ Adaptive Server Anywhere ➤ ODBC Administrator.

The ODBC Data Source Administrator dialog appears.

2. Click Add.

The Create New Data Source wizard appears.

3. From the list of drivers, choose Adaptive Server Anywhere 9.0, and click Finish.

The ODBC Configuration for Adaptive Server Anywhere dialog appears.

lata <u>s</u> ource nam	e:	
escription:	E	
solation level:		
Microsoft app	lications (Keys in SQL	Statistics)
Delp <u>h</u> i applica	ations	
Supp <u>r</u> ess feto	h warnings	
- Provent drive	r not canable errors	
Tieveur duve	r not capable entris	
Delay AutoCo	immit until statement c	lose
Delay AutoCo Describe Curso	nmit until statement c r Behavior	lose
T Delay A <u>u</u> toCo Describe Curso C <u>N</u> ever	mmit until statement o r Behavior If required	lose C Al <u>w</u> ays
Delay A <u>u</u> toCo Describe Curso C <u>N</u> ever ranslator: KNo	mmit until statement o r Behavior I translator>	lose O Always
Delay A <u>u</u> toCo Describe Curso <u>N</u> ever ranslator: KNo	mmit until statement o r Behavior I frequired Translator>	lose C Al <u>w</u> ays S <u>e</u> lect Translator
Delay A <u>u</u> toCo Describe Curso <u>N</u> ever ranslator: < <u>No</u>	immit until statement o r Behavior If required Translator>	lose C Al <u>w</u> ays Select Translator Test Connection

Most of the fields in this window are optional. Click the question mark at the top right of the window and click a dialog field to find more information about that field.

For more information about the fields in the dialog, see "Configuring ODBC data sources using the ODBC Administrator" on page 56.

4. When you have specified the parameters you need, click OK to close the window and create the data source.

To edit a data source, find and select one in the ODBC Administrator main window and click Configure.

Creating an ODBC data
source from the
command lineYou can create User Data Sources using the *dbdsn* utility. You cannot create
File Data Sources or System Data Sources. File and System Data Sources
are limited to Windows operating systems only, and you can use the ODBC
Administrator to create them.

To create an ODBC data source (Command line)

- 1. Open a command prompt.
- 2. Type a *dbdsn* command, specifying the connection parameters you wish to use.

For example, the following command creates a data source for the Adaptive Server Anywhere sample database. The command must be typed on one line:

```
dbdsn -w "My DSN"
"uid=DBA;pwd=SQL;dbf=c:\Program Files\Sybase\SQL Anywhere 9\
asademo.db"
```

For more information on the *dbdsn* utility, see "The Data Source utility" on page 472.

Configuring ODBC data sources using the ODBC Administrator

This section describes the meaning of each of the options on the ODBC configuration dialog.

ODBC tab

This tab provides fields related to ODBC data sources and appears only when you are creating or modifying ODBC data sources.

Data source name The Data Source Name is used to identify the ODBC data source. You can use any descriptive name for the data source (spaces are allowed) but it is recommended that you keep the name short, as you may need to type it in connection strings.

For more information, see "DataSourceName connection parameter [DSN]" on page 186.

Description You may type an optional longer description of the Data Source to help you or end users to identify this data source from among their list of available data sources. This field is optional.

Isolation level Type the desired isolation level for this data source:

- **0** Dirty reads, non-repeatable reads and phantom rows may occur. This is the default isolation level.
- 1 Non-repeatable rows and phantom rows may occur. Dirty reads are prevented.
- 2 Phantom rows may occur. Dirty reads and non-repeatable rows are prevented.
• 3 Dirty reads, non-repeatable reads and phantom rows are prevented.

For more information, see "Choosing isolation levels" [ASA SQL User's Guide, page 112].

Microsoft applications (Keys in SQLStatistics) Check this box if you want foreign keys to be returned by the **SQLStatistics** function. The ODBC specification states that SQLStatistics should not return primary and foreign keys. However, some Microsoft applications (such as Visual Basic and Access) assume that primary and foreign keys are returned by **SQLStatistics**.

Delphi applications Check this box to improve performance for Borland Delphi applications. When this option is checked, one bookmark value is assigned to each row, instead of the two that are otherwise assigned (one for fetching forwards and a different one for fetching backwards).

Delphi cannot handle multiple bookmark values for a row. If the option is unchecked, scrollable cursor performance can suffer since scrolling must always take place from the beginning of the cursor to the row requested in order to get the correct bookmark value.

Suppress fetch warnings Check this box to suppress warning messages that are returned from the database server on a fetch.

Versions 8.0 and later of the database server return a wider range of fetch warnings than earlier versions of the software. For applications that are deployed with an earlier version of the software, you can select this option to ensure that fetch warnings are handled properly.

Prevent driver not capable errors The Adaptive Server Anywhere ODBC driver returns a Driver not capable error code because it does not support qualifiers. Some ODBC applications do not handle this error properly. Check this box to disable this error code, allowing such applications to work.

Delay AutoCommit until statement close Check this box if you want the Adaptive Server Anywhere ODBC driver to delay the commit operation until a statement has been closed.

Describe Cursor Behavior Choose how often you wish a cursor to be redescribed when a procedure is executed or resumed. The default setting is If Required.

- Never Select this option if you know that your cursors do not have to be redescribed because redescribing cursors is expensive and can decrease performance.
- If required When you select this option, the ODBC driver determines

whether a cursor must be redescribed. The presence of a RESULT clause in your procedure prevents ODBC applications from redescribing the result set after a cursor is opened.

♦ Always The cursor is redescribed each time it is opened. If you use Transact-SQL procedures or procedures that return multiple result sets, you must redescribe the cursor each time it is opened.

Translator The Translator converts characters between ANSI and OEM code pages. Most databases do not require a translator because the ODBC Driver Manager automatically performs a conversion between the client's character set and the database's character set. If your database uses an ANSI code page (the default) do not select a translator.

• Select Translator Opens the Select Translator dialog where you can select the ODBC translator that you want to use from a list of installed translators. Use this option sparingly, since the ODBC driver and database server do automatic character set translation.

Test connection Tests whether the information provided results in a proper connection. In order for the test to work, a user ID and password must have been specified.

Login tab

This tab provides fields related to user authentication and logins.

Use integrated login Connects using an integrated login. The user ID and password do not need to be specified: instead, your operating system user id and password are supplied to the Adaptive Server Anywhere integrated login mechanism.

To use this type of login users must have been granted integrated login permission. The database being connected to must also be set up to accept integrated logins. Only users with DBA access may administer integrated login permissions.

For more information, see "Using integrated logins" on page 85.

User ID Provides a place for you to type the user ID for the connection.

For more information, see "Userid connection parameter [UID]" on page 199.

Password Provides a place for you to type the password for the connection.

For more information, see "Password connection parameter [PWD]" on

page 196.

Encrypt password Check this box if you want the password to be stored in encrypted form in the profile.

For more information, see "EncryptedPassword connection parameter [ENP]" on page 188.

Database tab

This tab provides fields for identifying the database server and database to which you are connecting.

Server name Provides a place for you to type the name of the Adaptive Server Anywhere personal or network server to which you want to connect.

For more information, see "EngineName connection parameter [ENG]" on page 187.

Start line Type the name of the server executable that should be started. Only provide a **StartLine (START)** connection parameter if the database server you are connecting to is not currently running. For example:

C:\Program Files\Sybase\ SQL Anywhere 9\win32\dbeng9.exe -c 8m

For more information, see "StartLine connection parameter [START]" on page 199.

Database name Provides a place for you to type the name of the Adaptive Server Anywhere database that you wish to connect to.

For more information, see "DatabaseName connection parameter [DBN]" on page 184.

Database file Provides a place for you to type the full path and name of the Adaptive Server Anywhere database file on the server PC. You may also click Browse to locate the file. For example:

C:\Program Files\Sybase\SQL Anywhere 9\asademo.db

For more information, see "DatabaseFile connection parameter [DBF]" on page 183.

Encryption key If your database is strongly encrypted, you must supply the correct encryption key to access any part of the database.

Start database automatically Causes the database to start automatically (if it is not already running) when you start a new session. This option is enabled when you supply a database file or start line.

Stop database after last disconnect Causes the automatic shutdown of

the server after the last user has disconnected.

For more information, see "AutoStop connection parameter [ASTOP]" on page 177.

Network tab

This tab provides fields for controlling client/server communications over a network.

Select the network protocols and options These checkboxes specify what protocol or protocols the ODBC DSN uses to access a network database server. In the adjacent boxes, you may enter communication parameters that establish and tune connections from your client application to a database.

TCP/IP If you want to use ECC_TLS (Certicom) or RSA_TLS encryption for network packets, you must select the TCP/IP protocol to access the network database server.

For example, you could enter the following communication parameters for a TCP/IP connection **HOST=my_server;PORT=4563**.

- SPX You can select the SPX protocol to connect to databases on Novell NetWare networks. NetWare also supports the TCP/IP protocol.
- Named pipes The Named Pipes protocol is used for client/server communication on the same machine. If you want to run under a certified security environment, you can use the Named Pipes protocol. It is only provided on Windows NT/2000/XP.
- Shared memory The shared memory protocol is used for communication between a client and server running under the same operating system on the same machine.

For more information, see "CommLinks connection parameter [LINKS]" on page 179, and "Network communications parameters" on page 201.

Liveness timeout A liveness packet is sent across a client/server to confirm that a connection is intact. If the client runs for the liveness timeout period without detecting a liveness packet, the communication will be severed. This parameter works only with the network server and TCP/IP and SPX communications protocols. The default is 120 seconds.

For more information, see "LivenessTimeout connection parameter [LTO]" on page 194.

Idle timeout Set the amount of client idle time before the connection is terminated. If a client runs for the idle timeout period without submitting a request, the connection is severed.

The default client idle time is 240 minutes.

For more information, see "Idle connection parameter [IDLE]" on page 192.

Buffer size Sets the maximum size of communication packets, in bytes. The default buffer size is 1460.

For more information, see "CommBufferSize connection parameter [CBSIZE]" on page 178.

Compress network packets Sets compression for the connection. Using compression for a connection can significantly improve Adaptive Server Anywhere performance under some circumstances.

For more information, see "Compress connection parameter [COMP]" on page 181.

Select the method for encryption of network packets Allows the encryption of packets transmitted from the client machine over the network.

- **None** Communication packets transmitted from the client are not encrypted. This is the default setting.
- Simple Communication packets transmitted from the client are encrypted with simple encryption. Simple encryption is supported on all platforms, as well as on previous versions of Adaptive Server Anywhere.
- ECC TLS Communication packets transmitted from the client are encrypted with ECC_TLS (formerly Certicom) encryption. This is a type of strong encryption. It is only available over the TCP/IP protocol.

In the adjacent field you must supply a trusted certificate value.

♦ RSA TLS Communication packets transmitted from the client are encrypted with RSA_TLS encryption. This is a type of strong encryption. It is only available over the TCP/IP protocol.

In the adjacent field you must supply a trusted certificate value.

For more information, see "Encryption connection parameter [ENC]" on page 188.

Advanced tab

This tab provides advanced fields that are for special purposes and are not

required by the majority of users.

Connection name The name of the connection that is being created.

Character set Lets you specify a character set (a set of 256 letters, numbers, and symbols specific to a country or language). The ANSI character set is used by Microsoft Windows. An OEM character set is any character set except the ANSI character set.

Allow multiple record fetching Enables multiple records to be retrieved at one time instead of individually. By default, multiple record fetching is allowed.

Display debugging information in a log file Select this option to specify the name of the file in which the debugging information is to be saved.

Additional connection parameters Type any additional options here. Parameters set throughout the remainder of this dialog take precedence over parameters typed here.

For more information, see "Connection parameters" on page 174.

Using file data sources on Windows

	On Windows operating systems, ODBC data sources are typically stored in the system registry. File data sources are an alternative, which are stored as files. In Windows, file data sources typically have the extension <i>.dsn.</i> They consist of sections, each section starting with a name enclosed in square brackets. DSN files are very similar in layout to initialization files.
	To connect using a File Data Source, use the FileDataSourceName (FILEDSN) connection parameter. You cannot use both DataSourceName (DSN) and FileDataSourceName (FILEDSN) in the same connection.
File data sources can be distributed	One benefit of file data sources is that you can distribute the file to users. If the file is placed in the default location for file data sources, it is picked up automatically by ODBC. In this way, managing connections for many users can be made simpler.
	Embedded SQL applications can also use ODBC file data sources.
*	To create an ODBC file data source (ODBC Administrator)
	1. Start the ODBC Administrator, click the File DSN tab and click Add.
	2. Select Adaptive Server Anywhere 9.0 from the list of drivers, and click Next.
	3. Follow the instructions to create the data source.

Using ODBC data sources on UNIX

On UNIX operating systems, ODBC data sources are held in a file named *.odbc.ini*. A sample file looks like this:

```
[My Data Source]
ENG=myserver
CommLinks=tcpip(Host=hostname)
uid=DBA
pwd=SQL
```

You can type any connection parameter in the .odbc.ini file.

For a complete list or connection parameters, see "Connection parameters" on page 174.

Network communications parameters are added as part of the **CommLinks** (LINKS) parameter.

For a complete list of network communications parameters, see "Network communications parameters" on page 201.

You can create and manage ODBC data sources on UNIX using the *dbdsn* utility.

For more information, see "Creating an ODBC data source" on page 53, and "The Data Source utility" on page 472.

File location The database server looks for the *.odbc.ini* file in the following locations:

- 1. ODBCINI environment variable
- 2. ODBCHOME and HOME environment variables
- 3. The user's home directory
- 4. The path

Using ODBC data sources on Windows CE

Windows CE does not provide an ODBC driver manager or an ODBC Administrator. On this platform, Adaptive Server Anywhere uses ODBC data sources stored in files. You can specify either the **DSN** or the **FILEDSN** keyword to use these data source definitions—on Windows CE (only), **DSN** and **FILEDSN** are synonyms.

Data source location Windows CE searches for the data source files in the following locations:

1. The directory from which the ODBC driver (*dbodbc9.dll*) was loaded. This is usually the Windows directory. 2. The directory specified in Location key of the Adaptive Server Anywhere section of the registry. This is usually the same as the Adaptive Server Anywhere installation directory. The default installation directory is:

```
\Program Files\Sybase\ASA
```

Each data source itself is held in a file. The file has the same name as the data source, with an extension of *.dsn*.

For more information about file data sources, see "Using file data sources on Windows" on page 62.

Connecting from desktop applications to a Windows CE database

You can connect from applications running on a desktop PC, such as Sybase Central or Interactive SQL, to a database server running on a Windows CE device. The connection uses TCP/IP over the ActiveSync link between the desktop machine and the Windows CE device.

If you are using an Ethernet connection between your desktop machine and the Windows CE device, or if you are using Windows CE Services 2.2, the following procedure works. If you are using a serial cable connection and ActiveSync 3.0, see the following section.

To connect from a desktop application to a Windows CE database server

 Determine the IP address of the server. Start the server on the Windows CE device with the -z option (output extra debug information).

For example:

dbsrv9 -z -x tcpip -n TestServer asademo.db

With the -z option, the server writes out its IP address during startup. The address may change if you disconnect your HPC from the network and then re-connect it.

To change between static and dynamic IP assignment for the HPC, configure the settings in the Windows Control Panel. Open Network, and choose the Services tab. Select Remote Access Service and click Properties ➤ Network ➤ TCP/IP Configuration.

2. Create an ODBC profile on your desktop machine.

Open the ODBC Administrator, and click Add. Choose Adaptive Server Anywhere 9.0 from the list of drivers and click Finish. The ODBC Configuration for Adaptive Server Anywhere dialog appears.

- On the Login tab, type a user ID and password.
- On the Database tab, type the server name.
- On the Network tab, check TCP/IP, and type the following in the adjacent field:

dobroadcast=DIRECT;host=XXX.XXX.XXX.XXX

where XXX.XXX.XXX.XXX is the server IP address.

• On the ODBC tab, click Test Connection to confirm that your ODBC data source is properly configured.

5. Exit the ODDC Auffillistrator.	3.	Exit the	ODBC	Administrator.
-----------------------------------	----	----------	------	----------------

- 4. Ensure the database server is running on your Windows CE machine.
- 5. On your desktop machine, start an application such as Interactive SQL and select the ODBC data source you have created. The application connects to the Windows CE database.

Using ActiveSync 3.0 and a serial cable

To connect to a Windows CE device from your desktop over a serial cable, Adaptive Server Anywhere uses the TCP/IP protocol. For TCP/IP to work in this context, your ActiveSync installation must be set up to provide a Remote Access Service (RAS) link between desktop machine and Windows CE device.

ActiveSync 2.2 automatically installs and configures RAS, and you can connect in a straightforward manner.

For information, see "Connecting from desktop applications to a Windows CE database" on page 65.

ActiveSync 3.0 does not install and configure RAS. You must install RAS yourself to obtain TCP/IP connectivity to your device over a serial connection.

Instructions for installing RAS are provided by Microsoft. They are available at *http://support.microsoft.com/support/kb/articles/Q241/2/16.ASP* (Microsoft Knowledge Base article Q241216). You must follow these instructions *exactly*, including re-installing any Windows NT service packs, and granting your user account dial-in access using User Manager.

As you follow the instructions, where it says to install your modem, choose Dial-up Networking Serial Cable between 2 PCs instead.

Using RAS with ActiveSync

The following list includes suggestions for enabling ActiveSync 3.0 connections over a serial connection using RAS:

- 1. In the ActiveSync **Connection Settings**, select the checkbox Allow network (Ethernet) and Remote Access Service (RAS) server connection with desktop computer. You may need to turn off the checkbox Allow serial cable or infrared connection to this COM port.
- On the desktop, using Remote Access Administrator (under Administrative Tools on Windows NT), start RAS on COM1.
- 3. On the Windows CE device, run the ActiveSync client (*repllog.exe* on a Windows CE PC). Choose serial connection.

- 4. Wait for up to one minute for a connection to be established.
- 5. As a test, run the *ipconfig* utility on Windows NT, and see the 192.168.55.100 static IP of the device. This is the IP you would use when connecting to an Adaptive Server Anywhere database server (for example) running on the CE device.
- 6. If you switch devices, Stop and Restart the RAS service (or reboot).
- 7. If everything is set up as above, but you still fail to get a connection from the device to the desktop, you should make sure your Port settings match the baud rates in the Modems Control Panel applet.

Connecting to a database using OLE DB

OLE DB uses the Component Object Model (COM) to make data from a variety of sources available to applications. Relational databases are among the classes of data sources that you can access through OLE DB.

This section describes how to connect to an Adaptive Server Anywhere database using OLE DB from the following environments:

- Sybase PowerBuilder can access OLE DB data sources, and you can use Adaptive Server Anywhere as a PowerBuilder OLE DB database profile.
- Microsoft ActiveX Data Objects (ADO) provides a programming interface for OLE DB data sources. You can access Adaptive Server Anywhere from programming tools such as Microsoft Visual Basic.

This section is an introduction to how to use OLE DB from Sybase PowerBuilder and Microsoft ADO environments such as Visual Basic. It is not complete documentation on how to program using ADO or OLE DB. The primary source of information on development topics is your development tool documentation.

For more information about OLE DB, see "Introduction to OLE DB" [*ASA Programming Guide*, page 314].

OLE DB providers

You need an OLE DB provider for each type of data source you wish to access. Each **OLE DB provider** is a dynamic-link library. There are two OLE DB providers you can use to access Adaptive Server Anywhere:

 Sybase ASA OLE DB provider The Adaptive Server Anywhere OLE DB provider provides access to Adaptive Server Anywhere as an OLE DB data source without the need for ODBC components. The short name for this provider is ASAProv.

When the **ASAProv** provider is installed, it registers itself. This registration process includes making registry entries in the COM section of the registry so that ADO can locate the DLL when the **ASAProv** provider is called. If you change the location of your DLL, you must reregister it.

For more information about OLE DB providers, see "Introduction to OLE DB" [ASA Programming Guide, page 314].

• Microsoft OLE DB provider for ODBC Microsoft provides an OLE DB provider with a short name of MSDASQL.

The **MSDASQL** provider makes ODBC data sources appear as OLE DB data sources. It requires the Adaptive Server Anywhere ODBC driver.

Connecting from ADO

Example

ADO is an object-oriented programming interface. In ADO, the **Connection** object represents a unique session with a data source.

You can use the following **Connection** object features to initiate a connection:

- The Provider property that holds the name of the provider. If you do not supply a Provider name, ADO uses the MSDASQL provider.
- The ConnectionString property that holds a connection string. This property holds an Adaptive Server Anywhere connection string, which is used in just the same way as the ODBC driver. You can supply ODBC data source names, or explicit UserID, Password, DatabaseName, and other parameters, just as in other connection strings.

For more information about connection parameters, see "Connection parameters" on page 70.

• The **Open** method initiates a connection.

For more information about ADO, see "ADO programming with Adaptive Server Anywhere" [*ASA Programming Guide*, page 315].

The following Visual Basic code initiates an OLE DB connection to Adaptive Server Anywhere:

```
' Declare the connection object
Dim myConn as New ADODB.Connection
myConn.Provider = "ASAProv"
myConn.ConnectionString = "Data Source=ASA 9.0 Sample"
myConn.Open
```

Connection parameters

The following table lists the Adaptive Server Anywhere connection parameters.

For more information about each of these connection parameters, see "Connection and Communication Parameters" on page 173. For character set issues in connection strings, see "Connection strings and character sets" on page 318.

Parameter	Short form	Argument
"AppInfo connection pa- rameter [APP]" on page 175	APP	String
"AutoStart connection pa- rameter [ASTART]" on page 176	ASTART	Boolean
"AutoStop connection pa- rameter [ASTOP]" on page 177	ASTOP	Boolean
"CharSet connection param- eter [CS]" on page 177	CS	String
"CommBufferSize connec- tion parameter [CBSIZE]" on page 178	CBSIZE	Integer
"CommLinks connection parameter [LINKS]" on page 179	LINKS	String
"Compress connection parameter [COMP]" on page 181	СОМР	String
"CompressionThreshold connection parameter [COMPTH]" on page 182	СОМРТН	Integer
"ConnectionName connec- tion parameter [CON]" on page 183	CON	String
"DatabaseFile connec- tion parameter [DBF]" on page 183	DBF	String

Parameter	Short form	Argument
"DatabaseName connec- tion parameter [DBN]" on page 184	DBN	String
"DatabaseSwitches connec- tion parameter [DBS]" on page 185	DBS	String
"DataSourceName connec- tion parameter [DSN]" on page 186	DSN	String
"DisableMultiRowFetch connection parameter [DMRF]" on page 187	DMRF	Boolean
"EncryptedPassword con- nection parameter [ENP]" on page 188	ENP	Encrypted string
"Encryption connection pa- rameter [ENC]" on page 188	ENC	String
"EngineName connec- tion parameter [ENG]" on page 187 / "Server- Name connection parameter [ENG]" on page 198	ENG	String
"FileDataSourceName connection parameter [FILEDSN]" on page 191	FILEDSN	String
"ForceStart connection pa- rameter [FORCESTART]" on page 191	FORCE	Boolean
"Idle connection parameter [IDLE]" on page 192	IDLE	Integer
"Integrated connection pa- rameter [INT]" on page 192	INT	Boolean
"LazyClose connection pa- rameter [LCLOSE]" on page 193	LCLOSE	Boolean

Parameter	Short form	Argument
"LivenessTimeout connec- tion parameter [LTO]" on page 194	LTO	Integer
"Logfile connection param- eter [LOG]" on page 194	LOG	String
"Password connection parameter [PWD]" on page 196 *	PWD	String
"PrefetchBuffer connec- tion parameter [PBUF]" on page 197	PBUF	Integer
"PrefetchRows connection parameter [PROWS]" on page 198	PROWS	Integer
"StartLine connection pa- rameter [START]" on page 199	START	String
"Unconditional connec- tion parameter [UNC]" on page 199	UNC	Boolean
"Userid connection param- eter [UID]" on page 199 *	UID	String

* Verbose form of keyword not used by ODBC connection parameters

- ♦ Boolean values Boolean (true or false) arguments are either YES, ON, 1, or TRUE if true, or NO, OFF, 0, or FALSE if false.
- **Case sensitivity** Connection parameters are case insensitive.
- The connection parameters used by the interface library can be obtained from the following places (in order of precedence):
 - **Connection string** You can pass parameters explicitly in the connection string.
 - **SQLCONNECT environment variable** The SQLCONNECT environment variable can store connection parameters.
 - Data sources ODBC data sources can store parameters.

Notes

• Character set restrictions The server name must be composed of the ASCII character set in the range 1 to 127. There is no such limitation on other parameters.

For more information on the character set issues, see "Connection strings and character sets" on page 318.

- **Priority** The following rules govern the priority of parameters:
 - The entries in a connect string are read left to right. If the same parameter is specified more than once, the last one in the string applies.
 - If a string contains a data source or file data source entry, the profile is read from the configuration file, and the entries from the file are used if they are not already set. For example, if a connection string contains a data source name and sets some of the parameters contained in the data source explicitly, then in case of conflict the explicit parameters are used.

Connection parameter tips

Connection parameters often provide more than one way of accomplishing a given task. This is particularly the case with embedded databases, where the connection string starts a database server. For example, if your connection starts a database, you can specify the database name using the **DatabaseName (DBN)** connection parameter or using the **(DatabaseSwitch DBS)** parameter.

Here are some recommendations and notes for situations where connection parameters conflict:

- Specify database files using DBF You can specify a database file on the StartLine (START) parameter or using the DatabaseFile (DBF) connection parameter (recommended).
- Specify database names using DBN You can specify a database name on the StartLine (START) parameter, the DatabaseSwitch (DBS) connection parameter, or using the DatabaseName (DBN) connection parameter (recommended).
- Use the Start parameter to specify cache size Even though you use the DatabaseFile (DBF) connection parameter to specify a database file, you may still want to tune the way in which it starts. You can use the StartLine (START) connection parameter to do this.

For example, if you are using the Java features of Adaptive Server Anywhere, you should provide additional cache memory on the **StartLine (START)** connection parameter. The following sample set of embedded database connection parameters describes a connection that may use Java features:

```
DBF=path\asademo.db
dbn=Sample
ENG=Sample Server
uid=DBA
pwd=SQL
Start=dbeng9 -c 8M
```

Troubleshooting connections

Who needs to read this section?

In many cases, establishing a connection to a database is straightforward using the information presented in the first part of this chapter.

However, if you are having problems establishing connections to a server, you may need to understand the process by which Adaptive Server Anywhere establishes connections in order to resolve your problems. This section describes how Adaptive Server Anywhere connections work.

For more information about network-specific issues, including connections across firewalls, see "Client/Server Communications" on page 93.

The software follows exactly the same procedure for each of the following types of client application:

- ODBC Any ODBC application using the SQLDriverConnect function, which is the common method of connection for ODBC applications. Many application development systems, such as Sybase PowerBuilder and Power++, belong to this class of application.
- Embedded SQL Any client application using Embedded SQL and using the recommended function for connecting to a database (db_string_connect).

The SQL CONNECT statement is available for Embedded SQL applications and in Interactive SQL. It has two forms: CONNECT AS... and CONNECT USING. All the database administration tools, including Interactive SQL, use db_string_connect.

For more troubleshooting tips, see "Troubleshooting server startup" on page 34, and "Troubleshooting network communications" on page 105.

The steps in establishing a connection

To establish a connection, Adaptive Server Anywhere carries out the following steps in order:

- 1. Locate the interface library The client application must locate the ODBC driver or Embedded SQL interface library.
- 2. Assemble a list of connection parameters Since connection parameters may appear in several places (such as data sources, a connection string assembled by the application, or an environment variable) Adaptive Server Anywhere assembles the parameters into a single list.
- 3. Locate a server Using the connection parameters, Adaptive Server Anywhere locates a database server on your machine or over a network.

- 4. Locate the database Adaptive Server Anywhere locates the database you want to connect to.
- 5. **Start a personal server** If Adaptive Server Anywhere fails to locate a server, it attempts to start a personal database server and load the database.

The following sections describe each of these steps in detail.

Locating the interface library

	The client application makes a call to one of the Adaptive Server Anywhere interface libraries. In general, the location of this DLL or shared library is transparent to the user. Here we describe how to locate the library in case of problems.
ODBC driver location	For ODBC, the interface library is also called an ODBC driver. An ODBC client application calls the ODBC driver manager, and the driver manager locates the Adaptive Server Anywhere driver.
	The ODBC driver manager looks in the supplied data source in the <i>odbc.ini</i> file or registry to locate the driver. When you create a data source using the ODBC Administrator, Adaptive Server Anywhere fills in the current location for your ODBC driver.
Embedded SQL interface library location	Embedded SQL applications call the interface library by name. The name of the Adaptive Server Anywhere Embedded SQL interface library is as follows:
	Windows NT/2000/XP and Windows 95/98/Me dblib9.dll
	• UNIX <i>dblib9</i> with an operating-system-specific extension
	◆ NetWare dblib9.nlm
	The locations that are searched depend on the operating system:
	• PC operating systems On PC operating systems such as Windows, files are looked for in the current directory, in the system path, and in the <i>Windows</i> and <i>Windows</i> \system directories.
	• UNIX operating systems On UNIX, files are looked for in the system path and the user library path.
	• NetWare On NetWare, files are looked for in the search path, and in the <i>sys:system</i> directory.
When the library is located	Once the client application locates the interface library, it passes a connection string to it. The interface library uses the connection string to

assemble a list of connection parameters, which it uses to establish a connection to a server.

Assembling a list of connection parameters

The following figure illustrates how the interface libraries assemble the list of connection parameters they use to establish a connection.



Notes

Key points from the figure include:

- **Precedence** Parameters held in more than one place are subject to the following order of precedence:
 - 1. Connection string
 - 2. SQLCONNECT
 - 3. Data source

That is, if a parameter is supplied both in a data source and in a connection string, the connection string value overrides the data source value.

- Failure Failure at this stage occurs only if you specify in the connection string or in SQLCONNECT a data source that does not exist in the client connection file.
- Common parameters Depending on other connections already in use, some connection parameters may be ignored, including:

- AutoStop Ignored if the database is already loaded.
- **CommLinks** The specifications for a network protocol are ignored if another connection has already set parameters for that protocol.
- **CommBufferSize** If CommBufferSize is specified on the client, the connection uses the minimum value of either the engine's buffer size or the CommBufferSize value. If it is not specified on the client, the connection uses the engine's buffer size.
- **Unconditional** Ignored if the database is already loaded or if the server is already running.

The interface library uses the completed list of connection parameters to attempt to connect.

Locating a server

In the next step towards establishing a connection, Adaptive Server Anywhere attempts to locate a server. If the connection parameter list includes a server name (**EngineName** (**ENG**) connection parameter), it carries out a search first for a local server (personal server or network server running on the same machine) of that name, followed by a search over a network. If no **EngineName** (**ENG**) connection parameter is supplied, Adaptive Server Anywhere looks for a default server.



Final Adaptive Server Anywhere locates a server, it tries to locate or load the required database on that server. For information, see "Locating the database" on page 80.

Figure 3 If Adaptive Server Anywhere cannot locate a server, it attempts to start a personal server. For information, see "Starting a personal server" on page 81.

- For local connections, locating a server is simple. For connections over a network, you can use the CommLinks (LINKS) connection parameter to tune the search in many ways by supplying network communication parameters.
- The network search involves a search over one or more of the protocols supported by Adaptive Server Anywhere. For each protocol, the network

Notes

library starts a single port. All connections over that protocol at any one time use a single port.

- You can specify a set of network communication parameters for each network port in the argument to the **CommLinks** (**LINKS**) connection parameter. Since these parameters are necessary only when the port first starts, the interface library ignores any connection parameters specified in the **CommLinks** (**LINKS**) connection parameter for a port already started.
- Each attempt to locate a server (the local attempt and the attempt for each network port) involves two steps. First, Adaptive Server Anywhere looks in the server name cache to see if a server of that name is available. Second, it uses the available connection parameters to attempt a connection.

Locating the database

If Adaptive Server Anywhere successfully locates a server, it then tries to locate the database. For example:



Starting a personal server

If no server can be located, the interface libraries attempt to start a personal server using other parameters. The START and DBF parameters can be used to start a personal server.



The **StartLine** (**START**) connection parameter takes a personal database server command line. If a Start parameter is unavailable, Adaptive Server Anywhere attempts to start a personal server on the file indicated by the **DatabaseFile** (**DBF**). If both an **EngineName** (**ENG**) parameter and a **DatabaseFile** (**DBF**) parameter appear, the **EngineName** (**ENG**) parameter becomes the name of the server.

Server name caching for faster connections

	When the DoBroadcast (DOBROAD) communication parameter is set to DIRECT or ALL , the network library looks for a database server on a network by broadcasting over the network using the CommLinks (LINKS) connection parameter.
Tuning the broadcast	The CommLinks (LINKS) parameter takes as argument a string listing the protocols to use and, optionally for each protocol, a variety of network communication parameters that tune the broadcast.
	For more information about network communications parameters, see "Network communications parameters" on page 201.
Caching server information	Broadcasting over large networks searching for a server of a specific name can be time-consuming. Leaving the DoBroadcast (DOBROAD) communication parameter at the default value (ALL) speeds up network

connections by saving the protocol the first connection to a server was found on, and its address, to a file and using that information for subsequent connections.

The server information is saved in a cached file named *asasrv.ini*. The file contains a set of sections, each of the following form:

```
[Server name]
Link=protocol_name
Address=address_string
```

The *asasrv.ini* file is located in your Adaptive Server Anywhere executable directory (the same directory as the ODBC/ DBLib DLL).

```
Note:
```

It is very important that each server has a unique name. Giving different servers the same name can lead to identification problems.

How the cache is used If the server name and protocol in the cache match the connection string, Adaptive Server Anywhere tries to connect using the cached address first. If that fails, or if the server name and protocol in the cache do not match the connection string, the connection string information is used to search for the server using a broadcast. If the broadcast is successful, the server name entry in the cache is overwritten. If no server is found, the server name entry in the cache is removed.

Interactive SQL connections

The Interactive SQL utility has a different behavior from the default Embedded SQL behavior when a CONNECT statement is issued while already connected to a database. If no database or server is specified in the CONNECT statement, Interactive SQL connects to the current database, rather than to the default database. This behavior is required for database reloading operations.

For more information, see "CONNECT statement [ESQL] [Interactive SQL]" [*ASA SQL Reference*, page 287].

Testing that a server can be found

The *dbping* utility is provided to help in troubleshooting connections. In particular, you can use it to test if a server with a particular name is available on your network.

The *dbping* utility takes a connection string as an option, but by default only those pieces required to locate a server are used. It does not attempt to start a server.

Examples The following command line tests to see if a server named Waterloo is available over a TCP/IP connection:

```
dbping -c "eng=Waterloo;CommLinks=tcpip"
```

The following command tests to see if a default server is available on the current machine.

dbping

For more information on *dbping* options, see "The Ping utility" on page 514.

Using integrated logins

	The integrated login feature allows you to maintain a single user ID and password for both database connections and operating system and/or network logins. This section describes the integrated login feature.
Operating systems supported	Integrated login capabilities are available for the Windows NT server only. It is possible for Windows 95/98/Me clients, as well as Windows NT/2000/XP clients, to use integrated logins to connect to a network server running on Windows NT/2000/XP.
Benefits of an integrated login	An integrated login is a mapping from one or more Windows NT/2000/XP user profiles to an existing user in a database. A user who has successfully navigated the security for that user profile and logged in to their machine can connect to a database without providing an additional user ID or password.
	To accomplish this, the database must be enabled to use integrated logins and a mapping must have been granted between the user profile used to log in to the machine and/or network, and a database user.
	Using an integrated login is more convenient for the user and permits a single security system for database and network security. Its advantages include:
	• When connecting to a database using an integrated login, the user does not need to type a user ID or password.
	• If you use an integrated login, the user authentication is done by the operating system, not the database: a single system is used for database security and machine or network security.
	• Multiple user profiles can be mapped to a single database user ID.
	• The name and password used to login to the Windows NT/2000/XP machine do not have to match the database user ID and password.
	Caution Integrated logins offer the convenience of a single security system but there are important security implications which database administrators should be familiar with.
	For more information about security and integrated logins, see "Security concerns: unrestricted database access" on page 89.

Using integrated logins

Several steps must be implemented in order to connect successfully via an integrated login.

To use an integrated login

- 1. Enable the integrated login feature in a database by setting the value of the LOGIN_MODE database option to either Mixed or Integrated (the option is case insensitive), in place of the default value of Standard. This step requires DBA authority).
- 2. Create an integrated login mapping between a user profile and an existing database user. This can be done using a SQL statement or a wizard in Sybase Central. In Sybase Central, all users with integrated login permission appear in the Integrated Logins folder.
- 3. Connect from a client application in such a way that the integrated login facility is triggered.

Each of these steps is described in the sections below.

Enabling the integrated login feature

The LOGIN_MODE database option determines whether the integrated login feature is enabled. As database options apply only to the database in which they are found, different databases can have a different integrated login setting even if they are loaded and running within the same server.

The LOGIN_MODE database option accepts one of following three values (which are case insensitive).

- **Standard** With this setting, integrated logins are not permitted. This is the default setting. An error occurs if an integrated login connection is attempted.
- **Mixed** With this setting, both integrated logins and standard logins are allowed.
- **Integrated** With this setting, all logins to the database must be made using integrated logins.

Caution

Setting the LOGIN_MODE database option to Integrated restricts connections to only those users who have been granted an integrated login mapping. Attempting to connect using a user ID and password generates an error. The only exception to this is users with DBA authority (full administrative rights).

Example

The following SQL statement sets the value of the LOGIN_MODE database option to Mixed, allowing both standard and integrated login connections:

Creating an integrated login

User profiles can only be mapped to an existing database user ID. When that database user ID is removed from the database, all integrated login mappings based on that database user ID are automatically removed.

A user profile does not have to exist for it to be mapped to a database user ID. More than one user profile can be mapped to the same user ID.

Only users with DBA authority are able to create or remove an integrated login mapping.

An integrated login mapping is made either using a wizard in Sybase Central or a SQL statement.

To map an integrated login (Sybase Central)

- 1. Connect to a database as a user with DBA authority.
- 2. Click the New Integrated Login button on the toolbar.
- 3. On the first page of the wizard, specify the name of the system (computer) user for whom the integrated login is to be created.

Also, select the database user ID this user maps to. The wizard displays the available database users. You must select one of these. You cannot add a new database user ID.

4. Follow the remaining instructions in the wizard.

To map an integrated login (SQL)

- 1. Connect to a database with DBA authority.
- 2. Execute a GRANT INTEGRATED LOGIN TO statement.

The following SQL statement allows Windows NT users **fran_whitney** and **matthew_cobb** to log in to the database as the user **DBA**, without having to know or provide the DBA user ID or password.

```
GRANT INTEGRATED LOGIN
TO fran_whitney, matthew_cobb
AS USER DBA
```

For more information, see "GRANT statement" [ASA SQL Reference, page 456].

Revoking integrated login permission

You can remove an integrated login mapping using either Sybase Central or Interactive SQL.

Example

	*	To revoke an integrated login permission (Sybase Central)
		1. Connect to a database with DBA authority.
		2. Open the Integrated Logins folder.
		3. In the right pane, right-click the user/group you would like to remove the integrated login permission from and choose Delete from the popup menu.
	*	To revoke an integrated login permission (SQL)
		1. Connect to a database with DBA authority.
		2. Execute a REVOKE INTEGRATED LOGIN FROM statement.
Example		The following SQL statement removes integrated login permission from the Windows NT user pchin.
		REVOKE INTEGRATED LOGIN FROM pchin
		For more information, see the "GRANT statement" [ASA SQL Reference, page 456].

Connecting from a client application

A client application can connect to a database using an integrated login in one of the following ways:

- Set the **Integrated** (**INT**) parameter in the list of connection parameters to YES.
- Specify neither a user ID nor a password in the connection string or Connect dialog. This method is available only for Embedded SQL applications, including the Adaptive Server Anywhere administration utilities.

If Integrated=YES is specified in the connection string, an integrated login is attempted. If the connection attempt fails and the LOGIN_MODE database option is set to Mixed, the server attempts a standard login.

If an attempt to connect to a database is made without providing a user ID or password, an integrated login is attempted. The attempt succeeds or fails depending on whether the current user profile name matches an integrated login mapping in the database.

Interactive SQLFor example, a connection attempt using the following Interactive SQLExamplesstatement will succeed, providing the user has logged on with a user profile

name that matches a integrated login mapping in a default database of a server:

```
CONNECT USING 'INTEGRATED=yes'
```

The Interactive SQL statement

CONNECT

can connect to a database if all the following are true:

- A server is currently running.
- The default database on the current server is enabled to accept integrated login connections.
- An integrated login mapping has been created that matches the current user's user profile name.
- If the user is prompted with a dialog by the server for more connection information (such as occurs when using the Interactive SQL utility), the user clicks OK *without* providing more information.

Integrated logins viaA client application connecting to a database via ODBC can use anODBCintegrated login by including the Integrated (INT) parameter among other
attributes in its Data Source configuration.

Setting the attribute Integrated=YES in an ODBC data source causes database connection attempts using that DSN to attempt an integrated login. If the LOGIN_MODE database option is set to Standard, the ODBC driver prompts the user for a database user ID and password.

Security concerns: unrestricted database access

The integrated login feature works using the login control system of Windows NT/2000/XP in place of the Adaptive Server Anywhere security system. Essentially, the user passes through the database security if they can log in to the machine hosting the database, and if other conditions, outlined in "Using integrated logins" on page 85, are met.

If the user successfully logs in to the Windows NT/2000/XP server as "dsmith", they can connect to the database without further proof of identification provided there is either an integrated login mapping or a default integrated login user ID.

When using integrated logins, database administrators should give special consideration to the way Windows NT/2000/XP enforces login security in order to prevent unwanted access to the database.

In particular, be aware that by default a "Guest" user profile is created and enabled when Windows NT Workstation or Server is installed.

Caution

Leaving the user profile Guest enabled can permit unrestricted access to a database that is hosted by that server.

If the Guest user profile is enabled and has a blank password, any attempt to log in to the server will be successful. It is not required that a user profile exist on the server, or that the login ID provided have domain login permissions. Literally any user can log in to the server using any login ID and any password: they are logged in by default to the Guest user profile.

This has important implications for connecting to a database with the integrated login feature enabled.

Consider the following scenario, which assumes the Windows NT server hosting a database has a Guest user profile that is enabled with a blank password.

An integrated login mapping exists between the user fran_whitney and the database user ID DBA. When the user fran_whitney connects to the server with her correct login ID and password, she connects to the database as DBA, a user with full administrative rights.

But anyone else attempting to connect to the server as **fran_whitney** will successfully log in to the server regardless of the password they provide because Windows NT will default that connection attempt to the Guest user profile. Having successfully logged in to the server using the **fran_whitney** login ID, the unauthorized user successfully connects to the database as **DBA** using the integrated login mapping.

Disable the Guest user profile for security

The safest integrated login policy is to disable the Guest user profile on any Windows NT machine hosting an Adaptive Server Anywhere database. This can be done using the Windows NT User Manager utility.

Setting temporary public options for added security

Setting the value of the LOGIN_MODE option for a given database to Mixed or Integrated using the following SQL statement permanently enables integrated logins for that database.

SET OPTION Public.LOGIN_MODE = Mixed

If the database is shut down and restarted, the option value remains the same and integrated logins are still enabled. Changing the LOGIN_MODE option temporarily will still allow user access via integrated logins. The following statement will change the option value temporarily:

```
SET TEMPORARY OPTION Public.LOGIN_MODE = Mixed
```

If the permanent option value is Standard, the database will revert to that value when it is shut down.

Setting temporary public options can be considered an additional security measure for database access since enabling integrated logins means that the database is relying on the security of the operating system on which it is running. If the database is shut down and copied to another machine (such as a user's machine) access to the database reverts to the Adaptive Server Anywhere security model and not the security model of the operating system of the machine where the database has been copied.

For more information on using the SET OPTION statement see "SET OPTION statement" [ASA SQL Reference, page 556].

Network aspects of integrated logins

If the database is located on a network server, then one of two conditions must be met for integrated logins to be used:

The user profile used for the integrated login connection attempt must exist on both the local machine and the server. As well as having identical user profile names on both machines, the passwords for both user profiles must also be identical.

For example, when the user jsmith attempts to connect using an integrated login to a database loaded on a network server, identical user profile names and passwords must exist on both the local machine and application server hosting the database. jsmith must be permitted to login to both the local machine and the server hosting the network server.

• If network access is controlled by a Microsoft Domain, the user attempting an integrated login must have domain permissions with the Domain Controller server and be logged in to the network. A user profile on the network server matching the user profile on the local machine is not required.

Creating a default integrated login user

A default integrated login user ID can be created so that connecting via an integrated login will be successful even if no integrated login mapping exists for the user profile currently in use.

For example, if no integrated login mapping exists for the user profile name JSMITH, an integrated login connection attempt will normally fail when JSMITH is the user profile in use.

However, if you create a user ID named Guest in a database, an integrated login will successfully map to the Guest user ID if no integrated login mapping explicitly identifies the user profile JSMITH.

The default integrated login user permits anyone attempting an integrated login to successfully connect to a database if the database contains a user ID named Guest. The authorities granted to the Guest user ID determine the permissions and authorities granted to the newly-connected user.
CHAPTER 3

Client/Server Communications

About this chapter	Each network environment has its own peculiarities. This chap those aspects of network communication that are relevant to the functioning of your database server, and provides some tips for network communication problems. It describes how networks of provides hints on running the network database server under ea	ter describes e proper diagnosing operate, and ch protocol.
	Network database server only The material in this chapter applies only to the network server need to read this chapter if you are using the personal database	You do not e server.
Contents	Торіс:	page
	Supported network protocols	94
	Using the TCP/IP protocol	95
	Using the SPX protocol	100
	Using Named Pipes	101
	Adjusting communication compression settings to improve perfor- mance	102
	Troubleshooting network communications	105

Supported network protocols

Properly configured Adaptive Server Anywhere servers run under the following networks and protocols:

- Windows NT/2000/XP and Windows 95/98/Me TCP/IP or SPX protocols.
- Windows CE TCP/IP protocol.
- NetWare All Novell networks using the SPX or TCP/IP protocols.
- ◆ UNIX TCP/IP protocol.

In addition, the Named Pipes protocol is provided for communication from a client application running on the same machine but built for a different operating system.

 Windows NT/2000/XP Named Pipes is used for same-machine communication between any application and the server, but is generally not recommended for this purpose. Named Pipes is not used for network communications.

The client library for each platform supports the same protocols as the corresponding server. In order for Adaptive Server Anywhere to run properly, the network protocols (TCP/IP and/or SPX) must be installed and configured properly on both the client and server computers.

Using the TCP/IP protocol

TCP/IP is a suite of protocols originally implemented by the University of California at Berkeley for BSD UNIX. TCP/IP has gained widespread use with the expansion of the Internet and the World wide web.

UDP is a transport layer protocol that sits on top of IP. Adaptive Server Anywhere uses UDP on top of IP to do initial server name resolution and TCP for connection and communication after that.

When you use the TCP/IP protocol, you can force strong encryption of client/server communications using ECC_TLS (formerly Certicom) or RSA_TLS encryption technology.

For more information about strong encryption, see "Encrypting client/server communications" on page ??.

Using TCP/IP with Windows

The TCP/IP implementation for Windows NT/2000/XP is written to the Winsock 2.0 standard, and the implementation for Windows 95/98/Me and Windows CE uses the Winsock 1.1 standard.

If you do not have TCP/IP installed, chose TCP/IP Protocol from the Control Panel, Network Settings.

Tuning TCP/IP performance

Increasing the packet size may improve query response time, especially for queries transferring a large amount of data between a client and a server process. You can set the packet size using the -p option in the database server command, or by setting the **CommBufferSize (CBSIZE)** connection parameter in your connection profile.

For more information, see "-p server option" on page 154, or "CommBufferSize connection parameter [CBSIZE]" on page 178.

Connecting across a firewall

There are restrictions on connections when the client application is on one side of a firewall, and the server is on the other. Firewall software filters network packets according to network port. Also, it is common to disallow UDP packets from crossing the firewall.

When connecting across a firewall, you must use a set of communication parameters in the **CommLinks** (**LINKS**) connection parameter of your application's connection string.

	• ClientPort Set this parameter to a range of allowed values for the client application to use. You can then configure your firewall to allow these packets across. You can use the short form CPort.
	• Host Set this parameter to the host name on which the database server is running. You can use the short form IP.
	• ServerPort If your database server is not using the default port of 2638, you must specify the port it is using. You can use the short form Port.
	• DoBroadcast=NONE Set this parameter to prevent UDP from being used when connecting to the server.
Example	The following connection string fragment restricts the client application to ports 5050 through 5060, and connects to a server named myeng running on the machine at address myhost using the server port 2020. No UDP broadcast is carried out because of the DoBroadcast option.
	Eng=myeng;Links=tcpip(ClientPort=5050- 5060;Host=myhost;Port=2020;DoBroadcast=NONE)
	For more information, see the following:
	• "CommLinks connection parameter [LINKS]" on page 179
	• "ClientPort communication parameter [CPORT]" on page 204
	• "ServerPort communication parameter [PORT]" on page 215
	 "Host communication parameter [IP]" on page 207
	• "DoBroadcast communication parameter [DOBROAD]" on page 206
Connecting on a d	lial-up network connection
	You can use connection and communications parameters to assist with connecting to a database across a dial-up link.

On the client side, you should specify the following communications parameters:

• Host parameter You should specify the host name or IP address of the database server using the Host (IP) communication parameter.

For more information, see "Host communication parameter [IP]" on page 207.

• **DoBroadcast parameter** If you specify the **Host** (**IP**) communication parameter, there is no need to do a broadcast search for the database server. For this reason, use direct broadcasting.

For more information, see "DoBroadcast communication parameter [DOBROAD]" on page 206.

• MyIP parameter You should set MyIP=NONE on the client side.

For more information, see "MyIP communication parameter [ME]" on page 213.

 TIMEOUT parameter Set the TIMEOUT (TO) communication parameter to increase the time the client will wait while searching for a server.

For more information, see the "Timeout communication parameter [TO]" on page 217.

A typical CommLinks (LINKS) connection parameter may look as follows:

Links=tcpip(MyIP=NONE;DoBroadcast=DIRECT;Host=server_ip)

Encrypting client/server communications over TCP/IP

By default, communication packets are not encrypted; this poses a potential security risk. For greater security, you can force client/server network communications over the TCP/IP port to be encrypted using ECC_TLS (formerly Certicom) or RSA_TLS encryption technology. TCP/IP is the only network protocol that supports this strong encryption.

For information about forcing encryption of client/server communications from the server, see "-ec server option" on page 141.

For information about forcing encryption of client/server communications from a particular client, see "Encryption connection parameter [ENC]" on page 188.

Connecting using an LDAP server

You can specify a central LDAP server to keep track of all servers in an enterprise if you are operating on a Windows (except CE and Win64), Unix or NetWare platform. When the database server registers itself with an LDAP server, clients can query the LDAP server and find the server they are looking for regardless of whether they are on a WAN or LAN, or going through firewalls. They do not need to specify an IP address (HOST=) either. The Server Location utility [dblocate] can use the LDAP server to find other such servers as well.

LDAP is only used with TCP/IP, and only on network servers.

To enable this feature, a file containing information on how to find and connect to the LDAP server must be created on both the server machine and on each client machine. By default the name of this file is *asaldap.ini*, but it is configurable. If this file doesn't exist, LDAP support is silently disabled.

The file must be located in the same directory as the ASA executables (for example, *%asany%*|*win32* on Windows) unless a full path is specified with the LDAP parameter. The file must be in the following format:

```
[LDAP]
server=<machine running LDAP server>
port=<port number of LDAP server>
basedn=<Base DN>
authdn=<Authentication DN>
password=<password for authdn>
search_timeout=<age of timestamps to be ignored>
update_timeout=<frequency of timestamp updates>
```

server the name or IP address of the machine running the LDAP server. Required on NetWare and Unix. If this entry is missing on Windows, Windows looks for an LDAP server running on the local domain controller.

port the port number used by the LDAP server. The default is 389.

basedn the domain name of the subtree where the Adaptive Server Anywhere entries are stored. This defaults to the root of the tree.

authdn the authentication domain name. The domain name must specify an existing user object in the LDAP directory that has write access to the basedn. This is required for the server, and ignored on the client.

password the password for authdn. This is required for the server, and ignored on the client.

search_timeout the age of timestamps at which the timestamp will be ignored by the client and/or the Server Location [dblocate] utility. A value of 0 disables this option so that all entries are assumed to be current. Default is 600 seconds (10 minutes).

update_timeout The frequency of timestamp updates in the LDAP directory. A value of 0 disables this option so that the server never updates the timestamp. The default is 120 seconds (2 minutes).

The following is a sample asaldap.ini file:

```
[LDAP]
server=ldapserver
basedn=dc=iAnywhere,dc=com
authdn=cn=ASAServer,ou=iAnywhereASA,dc=iAnywhere,dc=com
password=secret
```

The entries are stored in a subtree of the basedn called iAnywhereASA. This entry must be created before Adaptive Server Anywhere can use LDAP. To

Example

create the subtree, you can use the LDAPADD utility, supplying the following information:

```
dn: ou=iAnywhereASA,<basedn>
objectClass: organizationalUnit
objectClass: top
ou: iAnywhereASA
```

When the server starts up, it checks for an existing entry with the same name in the LDAP file. If one is found, it is replaced if either

- the location entries in LDAP match the server attempting to start, or
- the timestamp field in the LDAP entry is more than 10 minutes old (the timeout value is configurable)

If neither of these is true, then there is another server running with the same name as the one attempting to start, and startup fails.

To ensure that entries in LDAP are up-to-date, the server updates a timestamp field in the LDAP entry every 2 minutes. If an entry's timestamp is older than 10 minutes, clients ignore the LDAP entry. Both of these values are configurable.

On the client, the LDAP directory is searched before doing any broadcasting, so if the server is found, no broadcasts are sent. The LDAP search is very fast, so if it fails, there is no discernible delay.

The Server Location utility [dblocate] also uses LDAP - all servers listed in LDAP are added to the list of servers returned. This allows the Server Location utility [dblocate] to list servers that wouldn't be returned normally, for example, those which broadcasts wouldn't reach. Entries with timestamps older than 10 minutes are not included.

Using the SPX protocol

SPX is a protocol from Novell. Adaptive Server Anywhere for NetWare, Windows NT/2000/XP, and Windows 95/98/Me can all employ the SPX protocol. This section provides some tips for using SPX under different operating systems.

SPX is not supported on 64-bit operating systems.

Connecting via SPX Some machines can use the NetWare bindery. These machines are NetWare servers or Windows NT/2000/XP or Windows 95/98/Me machines where the Client Service for NetWare is installed. A client application on one of these machines does not need to use broadcasts to connect to the server if the server to which it is connecting is also using the bindery.

Applications running on machines not using the bindery must connect using one of the following:

♦ An explicit address You can specify an explicit address using the HOST (IP) communication parameter.

For more information, see the "Host communication parameter [IP]" on page 207.

 Broadcast If a server is not found in the bindery, the client will attempt to find it using a broadcast. This can be disabled with by setting the DoBroadcast (DOBROAD) communication parameter to NONE (client-side) or NO (server-side).

For more information, see the "DoBroadcast communication parameter [DOBROAD]" on page 206.

Using Named Pipes

Named Pipes is a facility for interprocess communication. Named Pipes can be used for communication between processes on the same computer or on different computers.

Adaptive Server Anywhere for Windows NT/2000/XP uses local Named Pipes for same-machine communication.

For more information, see "Supported network protocols" on page 94.

Adaptive Server Anywhere does not use Named Pipes for client/server communications between different machines.

Adjusting communication compression settings to improve performance

Enabling compression for one or all connections, as well as setting the minimum size at which packets are compressed can significantly improve Adaptive Server Anywhere performance in some circumstances.

To determine if enabling compression will help in your particular situation, we recommend that you conduct a performance analysis on your particular network and using your particular application before using communication compression in a production environment. Performance results will vary according to the type of network you are using, your applications, and the data you transfer.

The most basic way of tuning compression is as simple as enabling or disabling the **Compression** (**COMP**) connection parameter on either a connection or server level. More advanced methods of fine tuning compression performance include adjusting the **CommBufferSize** (**CBSIZE**) and/or the **CompressionThreshold** (**COMPT**) connection parameters.

Enabling compression increases the quantity of information stored in data packets, thereby reducing the number of packets required to transmit a particular set of data. By reducing the number of packets, the data can be transmitted more quickly.

For more information about performance analysis, see "Performance Monitor statistics" on page 638, or "sa_conn_compression_info system procedure" [ASA SQL Reference, page 708].

Enabling Compression Enabling compression for a connection (or all connections) can significantly improve Adaptive Server Anywhere performance under some circumstances, including:

- when used over slow networks such as some wireless networks, some modems, serial links and some WANs.
- when used in conjunction with Adaptive Server Anywhere encryption over a slow network with built-in compression, since packets are compressed before they are encrypted.

Enabling compression, however, can sometimes also cause slower performance. For instance,

 communication compression uses more memory and more CPU. It may cause slower performance, especially for LANs and other fast networks.

	 most modems and some slow networks already have built-in compression. In these cases, Adaptive Server Anywhere communication compression will not likely provide additional performance benefits unless you are also encrypting the data.
	For more information about compression, see "Compress connection parameter [COMP]" on page 181, "-pc server option" on page 154.
Modifying the CommBufferSize setting	While CommBufferSize (CBSIZE) is not a compression parameter in itself, adjusting this parameter can benefit compression, especially if you are transferring a large amount of data between a client and server. A larger packet size can improve performance for multi-row fetches and fetches of larger rows, as well as inserting or retrieving BLOBs.
	As always, however, there are tradeoffs with performance improvements. Since each connection has its own pool of buffers, a large buffer size also increases the memory usage. Be sure to analyze your particular situation to make sure that the benefits of increasing the CommBufferSize outweigh the costs.
	For more information see "Tuning TCP/IP performance" on page 95, the "CommBufferSize connection parameter [CBSIZE]" on page 178, or the "-p server option" on page 154.
Modifying the compression threshold	You can also adjust the compression threshold to improve Adaptive Server Anywhere performance. For most networks, the compression threshold does not need to be changed.
	When compression is enabled, individual packets may or may not be compressed, depending on their size. For example, Adaptive Server Anywhere does not compress packets smaller than the compression threshold, even if communication compression is enabled. As well, small packets (less than about 100 bytes) usually do not compress at all. Since CPU time is required to compress packets, attempting to compress small packets could actually decrease performance.
	Generally speaking, lowering the compression threshold value may improve performance on very slow networks, while raising the compression threshold may improve performance by reducing CPU usage. However, since lowering the compression threshold value will increase CPU usage on both the client and server, a performance analysis should be done to determine whether or not changing the compression threshold is beneficial.
	For more information see "CompressionThreshold connection parameter [COMPTH]" on page 182 and "-pt server option" on page 155.

To adjust Adaptive Server Anywhere compression settings

1. Enable communication compression.

Large data transfers with highly compressible data and larger packet sizes tend to get the best compression rates.

For more information about enabling compression, see "Compress connection parameter [COMP]" on page 181 and "-pc server option" on page 154.

2. Adjust the CommBufferSize setting.

Increasing Adaptive Server Anywhere's packet size can improve compression performance.

For more information about adjusting the CommBufferSize setting, see "CommBufferSize connection parameter [CBSIZE]" on page 178 or "-p server option" on page 154.

3. Adjust the CompressionThreshold setting.

Lowering the compression threshold value may improve performance on very slow networks, while raising the compression threshold may improve performance by reducing CPU usage.

For more information about adjusting the **CompressionThreshold** (**COMPT**) connection parameter, see "CompressionThreshold connection parameter [COMPTH]" on page 182 and "-pt server option" on page 155.

Troubleshooting network communications

Network software involves several different components, increasing the likelihood of problems. Although we provide some tips concerning network troubleshooting here, the primary source of assistance in network troubleshooting should be the documentation and technical support for your network communications software, as provided by your network communications software vendor.

Ensure that you are using compatible protocols

Ensure that the client and the database server are using the same protocol. The -x option for the server selects a list of protocols for the server to use, and the **CommLinks (LINKS)** connection parameter does the same for the client application.

You can use these options to ensure that each application is using the same protocol.

By default, both the database server and client library use all available protocols. The server supports client requests on any active protocol, and the client searches for a server on all active protocols.

For more information about the -x option, see the "-x server option" on page 162.

For information about the **CommLinks** (**LINKS**) connection parameter, see "CommLinks connection parameter [LINKS]" on page 179.

Ensure that you have current drivers

Old network adapter drivers can be a source of communication problems. You should ensure that you have the latest version of your network adapter. You should be able to obtain current network adapter drivers from the manufacturer or supplier of the card.

Testing the TCP/IP protocol

	The ping utility can be useful for testing that TCP/IP is installed and configured properly.
Using ping to test the IP layer	Each IP layer has an associated address—a four-integer, period-separated number (such as 191.72.109.12). Ping takes as an argument an IP-address and attempts to send a single packet to the address.
	First, determine if your own machine is configured correctly by "pinging" yourself. If your IP-address is 191.72.109.12, you would type:

ping 191.72.109.12

at the command prompt and wait to see if the packets are routed at all. If they are, the output will appear similar to the following:

```
c:> ping 191.72.109.12
Pinging 191.72.109.12 with 32 bytes of data:
Reply from 191.72.109.12: bytes=32 time<.10ms TTL=32
Reply from 191.72.109.12: bytes=32 time<.10ms TTL=32
Reply from 191.72.109.12: bytes=32 time<.10ms TTL=32
...</pre>
```

If this works, it means that the computer is able to route packets to itself. This is reasonable assurance that the IP layer is set up correctly. You could also ask someone else running TCP/IP for their IP address and try pinging them.

You should ensure that you can ping the computer running the database server from the client computer before proceeding.

Diagnosing wiring problems

Faulty network wiring or connectors can cause problems that are difficult to track down. Try recreating problems on a similar machine with the same configuration. If a problem occurs on only one machine, it may be a wiring problem or a hardware problem.

A checklist of common problems

The following list presents some common problems and their solutions.

For information about troubleshooting connections to the database or database server, see "Troubleshooting connections" on page 75 and "Troubleshooting server startup" on page 34.

If you receive the message Unable to start--server not found when trying to start the client, the client cannot find the database server on the network. Check for the following problems:

- Under the TCP/IP protocol, clients search for database servers by broadcasting a request. Such broadcasts will typically not pass through gateways, so any database server on a machine in another (sub)network, will not be found. If this is the case, you must supply the host name of the machine on which the server is running using the -x option.
- A firewall between the client and server may be preventing the connection. For more information, see "Connecting across a firewall" on page 95.

- Your network drivers are not installed properly or the network wiring is not installed properly.
- ◆ If you receive the message Unable to initialize any communication links, no link can be established. The probable cause is that your network drivers have not been installed. The server tries to start communication links using all available protocols unless you have specified otherwise using the -x option. Check your network documentation to find out how to install the driver you wish to use.
- You must use the TCP/IP protocol if you are connecting via jConnect.
- If you are trying to connect to a database server on your local machine, make sure the Search Network For Database Servers option on the Database tab of the Connect dialog is cleared. You can select this option if you are trying to connect to a database server running on a machine other than your local machine.

For more information about network communication parameters, see "Network communications parameters" on page 201.

Adjusting timeout values

If you are experiencing problems with connections unexpectedly terminating, consider adjusting either the liveness or the idle timeout values.

For more information about liveness timeout values, see "LivenessTimeout connection parameter [LTO]" on page 194, and the "-tl server option" on page 159.

For more information about connection timeouts, see the "Idle connection parameter [IDLE]" on page 192, and the "-ti server option" on page 159.

CHAPTER 4

Adaptive Server Anywhere as an Open Server

About this chapter	Adaptive Server Anywhere can appear to client application Server. This feature enables Sybase Open Client application natively to Adaptive Server Anywhere databases.	is as an Open ons to connect
	This chapter describes how to use Adaptive Server Anywh Server, and how to configure Open Client and Adaptive Se work together.	ere as an Open rver Anywhere to
Contonto	For information on developing Open Client application Adaptive Server Anywhere, see "The Open Client Interfact <i>Programming Guide</i> , page 447].	ns for use with e" [<i>ASA</i>
Contents	Торіс:	page
	Open Clients, Open Servers, and TDS	110
	Setting up Adaptive Server Anywhere as an Open Server	112
	Configuring Open Servers	114
	Characteristics of Open Client and jConnect connections	120

Open Clients, Open Servers, and TDS

	This chapter describes how Adaptive Server Anywhere fits into the Sybase Open Client/Open Server client/server architecture. This section describes the key concepts of this architecture, and provides the conceptual background for the rest of the chapter.
	If you simply wish to use a Sybase application with Adaptive Server Anywhere, you do not need to know any details of Open Client, Open Server, or TDS. However, an understanding of how these pieces fit together may be helpful for configuring your database and setting up applications. This section explains how the pieces fit together, but avoids any discussion of the internal features of the pieces.
Open Clients and Open Servers	Adaptive Server Anywhere and other members of the Adaptive Server family act as Open Servers . This means that you can develop client applications using the Open Client libraries available from Sybase. Open Client includes both the Client Library (CT-Library) and the older DB-Library interfaces.
Tabular Data Stream	Open Clients and Open Servers exchange information using an application protocol called the tabular data stream (TDS). All applications built using the Sybase Open Client libraries are also TDS applications because the Open Client libraries handle the TDS interface. However, some applications (such as Sybase jConnect) are TDS applications even though they do not use the Sybase Open Client libraries—they communicate directly using TDS protocol.
	While many Open Servers use the Sybase Open Server libraries to handle the interface to TDS, some applications have a direct interface to TDS of their own. Sybase Adaptive Server Enterprise and Adaptive Server Anywhere both have internal TDS interfaces. They appear to client applications as an Open Server, but do not use the Sybase Open Server libraries.
Programming interfaces and application protocols	Adaptive Server Anywhere supports two application protocols. Open Client applications and other Sybase applications such as Replication Server and OmniConnect use TDS. ODBC and Embedded SQL applications use a separate application protocol specific to Adaptive Server Anywhere.
TDS uses TCP/IP	Application protocols such as TDS sit on top of lower-level communications protocols that handle network traffic. Adaptive Server Anywhere supports TDS only over the TCP/IP network protocol. In contrast, the Adaptive Server Anywhere-specific application protocol supports several network protocols, as well as a shared memory protocol designed for same-machine communication.

Sybase applications and Adaptive Server Anywhere

	The ability of Adaptive Server Anywhere to act as an Open Server enables Sybase applications such as Replication Server and OmniConnect to work with Adaptive Server Anywhere.
Replication Server support	The Open Server interface enables support for Sybase Replication Server: Replication Server connects through the Open Server interface, enabling databases to act as replicate sites in Replication Server installations.
	For your database to act as a primary site in a Replication Server installation, you must also use the Replication Agent for Sybase Adaptive Server Anywhere, also called a Log Transfer Manager .
	For information on the Replication Agent, see "Replicating Data with Replication Server" on page 425.
OmniConnect support	Sybase OmniConnect provides a unified view of disparate data within an organization, allowing users to access multiple data sources without having to know what the data looks like or where to find it. In addition, OmniConnect performs heterogeneous joins of data across the enterprise, enabling cross-platform table joins of targets such as DB2, Sybase Adaptive Server Enterprise, Oracle, and VSAM.
	Using the Open Server interface, Adaptive Server Anywhere can act as a data source for OmniConnect.

Setting up Adaptive Server Anywhere as an Open Server

This section describes how to set up an Adaptive Server Anywhere server to receive connections from Open Client applications.

System requirements

There are separate requirements at the client and server for using Adaptive Server Anywhere as an Open Server. Server-side requirements You must have the following elements at the server side to use Adaptive Server Anywhere as an Open Server: ♦ Adaptive Server Anywhere server components You must use the network server (dbsrv9.exe) if you want to access an Open Server over a network. You can use the personal server (dbeng9.exe) as an Open Server only for connections from the same machine. ◆ **TCP/IP** You must have a TCP/IP protocol stack to use Adaptive Server Anywhere as an Open Server, even if you are not connecting over a network. You need the following elements to use Sybase client applications to connect Client-side requirements to an Open Server (including Adaptive Server Anywhere): • Open Client components The Open Client libraries provide the network libraries your application needs to communicate via TDS, if your application uses Open Client. jConnect If your application uses JDBC, you need jConnect and a Java runtime environment. • **DSEdit** You need *DSEdit*, the directory services editor, to make server names available to your Open Client application. On UNIX platforms, this utility is called sybinit. DSEdit is not included with SQL Anywhere Studio, but is included with

Open Server software.

Starting the database server as an Open Server

If you wish to use Adaptive Server Anywhere as an Open Server, you must ensure that you start it using the TCP/IP protocol. By default, the server starts all available communications protocols, but you can limit the protocols started by listing them explicitly in the command. For example, the following commands are both valid:

	dbsrv9 -x tcpip,spx asademo.db dbsrv9 -x tcpip -n myserver asademo.db
	The first command uses both TCP/IP and SPX protocols, of which TCP/IP is available for use by Open Client applications. The second line uses only TCP/IP.
	You can use the personal database server as an Open Server for communications on the same machine because it supports the TCP/IP protocol.
	The server can serve other applications through the TCP/IP protocol or other protocols using the Adaptive Server Anywhere-specific application protocol at the same time as serving Open Client applications over TDS.
Port numbers	Every application using TCP/IP on a machine uses a distinct TCP/IP port so that network packets end up at the right application. The default port for Adaptive Server Anywhere is port 2638. It is recommended that you use the default port number as Adaptive Server Anywhere has been granted that port number by the Internet Assigned Numbers Authority (IANA). If you wish to use a different port number, you can specify which one using the ServerPort (PORT) communication parameter:
	dbsrv9 -x tcpip(ServerPort=2629) -n myserver asademo.db
	You may also need to supply an EngineName if more than one local database server is running, or if you wish to connect to a network server.
Open Client settings	To connect to this server, the interfaces file at the client machine must contain an entry specifying the machine name on which the database server is running, and the TCP/IP port it uses.
	For details on setting up the client machine, see "Configuring Open Servers" on page 114.

Configuring Open Servers

Adaptive Server Anywhere can communicate with other Adaptive Servers, Open Server applications, and client software on the network. Clients can talk to one or more servers, and servers can communicate with other servers via remote procedure calls. For products to interact with one another, each needs to know where the others reside on the network. This network service information is stored in the interfaces file.

The interfaces file

The interfaces file is usually named *SQL.ini* on PC operating systems and *interfaces*, or *interfac* on UNIX operating systems.

Like an address book, the interfaces file lists the name and address of every database server known to Open Client applications on your machine. When you use an Open Client program to connect to a database server, the program looks up the server name in the interfaces file and then connects to the server using the address.

The name, location, and contents of the interfaces file differ between operating systems. Also, the format of the addresses in the interfaces file differs between network protocols.

When you install Adaptive Server Anywhere, the setup program creates a simple interfaces file that you can use for local connections to Adaptive Server Anywhere over TCP/IP. It is the System Administrator's responsibility to modify the interfaces file and distribute it to users so that they can connect to Adaptive Server Anywhere over the network.

Using the DSEdit utility

The *DSEdit* utility is a Windows utility that allows you to configure the interfaces file (*SQL.ini*). The following sections explain how to use the *DSEdit* utility to configure the interfaces file.

These sections describe how to use *DSEdit* for those tasks required for Adaptive Server Anywhere. It is not complete documentation for the *DSEdit* utility. For more information on *DSEdit*, see the *Utility Programs* book for your platform, included with other Sybase products.

Starting DSEdit

The *DSEdit* executable is in the *SYBASE**bin* directory, which is added to your path on installation. You can start *DSEdit* either from the command prompt or from the Explorer in the standard fashion.

When you start DSEdit, the Select Directory Service dialog appears.

5 Name: terfacesDriver	Οκ
	Cancel
	1

Opening a Directory Services session

The Select Directory Service window allows you to open a session with a directory service. You can open a session to edit the interfaces file (*SQL.ini*), or any directory service that has a driver listed in the *libtcl.cfg* file.

To open a session

1. Click the local name of the directory service you want to connect to, as listed in the DS Name box, and click OK.

For Adaptive Server Anywhere, select the Interfaces Driver.

SYBASE environment variable must be set

The *DSEdit* utility uses the SYBASE environment variable to locate the *libtcl.cfg* file. If the SYBASE environment variable is incorrect, *DSEdit* cannot locate the *libtcl.cfg* file.

DSEDIT1 - Inte	rfacesDriver		X
Server	Attributes	Values	
	_		
Path			

You can add, modify, or delete entries for servers, including Adaptive Server Anywhere servers, in this window.

Adding a server entry

To add a server entry

1. Choose Add from the Server Object menu.

The Input Server Name window appears.

2. Type a server name in the Server Name box, and click OK to enter the server name.

The server name entry must match the name you used to start Adaptive Server Anywhere with. The server *address* is used to identify and locate the server. The server name field is an identifier for Open Client. For Adaptive Server Anywhere, if the server has more than one database loaded, the *DSEdit* server name entry identifies which database to use.

The server entry appears in the Server box. To specify the attributes of the server, you must modify the entry.

Adding or changing the server address

Once you have entered a Server Name, you need to modify the Server Address to complete the interfaces file entry.

To enter a Server Address

- 1. Select a server entry in the Server box.
- 2. Right-click the Server Address in the Attributes box.
- 3. Choose Modify Attribute from the popup menu. A window appears, showing the current value of the address. If you have no address entered, the box is empty.

 letwork Addre	ess				i Down
Add		Edit	1	Delete	Ì

- 4. Click Add. The Network Address for Protocol window appears.
- 5. Select NLWNSCK from the Protocol list box (this is the TCP/IP protocol) and enter a value in the Network Address text box.

:5:
Cancel

For TCP/IP addresses, use one of the following two forms:

- computer name, name, port number
- IP-address, portnumber

	The address or computer name is separated from the port number by a comma.
Machine name	A name (or an IP address) identifies the machine on which the server is running. On Windows operating systems, you can find the machine name in Network Settings, in the Control Panel.
	If your client and server are on the same machine, you must still enter the machine name. In this case, you can use localhost to identify the current machine.
Port Number	The port number you enter must match the port number you used to start the Adaptive Server Anywhere database server with, as described in "Starting the database server as an Open Server" on page 112. The default port number for Adaptive Server Anywhere servers is 2638. This number has been assigned to Adaptive Server Anywhere by the Internet Adapter Number Authority (IANA), and use of this port is recommended unless you have good reasons for explicitly using another port.
	The following are valid server address entries:

elora,2638 123.85.234.029,2638

Verifying the server address

You can verify your network connection using the Ping command from the Server Object menu.

Database connections not verified

Verifying a network connection confirms that a server is receiving requests on the machine name and port number specified. It does not verify anything about database connections.

To ping a server

- 1. Ensure that the database server is running.
- 2. Click the server entry in the Server box of the DSEdit session window.
- 3. Choose Ping Server from the Server Object menu. The Ping window appears.
- 4. Select the address you want to ping. Click Ping.

A message box appears, notifying you whether or not the connection is successful. A message box for a successful connection states that both Open Connection and Close Connection succeeded.

Renaming a server entry

You can rename server entries from the DSEdit session window.

To rename a server entry

- 1. Select a server entry in the Server box.
- 2. Choose Rename from the Server Object menu. The Input Server Name window appears.
- 3. Type a new name for the server entry in the Server Name box.
- 4. Click OK to make the change.

Deleting server entries

You can delete server entries from the DSEdit session window.

To delete a server entry

- 1. Click a server entry in the Server box.
- 2. Choose Delete from the Server Object menu.

Configuring servers for JDBC

The JDBC connection address (URL) contains all the information required to locate the server.

For information on the JDBC URL, see "Supplying a URL for the server" [*ASA Programming Guide*, page 112].

Characteristics of Open Client and jConnect connections

When Adaptive Server Anywhere is serving applications over TDS, it automatically sets relevant database options to values compatible with Adaptive Server Enterprise default behavior. These options are set temporarily, for the duration of the connection only. The client application can override them at any time.

Default settings

The database options set on connection using TDS include:

Option	Set to
ALLOW_NULLS_BY_DEFAULT	OFF
ANSINULL	OFF
ANSI_BLANKS	ON
ANSI_INTEGER_OVERFLOW	ON
AUTOMATIC_TIMESTAMP	ON
CHAINED	OFF
CONTINUE_AFTER_RAISERROR	ON
DATE_FORMAT	YYYY-MM-DD
DATE_ORDER	MDY
ESCAPE_CHARACTER	OFF
ISOLATION_LEVEL	1
FLOAT_AS_DOUBLE	ON
QUOTED_IDENTIFIER	OFF
TIME_FORMAT	HH:NN:SS.SSS
TIMESTAMP_FORMAT	YYYY-MM-DD HH:NN:SS.SSS
TSQL_HEX_CONSTANT	ON
TSQL_VARIABLES	ON

How the startup options The default database options are set for TDS connections using a system procedure named sp_tsql_environment. This procedure sets the following options:

```
SET TEMPORARY OPTION TSQL_VARIABLES='ON';
SET TEMPORARY OPTION ANSI BLANKS='ON';
SET TEMPORARY OPTION TSQL_HEX_CONSTANT='ON';
SET TEMPORARY OPTION CHAINED='OFF';
SET TEMPORARY OPTION QUOTED_IDENTIFIER='OFF';
SET TEMPORARY OPTION ALLOW_NULLS_BY_DEFAULT='OFF';
SET TEMPORARY OPTION AUTOMATIC TIMESTAMP='ON';
SET TEMPORARY OPTION ANSINULL='OFF';
SET TEMPORARY OPTION CONTINUE_AFTER_RAISERROR='ON';
SET TEMPORARY OPTION FLOAT AS DOUBLE='ON';
SET TEMPORARY OPTION ISOLATION LEVEL='1';
SET TEMPORARY OPTION DATE_FORMAT='YYYY-MM-DD';
SET TEMPORARY OPTION TIMESTAMP_FORMAT='YYYY-MM-DD HH:NN:SS.SSS';
SET TEMPORARY OPTION TIME_FORMAT='HH:NN:SS.SSS';
SET TEMPORARY OPTION DATE ORDER='MDY';
SET TEMPORARY OPTION ESCAPE_CHARACTER='OFF'
```

Do not edit the sp_tsql_environment procedure

Do not alter the sp_tsql_environment procedure yourself. It is for system use only.

The procedure sets options only for connections that use the TDS communications protocol. This includes Open Client and JDBC connections using jConnect. Other connections (ODBC and Embedded SQL) have the default settings for the database.

You can change the options for TDS connections.

To change the option settings for TDS connections

1. Create a procedure that sets the database options you want. For example, you could use a procedure such as the following:

```
CREATE PROCEDURE my_startup_procedure()
BEGIN
IF connection_property('CommProtocol')='TDS' THEN
SET TEMPORARY OPTION QUOTED_IDENTIFIER='OFF';
END IF
END
```

This particular procedure example changes only the QUOTED_IDENTIFIER option from the default setting.

2. Set the LOGIN_PROCEDURE option to the name of a new procedure:

SET OPTION LOGIN_PROCEDURE= 'DBA.my_startup_procedure'

Future connections will use the procedure. You can configure the procedure differently for different user IDs.

Tor more information about database options, see "Database Options" on page 555.

F

CHAPTER 5

The Database Server

About this chapter	This chapter describes the line options for database server.	the Adaptive Server Anywhere
	It also contains information for the option for compatibility with Version 5 software	s of the client executable (provided)).
Contents	Торіс:	page
	The database server	124

The database server

Function

Start a personal database server or network database server.

1

Syntax	{ dbeng9 dbsrv9 } [server-options] [database-file [database-options]]
NetWare syntax	load dbsrv9 [server-options] [database-file [database-options]]

Server options

Server Option	Description
@environment- variable	Read in options from an environment variable. See "@environment-variable server option" on page 131.
@filename	Read in options from a configuration file. See "@file- name server option" on page 131.
-?	Display usage information. See "-? server option" on page 133.
-b	Run in bulk operations mode. See "-b server option" on page 133.
-c size	Set initial cache size. See "-c server option" on page 133.
-ca 0	Disable dynamic cache sizing [Windows NT/2000/XP, Windows 95/98/Me, UNIX]. See "-ca server option" on page 135.
-ch size	Set the cache size upper limit [Windows NT/2000/XP, Windows 95/98/Me]. See "-ch server option" on page 136.
-cl size	Set the cache size lower limit [Windows NT/2000/XP]. See "-cl server option" on page 136.
-cs	Display cache usage in database server window. See "-cs server option" on page 137.
-ct { + - }	Turn character-set translation on and off [not NetWare or Windows CE]. See "-ct server option" on page 137.

Server Option	Description
-cw	Enable use of Address Windowing Extensions on Win- dows 2000 and Windows XP for setting the size of the database server cache. See "-cw server option" on page 138.
-d	Use POSIX I/O [NetWare]. See "-d server option" on page 141.
-ec encryption- options	Enable packet encryption [network server]. See "-ec server option" on page 141.
-ер	Prompt for encryption key. See "-ep server option" on page 143.
-ga	Automatically unload the database after the last con- nection closed. In addition, shut down after the last database is closed [Not NetWare]. See "-ga server option" on page 144.
-gb level	Set database process priority class to <i>level</i> [Win- dows NT/2000/XP]. See "-gb server option" on page 144.
-gc num	Set maximum checkpoint timeout period to <i>num</i> min- utes. See "-gc server option" on page 144.
-gd level	Set database starting permission. See "-gd server option" on page 145.
-ge size	Set the stack size for threads that run external functions [not UNIX]. See "-ge server option" on page 146.
-gf	Disable firing of triggers. See "-gf server option" on page 146.
-gk level	Set the permission required to stop the server. See "-gk server option" on page 146.
-gl level	Set the permission required to load or unload data. See "-gl server option" on page 147.
-gm num	Set the maximum number of connections. See "-gm server option" on page 147.
-gn num	Set the maximum number of concurrent requests the database server can handle at one time. See "-gn server option" on page 148.

Server Option	Description
-gp size	Set the maximum page size to <i>size</i> bytes. See "-gp server option" on page 149.
-gr minutes	Set the maximum recovery time to <i>num</i> minutes. See "-gr server option" on page 149.
-gss size	Set the thread stack size to <i>size</i> bytes [not applicable to Windows]. See "-gss server option" on page 149.
-gt num	Set the number of operating processors used by the database server. See "-gt server option" on page 150.
-gu level	Set the permission level for utility commands: utilitydb , all , none , or DBA . See "-gu server option" on page 150.
-gx	Set the number of operating system threads assigned to the database server process. [Windows NT/2000/XP, Windows 95/98/Me]. See "-gx server option" on page 150.
-m	Truncate the transaction log after each checkpoint, for all databases. See "-m server option" on page 151.
-n name	Use <i>name</i> as the name of the database server. Note that the -n switch is positional. See "-n server option" on page 152.
-o filename	Output messages to the specified file. See "-o server option" on page 153.
-oe filename	Filename to log startup errors, fatal errors and assertions to. See "-oe server option" on page 153.
-os size	Limit the size of the log file for messages. See "-os server option" on page 154.
-p packet-size	Set the maximum network packet size [network server]. See "-p server option" on page 154.
-рс	Compress all connections except same-machine connec- tions. See "-pc server option" on page 154.
-pt size_in_bytes	Set the minimum network packet size to compress. See "-pt server option" on page 155.
-qi	Do not display database server tray icon or screen. See "-qi server option" on page 156.

Server Option	Description
-qp	Suppress messages about performance in the database server window. See "-qp server option" on page 156.
-qs	Suppress startup error dialogs. See "-qs server option" on page 156.
-qw	Do not display database server screen. See "-qw server option" on page 157.
-r	Opens database in read-only mode. See "-r server option" on page 157.
-S	Set the syslog facility ID [UNIX]. See "-s server option" on page 158.
-sb { 0 1 }	Specify how the server reacts to broadcasts. See "-sb server option" on page 158.
-sc	Disable the shared memory port, and enable Named Pipes. [Windows NT/2000/XP]. See "-sc server option" on page 158.
-ti minutes	Client idle time before shutdown—default 240 minutes. See "-ti server option" on page 159.
-tl seconds	Default liveness timeout for clients in seconds—default 120 seconds. See "-tl server option" on page 159.
-tmf	Force transaction manager recovery for distributed transactions [Windows NT/2000/XP]. See "-tmf server option" on page 160.
-tmt milliseconds	Set the reenlistment timeout for distributed transactions [Windows NT/2000/XP]. See "-tmt server option" on page 160.
-tq time	Set quitting time [network server]. See "-tq time server option" on page 161.
-u	Use buffered disk I/O. See "-u server option" on page 161.
-ud	Run as a daemon [UNIX]. See "-ud server option" on page 162.
-ut minutes	Touch temporary files every <i>min</i> minutes [UNIX]. See "-ut server option" on page 162.

Server Option	Description
-v	Display database server version and stop. See "-v server option" on page 162.
-x list	Comma-separated list of communication links to try. See "-x server option" on page 162.
- y	Run as a Windows 95/98/Me service [Win- dows 95/98/Me]. See "-y server option" on page 165.
-Z	Provide diagnostic information on communication links [network server]. See "-z server option" on page 165.
-zl	Turn on capturing of the most recently-prepared SQL statement for each connection. See "-zl server option" on page 166.
-zo filename	Redirect request-level logging information to a separate file. See "-zo server option" on page 166.
-zr { all SQL none }	Turn on logging of SQL operations. The default is NONE. See "-zr server option" on page 167.
-zs size	Limit the size of the log file used for request-level logging. See "-zs server option" on page 167.

Recovery options

Recovery Option	Description
-a filename	Apply the named transaction log file. See "-a recovery option" on page 167.
-f	Force the database to start without a transaction log. See "-f recovery option" on page 168.

Database options
	Database Option	Description
	-ek key	Specify encryption key. See "-ek database option" on page 168.
	-m	Truncate (delete) the transaction log after each check- point for the specified database. See "-m database option" on page 169.
	-n name	Name the database. Note that the -n switch is positional. See "-n database option" on page 170.
	-r	Opens the specified database(s) in read-only mode. Database modifications not allowed. See "-r database option" on page 170.
Description	The dbeng9 comm command starts a n	and starts a personal database server. The dbsrv9 etwork database server.
Cache size	The amount of cach factor in affecting p of cache memory th value.	The memory available to the database server can be a key performance. The database server takes an initial amount that is either specified by the $-c$ option or is a default
	For information page 133.	n on the default cache size, see "-c server option" on
	On Windows NT/2 server automaticall determined by a he	000/XP, Windows 95/98/Me, and UNIX, the database y takes more memory for use in the cache as needed, uristic algorithm.
	For more information [ASA SQL User's Guilding	mation, see "Using the cache to improve performance" <i>ide</i> , page 176].
	You can use databa server option" on p initial amount: see	se options to configure the upper cache limit: see "-ch age 136. As well, you can force the cache to remain at its "-ca server option" on page 135.
Server differences	The personal databa uses at most two Cl client/server connect	ase server has a maximum of ten concurrent connections, PUs for request processing, and does not support network ctions.
	In addition, there an level that is require execute the CHECH	re other minor differences, such as the default permission d to start new databases, or the permissions required to KPOINT statement.
Platform availability	Both personal and a operating system, v	network database servers are supplied for each supported with the following exceptions:
	Novell NetWare	Only the network server is supplied.

	• Windows CE Only the network server is supplied. The support for TCP/IP in the network server enables you to carry out tasks from your desktop machine, including database management, with Sybase Central.
NetWare notes	In NetWare, the database file and the transaction log file must be on a NetWare volume, and the paths must be fully specified. NetWare allows you to have volumes that span two or more hard disks.
	Database file The <i>database-file</i> specifies the database filename. If <i>database-file</i> is specified without a file extension, Adaptive Server Anywhere looks first for <i>database-file</i> with extension <i>.wrt</i> (a write file) followed by <i>database-file</i> with extension <i>.db</i> .
	If you use a relative path, it is read relative to the current working directory. You can supply a full path. Also, you can supply a path that conforms to the Universal Naming Convention (UNC) format:
	<pre>\\server\volume\path\file.ext</pre>
Suppressing Windows event log	If you run the database server as a Windows service, you can suppress Windows event log entries by setting a registry entry. The registry entry is
messages	Software\Sybase\Adaptive Server Anywhere\9.0
	To control event log entries, set the EventLogMask key, which is of type REG_DWORD. The value is a bit mask containing the internal bit values for the different types of event messages:
	errors EVENTLOG_ERROR_TYPE 0x0001 warnings EVENTLOG_WARNING_TYPE 0x0002 information EVENTLOG_INFORMATION_TYPE 0x0004
	For example, if the EventLogMask key is set to zero, no messages appear at all. A better setting would be 1, so that informational and warning messages do not appear, but errors do. The default setting (no entry present) is for all message types to appear.
See also	"Running the Database Server" on page 3
	"Network communications parameters" on page 201
Operating quietly	The database server supports quiet mode. You determine how quiet you want the server to operate, ranging from suppressing messages or the icon in the system tray, to complete silence. To operate a completely silent database server on Windows, specify the -qi -qs options. With these options set, there is no visual indication that the server is running, as all icons and all possible startup error messages are suppressed. If you run the database server in quiet mode, you can use either (or both) the -o or -oe options to diagnose errors.
	Note that the -qi and -qs options do not suppress error dialogs caused by the

-v (version) and -ep (prompt for database encryption password) server options.

Database server options

These options apply to the server as a whole, not just to an individual database.

@environment-variable server option

Function	Read in server options from the supplied environment variable.
Syntax	{ dbsrv9 dbeng9 } @env-var
Applies to	All operating systems and servers.
Description	The environment variable may contain any set of options. For example, this statement (typed all on one line) sets an environment variable that holds options for a database server that starts with a cache size of 4 Mb and loads the sample database.
	set envvar=-c 4096 "c:\Program Files\Sybase\SQL Anywhere 9\ asademo.db"

This statement starts the database server using the environment variable.

dbsrv9 @envvar

Environment variable given priority

If you have both a file and an environment variable with the value of your @ option, the environment variable is used.

@filename server option

Function	Read in server options from the supplied file. The number sign (#) must precede comments.
Syntax	{ dbsrv9 dbeng9 } @filename
Applies to	All operating systems and servers, as well as all database utilities except Interactive SQL [dbisql], the Debugger utility [dbprdbg], the Language utility [dblang], and the Console utility [dbconsole].
Description	The command file may contain line breaks, and may contain any set of options.
	If you want to protect the information in a configuration file (for example, because it contains passwords) you can use the File Hiding [dbfhide] utility

	to obfuscate the contents of configuration files.
	For more information about the File Hiding utility, see "The File Hiding utility" on page 466.
	All database utilities except Interactive SQL [dbisql], the Debugger utility [dbprdbg], the Language utility [dblang], and the Console utility [dbconsole] accept the @filename parameter. The filename can occur at any point in the command line, and parameters contained in the file are inserted at that point. Multiple files can be specified, and the file specifier can be used with command line switches. The @file syntax is not supported within a file.
	@file parameters will be left untouched if they occur after the -w switch. This can be useful, for example, if you want to use the Service Creation utility [dbsvc] and to create a service where the server (or MobiLink) reads the parameter file itself.
Examples	The following reads all parameters from dbinit.in
	dbinit @dbinit.in
	The following example is the same as dbinit -q -ja -p 2048 -b new.db
	echo "-ja -p 2048" > dbinit.in dbinit -q @dbinit.in -b new.db
	In the following example, dbsvc_parms will be read by dbsvc, but dbsrv_parms will not. The service will be created with the command line dbsrv9.exe @dbsrv_parms.
	dbsvc @dbsvc_parms -w mysvc dbsrv9.exe @dbsrv_parms
	The following command file holds a set of options for a server named myserver that starts with a cache size of 4 Mb and loads the sample database:
	-c 4096 -n myserver "c:\Program Files\Sybase\SQL Anywhere 9\asademo.db"

If this configuration file is saved as *c:\config.txt*, it can be used in a command as follows:

dbsrv9 @c:\config.txt

The following command file contains comments:

```
#This is the server name:
-n MyServer
#These are the protocols:
-x tcpip
#This is the database file
my.db
```

-? server option

Function	Display usage information.
Syntax	{ dbsrv9 dbeng9 } -?
Applies to	All operating systems and servers.
Description	Display a short description of each server option. The database does not carry out any other task.

-b server option

Function	Use bulk operation mode.
Syntax	{ dbsrv9 dbeng9 } -b
Applies to	All operating systems and servers.
Description	This is useful for using the Interactive SQL INPUT command to load large quantities of data into a database.
	The -b option should not be used if you are using LOAD TABLE to bulk load data.
	When you use this option, the database server allows only one connection by one application. It keeps a rollback log, but it does not keep a transaction log. The multi-user locking mechanism is turned off.
	When you first start the database server after loading data with the -b option, you should use a new log file.
	Bulk operation mode does not disable the firing of triggers.
-c server option	
Function	Set the initial memory reserved for caching database pages and other server information.

Syntax	{ dbsrv9 dbeng9 } -c { integer integerG integerK integerP }
Applies to	All operating systems and servers.
Description	The amount of memory available for use as a database server cache is one of the key factors controlling performance. You can set the initial amount of cache memory using the $-c$ server option
	The more cache memory that can be given the server, the better its performance.
	The units G , K , and M can be either lower case or upper case. If G , K , or M is not supplied, any integer less than 10 000 is assumed to be in kilobytes, and any integer 10 000 or greater is assumed to be in bytes. For example, $-c$ 4096 means 4096 kb or 4 194 304 bytes, whereas $-c$ 200 000 means (an unreasonably small) cache of 200 000 bytes.
	The unit P is a percentage of the physical system memory, and if you use this, the argument is a percentage. You can use % as an alternative to P, but as most non-UNIX operating systems use % as an environment variable escape character, you must escape the % character. To use 50 percent of the physical system memory, you would use the following:
	dbeng9 -c 50%%
	If no $-c$ option is provided, the database server computes the initial cache allocation as follows:
	 It uses the following operating-system-specific default cache sizes: Windows CE 600K
	• Windows NT/2000/XP, Windows 95/98/Me, NetWare 2 Mb
	◆ UNIX 8 Mb
	 2. It computes a runtime-specific minimum default cache size, which is the lesser of the following items: 25% of the machine's physical memory The sum of the sizes of the main database files specified on the command line. Additional dbspaces apart from the main database files are not included in the calculation. If no files are specified, this value is zero.

3. It allocates the greater of the two values computed.

	NetWare database server There is a tradeoff between memory for the database server and memory for the NetWare file system buffers. A larger database server cache will improve database server performance at the expense of NetWare file system performance. If the database server cache is too big, NetWare will report an error that there is insufficient memory for cache buffers. NetWare memory requirements increase with every new directory and file on the file server. To track memory usage on the NetWare server, load monitor.nlm (if it is not already loaded) and select "Resource Utilization". Extra memory for your NetWare server computer could improve database
Example	The following example, entered all on one line, starts a server named myserver that starts with a cache size of 3 Mb and loads the sample database:
	dbeng9 -c 3M -n myserver "C:\Program Files\Sybase\SQL Anywhere 9\asademo.db"
See also	"-ch server option" on page 136
-ca server option	
Function	Enforces a static cache size. The zero argument is required.
Syntax	{ dbsrv9 dbeng9 } -ca 0
Applies to	Windows NT/2000/XP, Windows 95/98/Me, UNIX
Description	You can disable automatic cache increase due to high server load by specifying -ca 0 on the command line. If you do not set -ca 0, the database server automatically takes additional cache as needed. The cache size still increases if the database server would otherwise run into the error Fatal Error: dynamic memory exhausted, or if the Java VM requires memory that would lead to a fatal error.
	This server option should be used only in the form $-ca$ 0. If $-ca$ is omitted, automatic cache growth is enabled.
Example	The following example starts a server named myserver that has a static cache that is 40% of the available physical memory and loads the sample database:
	dbsrv9 -c 40P -ca 0 -n myserver "C:\Program Files\Sybase\SQL Anywhere 9\asademo.db"
See also	"-c server option" on page 133

"-ch server option" on page 136

-ch server option

Function	Sets a maximum cache size, as a limit to automatic cache growth.
Syntax	{ dbsrv9 dbeng9 } -ch { integer integerG integerK integerP }
Applies to	Windows NT/2000/XP, Windows 95/98/Me, and UNIX
Description	This option limits the cache that the database server can take during automatic cache growth. By default the upper limit is approximately the lower of 256 Mb and 90% of the physical memory of the machine.
	For the meaning and usage of the cache size arguments and the G , K , M , and P characters, see "-c server option" on page 133.
Example	The following example starts a server named silver that has a maximum cache size of 2 Mb and loads the sample database:
	dbeng9 -ch 2M -n silver "C:\Program Files\Sybase\SQL Anywhere 9\ asademo.db"
See also	"-c server option" on page 133
	"-ca server option" on page 135
	"-cl server option" on page 136
-cl server option	
Function	Sets a minimum cache size as a lower limit to automatic cache resizing.
Syntax	{ dbsrv9 dbeng9 } -cl {integer integerG integerK integerM integerP }
Applies to	Windows NT/2000/XP, Windows 95/98/Me, UNIX
Description	This option sets a lower limit to the cache. The default minimum cache size is the initial cache size.
	For the meaning and usage of the cache size arguments and the G , K , M , and P characters, see "-c server option" on page 133.

Example	The following example starts a server named silver that has a minimum cache size of 5 Mb and loads the sample database:
	dbeng9 -cl 5M -n silver "C:\Program Files\Sybase\SQL Anywhere 9\ asademo.db"
See also	"-c server option" on page 133
	"-ca server option" on page 135
	"-ch server option" on page 136

-cs server option

Function	Display cache size changes in the database server window.
Syntax	{ dbsrv9 dbeng9 } -cs
Applies to	All platforms with dynamic cache sizing.
Description	For troubleshooting purposes, display cache information in the database server window whenever the cache size changes.

-ct server option

Function	To turn character set translation on and off.
Syntax	{ dbsrv9 dbeng9 } -ct { + - }
Applies to	All operating systems except Windows CE.
Description	By default, character set translation is turned on. Character set translation converts strings between character sets that represent the same characters, but at different values. This is useful when the client machine and the database use different character sets. Character set translation is disabled with the $-ct$ - server option. If this argument is supplied as $-ct$ +, character set translation is turned on.
	In version 7.x and earlier of Adaptive Server Anywhere, the $+$ or $-$ values are not accepted: specifying the $-ct$ option enabled character set translation.
Example	The following example starts a server with character set translation disabled and loads the sample database:
	dbeng9 -ct- "C:\Program Files\Sybase\SQL Anywhere 9\asademo.db"
See also	"Turning off character set translation on a database server" on page 329
	Starting a database server using character set translation on page 329

-cw server option

Function	Enables use of Address Windowing Extensions (AWE) on Windows 2000, Windows XP, and Windows Server 2003 for setting the size of the database server cache.
Syntax	{ dbsrv9 dbeng9 } -cw
Applies to	Windows 2000, Windows XP, Windows Server 2003, and higher.
Description	The amount of memory available for use as a database server cache is one of the key factors controlling performance. Because Windows 2000, Windows XP, and Windows Server 2003 support Address Windowing Extensions, you can use the -cw option to take advantage of large cache sizes based on the maximum amount of physical memory in the system.

Operating system	Maximum non- AWE cache size	Maximum amount of phys- ical memory supported by Windows
Windows 2000 Profes- sional	1.8 Gb	4 Gb
Windows 2000 Server	1.8 Gb*	4 Gb
Windows 2000 Ad- vanced Server	2.7 Gb*	8 Gb
Windows 2000 Datacen- ter Server	2.7 Gb*	64 Gb
Windows XP Home Edition	1.8 Gb	2 Gb
Windows XP Profes- sional	1.8 Gb	4 Gb
Windows Server 2003, Web Edition	1.8 Gb	2 Gb
Windows Server 2003, Standard Edition	1.8 Gb	4 Gb
Windows Server 2003, Enterprise Edition	2.7 Gb*	32 Gb
Windows Server 2003, Datacenter Edition	2.7 Gb*	64 Gb

*You must boot the operating system using the / 3GB option to use a cache of this size.

When using an AWE cache, almost all of the available physical memory in the system can be allocated for the cache.

If you can set a cache of the desired size using a non-AWE cache, this is recommended because AWE caches allocate memory that can only be used by the database server. This means that while the database server is running, the operating system and other applications cannot use the memory allocated for the database server cache. AWE caches do not support dynamic cache sizing. Therefore, if an AWE cache is used and you specify the -ch or -cl options to set the upper and lower cache size, they are ignored.

For information about specifying the cache size, see "-c server option" on page 133.

To start a database server with an AWE cache, you must do the following

- Have at least 130 Mb of memory available on your system.
- If your system has between 2 Gb and 16 Gb of memory, add the /3GB option to the Windows boot line in the "[operating systems]" section of the *boot.ini* file.

If your system has more than 16 Gb of memory, do not add the /3GB option to the Windows boot line in the "[operating systems]" section of the *boot.ini* file because Windows will not be able to address memory beyond 16 Gb.

- If your system has more than 4 Gb of memory, add the /PAE option to the Windows boot line in the "[operating systems]" section of the *boot.ini* file.
- Grant the "Lock pages in memory" privilege to the user ID under which the server is run. The following steps explain how to do this on Windows 2000.
 - 1. Log on to Windows as Administrator.
 - 2. From the Start menu, choose Settings \succ Control Panel.
 - 3. Open the Administrative Tools folder.
 - 4. Double-click Local Security Policy.
 - 5. Open Local Policies in the left pane.
 - 6. Double-click User Rights Assignment in the left pane.
 - Double-click the Lock Pages In Memory policy in the right pane. The Local Security Policy Setting dialog appears.

8.	In the Local Security Policy Setting dialog, click Add
	The Select Users or Groups dialog appears.

- 9. Select the user ID from the list and click Add.
- 10. In the Local Security Policy Setting dialog, click OK.
- 11. Restart the computer for the setting to take effect.

If you specify the -cw option and the -c option on the command line, the database server attempts the initial cache allocation as follows:

- 1. The AWE cache is no larger than the cache size specified by the -c option. If the value specified by the -c option is less than 2 Mb, AWE is not used.
- 2. The AWE cache is no larger than all available physical memory less 128 Mb.
- 3. The AWE cache is no smaller than 2 Mb. If this minimum amount of physical memory is not available, AWE is not used.

When you specify the -cw option and do not specify the -c option, the database server attempts the initial cache allocation as follows:

1.	The AWE cache uses 100% of all available memory except for 128 M	1b
	hat is left free for the operating system.	

- 2. The AWE cache is no larger than the sum of the sizes of the main database files specified on the command line. Additional dbspaces apart from the main database files are not included in the calculation. If no files are specified, this value is zero.
- 3. The AWE cache is no smaller than 2 Mb. If this minimum amount of physical memory is not available, an AWE cache is not used.

When the server uses an AWE cache, the cache page size is at least 4 Kb and dynamic cache sizing is disabled. On 64-bit Windows platforms, the cache page size is at least 8 Kb.

For more information about dynamic cache sizing, see "Using the cache to improve performance" [ASA SQL User's Guide, page 176].

Example The following example starts a server named **myserver** that starts with a cache size of 12 Gb and loads the sample database:

dbeng9 -c 12G -cw asademo.db

See also "-c server option" on page 133

-d server option

Function	Use POSIX I/O.
Syntax	{ dbsrv9 dbeng9 } -d
Applies to	NetWare. The use of this option on Windows platforms has been deprecated.
Description	The -d option forces the use of POSIX I/O rather than DFS (Direct File System) I/O. This switch is provided as a workaround for bugs in old versions of DFS. Note that asynchronous I/O is not available when using POSIX I/O.
	This option applies to NetWare systems only.
-ec server option	
Function	Encrypt all native Adaptive Server Anywhere packets (DBLib, ODBC, and OLEDB) transmitted to and from all clients. TDS packets are not encrypted.
Syntax	{ dbsrv9 dbeng9 } -ec encryption-options
	encryption-options: { NONE
	SIMPLE
	ECC_TLS (CERTIFICATE=filename; CERTIFICATE_PASSWORD=password)
	RSA_TLS (CERTIFICATE=filename; CERTIFICATE_PASSWORD=password)
	ALL } ,
Description	The -ec option instructs the database server to accept <i>only</i> connections from ODBC, OLE DB, or embedded SQL interfaces that are encrypted using one of the specified types. Connections over the TDS protocol, which include Java applications using jConnect, are always accepted, regardless of encryption.
	By default, communication packets are not encrypted, which poses a potential security risk. If you are concerned about the security of network packets, use the -ec option. Encryption affects performance only marginally. The -ec option controls the server's encryption settings and requires one or more of the following parameters in a comma-separated list:
	none accepts only connections that are not encrypted.
	simple accepts connections that are encrypted with simple encryption. This type of encryption is supported on all platforms, as well as on previous

versions of Adaptive Server Anywhere. Simple encryption is not as strong as Certicom encryption.

ECC_TLS Formerly called Certicom encryption, this parameter accepts connections that are encrypted using the elliptic curve-based Certicom encryption technology. To use this type of encryption, both the server and the client must be operating on Solaris, Linux, NetWare, or any supported Windows platform except Windows CE, and the connection must be over the TCP/IP port. UNIX platforms, except for Solaris and Linux, do not recognize the client or server **ECC_TLS** parameter. Specifying **CERTICOM** is accepted to mean **ECC_TLS**. This parameter accepts the following arguments:

- **certificate** the file name of the certificate. The default value is *sample.crt*.
- ◆ certificate_password the password for the certificate named above. The password for *sample.crt* is tJ1#m6+W.

RSA_TLS this parameter accepts connections that are encrypted using RSA-based encryption technology. To use this type of encryption, both the server and the client must be operating on Solaris, Linux, NetWare, or any supported Windows platform except Windows CE, and the connection must be over the TCP/IP port. UNIX platforms, except for Solaris and Linux, do not recognize the client or server **RSA_TLS** parameter. This parameter accepts the following arguments:

- **certificate** the file name of the certificate. The default value is *rsaserver.crt.*
- **certificate_password** the password for the certificate named above. The password for *rsaserver.crt* is **test**.

Caution

The sample certificate should be used for testing purposes only. The sample certificate provides no security in deployed situations because it and the corresponding password are widely distributed with Sybase software. To protect your system, you must create your own certificate.

You can use the **gencert** utility provided with SQL Anywhere Studio to generate new certificates in Adaptive Server Anywhere.

For information about the **gencert** utility and creating certificates, see "Self-signed certificates" [*MobiLink Synchronization User's Guide*, page 347].

all accepts connections that are not encrypted (none), or encrypted with simple encryption (simple). This is the default.

	The <i>dbtls9.dll</i> and <i>dbrsa9.dll</i> files contain the Certicom code used for encryption and decryption. When you connect to the server, if the appropriate file cannot be found, or if an error occurs, a message appears on the server console in debug mode. The server does not start if the types of encryption specified cannot be initiated.
	The client's and the server's encryption settings must match or the connection will fail. The server automatically encrypts client transmissions that request encryption if the none parameter is not used with the -ec option.
Example 1	dbsrv9 -ec simple,certicom(certificate=sample.crt; certificate_ password=tJ1#m6+W) -x tcpip asademo.db
Example 2	dbsrv9 -ec ecc_tls(certificate=sample.crt; certificate_ password=tJ1#m6+W) -x tcpip asademo.db
Example 3	dbsrv9 -ec rsa_tls(certificate=rsaserver.crt; certificate_ password=test) -x tcpip asademo.db
See also	"Encryption connection parameter [ENC]" on page 188
	"Encrypting client/server communications" on page ??
-ep server option	
Function	To prompt the user for the encryption key upon starting a strongly encrypted database.
Syntax	{ dbsrv9 dbeng9 } -ep
Description	The -ep option instructs the database server to display a dialog box for the user to enter the encryption key. This server option provides an extra measure of security by never allowing the encryption key to be seen in clear text.
	When used with supported tools, this option always prompts the user for the encryption key, even if a key is not necessary. If the dialog box prompt appears and you know that a key is not necessary, you can click Cancel to continue.
	When used with the engine, this option prompts the user for the encryption key only if <i>all</i> of the following are true:
	 the -ep option is specified
	 the engine is either a Windows personal server, or the engine is just starting up
	• a key is required to start a database

	 the engine is either not a Windows service, or it is a Windows service with the interact with desktop option turned ON
	• the engine is not a daemon (UNIX)
Example	dbsrv9 -ep -x tcpip myencrypted.db
See also	"Encryption connection parameter [ENC]" on page 188
	"Encrypting client/server communications" on page ??
	"-ek database option" on page 168
-ga server option	
Function	Unload database after last connection is dropped.
Syntax	{ dbsrv9 dbeng9 } -ga
Applies to	All operating systems except NetWare.
Description	Specifying this option on the network server causes each database to be unloaded after the last connection to it is dropped. In addition to unloading each database after the last connection is dropped, the personal server shuts down when the last database is stopped.
-gb server option	
Function	Set the database process priority class.
Syntax	{ dbsrv9 dbeng9 } -gb { idle normal high maximum }
Applies to	Windows NT/2000/XP
Description	Set the database process priority class. The value idle is provided for completeness, and maximum may interfere with the running of your computer. Normal and high are the commonly used settings.
-gc server option	
Function	Set maximum desired interval between checkpoints.
Syntax	{ dbsrv9 dbeng9 } -gc integer
Applies to	All operating systems and servers.
Description	Set the <i>maximum</i> desired length of time in minutes that the database server runs without doing a checkpoint on each database.
	The default value is 60 minutes.

	When a database server is running with multiple databases, the checkpoint time specified by the first database started is used unless overridden by this option. If a value of 0 is entered, the default value of 60 minutes is used.
	Checkpoints generally occur more frequently than the specified time.
	For more information, see "How the database server decides when to checkpoint" on page 363.
See also	"CHECKPOINT_TIME option [database]" on page 582
	"Checkpoints and the checkpoint log" on page 360
-gd server option	
Function	Set permissions required to start or stop a database.
Syntax	{ dbsrv9 dbeng9 } -gd { DBA all none }
Applies to	All operating systems and servers.
Description	This is the permission required for a user to cause a new database file to be loaded by the server, or to stop a database on a running database server. The level can be one of the following:
	• DBA Only users with DBA authority can start or stop databases.
	• all All users can start or stop databases.
	• none Starting and stopping databases is not allowed apart from when the database server itself is started and stopped.
	The default setting is ALL for the personal database server and DBA for the network database server. Both uppercase and lowercase syntax is acceptable.
	Note that when this option is set to DBA, the client application must already have a connection to the server in order to start or stop a database. Providing a DBA username and password on a new connection is not sufficient.
Example	The following set of steps illustrates how to use the -gd option for the network database server.
	1. Enter a password in the <i>util_db.ini</i> file in the directory that holds the database server executable:
	[UTILITY_DB] pwd=mypwd
	2. Start the network database server:

```
dbsrv9 -x tcpip -n myserver -gd DBA
```

3. Connect to the utility database from Interactive SQL.

The following command assumes that **myserver** is the default database server:

dbisql -c "uid=DBA;pwd=mypwd;dbn=utility_db "

4. Start a database:

start database asademo
on myserver;

5. Connect to the database you have started:

connect to myserver database asademo user DBA identified by SQL

-ge server option

Function	Set stack size for external functions.
Syntax	{ dbsrv9 dbeng9 } -ge integer
Applies to	Windows 95/98/Me, Windows NT/2000/XP, NetWare
Description	Sets the stack size for threads running external functions, in bytes. The default is 32K.
-gf server option	
Function	Disable firing of triggers by the server.
Syntax	{ dbsrv9 dbeng9 } -gf
Applies to	All operating systems and servers.
Description	The -gf server option instructs the server to disable the firing of triggers.
See also	"FIRE_TRIGGERS option [compatibility]" on page 592
-gk server option	

Function	Set the permission required to stop the network server and personal server using <i>dbstop</i> .
Syntax	{ dbsrv9 dbeng9 } -gk { DBA all none }
Applies to	All operating systems and servers.
Description	The allowed values include:

	• DBA Only users with DBA authority can use <i>dbstop</i> to stop the server. This is the default for the network server.
	• all All users can use <i>dbstop</i> to stop the server. This is the default for the personal server.
	• none The server cannot be stopped using <i>dbstop</i> .
	Both uppercase and lower case syntax is acceptable.
-gl server option	
Function	Set the permission required to load data using LOAD TABLE, and to unload data using UNLOAD or UNLOAD TABLE.
Syntax	{ dbsrv9 dbeng9 } -gl { DBA all none }
Applies to	All operating systems and servers.
Description	Using the UNLOAD TABLE or UNLOAD statements places data in files on the database server machine, and the LOAD TABLE statement reads files from the database server machine.
	To control access to the file system using these statements, the -gl server option allows you to control the level of database permission that is required to use these statements.
	The allowed values are as follows:
	• DBA Only users with DBA authority can load or unload data from the database.
	• all All users can load or unload data from the database.
	• none Data cannot be unloaded or loaded.
	Both uppercase and lower case syntax is acceptable.
	The default settings are all for personal database servers on non-UNIX operating systems, and DBA for the network database server and the UNIX personal server. These settings reflect the fact that, on non-UNIX platforms, the personal database server is running on the current machine, and so the user already has access to the file system.
-gm server option	
Function	Limit the number of concurrent connections to the server.
Syntax	{ dbsrv9 dbeng9 } -gm integer

Applies to	All operating systems and servers.
Description	Defines the connection limit for the server. If this number is greater than the number that is allowed under licensing and memory constraints, it has no effect.
	The database server allows one extra DBA connection above the connection limit to allow a DBA to connect to the server and drop other connections in an emergency.
-gn server option	
Function	Set the maximum number of active requests (both user and system) that the database server can handle concurrently.
Syntax	{ dbsrv9 dbeng9 } -gn integer
Applies to	All operating systems and servers.
Description	Set the number of active user and system requests that the database server can handle concurrently. If the maximum number of active requests are already in use when the database server receives an additional request, the new request must wait until another request is completed.
	Each connection uses a thread for each request, and when the request is completed the thread is returned to the pool for use by other connections. As no connection can have more than one request in progress at one time, no connection uses more than one thread at a time. An exception to this rule is if a Java application uses threads. Each thread in the Java application is a database server execution thread.
	The default is 20 threads for the network database server, and 10 threads for the personal database server. The number of threads cannot be greater than the number of server connections, so the maximum number of threads for the personal database server is also 10.
	You may want investigate the Performance Monitor readings for Requests: Active and Requests: Unscheduled . If the number of active requests is always less than -gn, you can lower -gn. If the number of total requests (active + unscheduled) is often larger than -gn, then you might want to increase -gn. The Performance Monitor is not available for UNIX or Linux platforms.
See also	"Controlling threading from the command line" on page 14

-gp server option

Function	Set the maximum allowed database page size.
Syntax	{ dbsrv9 dbeng9 } -gp { 1024 2048 4096 8192 16384 32768 }
Applies to	All operating systems and servers.
Description	Database files with a page size larger than the page size of the server cannot be loaded. This option explicitly sets the page size of the server, in bytes.
	If you do not use this option, then the page size of the first database on the command line is used.
	The minimum page size on all UNIX platforms is 2048 bytes. You can still use databases with smaller page sizes, but cache memory is used very inefficiently. If you do not use this option and start a server with no databases loaded, the default value is 2048.
	On all other platforms, if you do not use this option and start a server with no databases loaded, the default value is 1024.
-gr server option	
Function	Set the maximum length of time (in minutes) for recovery from system failure.
Syntax	{ dbsrv9 dbeng9 } -gr integer
Applies to	All operating systems and servers.
Description	When a database server is running with multiple databases, the recovery time that is specified by the first database started is used unless overridden by this option.
	For more information, see "RECOVERY_TIME option [database]" on page 621.
-gss server option	
Function	Set the stack size per internal execution thread in the server.
Syntax	{ dbsrv9 dbeng9 } -gss { integer integerK integerM }
Applies to	This option has no effect on Windows operating systems.
Description	The number of internal execution threads is controlled by the -gn option, and has a default value of 20. The default stack size per internal execution

	thread is 64 kb for NetWare and UNIX, and the maximum stack size is 1 M. The -gss option allows you to lower the memory usage of the database server in environments with limited memory.
See also	"Controlling threading from the command line" on page 14
-gt server option	
Function	Sets the maximum number of requests that can run simultaneously by the database server. This option is only useful on multiprocessor systems.
Syntax	{ dbsrv9 dbeng9 } -gt integer
Applies to	All operating systems and servers except NetWare.
Description	By default, on Windows operating systems, the network database server uses all CPUs available on the machine, the personal database server is limited to two processors, and the runtime database server uses a single processor. On UNIX, all database servers use all available processors by default.
See also	"Controlling threading from the command line" on page 14
-gu server option	
Function	Set permission levels for utility commands.
Syntax	{ dbsrv9 dbeng9 } -gu { all none DBA utility_db }
Applies to	All operating systems and servers.
Description	Sets permission levels for utility commands such as CREATE DATABASE and DROP DATABASE. The level can be set to one of the following: utility_db, all, none, DBA . The default is DBA .
	The utility_db level restricts the use of these commands to only those users who can connect to the utility database. The all , none , and DBA levels permit all users, no users, or users with DBA authority to execute utility commands.
-gx server option	
Function	Set the maximum number of requests that can concurrently execute blocking system calls.
Syntax	{ dbsrv9 dbeng9 } -gx integer
Applies to	Windows 95/98/Me, Windows NT/2000/XP

Description	This option is only of use on Windows NT. On other platforms the -gn value is equivalent to the -gt value.
	Meaningful values are in the range specified by the -gt option value and the -gn option value. By default, this option is set to a value one greater than the number of CPUs on the machine. The additional process permits the use of remote data access between databases on a single database server.
	The primary cost of increasing -gx is reduced address space (not physical memory) available for the cache. This option is only of use if you are using omni, java, or external stored procedures (and in the case of java/external stored procedures, those if features are being used to implement servers). If you are using omni/java/external stored procedures and the engine hangs for no apparent reason, increasing -gx might solve the problem. Setting the -gx option beyond the -gn option has no benefit.
	You may want to increase the option setting beyond the default to reserve capacity for external tasks, separate from standard database tasks. For example, you may want to increase the value of -gx by one for each concurrent connection using remote data access, or for connections using Java in the database to listen on an external port. In other circumstances, this option should be left at the default value.
	On UNIX, each task is executed in its own thread, so that the number of tasks (-gn) also determines the number of threads.
See also	"Controlling threading from the command line" on page 14
-m server option	
Function	Delete the transaction log when a checkpoint is done.
Syntax	{ dbsrv9 dbeng9 } -m
Applies to	All operating systems and servers.
Description	This option deletes the transaction log when a checkpoint is done, either at shutdown or as a result of a checkpoint scheduled by the server.
	Caution When this option is selected, there is no protection against media failure on the device that contains the database files.
	This provides a way to automatically limit the growth of the transaction log. Checkpoint frequency is still controlled by the CHECKPOINT_TIME and

The -m option is useful for limiting the size of the transaction log in

RECOVERY_TIME options (which you can also set on the command line).

situations where high volume transactions requiring fast response times are being processed, and the contents of the transaction log are not being relied upon for recovery or replication. The -m option provides an alternative to operating without a transaction log at all, in which case a checkpoint would be required following each COMMIT and performance would suffer as a result. When the -m option is selected, there is no protection against media failure on the device that contains the database files. Other alternatives for managing the transaction log (for example, using the BACKUP statement and events) should be considered before using the -m option.

To avoid database file fragmentation, it is recommended that where this option is used, the transaction log be placed on a separate device or partition from the database itself.

Replicated and Synchronized databases

Do not use the -m option with databases that are being replicated or synchronized. Replication and Synchronization, used by SQL Remote and MobiLink, inherently rely on transaction log information.

-n server option

Function	Set the name of the database server.
Syntax	{ dbsrv9 dbeng9 } -n database-file-name
Applies to	All operating systems and servers.
Description	By default, the database server receives the name of the database file with the path and extension removed. For example, if the server is started on the file <i>c</i> :\ <i>Program Files</i> \ <i>Sybase</i> \ <i>SQL Anywhere 9</i> \ <i>asademo.db</i> and no -n option is specified, the name of the server is asademo .
	The server name is interpreted according to the character set of the machine, as no database collation exists at startup time. Multi-byte characters are not recommended in server names.
	Names must be a valid identifier. Long engine names are truncated to

different lengths depending on the protocol.

Protocol	Truncation Length
Unix shared memory	31 bytes
non-Unix shared memory	40 bytes
TCP/IP	40 bytes
SPX	32 bytes
Named Pipes	8 bytes

i.

The server name specifies the name to be used in the **EngineName (ENG)** connection parameter of client application connection strings or profiles. With shared memory, there is a default database server that will be used if no server name is specified provided that at least one database server is running on the computer.

Running multiple servers with the same name is not recommended.

	There are two -n options The -n option is positional. If it appears before a database file name, it is a server option and names the server. If it appears after a database file name, it is a database option and names the database.
	For example, the following command names the server SERV and the database DATA:
	dbsrv9 -n SERV asademo.db -n DATA
	For more information, see "-n database option" on page 170.
See also	"Identifiers" [ASA SQL Reference, page 7]
	"EngineName connection parameter [ENG]" on page 187
-o server option	
Function	Print all server window output to a file.
Syntax	{ dbsrv9 dbeng9 } -o filename
Applies to	All operating systems and servers.
Description	Print all server message window output to a file.

-oe server option

Function	Specifies a filename	to log startup errors,	fatal errors and assertions.
----------	----------------------	------------------------	------------------------------

-	
Syntax	{ dbsrv9 dbeng9 } -oe filename
Applies to	All operating systems and servers.
Description	Each line in this file is prefixed with the date and time. Startup errors include such errors as:
	 Could not open/read database file: <database file=""></database>
	• A database server with that name has already started
	Fatal errors and assertions are logged to the Windows Application Event Log (except on Window CE) or the Unix system log regardless of whether -oe is specified.
-os server option	
Function	Limit the file size for the server window output.
Syntax	{ dbsrv9 dbeng9 } -os { integer integerG integerK integerM }
Applies to	All operating systems and servers.
Description	Limits the size of the log file used by the $-\circ$ option. The units G , K , and M can be either lower case or upper case. If G , K , or M is not supplied, any integer less than 10 000 is assumed to be in kilobytes, and any integer 10 000 or greater is assumed to be in bytes.
See also	"-o server option" on page 153
-p server option	
Function	Set the maximum size of communication packets.
Syntax	{ dbsrv9 dbeng9 } -p integer
Applies to	All operating systems and servers.
Description	The default is 1460 bytes. The minimum value is 300 bytes and the maximum is 16 000.
See also	"CommBufferSize connection parameter [CBSIZE]" on page 178
-pc server option	
Function	Specifying this option compresses all connections except for same-machine connections. This option can be overridden for a particular client by specifying COMPRESS=NO in the client's connection parameters.
Syntax	{ dbsrv9 } -pc

Applies to	All operating systems and network servers.
Description	The packets sent between an Adaptive Server Anywhere client and server can be compressed using the $-pc$ option. Compressing a connection may improve performance under some circumstances. Large data transfers with highly compressible data tend to get the best compression rates.
	The default is not to compress connections. Specifying the -pc option compresses all connections except same-machine connections and TDS connections. TDS connections (including jConnect) do not support Adaptive Server Anywhere communication compression.
	Same-machine connections over any communication link will not enable compression, even if the -pc option or COMPRESS=YES connection parameter is used.
See also	"Adjusting communication compression settings to improve performance" on page 102
	"Compress connection parameter [COMP]" on page 181
	"Try using Adaptive Server Anywhere's compression features" [ASA SQL User's Guide, page 166]
-pt server option	
Function	To increase or decrease the size limit at which packets are compressed.
Syntax	{ dbsrv9 } -pt size_in_bytes
Applies to	All operating systems and network servers.
Description	This parameter takes an integer value representing the minimum byte-size of packets to be compressed. Values less than 80 are not recommended. The default is 120 bytes.
	Under some circumstances, changing the compression threshold can help performance of a compressed connection by allowing you to only compress packets when compression will increase the speed at which the packets are transferred. The default setting should be appropriate for most cases.
	If both client and server specify different compression threshold settings, the client setting applies.
See also	"Adjusting communication compression settings to improve performance" on page 102

"Try using Adaptive Server Anywhere's compression features" [ASA SQL User's Guide, page 166]

-qi server option

Function	Do not display the database server tray icon and window.
Syntax	{ dbsrv9 dbeng9 } -qi
Applies to	Windows platforms, excluding Windows CE
Description	This option leaves no visual indication that the engine is running, other than possible startup error dialogs. You can use either (or both) the -o or -oe logs to diagnose errors.
-qp server option	
Function	Do not display messages about performance in the database server window. Messages that are suppressed include the following:
	 No unique index or primary key for table 'table_name'
	• Database file "mydatabase.db" consists of nnn fragments
Syntax	{ dbsrv9 dbeng9 } -qp
Applies to	All operating systems and servers.
Description	Do not display messages about performance in the database server window. Messages that are suppressed include the following:
	No unique index or primary key for table 'table_name'
	• Database file "mydatabase.db" consists of nnn fragments
-qs server option	
Function	Suppress startup error dialogs.
Syntax	{ dbsrv9 dbeng9 } -qs
Applies to	Windows only
Description	This option suppresses startup error dialogs. Startup errors include errors such as:
	 Could not open/read database file: <database file=""></database>
	• A database server with that name has already started

On Windows platforms, if the engine is not being autostarted, these errors appear in a dialog and must be cleared before the server stops. These dialogs do not appear if the -qs switch is used.

If there is an error loading the language DLL, no dialog appears if **-qs** was specified on the command line and not in @environment-variable or @filename syntaxes. This error is not logged to the **-o** or **-oe** logs, but rather to the Windows Application Event Log (except on Window CE).

-qw server option

Function	Do not display database server screen.
Syntax	{ dbsrv9 dbeng9 } -qw
Applies to	All operating systems and servers except NetWare.
Description	This option suppresses the database server window (Windows platforms) and display messages on the console (non-Windows platforms).
-r server option	
Function	All databases started on the server are read-only. No changes to the database(s) are allowed: the server does not modify the database file(s) and transaction log files. This option is position dependent.
Syntax	{ dbsrv9 dbeng9 } -r
Applies to	All operating systems and servers.
Description	Opens all database files as read-only with the exception of the temporary file when the option is specified before any database names on the command line. If the $-r$ server option is specified after a database name, only that specific database is read-only. You can make changes on temporary tables, but ROLLBACK has no effect, since the transaction and rollback logs are disabled.
	Databases distributed on CD-ROM devices, and compressed databases are examples of database files that cannot be modified. You can either create a write file to allow changes to the database outside the database file, or run in read-only mode.
	If you attempt to modify the database, for example with an INSERT or DELETE statement, a SQLSTATE_READ_ONLY_DATABASE error is returned.
	Databases that require recovery cannot be started in read-only mode. For example, database files created using an online backup cannot be started in

read-only mode if there were any open transactions when the backup was started, since these transactions would require recovery when the backup copy is started.

Example

To open two databases in read-only mode

dbeng9 -r database1.db database2.db

To open only the first of two databases in read-only mode.

dbeng9 database1.db -r database2.db

-s server option

Function	Set the user ID for syslog messages.
Syntax	{ dbsrv9 dbeng9 } -s { none user login-id }
Applies to	UNIX
Description	Sets the system user ID used in messages to the syslog facility. The default is user , which uses the user ID for the database server process. A value of none prevents any syslog messages from being logged.

-sb server option

Function	Specify how the server reacts to broadcasts.
Syntax	{ dbsrv9 dbeng9 } -sb { 0 1}
Applies to	SPX, TCP/IP
Description	Using -sb 0 causes the server not to start up any TCP/UDP broadcast listeners. In addition to forcing clients to use the DoBroadcast=NONE and HOST= options to connect to the server, this option causes the server to be unlisted when using dblocate.
	Using -sb1 causes the server to not respond to broadcasts from dblocate, while leaving connection logic unaffected. You can connect to the server by specifying LINKS=tcpip and ENG= <name>.</name>
-sc server option	
Function	Disable the shared memory communications protocol and use Named Pipes.
Syntax	{ dbsrv9 dbeng9 } -sc
Applies to	Windows NT/2000/XP

Description	This option disables the shared memory communications protocol that is used for same-machine communications, and starts the Named Pipes protocol.
	This option is implemented as part of an initiative to obtain C^2 security

This option is implemented as part of an initiative to obtain C2 security certification. It is likely to be of general use only for customers wanting to run in a C2-certified environment.

-ti server option

Function	Disconnect inactive connections.
Syntax	{ dbsrv9 dbeng9 } -ti minutes
Applies to	All operating systems and servers.
Description	Disconnect connections that have not submitted a request for <i>minutes</i> . The default is 240 (4 hours). A client machine in the middle of a database transaction holds locks until the transaction is ended or the connection is terminated. The -ti option is provided to disconnect inactive connections, freeing their locks.
	The -ti option is very useful when used in conjunction with <i>dbsrv9</i> , since most connections will be over network links (TCP or SPX).
	The -ti option is useful with <i>dbeng9</i> only for local TCPIP connections. Using -ti has no effect on connections to a local engine using shared memory.
	Setting the value to zero disables checking of inactive connections, so that no connections are disconnected.
See also	"sa_server_option system procedure" [ASA SQL Reference, page 739]
-tl server option	
Function	Set the period at which to send liveness packets.
Syntax	{ dbsrv9 dbeng9 } -tl seconds
Applies to	All database servers using TCP/IP or SPX.
See also	"sa_server_option system procedure" [ASA SQL Reference, page 739]
Description	A liveness packet is sent periodically across a client/server TCP/IP or SPX communications protocol to confirm that a connection is intact. If the server runs for a LivenessTimeout period (default 2 minutes) without detecting a liveness packet, on a connection, the communication is severed, and the

server drops the connection associated with that client. UNIX non-threaded clients and TDS connections do not do liveness checking.

The -tl option on the server sets the LivenessTimeout value for all clients that do not specify a liveness period.

Liveness packets are sent when a connection has not sent any packets for between one third and two thirds of the LivenessTimeout value.

When there are more than 200 connections, the server automatically calculates a higher LivenessTimeout value based on the stated LivenessTimeout value. This enables the server to handle a large number of connections more efficiently. Liveness packets are sent between one third and two thirds of the LivenessTimeout on each idle connection. Large number of liveness packets are not sent at the same time. If liveness packets take a long time to send (depending on the network, the machine's hardware, and the CPU and network load on the machine), it is possible that liveness packets will sent after two thirds of the LivenessTimeout. A warning appears in the engine console if the livenessTimeout value.

Although it is not generally recommended, you can disable liveness by specifying:

dbsrv9 -tl 0

Rather than disabling the LivenessTimeout option, consider increasing the value to 1 hour as follows:

dbsrv9 -tl 6000

-tmt server option

Function	To set a reenlistment timeout for participation in distributed transactions.
Syntax	{ dbsrv9 dbeng9 } -tmt milliseconds
Applies to	Windows NT/2000/XP only.
Description	Used during recovery of distributed transactions. The value specifies how long the database server should wait to be reenlisted. By default there is no timeout (the database server waits indefinitely).
-tmf server option	

Function	For recovery from distributed transactions in unusual circumstances.
Syntax	{ dbsrv9 dbeng9 } -tmf

Applies to	Windows NT/2000/XP only.
Description	Used during recovery of distributed transactions when the distributed transaction coordinator is not available. It could also be used if starting a database with distributed transactions in the transaction log, on a platform where the distributed transaction coordinator is not available.
	Caution
	If you use this option, distributed transactions are not recovered properly. It is not for routine use.
-tq time server option	
Function	Shut down the server at a specified time.
Syntax	{ dbsrv9 dbeng9 } -tq { datetime time }
Applies to	All operating systems and servers.
Description	This is useful for setting up automatic off-line backup procedures (see "Backup and Data Recovery" on page 337). The format for the time is in <i>hh:mm</i> (24 hour clock), and can be preceded by an optional date. If a date is specified, the date and time must be enclosed in double quotes and be in the format <i>YYYY/MM/DD HH:MM</i> .
See also	"sa_server_option system procedure" [ASA SQL Reference, page 739]
-u server option	
Function	Open files using the operating system disk cache.
Syntax	{
Applies to	Windows NT/2000/XP, Windows 95/98/Me, UNIX
Description	Files are opened using the operating system disk cache in addition to the

While the operating system disk cache may improve performance in some cases, in general better performance is obtained without this option, using

If the server is running on a dedicated machine, you should not use the -u option, as the database cache itself is generally more efficient. You may want to use the -u option if the server is running on a machine with several other applications (so that a large database cache may interfere with other applications) and yet IO-intensive tasks are run intermittently on the server

database cache.

the database cache only.

(so that a large cache will improve performance).

-ud server option

Function	Run as a daemon.
Syntax	{ dbsrv9 dbeng9 } -ud
Applies to	UNIX
Description	Using this option lets you run the server so that it continues running after the current operating system session ends.
-ut server option	
Function	Touch temporary files.
Syntax	{ dbsrv9 dbeng9 } -ut minutes
Applies to	UNIX
Description	This option causes the server to touch temporary files at specified intervals.
-v server option	
Function	Display the software version.
Syntax	{ dbsrv9 dbeng9 } -v
Applies to	All operating systems and servers.
Description	Supplies the database server version in a message box, and then stops.
-x server option	
Function	To specify server side network communications protocols.
Syntax 1	dbsrv9 -x { all none srv-protocols }
	srv-protocols: { [namedpipes spx tcpip] parmlist }, parmlist: (parm=value;)
Syntax 2	dbeng9 -x { all none eng-protocols }
	eng-protocols: { namedpipes tcpip [<i>parmlist</i>] }, parmlist: (parm=value;)

Applies to All operating systems and servers.

Description

Use the -x option to specify which communications protocols, in addition to shared memory, you want to use to listen for client connection broadcasts.

If you do not specify the -x option, the server attempts to listen for client connection broadcasts using all protocols supported by the database server running on your operating system, including shared memory.

If you specify the -x option with one or more protocols, the server attempts to listen for client connection broadcasts using the specified protocol(s) and also using a shared-memory protocol.

If you are running Windows CE and specify the -x option, the server only attempts to listen for client connection broadcasts using the TCP/IP protocol unless you explicitly request otherwise.

Regardless of which settings you choose for the -x option, the server always listens for connection broadcasts using the shared memory protocol. In addition to the shared memory protocol, you can also specify the following:

- ♦ ALL Listen for connection attempts by the client using all communications protocols that are supported by the server on this platform, including shared memory. This is the default.
- NamedPipes (NP) Listen for connection attempts by the client using the NamedPipes protocol. NamedPipes is supported on Windows NT/2000/XP as an alternative means of same-machine communication.
- **NONE** Listen for connection attempts by the client using only the shared memory protocol.
- ◆ SPX Listen for connection attempts by the client using the SPX protocol. The SPX protocol is supported by NetWare, Windows NT/2000/XP, and Windows 95/98/Me network servers.
- ◆ TCPIP (TCP) Attempt to connect to the client using the TCP/IP protocol. The TCP/IP protocol is supported by the network server on all operating systems, and by the personal database server for same-machine communications.

	By default, the database server listens for broadcasts on port 2638, and redirects them to the appropriate port. This ensures a connection in most cases.
	You can override this default and cause the server not to listen on port 2638 by setting the option -sb 0, or by turning off the BroadcastListener option (BroadcastListener=0). Additionally, if the client and server are communicating through a firewall, the client must send the packet to the exact port the server is listening on by specifying DoBroadcast=None and Host=.
	For information, see "ServerPort communication parameter [PORT]" on page 215.
	For some protocols, additional parameters may be provided, in the format
	-x tcpip(PARM1=value1;PARM2=value2;)
	For a description of available parameters, see "Network communications parameters" on page 201.
	For UNIX, quotation marks are required if more than one parameter is supplied:
	<pre>-x "tcpip(PARM1=value1;PARM2=value2;)"</pre>
Examples	Allow only shared memory, TCP/IP and SPX communications:
	-x tcpip,spx
See also	"CommLinks connection parameter [LINKS]" on page 179
-xs server option	
Function	To specify server side web services communications protocols.
Syntax 1	{ dbeng9 dbsrv9 } -xs { all none web-protocols }
	web-protocols: { [http https] parmlist }, parmlist: (parm=value;)
Applies to	All operating systems and servers.
Description	Use the $-xs$ option to specify which web protocols you want to use to listen for client connection broadcasts.
	If you do not specify the $-xss$ option, the server does not attempt to listen for web requests.
If you specify the -xs option with one or more protocols, the server attempts	

to listen for client requests using the specified protocol(s).	

	Regardless of which settings you choose for the -xs option, the server always listens for connection broadcasts using the shared memory protocol. You can specify any of the following:
	♦ ALL Listen for connection attempts by the client using all communications protocols that are supported by the server on this platform, including shared memory.
	• HTTP Listen for connection attempts by the client using the HTTP protocol. The default port on which to list is 80.
	• HTTPS Listen for connection attempts by the client using the HTTPS protocol. The default port on which to list is 443.
	• NONE Listen for connection attempts by the client using only the shared memory protocol. This is the default.
	For a description of available parameters, see "Network communications parameters" on page 201.
	For UNIX, quotation marks are required if more than one parameter is supplied:
	<pre>-xs "http(PARM1=value1;PARM2=value2;)"</pre>
Examples	Allow only shared memory, TCP/IP and SPX communications:
	dbeng9 web.db -xs http(port=80)
See also	"CommLinks connection parameter [LINKS]" on page 179
-y server option	
Function	Run as a Windows service.
Syntax	{ dbsrv9 dbeng9 } -y
Applies to	Windows 95/98/Me
Description	If the server registered as a Windows service, it continues to operate whether users log on or off, and shutdown commands are ignored.
-z server option	

Function	Display communications operations on startup for troubleshooting purposes.
Syntax	{ dbsrv9 dbeng9 } -z

Applies to	All operating systems and servers.		
Description	This should only be used when tracking problems. The information appears in the database server window.		
-zl server option			
Function	Turn on capturing of the most recently-prepared SQL statement for each connection to databases on the server.		
Syntax	{ dbsrv9 dbeng9 } -zl		
Applies to	All operating systems and servers.		
Description	This feature can also be turned on using the remember_last_statement server setting. You can obtain the most recently-prepared SQL statement for a connection using the LastStatement property function. The sa_conn_activity stored procedure allows you to obtain the most recently-prepared SQL statement for all current connections to databases on the server.		
	For stored procedure calls, only the outermost procedure call appears, not the statements within the procedure.		
See also	"Connection-level properties" on page 647		
	"sa_conn_activity system procedure" [ASA SQL Reference, page 708]		
	"sa_server_option system procedure" [ASA SQL Reference, page 739]		
-zo server option			
Function	Redirect request-level logging information to a file separate from the regular log file.		
Syntax	{ dbsrv9 dbeng9 } -z		
Applies to	All operating systems and servers.		
Description	Request-level logging is turned on using the $-zr$ option. You can direct the output from this file to a separate file from that specified on a -0 option.		
	This option also prevents request-level logging from appearing in the console.		
See also	"-zr server option" on page 167		
	"-zs server option" on page 167		

-zr server option

Function	Enable request-level logging of operations.	
Syntax	{ dbsrv9 dbeng9 } -zr { all SQL none SQL+hostvars }	
Applies to	All operating systems and servers.	
Description	This should only be used when tracking problems. The information appears in the database server window or is sent to the logging file.	
See also	"sa_server_option system procedure" [ASA SQL Reference, page 739]	
-zs server option		
Function	Limit the size of the request-level logging file.	
Syntax	{ dbsrv9 dbeng9 } -zs size	
Applies to	All operating systems and servers.	
Description	Request-level logging is turned on using the -zr option, and redirected to a separate file using the -zo option. You can limit the size of the file using the -zs option.	
	By default there is no limit. The value is in kilobytes.	
See also	"-zo server option" on page 166	
	"-zr server option" on page 167	

Recovery options

These options are for use in recovery situations only.

-a recovery option

Function	Apply the named transaction log.	
Syntax	{ dbsrv9 dbeng9 } -a log-filename	
Applies to	All operating systems and servers.	
Description	This is used to recover from media failure on the database file. When this option is specified, the database server applies the log and then terminates—it will not continue to run.	
	Specifying a cache size when starting the server can reduce recovery time.	

For information on recovery, see "Backup and Data Recovery" on page 337.
 Example
 The following example, entered all on one line, applies the log file asademo.log to the sample database.
 dbeng9 "C:\Program Files\Sybase\SOL Anywhere 9\asademo.db" -

dbeng9 "C:\Program Files\Sybase\SQL Anywhere 9\asademo.db" -a
 "C:\backup\asademo.log"

-f recovery option

Function	Force the database server to start after the transaction log has been lost.	
Syntax	{	
Applies to	All operating systems and servers.	
Description	If there is no transaction log, the database server carries out a checkpoint recovery of the database and then terminates—it does not continue to run. You can then restart the database server without the $-f$ option for normal operation.	
	If there is a transaction log in the same directory as the database, the database server carries out a checkpoint recovery, and a recovery using the transaction log, and then terminates—it does not continue to run. You can then restart the database server without the $-f$ option for normal operation.	
	During recovery, you should use the same server that you use in production. For example, if you use the network database server (<i>dbsrv9.exe</i>) during production, then you should use the network server for recovery because you may encounter problems as a result of the personal server's 10 connection limit. Specifying a cache size when starting the server can reduce recovery time.	
	For more information see "Backup and Data Recovery" on page 337.	
Example	dbeng9 "c:\Program Files\Sybase\SQL Anywhere 9\asademo.db" -f	
Database options		
	These options are entered after the database name, and apply only to that database.	

-ek database option

Function	To specify the key for a strongly encrypted database	
Syntax	{ dbsrv9 dbeng9 } [server-options] database-file -ek key	

Description	The -ek database option must be provided after a database filename on the command line. You must provide the KEY value with the -ek option to start an encrypted database. The KEY is a string, including mixed cases, numbers, letters, and special characters.
Example	dbsrv9 -x tcpip asademo.db -ek "Akmm9u70y"
See also	"Encryption connection parameter [ENC]" on page 188
	"Encryption Key connection parameter [DBKEY]" on page 190
	"Encrypting client/server communications" on page ??
	"-ep server option" on page 143
-m database option	
Function	Truncate the transaction log when a checkpoint is done.
Syntax	{ dbsrv9 dbeng9 } [server-options] database-file -m
Applies to	All operating systems and servers.
Description	Truncate (delete) the transaction log when a checkpoint is done, either at shutdown or as a result of a checkpoint scheduled by the server. This provides a way to automatically limit the growth of the transaction log. Checkpoint frequency is still controlled by the CHECKPOINT_TIME and RECOVERY_TIME options (which you can also define on the command line).
	The $-m$ option is useful where high volume transactions requiring fast response times are being processed, and the contents of the transaction log are not being relied upon for recovery or replication. When this option is selected, there is no protection against media failure on the device that contains the database files.
	To avoid database file fragmentation, it is recommended that where this option is used, the transaction log be placed on a separate device or partition from the database itself.
	This option is the same as the -m server option, but applies only to the current database or the database identified by the <i>database-file</i> variable.
	Replicated databases Do not use the -m option with databases that are being replicated. Replica- tion inherently relies on transaction log information.
Example	The following example starts a database server named silver and loads the

sample database. When a checkpoint is done, the transaction log is truncated.

```
dbsrv9 -n silver "c:\Program Files\Sybase\SQL Anywhere 9\
asademo.db" -m
```

-n database option

Function	Set the name of the database.
Syntax	{ dbsrv9 dbeng9 } [server-options] database-file -n string
Applies to	All operating systems and servers.
Description	Both database servers and databases can be named. Since a database server can load several databases, the database name is used to distinguish the different databases.
	By default, the database receives the name of the database file with the path and extension removed. For example, if the database is started on $c: asa asademo.db$ and no $-n$ option is specified, the name of the database is asademo .
Example	The following example starts the database server with a cache size of 3Mb, loads the sample database, and names the sample database "test". dbsrv9 -c 3Mb "c:\Program Files\Sybase\SQL Anywhere 9\ asademo.db" -n "test"
	 There are two -n options The -n option is positional. If it appears before a database file name, it is a server option and names the server. If it appears after a database file name, it is a database option and names the database. For example, the following command names the server SERV and the database DATA: dbsrv9 -n SERV asademo.db -n DATA For more information, see "-n server option" on page 152.

-r database option

Function

Starts the named database as read-only. No changes to the database(s) are allowed: the server does not modify the database file(s) and transaction log files. This option is position dependent.

Syntax	{ dbsrv9 dbeng9 } -r		
Applies to	All operating systems and servers.		
Description	Opens all database files as read-only with the exception of the temporary file when the option is specified before any database names on the command line. If the $-r$ server option is specified after a database name, only that specific database is read-only. You can make changes on temporary tables, but ROLLBACK has no effect, since the transaction and rollback logs are disabled.		
	Databases distributed on CD-ROM devices, and compressed databases are examples of database files that cannot be modified. You can either create a write file to allow changes to the database outside the database file, or run in read-only mode.		
	If you attempt to modify the database, for example with an INSERT or DELETE statement, a SQLSTATE_READ_ONLY_DATABASE error is returned.		
	Databases that require recovery cannot be started in read-only mode. For example, database files created using an online backup cannot be started in read-only mode if there were any open transactions when the backup was started, since these transactions would require recovery when the backup copy is started.		
Example	To open two databases in read-only mode		
	dbeng9 -r databasel.db database2.db		
	To open only the first of two databases in read-only mode.		
	dbeng9 database1.db -r database2.db		

CHAPTER 6

Connection and Communication Parameters

About this chapter	This chapter provides a reference for the parameters that establish and describe connections from client applications to a database.	
Contents	Торіс:	page
	Connection parameters	174
	Network communications parameters	201

Connection parameters

This section describes each connection parameter. Connection parameters are included in connection strings. They can be entered in the following places:

• In an application's connection string.

For more information, see "Assembling a list of connection parameters" on page 77.

• In an ODBC data source.

For more information, see "Working with ODBC data sources" on page 53.

• In the Adaptive Server Anywhere Connect dialog.

For more information, see "Connecting from Adaptive Server Anywhere utilities" on page 51.

The ODBC configuration dialog and the Adaptive Server Anywhere Connect dialog for Windows operating systems share a common format. Some of the parameters correspond to checkboxes and fields in these dialogs, while others can be entered in the text box at the end on the Advanced tab.

- Connection parameters are case insensitive.
- The Usage for each connection parameter describes the circumstances under which the parameter is to be used. Common usage entries include the following:
 - **Embedded databases** When Adaptive Server Anywhere is used as an embedded database, the connection starts a personal server and loads the database. When the application disconnects from the database, the database is unloaded and the server stops.
 - **Running local databases** This refers to the case where an Adaptive Server Anywhere personal server is already running, and the database is already loaded on the server.
 - **Network servers** When Adaptive Server Anywhere is used as a network server, the client application must locate a server already running somewhere on the network and connect to a database.
- You can use the dbping utility to test connection strings. For example, if a
 personal server with the name asademo is running the database asademo
 (which can be started with the command "dbeng9 asademo.db")

The following string returns the message Ping database successful:

Notes

dbping -d -c "eng=asademo;dbn=asademo;uid=db;pwd=sql"

The following command, however, returns the message Ping database failed - Database server not running:

dbping -d -c "eng=other_engine;uid=dba;pwd=sql"

For more information, see "The Ping utility" on page 514.

AppInfo connection parameter [APP]

Function	To assist administrators in identifying the origin of particular client connections from a database server.
Usage	Anywhere
Values	String
Default	Empty string
Description	This connection parameter is sent to the database server from Embedded SQL, ODBC, or OLE DB clients. It is not available from Open Client or jConnect applications such as Interactive SQL or Sybase Central.
	It consists of a generated string that holds information about the client process, such as the IP address of the client machine, the operating system it is running on, and so on. The string is associated in the database server with the connection, and you can retrieve it using the following statement:
	<pre>select connection_property('appinfo')</pre>
	Clients can also specify their own string, which is appended to the generated string. The AppInfo property string is a sequence of semi-colon-delimited key = <i>value</i> pairs. The valid keys are as follows:
	◆ API Either DBLIB or ODBC.
	• APPINFO If you specified AppInfo in the connection string, the string entered.
	• EXE The name of the client executable (Windows and NetWare only)
	• HOST The host name of the client machine
	• IP The IP address of the client machine (UNIX and NetWare only)
	• OS The operating system name and version number (for example, Windows NT 4.0, NetWare 5.1)
	◆ PID The process ID of the client (Windows and UNIX only)

	• THREAD The thread ID of the client (Windows and UNIX only)
	• TIMEZONEADJUSTMENT the number of minutes that must be added to the Coordinated Universal Time (UTC) to display time local to the connection.
	• VERSION The version of the connection protocol in use, including major and minor values, and a build number (for example 9.0.00.3642)
	If you specify a debug log file in your client connection parameters, the APPINFO string is added to the file.
Examples	• Connect to the default database from the C++ version of Interactive SQL:
	dbisqlc -c uid=DBA;pwd=SQL
	View the application information:
	<pre>select connection_property('appinfo')</pre>
	The result is as follows (in a single string):
	HOST=machine-name; OS=Windows NT 4.0; PID=0x11b; THREAD=0x102; VERSION=9.0.00.3642
	• Connect to the default database from the C++ version of Interactive SQL, appending your own information to the AppInfo property:
	dbisqlc -c "uid=DBA;pwd=SQL;app=ISQL connection"
	View the application information:
	<pre>select connection_property('appinfo')</pre>
	The result is as follows (in a single string):

HOST=machine-name; OS=Windows NT 4.0; PID=0x10e; THREAD=0xe1; VERSION=9.0.00.3642; APPINFO=ISQL connection

AutoStart connection parameter [ASTART]

Function	To prevent a local database server from being started if no connection is found.
Usage	Anywhere

Values	YES, NO
Default	YES
Description	By default, if no server is found during a connection attempt, and a database file is specified, then a database server is started on the same machine. You can turn this behavior off by setting the AutoStart (ASTART) connection parameter to NO or OFF in the connection string.

AutoStop connection parameter [ASTOP]

Function	To prevent a database from being stopped as soon as there are no more open connections.
Usage	Embedded databases
Values	YES, NO
Default	YES
Description	By default, any server that is started from a connection string is stopped when there are no more connections to it. Also, any database that is loaded from a connection string is unloaded as soon as there are no more connections to it. This behavior is equivalent to AutoStop=YES.
	If you supply AutoStop=NO, any database that you start in that connection remains running when there are no more connections to it. As a consequence, the database server remains operational as well.
	The AutoStop (ASTOP) connection parameter is used only if you are connecting to a database that is not currently running. It is ignored if the database is already started.

CharSet connection parameter [CS]

Function	To specify the character set to be used on this connection.
Usage	Anywhere
Values	String
Default	The local character set.
	For information on how this is determined, see "Determining locale information" on page 326.
Description	If you supply a value for CharSet, the specified character set is used for the current connection. Setting CharSet=none disables character set conversion for the connection.

For a list of valid character set values, see "Character set labels" on page 304.

CommBufferSize connection parameter [CBSIZE]

Function	To set the maximum size of communication packets, in bytes.
Usage	Anywhere
Values	Integer
Default	If no CommBufferSize value is set, the CommBufferSize is controlled by the setting on the server, which defaults to 1460 bytes.
Description	The CommBufferSize (CBSIZE) connection parameter specifies the size of communications packets, in bytes. The minimum value of CommBufferSize (CBSIZE) is 300, and the maximum is 16000.
	The protocol stack sets the maximum size of a packet on a network. If you set CommBufferSize to be larger than that permitted by your network, the largest buffers are broken up by the network software. You should set the buffer size to be somewhat smaller than that allowed by your network because the network software may add information to each buffer before sending it over the network.
	A larger packet size improves performance for multi-row fetches and fetches of larger rows. As each connection has its own pool of buffers, a large buffer size increases the memory usage. The application side uses four default-sized buffers per connection, and on the server side, there are two.
	If CommBufferSize is not specified on the client, the connection uses the engine's buffer size. If CommBufferSize is specified on the client, the connection uses the minimum of either the engine's buffer size or the CommBufferSize value.
	This corresponds to the SQL Anywhere Version 5 <i>dbclient</i> $-p$ option. Using the $-p$ option to set the CommBufferSize causes all clients which do not specify their own CommBufferSize, as well as all clients which specify a CommBufferSize larger than the engine's CommBufferSize, to use the size specified in the $-p$ option.
Example	• To set the buffer size to 400 bytes:
	 CommBufferSize=400
	Alternatively, you can set this parameter by entering its value in the Buffer

Alternatively, you can set this parameter by entering its value in the Buffe Size text box of the Network tab of the ODBC Configuration dialog.

CommLinks connection parameter [LINKS]

Function	To specify client side network communications protocols.
Usage	Anywhere. The CommLinks (LINKS) connection parameter is optional for connections to a personal server, and required for connections to a network server.
Values	String
Default	Use only the shared memory communications protocol to connect.
Description	If you do not specify a CommLinks (LINKS) connection parameter, the client searches for a server on the current machine only, and only using a shared memory connection. This is the default behavior, and is equivalent to CommLinks=ShMem. The shared memory protocol is the fastest communication link between a client and server running on the same machine, as is typical for applications connecting to a personal database server.
	If you specify CommLinks=ALL, the client searches for a server using all available communication protocols. Since there may be an impact on performance if you specify CommLinks=ALL, use this setting only when you don't know which protocol to use.
	If you specify one or more protocols in the CommLinks (LINKS) connection parameter, the client uses the named communication protocol(s), in the order specified, to search for a network database server. A connection error appears and the connection attempt aborts if the connection fails to connect using a specified protocol, even if there are protocols remaining in the list to try.
	CommLinks (LINKS) connection parameter values are case insensitive, and include:
	• SharedMemory (ShMem) Start the shared memory protocol for same-machine communication. This is the default setting. The client tries shared memory first if it appears in a list of protocols, regardless of the order in which protocols appear.
	♦ ALL Attempt to connect using the shared memory protocol first, followed by all remaining and available communications protocols. Use this setting if you are unsure of which communication protocol(s) to use.
	◆ NamedPipes (NP) For C2 security purposes, connect from a Windows NT/2000/XP client to a database server on the same machine that was started with the -sc option.

	• TCPIP (TCP) Start the TCP/IP communications protocol. TCP/IP is supported on all operating systems.
	• SPX Start the SPX communications protocol. The SPX protocol is supported for Windows and NetWare clients.
	Each of these values can have additional network communications parameters supplied.
	For a list of parameters, see "Network communications parameters" on page 201.
	You may wish to use a specific protocol, as opposed to ALL, for the following reasons:
	• The network library starts slightly faster if the client uses only necessary network protocols.
	• Connecting to the database may be faster.
	• You must specify the protocol explicitly if you wish to tune the broadcast behavior of a particular protocol by providing additional network communications parameters.
	The CommLinks (LINKS) connection parameter corresponds to the database server $-x$ option.
Examples	• The following connection string fragment starts the TCP/IP protocol only:
	CommLinks=tcpip
	• The following connection string fragment starts the shared memory protocol and searches for the database server over shared memory. If the search fails, it then starts the TCP/IP port and searches for the server on the local network. If that fails, it starts the SPX port and searches for the engine over SPX.
	CommLinks=tcpip, shmem, spx
	• The following connection string fragment starts the SPX port and searches for the engine over SPX. If the search fails, it then starts the TCP port and searches for the server on the local network, as well as the host kangaroo. Note that if the server is found over SPX, the TCP port is NOT started.

CommLinks=spx,tcpip(HOST=kangaroo)

See also	"Network communications parameters" on page 201
	"Client/Server Communications" on page 93
	"-x server option" on page 162
Compress connec	tion parameter [COMP]
Function	To turn compression ON or OFF for a connection. Compressing a connection may improve performance under some circumstances.
Usage	Anywhere except with TDS connections. TDS connections (including jConnect) do not support Adaptive Server Anywhere communication compression.
Values	YES, NO
	In the case of a difference between client and server settings, the client setting applies.
Default	NO
	If no Compress value is set, the compression status is controlled by the setting on the server, which defaults to no compression.
Description	The packets sent between an Adaptive Server Anywhere client and server can be compressed using the Compress (COMP) connection parameter. Large data transfers with highly compressible data tend to get the best compression rates.
	Available values of the Compress (COMP) connection parameter are case insensitive, and include:
	• YES Turn communication compression on for this connection.
	• NO Turn communication compression off for this connection.
	To save yourself time and possible disappointment, it is wise to conduct a performance analysis on the particular network and using the particular application before using communication compression in a production environment.
	To enable compression for all remote connections on the server, use the $-pc$ server option.
	Note that same-machine connections over any communication link will not enable compression, even if the -pc option or COMPRESS=YES parameter is used.
Examples	• The following connection string fragment turns packet compression ON:
	Compress=YES

	• The following connection string fragment turns packet compression OFF:
	Compress=NO
See also	"-pc server option" on page 154
	"Adjusting communication compression settings to improve performance" on page 102

CompressionThreshold connection parameter [COMPTH]

Function	To increase or decrease the size limit at which packets are compressed. Changing the compression threshold can help performance of a compressed connection by allowing you to only compress packets when compression will increase the speed at which the packets are transferred.
Usage	Anywhere except TDS. Only applies to compressed connections.
Values	<i>Integer</i> , representing the minimum byte-size of packets to be compressed. Values less than 80 are not recommended.
	If both client and server specify different compression threshold settings, the client setting applies.
Default	120
	If no CompressionThreshold value is set, the compression threshold value is controlled by the setting on the server, which defaults to 120 bytes.
Description	When compression is enabled, individual packets may or may not be compressed, depending on their size. For example, Adaptive Server Anywhere does not compress packets smaller than the compression threshold, even if communication compression is enabled. As well, small packets (less than about 100 bytes) usually do not compress at all. Since CPU time is required to compress packets, attempting to compress small packets could actually decrease performance.
	Generally speaking, lowering the compression threshold value may improve performance on very slow networks, while raising the compression threshold may improve performance by reducing CPU. However, since lowering the compression threshold value will increase CPU usage on both the client and server, a performance analysis should be done to determine whether or not changing the compression threshold is beneficial.
Example	• Connect, with a compression threshold of 100 bytes.
	CompressionThreshold=100
See also	"-pt server option" on page 155

"Adjusting communication compression settings to improve performance" on page 102

ConnectionName connection parameter [CON]

Function	Names a connection, to make switching to it easier in multi-connection applications.
Usage	Not available for ODBC.
Values	String
Default	No connection name.
Description	An optional parameter, providing a name for the particular connection you are making. You may leave this unspecified unless you are going to establish more than one connection, and switch between them.
	The connection name is not the same as the data source name.
Example	 Connect, naming the connection First_Con:
	CON=First_Con

DatabaseFile connection parameter [DBF]

Function	For use when starting a database that is not already running. The DatabaseFile connection parameter indicates which database file you want to
	load and connect to.
	If you want to connect to an already running database, use the DatabaseName (DSN) parameter.
Usage	Embedded databases
Values	String
Default	There is no default setting.
Description	The DatabaseFile (DBF) connection parameter is used to load and connect to a specific database file that is not already running on a database server.
	 If the database you want to connect to is not already running, use the DatabaseFile (DBF) connection parameter so the database can be started.
	• If the filename does not include an extension, Adaptive Server Anywhere looks for a file with the <i>.db</i> extension.
	• The path of the file is relative to the working directory of the database server. If you start the server from a command prompt, the working

	directory is the directory that you are in when entering the command. If you start the server from an icon or shortcut, it is the working directory that the icon or shortcut specifies. It is recommended that you supply a complete path and file name.
	• If you specify both the database file and the database name, the database file is ignored and is not used to try to connect to a database that is already running.
	You can also use UNC filenames and Novell NetWare Directory Services file names.
	For information about using UNC filenames and Novell NetWare Directory Services file names, see "The database server" on page 124.
	Caution The database file must be on the same machine as the database server. Starting a database file that is located on a network drive can lead to file corruption.
Example	◆ The DatabaseFile (DBF) connection parameter in the following example loads and connects to the sample database, <i>asademo.db</i> , installed in the directory c:\Program Files\Sybase\SQL Anywhere 9:
	DBF=c:\Program Files\Sybase\SQL Anywhere 9\asademo.db
	• The following two examples assume that you have started a database file named cities.db, and renamed the database Kitchener as follows:
	dbeng9 cities.db -n Kitchener
	• To successfully start and connect to a database and name it Kitchener:
	DBN=Kitchener;DBF=cities.db
	• Specifying DBF=cities.db would fail to connect to the running database named Kitchener.
See also	"DatabaseName connection parameter [DBN]" on page 184
DatabaseNam	e connection parameter [DBN]
Function	For use when connecting to a database that is already running. Identifies a loaded database to which a connection needs to be made.
	If you want to connect to a database that is not already running, use the DatabaseFile (DBF) parameter.
Usage	Running local databases or network servers.

Values	String
Default	There is no default setting.
Description	Whenever a database is started on a server, it is assigned a database name, either by the administrator using the -n option, or by the server using the base of the filename with the extension and path removed.
	If the database you want to connect to is already running, you should specify the database name rather than the database file.
	A connection will only occur if the name of the running database matches the name that is specified in the DatabaseName (DBN) parameter.
	Note If you specify both the database name and database file, the database file is ignored and is not used to try to connect to a database that is already running.
Example	• To start a database file named <i>cities.db</i> and rename the database <i>Kitchener</i> , you can use the following command:
	dbeng9 cities.db -n Kitchener
	 Assuming you have run the above command, you can successfully connect to the running database named Kitchener as follows: DBN=Kitchener
	 Alternatively, you could use the following to successfully connect to the running database named Kitchener:
	DBN=Kitchener;DBF=cities.db
	• However, specifying the following would fail to connect to the database named Kitchener:
	DBF=cities.db
DatabaseSwitches	s connection parameter [DBS]
Function	To provide database-specific options (switches) when starting a database.
Usage	Connecting to a server when the database is not loaded. This connection parameter autostarts an engine with the specified database and options if a server is not already running.

Values String

Default No options

Description	You should supply DatabaseSwitches only if you are connecting to a database that is not currently running. When the server starts the database specified by DatabaseFile, the server uses the supplied DatabaseSwitches to determine startup options for the database.
	Only database options (switches)
	can be supplied using this parameter. Server switches must be supplied using the StartLine connection parameter.
	For information about database options, see "Database options" on page 168.
Example	 The following command, entered all on one line at a command prompt, connects to the default database server, loads the database file asademo.db (DatabaseFile (DBF) connection parameter), names it as my_db (DBS parameter) and connects to the database of that name (DatabaseName (DBN) connection parameter).
	dbisql -c "uid=DBA;pwd=SQL;dbf=c:\Program Files\Sybase\SQL Anywhere 9\asademo.db;dbn=my_db;dbs=-n my_db"
See also	"The database server" on page 124
	"StartLine connection parameter [START]" on page 199

DataSourceName connection parameter [DSN]

Function	Tells the ODBC driver manager or Embedded SQL library where to look in the <i>odbc.ini</i> file or registry to find ODBC data source information.
Usage	Anywhere
Values	String
Default	There is no default data source name.
Description	It is common practice for ODBC applications to send only a data source name to ODBC. The ODBC driver manager and ODBC driver locate the data source, which contains the remainder of the connection parameters.
	In Adaptive Server Anywhere, Embedded SQL applications can also use ODBC data sources to store connection parameters.
Example	• The following parameter uses a data source name:
	DSN=Dynamo Demo
See also	"FileDataSourceName connection parameter [FILEDSN]" on page 191

DisableMultiRowFetch connection parameter [DMRF]

Function	To turn off multi-row fetches across the network
Usage	Anywhere
Values	YES, NO
Default	NO
Description	By default, when the database server gets a simple fetch request, the application asks for extra rows. You can disable this behavior by setting this parameter to ON .
	For more information, see "Using cursors in procedures and triggers" [ASA SQL User's Guide, page 646].
	Setting the DisableMultiRowFetch (DMRF) connection parameter to ON is equivalent to setting the PREFETCH database option to OFF . For more information, see "Prefetching rows" [<i>ASA Programming Guide</i> , page 42].
Example	• The following connection string fragment prevents prefetching:
	DMRF=YES

EngineName connection parameter [ENG]

Function	Synonym for ServerName. The name of a running database server to which you want to connect.
Usage	Network servers or running personal servers.
Values	String
Default	The default local database server.
Description	EngineName is not needed if you wish to connect to the default local database server.
	You need to supply an EngineName only if more than one local database server is running, or if you wish to connect to a network server. In the Connect dialog, and in the ODBC Administrator, this is the Server Name field.
	The server name is interpreted according to the character set of the client machine. Multi-byte characters are not recommended in server names.
	Names must be a valid identifier. Long engine names are truncated to different lengths depending on the protocol.

	Protocol	Truncation Length
	Unix shared memory	31 bytes
	non-Unix shared memory	40 bytes
	TCP/IP	40 bytes
	SPX	32 bytes
	Named Pipes	8 bytes
Example	• Connect to a server named Guelp	bh:
	ENG=Guelph	
See also	"Identifiers" [ASA SQL Reference, pag	e 7]
	"-n server option" on page 152	

EncryptedPassword connection parameter [ENP]

Function	To provide a password, stored in an encrypted fashion in a data source.
Usage	Anywhere
Values	String
Default	None
Description	Data sources are stored on disk as a file or in the registry. Storing passwords on disk may present a security problem. For this reason, when you enter a password into a data source, it is stored in an encrypted form.
	If both the Password (PWD) connection parameter and the EncryptedPassword (ENP) connection parameter are specified, Password (PWD) takes precedence.
Encryption connec	ction parameter [ENC]

Function	To encrypt packets sent between the client application and the server.
Usage	For ECC_TLS (Certicom), RSA_TLS, TCP/IP only.
	For none or simple, anywhere.
Values	String
Default	NONE
	If an Encryption value is not set, encryption is controlled by the setting on the server, which defaults to no encryption.

Description You can use this parameter if you are concerned about the security of network packets. Encryption does affect performance marginally. The **Encryption (ENC)** connection parameter accepts the following arguments:

none accepts communication packets that are not encrypted. This value is equivalent to NO in previous versions of Adaptive Server Anywhere.

simple accepts communication packets that are encrypted with simple encryption supported on all platforms and on previous versions of Adaptive Server Anywhere. This value is equivalent to YES in previous versions of Adaptive Server Anywhere.

ECC_TLS (formerly Certicom) accepts communication packets that are encrypted using Certicom encryption technology. To use this type of encryption, both the server and the client must be operating on Solaris, Linux, NetWare, and all supported Windows operating systems except Windows CE, and the connection must be over the TCP/IP port. UNIX platforms, except for Solaris and Linux, do not recognize the client or server **Certicom** parameter. To authenticate the server, the Certicom software verifies that the server's certificate values match any values you supply about the client using the following arguments:

- trusted_certificates specify the certificate file the client uses to authenticate the server.
- **certificate_company** specify the value for the organization field. The server's value and the client's value must match.
- **certificate_unit** specify the value for the organization unit field. The server's value and the client's value must match.
- **certificate_name** specify the certificate's common name. The server's value and the client's value must match.

RSA_TLS accepts communication packets that are encrypted using RSA encryption technology. To use this type of encryption, both the server and the client must be operating on Solaris, Linux, NetWare, and all supported Windows operating systems except Windows CE, and the connection must be over the TCP/IP port. UNIX platforms, except for Solaris and Linux, do not recognize the client or server **RSA_TLS** parameter. To authenticate the server, the Certicom software verifies that the server's certificate values match any values you supply about the client using the following arguments:

- trusted_certificates specify the certificate file the client uses to authenticate the server.
- certificate_company specify the value for the organization field. The server's value and the client's value must match.

	• certificate_unit specify the value for the organization unit field. The server's value and the client's value must match.
	• certificate_name specify the certificate's common name. The server's value and the client's value must match.
	Caution The sample certificate should be used for testing purposes only. The sample certificate provides no security in deployed situations because it and the corresponding password are widely distributed with Sybase software. To protect your system, you must create your own certificate.
	For information about certificates, see "Self-signed certificates" [<i>MobiLink Synchronization User's Guide</i> , page 347].
	You can use the connection_property system function to retrieve the encryption settings for the current connection. The function returns one of three values: none, simple, or Certicom, depending which type of encryption is being used.
	For information about using the connection_property system function, see "CONNECTION_PROPERTY function [System]" [ASA SQL Reference, page 106].
Example	• The following connection string fragment connects to a database server myeng with a TCP/IP link, using Certicom encryption and the sample trusted certificate:
	"ENG=myeng; LINKS=tcpip; Encryption=ECC_TLS (trusted_ certificates=sample.crt)"
	• The following connection string fragment connects to a database server myeng with a TCP/IP link, using RSA encryption and the sample trusted certificate:
	"ENG=myeng; LINKS=tcpip; Encryption=RSA_TLS (trusted_ certificates=sample.crt)"
See also	"-ec server option" on page 141
	"Encrypting client/server communications" on page ??
Encryption Key	connection parameter [DBKEY]
Function	To start an encrypted database with a connect request.
Usage	Anywhere

Values String

Default	None
Description	You must specify this parameter when you start an encrypted database with a connect request. You do not need to specify this parameter if you are connecting to an encrypted database that is already running.
	The encryption key is a string, including mixed cases, numbers, letters, and special characters.
Example	 The following fragment illustrates the use of the Encryption Key (DBKEY) connection parameter:
	"UID=dba;PWD=sql;ENG=myeng;DBKEY=V3moj3952B"
See also	"-ek database option" on page 168
	"-ep server option" on page 143
	"Encrypting client/server communications" on page ??

FileDataSourceName connection parameter [FILEDSN]

Function	The FileDataSourceName (FILEDSN) connection parameter tells the client library that there is an ODBC file data source holding information about the database to which you want to connect.
	Both ODBC and Embedded SQL applications can use File data sources
Usage	Anywhere
Values	String
Default	There is no default name.
Description	File data sources hold the same information as ODBC data sources stored in the registry. File data sources can be easily distributed to end users so that connection information does not have to be reconstructed on each machine.
See also	"DataSourceName connection parameter [DSN]" on page 186

ForceStart connection parameter [FORCESTART]

Function	To start a server without attempting to connect to one.
Usage	Only with the db_start_engine function
Values	YES, NO
Default	NO
See also	"db_start_engine function" [ASA Programming Guide, page 216]

Idle connection parameter [IDLE]

Function	To specify the connection's idle timeout period.
Usage	Anywhere except with TDS and Shared Memory connections. Shared Memory and TDS connections (including jConnect) ignore the Adaptive Server Anywhere Idle (IDLE) connection parameter.
Values	Integer
Default	240
Description	The Idle (IDLE) connection parameter applies only to the current connection. You can have multiple connections on the same server set to different timeout values.
	If no connection idle timeout value is set, the idle timeout value is controlled by the setting on the server, which defaults to 240 minutes. In case of a conflict between timeout values, the connection timeout value supercedes any server timeout value whether specified or unspecified.
	Optionally, the relevant server command line parameters could be included for both Idle and Liveness Timeout (-ti and -tl respectively).
Example	• The following connection string fragment sets the timeout value for this connection to 10 minutes:
	"ENG=myeng;LINKS=tcpip;IDLE=10"
See also	"-ti server option" on page 159
	e die en en en en filliti

Integrated connection parameter [INT]

Function	To use the integrated login facility.
Usage	Anywhere
Values	YES, NO
Default	NO
Description	The Integrated (INT) connection parameter has the following settings:
	• YES An integrated login is attempted. If the connection attempt fails and the LOGIN_MODE option is set to Mixed, a standard login is attempted.
	• NO This is the default setting. No integrated login is attempted.

For a client application to use an integrated login, the server must be running with the LOGIN_MODE database option set to Mixed or Integrated.

Example	 The following data source fragment uses an integrated login: INT=YES
See also	"LOGIN_MODE option [database]" on page 602
Language [LANG]	
Function	Specifies the language of the connection.
Usage	Anywhere
Values	The two-letter combination representing a language. For example, setting LANG=DE sets the default language to German.
Default	The language specified by (in order) ASLANG, dblang.exe, or the installer.
Description	This connection parameter establishes the language for the connection. Any errors or warnings from the server will be delivered in the specified language, assuming that the server supports the language.
	If no language is specified, the default language is used. The default language is the language specified by, in order, ASLANG, dblang.exe, or the installer.
	This connection parameter only affects the connection. Messages returned from Adaptive Server Anywhere's various tools and utilities will appear in the default language, while the messages returned from the server will appear in the connection's language.

LazyClose connection parameter [LCLOSE]

Function	Enabling this option causes the CLOSE <i>cursor-name</i> database request to be queued, and then sent to the server with the next database request. This eliminates a network request each time a cursor is closed.
Usage	Anywhere
Values	YES, NO
Default	NO
Description	When this parameter is enabled, cursors are not actually closed until the next database request. Any isolation level 1 cursor stability locks still apply to the cursor while the CLOSE <i>cursor-name</i> database request is queued.
	Enabling this option can improve performance if your:
	 network exhibits poor latency

• application sends many cursor open and close requests

Note that in rare circumstances, canceling the next request after the CLOSE *cursor-name* database request can leave the cursor in a state where it appears to be closed on the client side, but is not actually closed on the server side. Subsequent attempts to open another cursor with the same name will fail. Using LazyClose is not recommended if your application cancels requests frequently.

LivenessTimeout connection parameter [LTO]

Function	To control the termination of connections when they are no longer intact.
Usage	Network server only.
	All platforms except non-threaded UNIX applications.
Values	Integer (in seconds)
Default	120
	If no LivenessTimeout value is set, the LivenessTimeout is controlled by the setting on the server, which defaults to 120 seconds.
Description	A liveness packet is sent periodically across a client/server TCP/IP or SPX communications protocol to confirm that a connection is intact. If the client runs for the LivenessTimeout period without detecting a liveness request or response packet, the communication is severed.
	Liveness packets are sent when a connection has not sent any packets for between one third and two thirds of the LivenessTimeout value.
	When there are more than 200 connections to a server, the server automatically calculates a higher LivenessTimeout value based on the stated LivenessTimeout value. This enables the server to handle a large number of connections more efficiently.
Example	• The following sets a LivenessTimeout value of 10 minutes
	LTO=600
	Alternatively, you can set this parameter by entering its value in the LivenessTimeout text box of the Network tab of the ODBC Configuration dialog.

Logfile connection parameter [LOG]

Function	To send client error messages and debugging messages to a file.
Usage	Anywhere
Values	String

Default	No log file
Description	If you want to save client error messages and debugging messages in a file, use the Logfile (LOG) connection parameter.
	If the file name includes a path, it is relative to the current working directory of the client application.
	The LogFile (LOG) connection parameter is connection-specific, so from a single application you can set different LogFile arguments for different connections.
Example	The following command line starts Interactive SQL connecting to the ASA 9.0 Sample data source with a LogFile (LOG) connection parameter:
	dbisql -c "DSN=ASA 9.0 Sample;LOG=d:\logs\test.txt"
	Typical log file contents are as follows:

Fri Oct 27 2000 11:45 Application information: "HOST=NAME-PC;OS=Windows NT 4.0 (Service Pack 5); PID=0x14b; THREAD=0x148; EXE=C:\ASA90\WIN32\ DBPING.EXE;VERSION=9.0.0.1271;API=DBLIB;TIMEZONEADJUSTM ENT=-300" Attempting to connect using: UID=dba; PWD=***; ENG=name; DBG=YES; LOG=c:\temp\ cli.out;LINKS=shmem,tcpip Attempting to connect to a running server... Trying to start SharedMemory link ... SharedMemory link started successfully Attempting SharedMemory connection (no asasrv.ini cached address) Failed to connect over SharedMemory Trying to start TCPIP link ... Loading wsock32.dll Loading ws2_32.dll TCP using Winsock version 2.0 My IP address is 172.31.142.196 My IP address is 127.0.0.1 TCPIP link started successfully Attempting TCPIP connection (address 172.31.143.196:2638 found in asasrv.ini cache) Trying to find server at cached address 172.31.143.196:2638 without broadcasting Server not found (no reply received) Looking for server with name NAME I am in a class B network Sending broadcast to find server Using broadcast address of: 172.31.255.255:2638 I am in a class A network Sending broadcast to find server Using broadcast address of: 127.255.255.255:2638 Found database server at address 172.31.142.196:2638 Found database server NAME on TCPIP link Connected to server over TCPIP at address 172.31.142.196:2638 Writing server address 172.31.142.196:2638 to asasrv.ini cache Liveness timeout 120, liveness retransmit period 30 Connected to the server, attempting to connect to a running database... Connected to database successfully

Password connection parameter [PWD]

Function	To provide a password for the connection.
Usage	Anywhere
Values	String
Default	No password provided.

Description	Every user of a database has a password. The password must be supplied for the user to be allowed to connect to the database.
	The Password (PWD) connection parameter is not encrypted. If you are storing passwords in a data source, you should use the EncryptedPassword (ENP) connection parameter. Sybase Central and the Adaptive Server Anywhere ODBC configuration tool both use encrypted parameters.
	If both Password (PWD) connection parameter and the EncryptedPassword (ENP) connection parameter are specified, the Password (PWD) connection parameter takes precedence.
Example	• The following connection string fragment supplies the user ID DBA and password SQL .
	UID=DBA; PWD=SQL
	Alternatively, you can set these parameters in the User ID and Password text boxes in the Connect dialog and ODBC Administrator dialog.
See also	"EncryptedPassword connection parameter [ENP]" on page 188
PrefetchBuffer cor	nnection parameter [PBUF]
Function	Set the maximum amount of memory for buffering rows, in kilobytes.
Usage	Anywhere
Values	Integer
Default	16 (Windows CE) 64 (all other platforms)
Description	The PrefetchBuffer (PBUF) connection parameter controls the memory allocated on the client to store prefetched rows. In some circumstances, increasing the number of rows prefetched from the database server by the client can improve query performance. You can increase the number of rows prefetched using the PrefetchRows (PROWS) and PrefetchBuffer (PBUF) connection parameters.
	Increasing the PrefetchBuffer (PBUF) connection parameter increases the amount of memory used to buffer GET DATA requests. This may improve performance for some applications that process many GET DATA (SQLGetData) requests.
	For more information, see "PrefetchRows connection parameter [PROWS]" on page 198.
Examples	• The following connection string fragment could be used to determine if the PrefetchBuffer memory limit is reducing the number of rows prefetched.

... PrefetchRows=100; Debug=YES; Logfile=c:\client.txt

• The following string could be used to increase the memory limit to 256K:

... PrefetchRows=100; PrefetchBuffer=256

PrefetchRows connection parameter [PROWS]

Function	Set the maximum number of rows to prefetch when querying the database.
Usage	Anywhere
Values	Integer
Default	10
Description	Increasing the number of rows prefetched from the database server by the client can improve performance on cursors that only fetch relative 0 or 1, with either single row or wide fetches. Wide fetches include embedded SQL array fetches and ODBC block fetches.
	Improvements occur particularly under the following conditions:
	• The application fetches many rows (several hundred or more) with very few absolute fetches.
	• The application fetches rows at a high rate, and the client and server are on the same machine or connected by a fast network.
	 Client/server communication is over a slow network, such as a dial-up link or wide area network.
	The number of rows prefetched is limited both by the PrefetchRows (PROWS) connection parameter and the PrefetchBuffer (PBUF) connection parameter, which limits the memory available for storing prefetched rows.
	For more information, see "PrefetchBuffer connection parameter [PBUF]" on page 197.
Example	• The following connection string fragment sets the number of prefetched rows to 100:
	PrefetchRows=100;

ServerName connection parameter [ENG]

This is a synonym for the EngineName (ENG) connection parameter.

For more information, see "EngineName connection parameter [ENG]" on page 187.

StartLine connection parameter [START]

Function	To start a database server running from an application.
Usage	Embedded databases
Values	String
Default	No StartLine parameter
Description	You should supply a StartLine (START) connection parameter only if you are connecting to a database server that is not currently running. The StartLine (START) connection parameter is a command line to start a personal database server.
	For a detailed description of available command line options, see "The database server" on page 124.
Example	• The following data source fragment starts a personal database server with a cache of 8 Mb.
	StartLine=dbeng9 -c 8M;dbf=asademo.db

Unconditional connection parameter [UNC]

Function	To stop a server using <i>dbstop</i> even when there are connections to the server.
Usage	Anywhere
Values	YES, NO
Default	NO
Description	The <i>dbstop</i> utility shuts down a database server. If you specify UNC=YES in the connection string, the server is shut down even if there are active connections. If Unconditional is not set to YES, then the server is shut down only if there are no active connections.
Example	 The following command line shuts down the server unconditionally: dbstop -c "UID=DBA; PWD=SQL; ENG=server-name; UNC=YES"
See also	"Stopping a database server using the dbstop command-line utility" on page 525

Userid connection parameter [UID]

Function The user ID with which you log on to the database.

Usage	Anywhere
Values	String
Default	None
Description	You must always supply a user ID when connecting to a database.
Example	• The following connection string fragment supplies the user ID DBA and password SQL :

uid=DBA;pwd=SQL
Network communications parameters

Communication parameters (for both the client and the server) enable you to work around peculiarities of different network protocol implementations.

You can supply the network communication parameters in the server command. For example:

```
dbsrv9 -x tcpip(PARM1=value1;PARM2=value2;. . .),SPX
```

From the client side, you enter the communications parameters as the **CommLinks** (LINKS) connection parameter:

```
CommLinks=tcpip(PARM1=value1;PARM2=value2;. . .),SPX
```

If there are spaces in a parameter, the network communication parameters must be enclosed in quotation marks to be parsed properly by the system command interpreter:

```
dbsrv9 -x "tcpip(PARM1=value 1;PARM2=value 2;...),SPX"
CommLinks="tcpip(PARM1=value 1;PARM2=value 2;...),SPX"
```

The quotation marks are required under UNIX if more than one parameter is given because UNIX interprets the semicolon as a command separator.

Boolean parameters are turned on with YES, ON, TRUE, or 1, and are turned off with any of NO, OFF, FALSE, and 0. The parameters are case insensitive.

The examples provided should all be entered on a single line; you can also include them in a configuration file and use the @ server option to invoke the configuration file.

The parameters currently available for TCP/IP and SPX are as follows.

TCP/IP, HTTP, HTTPS, and SPX		I	I
communications parameters	TCP/IP	HTTP & HTTPS	SPX
	Broadcast [BCAST]		BroadcastListener [BLISTENER]
	BroadcastListener [BLISTENER]		DLL
		Certificate	
		Certificate_Password	
	ClientPort [CPORT]		DoBroadcast [DO- BROAD]

TCP/IP	HTTP & HTTPS	SPX
	DatabaseName [DBN]	
DLL		ExtendedName [ENAME]
DoBroadcast [DO- BROAD]		Host [IP]
Host [IP]		RegisterBindery [REG- BIN]
LocalOnly [LOCAL]	LocalOnly [LOCAL]	SearchBindery [BIN- SEARCH]
LDAP [LDAP]		
	LogFile [LOG]	
	LogMaxSize [LSize]	
	LogOptions [LOpt]	
	LogFormat [LF]	
	MaxConnections [Max- Conn]	
	MaxRequestSize [Max- Size]	
MyIP [ME]	MyIP [ME]	Timeout [TO]
ReceiveBufferSize [RCVBUFSZ]		
SendBufferSize [SND- BUFSZ]		
ServerPort [PORT]	ServerPort [PORT]	
TDS		
Timeout [TO]	Timeout [TO]	
VerifyServerName [VERIFY]		

Broadcast communication parameter [BCAST]

Usage

TCP/IP

Values	String (in the form of an IP address)
Default	Broadcasts to all addresses on the same subnet
Description	BROADCAST specifies the IP address used by your TCP/IP protocol implementation to identify a broadcast message.
	Broadcast addresses consist of the network IP address portion, with 255 as the remaining integers. For example
	 if the network portion is 95 (class A), then the broadcast address would be 95.255.255.255
	• if the network portion is 132.10 (class B), then the broadcast address would be 132.10.255.255
	 if the network portion is 197.31.175 (class C), then the broadcast address would be 197.31.175.255
	The broadcast IP address 255.255.255.255 broadcasts to all subnets.

BroadcastListener communication parameter [BLISTENER]

Usage	SPX, TCP/IP, Server side
Values	YES, NO
Default	YES
Description	This option allows you to turn broadcast listening OFF for this port.
	Using -sb 0 is the same as specifying BroadcastListener=NO on both TCP/IP and SPX.
Example	• Start a server that accepts both TCP/IP and SPX connections, but require that TCP/IP connections use BroadcastListener=NO:
	dbsrv9 -x tcpip(BroadcastListener=NO),spx
See also	"-sb server option" on page 158
Certificate	
Usage	HTTP, HTTPS
Values	String
Default	There is no default certificate name.
Description	This option allows you to specify the name of an encryption certificate. The password for this certificate must be specified with the Certificate_Password parameter.

Example

• Start a server that requires web connections to use a particular encryption certificate.

Certificate_Password

Usage	HTTP, HTTPS
Values	String
Default	There is no default certificate password.
Description	This option allows you to specify the password that matches the encryption certificate specified by the Certificate parameter.
Example	 Start a server that requires web connections to use a particular encryption certificate.
	dbsrv9 -xs http(Certificate=cert.file;Certificate_ Password=secret)

ClientPort communication parameter [CPORT]

Usage	TCP/IP. Client side only.
Values	Integer
Default	Assigned dynamically per connection by the networking implementation. If you do not have firewall restrictions, it is recommended that you do not use this parameter.
Description	This option is provided for connections across firewalls, as firewall software filters according to TCP/UDP port. It is recommended that you do not use this parameter unless you need to for firewall reasons.
	The ClientPort option designates the port number on which the client application communicates using TCP/IP. You may specify a single port number, or a combination of individual port numbers and ranges of port numbers.
	It is best to specify a list or a range of port numbers if you want to make multiple connections using a given Data Source or a given connect string. If you specify a single port number, then your application will be able to maintain only one connection at a time. In fact, even after closing the one connection, there is a short timeout period during which no new connection can be made using the specified port. When you specify a list and/or range of port numbers, the application keeps trying port numbers until it finds one to which it can successfully bind.

Examples	 The following string makes a connection from an application using port 6000 to a server named my_server using port 5000:
	CommLinks=tcpip{ClientPort=6000;ServerPort=5000}; ServerName=my_server
	 The following string makes a connection from an application that can use ports 5050 through 5060, as well as ports 5040 and 5070 for communicating with a server named my_server using the default server port:
	CommLinks=tcpip{ClientPort=5040,5050-5060,5070};ServerName=my_ server
See also	"Host communication parameter [IP]" on page 207
	"DoBroadcast communication parameter [DOBROAD]" on page 206
	"ServerPort communication parameter [PORT]" on page 215
	"Connecting across a firewall" on page 95

Database Name [DBN]

Usage	HTTP, HTTPS
Values	AUTO, REQUIRED, database-name
Default	AUTO
Description	Specifies the name of a database to use when processing web requests, or uses the REQUIRED or AUTO keyword to specify whether database names are required as part of the URI.
	If this parameter is set to REQUIRED, the URI must specify a database name.
	If this parameter is set to AUTO, the URI may specify a database name, but does not need to do so. If the URI contains no database name, the default database on the server is used to process web requests. Since the server must guess whether or not the URI contains a database name when set to AUTO, you should design your web site so as to avoid ambiguity.
	If this paramter is set to the name of a database, that database is used to process all web requests. The URI must not contain a database name.
Example	• The following command starts two databases, but permits only one of them to be accessed via HTTP.

dbsrv9 -xs http(dbn=web) asademo.db web.db

DLL communication parameter

Usage	TCP/IP, SPX (Windows 95/98/Me, Windows NT/2000/XP)
Values	String
Default	• On Windows NT/2000/XP, the default is ws2_32.dl 1 (Winsock 2.0).
	• On Windows 95/98/Me the default is wsock32.dll (Winsock 1.1).
Description	To support untested TCP/IP protocol stacks where the required networking interface functions are in DLLs that differ from the default protocol stack. The client or server looks for its required functionality in the named DLLs.
Example	• The following command starts a server using Winsock 1.1:
	dbsrv9 -x tcpip(dll=wsock32.dll) asademo

DoBroadcast communication parameter [DOBROAD]

Usage	TCP/IP (all platforms)
	SPX (all platforms except UNIX and CE)
Values	ALL, NONE, DIRECT (Client side)
	YES, NO (Server side)
Default	ALL
Description	Client usage With DoBroadcast=ALL (formerly DoBroadcast=YES) a broadcast is performed to search for a server. The broadcast goes first to the local subnet. If HOST= is specified, broadcast packets are also sent to each of the hosts. For TCP, all broadcast packets are UDP packets. For SPX, the broadcast is only performed if the server is not found in the bindery. For SPX, all broadcast packets are IPX packets.
	With DoBroadcast=DIRECT (formerly DoBroadcast=NO), no broadcast is performed to the local subnet to search for a database server. Broadcast packets are sent only to the hosts listed in the HOST (IP) communication parameter. If you specify DoBroadcast=DIRECT, the HOST (IP) communication parameter is required.
	Specifying DoBroadcast=NONE causes no UDP or IPX broadcasts to be used. A TCP/IP or SPX connection is made directly with the HOST/PORT specified, and the engine name is verified. With TCP/IP, you can choose not to verify the engine name by setting the VerifyServerName (VERIFY) communication parameter to NO . The HOST (IP) communication

parameter is a required parameter, while the **ServerPort (PORT)** communication parameter is optional.

For DIRECT and NONE, you must specify the server host with the HOST option.

Server usage Setting DoBroadcast=NO prevents the database server from broadcasting to find other servers with the same name. This is useful in certain rare circumstances, but it is not generally recommended.

 Example
 The following command starts a client without broadcasting to search for a database server. Instead, the server is looked for only on the computer named silver.

CommLinks=tcpip(DOBROADCAST=DIRECT;HOST=silver) asademo

ExtendedName communication parameter [ENAME]

Usage	SPX (platforms other than Windows 95/98/Me or Windows NT/2000/XP)
Values	YES, NO
Default	NO
Description	According to the Novell standard for legal SAP names, the following characters are not allowed:
	\
	If you start a server named "asademo-1", the default behavior is to strip out the - and try to start a server asademo1. By turning on ExtendedName, the name is left untouched.
	This is an SPX port parameter and is only useful when starting a server.
	Caution
	Users should be wary of using this option as it is contrary to the SAP standard.
Example	• The following command starts a NetWare server with the name asademo-1.
	<pre>load dbsrv9.nlm -x spx(ExtendedName=YES) asademo-1</pre>
Host communication	on parameter [IP]

Usage	TCP/IP, SPX (all platforms) Server and client sides
Values	String

Default	No additional machines.
Description	HOST specifies additional machines outside the immediate network to be searched by the client library. On the server, the search is carried out to avoid starting a server with a duplicate name.
	For TCP/IP, the HOST value the <i>hostname</i> or a dot-separated IP address may be used. You may optionally specify a PORT value as well.
	For SPX, an address of the form <i>a</i> : <i>b</i> : <i>c</i> : <i>d</i> : <i>e</i> : <i>f</i> / <i>g</i> : <i>h</i> : <i>i</i> : <i>j</i> is used, where <i>a</i> : <i>b</i> : <i>c</i> : <i>d</i> : <i>e</i> : <i>f</i> is the node number (Ethernet card address) of the server, and <i>g</i> : <i>h</i> : <i>i</i> : <i>j</i> is the network number. You may optionally specify a PORT value.
	The server prints addressing information to the database server window during startup if the $-z$ option is used. In addition, the application writes this information to its logfile if Debug is set to true and LogFile is specified.
	You can use a comma-separated list of addresses to search for more than one machine. You can also append a port number to an IP address, using a colon as separator. Alternatively, you can specify the host and server ports explicitly, as in $HOST=a:b:c:d:e:f/g;h:i:j;PORT=k$.
	To specify multiple values for a single parameter, use a comma-separated list. When you specify multiple ports and servers, you can associate a particular port with a specific server by specifying the port in the HOST (IP) communication parameter instead of the PORT parameter.
	IP and HOST are synonyms when using TCP/IP. For SPX, you must use HOST . PORT and ServerPort are synonyms for both protocols.
Examples	• The following connection string fragment instructs the client to look on the machines kangaroo and 197.75.209.222 (port 2369) to find a database server:
	LINKS=tcpip(IP=kangaroo,197.75.209.222:2369)
	• The following connection string fragment instructs the client to look on the machines my_server and kangaroo to find a database server. A connection is attempted to the first host that responds.
	LINKS=tcpip(HOST=my_server,kangaroo;PORT=2639)
	• The following connection string fragment instructs the client to look for a server on host1 running on port 1234 and for a server on host2 running on port 4567. The client does not look on host1 on port 4567 or on host2 on port 1234.
	LINKS=tcpip(HOST=host1:1234,host2:4567)
See also	"ClientPort communication parameter [CPORT]" on page 204

LDAP communication parameter [LDAP]

Usage	TCP/IP, server-side only
Values	YES, NO, or filename
Default	ON
	The default <i>filename</i> is asaldap.ini
Description	Having the database server register itself with an LDAP server allows clients (and the Locate utility [dblocate]) to query the LDAP server. This allows clients running over a WAN or through a firewall to find servers without specifying the IP address. It also allows the Locate utility [dblocate] to find such servers.
	Specifying <i>LDAP=filename</i> turns LDAP support on and uses the specified file as the configuration file. Specifying <i>LDAP=YES</i> turns LDAP support on and uses <i>asaldap.ini</i> as the configuration file.
	LDAP is only used with TCP/IP, and only on network servers.
See also	"Connecting using an LDAP server" on page 97
LocalOnly commu	nication parameter [LOCAL]
Usage	TCP/IP, HTTP, HTTPS
Values	YES, NO
Default	NO
Description	The LocalOnly (LOCAL) communication parameter allows a client to choose to connect only to a server on the local machine, if one exists. If no server with the matching engine name is found on the local machine, a server will not be autostarted.
	The LocalOnly (LOCAL) communication parameter is only useful if DoBroadcast=ALL (the default)
	LocalOnly=YES uses the regular broadcast mechanism, except that broadcast responses from servers on other machines are ignored.
	You can use the LocalOnly [LOCAL) communication parameter with the server to restrict connections to the local machine. Connection attempts from remote machines will not find this server, and the Locate [dblocate] utility will not see this server. Running a server with the LocalOnly [LOCAL) communication parameter set to YES allows the network server to run as a personal server without experiencing connection or CPU limits.

See also

"Broadcast communication parameter [BCAST]" on page 202

LogFile communication parameter [LOG]

Usage	HTTP, HTTPS
Values	Filename
Default	None
Description	Specify the name of the file to which the database server is to write information about web requests.
See also	"LogFormat communication parameter [LF]" on page 210, "LogMaxsize communication parameter [LSIZE]" on page 211, "LogOptions communication parameter [LOPT]" on page 211

LogFormat communication parameter [LF]

Usage	HTTP, HTTPS
Values	Format-string
Default	@T - @W - @I - @P - "@M @U @V" - @R - @L - @E
Description	This parameter controls the format of messages written to the log file and which fields appear in them. If they appear in the string, the current values are substituted for the the following codes as each message is written.
	• @ @ The @ character.
	• @B Date and time that processing of the request started, unless the request could not be queued due to an error.
	• @C Date and time that the client connected.
	• @D Name of the database associated with the request.
	• @E Text of the error message, if an error occurred.
	• @F Date and time that processing of the request finished.
	• @I IP address of the client.
	• @L Length of the response, in bytes, including headers and body.
	• @M HTTP request method.
	• @P Listener port associated with the request.
	 @Q Date and time that the request was queued for processing, unless the request could not be queued due to an error.

- **@R** Status code and delcriltion of the HTTP response.
- **@S** HTTP status code.
- **@T** Date and time that the current log entry was written.
- ◆ **@U** Requested URI.
- ◆ **@V** Requested HTTP version.
- ◆ **@W** Time taken to process the request (@F @B), or 0.000 if the request was not processed due to an error.

See Also "LogFile communication parameter [LOG]" on page 210, "LogMaxsize communication parameter [LSIZE]" on page 211, "LogOptions communication parameter [LOPT]" on page 211

LogMaxsize communication parameter [LSIZE]

Usage	HTTP, HTTPS
Values	Size
Default	0
Description	When the log file reaches the stated size, it is renamed and another log file is created. If LogMaxSize is zero, the log file size is unlimited.
See Also	"LogFile communication parameter [LOG]" on page 210, "LogFormat communication parameter [LF]" on page 210, "LogOptions communication parameter [LOPT]" on page 211

LogOptions communication parameter [LOPT]

Usage	HTTP, HTTPS
Values	NONE, OK, INFO, ERRORS, ALL, status-codes, REQHDRS, RESHDRS, HEADERS
Default	ALL
Description	The values available include keywords that select particular types of messages, and HTTP status codes. Multiple values may be specified, separated by commas.
	The following keywords control which categories of messages are logged:
	◆ NONE Log nothing.
	• OK Log requests that complete sucessfully (20x HTTP status codes).

- ◆ INFO Log requests that return oved or not modified status codes (30x HTTP status codes).
- **ERRORS** Log all errors (40x and 50x HTTP status codes)
- ◆ ALL Log all requests.

The following common HTTP status codes are also available. They can be used to log requests that return particular status codes:

- ◆ C200 OK
- ♦ C400 Bad request
- ◆ C401 Unauthorized
- ♦ C403 Forbidden
- ♦ C404 Not Found
- ◆ C408 Request Timeout
- ◆ C501 Not implemented
- ◆ C505 Service unavailable

In addition, the following keywords may be used to obtain more information about the logged messages:

- **REQHDRS** When logging requests, also write request headers to the log file.
- ♦ RESHDRS When logging requests, also write response headers to the log file.
- HEADERS When logging requests, also write both request and response headers to the log file (same as REQHDRS, RESHDRS).

See Also "LogFile communication parameter [LOG]" on page 210, "LogFormat communication parameter [LF]" on page 210, "LogMaxsize communication parameter [LSIZE]" on page 211

MaxConnections communication parameter [MAXCONN]

Usage	HTTP, HTTPS
Values	Size
Default	5 (personal server) or number of licesed connections (network server)

Description	The number of simultaneous connections accepted by the server. The value 0 indicates no limit.
See Also	"MaxRequestSize communication parameter [MAXSIZE]" on page 213

MaxRequestSize communication parameter [MAXSIZE]

Usage	HTTP, HTTPS
Values	Size
Default	100 kb
Description	The size of the largest request accepted by the server. If the size of a request exceeds this limit, the connection is closed and a 413 ENTITY TOO LARGE response is returned to the client. This value limits only the size of the request, not that of the response. The value 0 disables this limit, but should be used with extreme caution. Without this limit, a rogue client could overload the server or cause it to run out of memory.
See Also	"MaxConnections communication parameter [MAXCONN]" on page 212

MyIP communication parameter [ME]

Usage	TCP/IP, HTTP, HTTPS
Values	String
Description	The MyIP (ME) communication parameter is provided for machines with more than one network adapter.
	Each adapter has an IP address. By default, the database server uses the first network card it finds. If you want your database server to use more than one network card, specify the address of each card in the MyIP (ME) communication parameter.
	If the keyword NONE is supplied as the IP number, no attempt is made to determine the addressing information. The NONE keyword is intended for clients on machines where this operation is expensive, such as machines with multiple network cards or remote access (RAS) software and a network card. It is not intended for use on the server.
	Under Windows 95/98/Me or Windows NT/2000/XP, this option can be used multiple times for machines with multiple IP addresses.
	Separate multiple IP addresses with commas. Optionally, you can append a port number to an IP address, separated from the IP number by a colon.
Example	• The following command line (entered all on one line) instructs the server to use two network cards, one with a specified port number.

dbsrv9 -x tcpip(MyIP=192.75.209.12:2367,192.75.209.32) "c:\ Program Files\Sybase\SQL Anywhere 9\asademo.db"

• The following connection string fragment instructs the client to make no attempt to determine addressing information.

LINKS= tcpip(MyIP=NONE)

PreFetchOnOpen communication parameter

Usage	ODBC
Values	YES, NO
Default	NO
Description	Enabling this option sends a prefetch request with a cursor open request, thereby eliminating a network request to fetch rows each time a cursor is opened. Columns must already be bound in order for the prefetch to occur on the open. Rebinding columns between the cursor open and the first fetch when using PrefetchOnOpen will cause reduced performance.
	Calling ODBC's SQLExecute or SQLExecDirect on a query or stored procedure which returns a result set causes a cursor open.
	Enabling this option can improve performance if your:
	 network exhibits poor latency
	• application sends many cursor open and close requests
ReceiveBufferSize	communication parameter [RCVBUFSZ]
Usage	TCP/IP
Values	Integer
Default	Machine-dependent
Description	Sets the size for a buffer used by the TCP/IP protocol stack. You may want to increase the value if blob performance over the network is important.

RegisterBindery communication parameter [REGBIN]

Usage	SPX (Windows 95/98/Me and Windows NT/2000/XP only).
	Server side only.
Values	YES, TRUE, 1

	NO, FALSE, 0
Default	TRUE
Description	The database server attempts to register its name with any active binderies on the network when loading the SPX link. To disable this name registration, set RegisterBindery to NO, FALSE, or 0. In this case, the client library must be able to locate the database server over SPX by broadcasting packets.

SearchBindery communication parameter [BINSEARCH]

Usage	SPX (Windows 95/98/Me and Windows NT/2000/XP only)
Values	YES, TRUE, 1
	NO, OFF, 0
Default	YES
Description	With SEARCHBINDERY=NO, 0, or OFF, no NetWare bindery is searched for a database server.

SendBufferSize communication parameter [SNDBUFSZ]

Usage	TCP/IP
Values	Integer
Default	Machine-dependent
Description	Sets the size for a buffer used by the TCP/IP protocol stack. You may want to increase the value if blob performance over the network is important.

ServerPort communication parameter [PORT]

Usage	TCP/IP (all platforms), HTTP, HTTPS
Values	Integer
Default	The default value for TCP/IP is 2638. The default value for HTTP is 80. The default value for HTTPS is 443.
Description	The Internet Assigned Numbers Authority has assigned the Adaptive Server Anywhere database server port number 2638 to use for TCP/IP communications. However, applications are not disallowed from using this reserved port, and this may result in an addressing collision between the database server and another application.
	In the case of the database server, the ServerPort option designates the port number on which to communicate using TCP/IP.

In a data source, the ServerPort option informs the client of the port or ports on which database servers are listening for TCP/IP communication. The client broadcasts to every port that is specified on the **ServerPort (PORT)** communication parameter to find the server.

The database server always listens on port 2638, even if you specify a different port using a network communication parameter. Hence, applications can connect to the database server without specifying a port number. An exception is the HP-UX operating system, on which the server does not listen on port 2638 if it is started on another port.

Connecting on HP/DEC Unix platforms

In Adaptive Server Anywhere version 9.0.0 and higher, the default port behavior for HP/DEC Unix platforms is equivalent to the way database servers worked on all platforms in Adaptive Server Anywhere versions 6.x.

In version 9.0.0 and higher, on HP/DEC Unix platforms *only*, if no port number is specified, the TCP port starts up on the default port (2638). If port 2638 is already in use by another application (either TCP or UDP), TCP does not start at all - the server does NOT automatically choose a different port to use.

If -x topip is specified, and the default port number is in use, then the server does not start. If the -x option is omitted entirely, the server still starts up, but TCP is unavailable.

By default, the database server listens on the standard HTTP and HTTPS ports of 80 and 443, respectively.

Example

1. Start a network database server:

dbsrv9 -x tcpip -n server1

Port number 2638 is now taken.

2. Attempt to start another database server:

dbsrv9 -x tcpip -n server2

The default port is currently allocated, and so the server starts on another port.

3. If another web server on your machine is already using port 80 or you do not have permission to start a server on this low a port number, you may wish to start a server that listens on an alternate port, such as 8080:

dbsrv9 -xs http(port=8080) -n server3 web.db

TDS communication parameter

Usage	TCP/IP, NamedPipes	
	Server side only	
Values	YES, NO	
Default	YES	
Description	To disallow TDS connections to a database server, set TDS to NO. If you want to ensure that only encrypted connections are made to your server, these port options are the only way to disallow TDS connections.	
Example	 The following command starts a database server using the TCP/IP protocol, but disallowing connections from Open Client or jConnect applications. 	
	dbsrv9 -x tcpip(TDS=NO)	

Timeout communication parameter [TO]

Usage	TCP/IP, SPX (Windows 95/98/Me, Windows NT/2000/XP, NetWare), HTTP, HTTPS
Values	Integer, in seconds
Default	5
Description	Timeout specifies the length of time, in seconds, to wait for a response when establishing communications. It also specifies the length of time to wait for a response when disconnecting. You may wish to try longer times if you are having trouble establishing TCP/IP or SPX communications.
	In HTTP or HTTPS applications, this parameter specifies the maximum idle time permitted when receiving a request. If this limit is reached, the connection is closed and a 408 REQUEST TIMEOUT returned to the client. The value 0 disables idle timeout, but should be used with extreme caution. Without this limit, a rogue client could consume the server's resources and prevent other clients from connecting.
Example	 The following data source fragment starts a TCP/IP communications link only, with a timeout period of twenty seconds. CommLinks=tcpip(TO=20)

VerifyServerName communication parameter [VERIFY]

Usage	TCP/IP
	Client side only
Values	YES, NO
Default	YES
Description	When connecting over TCP using the DoBraodcast=NONE parameter, the client makes a TCP connection, then verifies that the name of the server found is the same as the one it's looking for. Specifying VerifyServerName=NO skips the verification of the server name. This allows Adaptive Server Anywhere clients to connect to an Adaptive Server Anywhere server if they know only an IP address/port. The server name must still be specified in the connect string, but it is ignored. The VerifyServerName (VERIFY) communication parameter is used only if DoBroadcast=NONE is specified.
	Note We recommend that you only use this parameter in the rare circumstances when it is not possible to give each server a unique server name, and use that unique name to connect. Giving each server a unique server name, and
	connecting to the server using that name is still the best way to connect.
See also	"DoBroadcast communication parameter [DOBROAD]" on page 206

CHAPTER 7

Using the Built-in Web Server

Errors

About this chapter	Adaptive Server Anywhere contains a built-in web server you to write embedded applications that have a HTTP or addition, it allows you to communicate to built-in applica SOAP format.	This server allow: XML interface. In tions using the
Contents	Торіс:	page
	About web services	220
	Quick start	221
	Creating web services	222
	Starting a database server that listens for web requests	225
	How URLs are interpreted	227
	Procedures that handle web requests	229
	Working with variables	232
	Automatic character-set conversion	234

235

About web services

When started with HTTP or HTTPS services enabled, Adaptive Server Anywhere acts much like a traditional web server, except that all requests are processed using web services defined within a database. GET, HEAD, and POST methods are supported.

Web services are stored within the database. The collection of web services within your database together define the available URIs. Each service provides a set of web pages. Typically, the content of these pages is generated by procedures that you write and store in your database, although they can be a single statement or, optionally, allow the users to provide statements of their own.

You can customize the properties of web services using optional attributes to the statements that permit you to create and alter web services. For example, you can control who can use a web service by several means:

- You can force all clients to provide a valid user name and password.
- You can create services that are accessible only through secure connections.
- You can permit all database users access to your service, or limit access to a group of your choosing.

Standard SOAP requests are also supported.

Quick start

The following procedure describes how to create a new database, start an Adaptive Anywhere Server with the HTTP server enabled, and access this database using any popular web browsers.

* To set up a simple web server

- 1. Ensure that you have no web server already running on port 80, or choose a different port number to use for the rest of this procedure.
- 2. Navigate to a convenient directory and create a new database named web.

dbinit web.db

3. Start a personal database server to run this new database. The -xs switch tells the database server to start all communication protocols, including http.

dbeng9 web.db -xs http(port=80)

- 4. Start Interactive SQL and connect to your new database using the user id **DBA** and the password **SQL**.
- 5. Using Interactive SQL, execute the following SQL statement, which adds an HTML service that exposes the SYS.SYSTABLE to HTTP users. Because authorization is off, no permissions are required to access the table from a web browser.

```
CREATE SERVICE tables TYPE 'HTML'
AUTHORIZATION OFF USER DBA
AS SELECT * FROM SYS.SYSTABLE
```

6. Start your favorite web browser and enter the URL http://localhost:80/tables.

Many properties of the service are controlled by parameters to the -xs command-line switch.

For more information, see "Network communications parameters" on page 201.

As well as starting the database server with the appropriate -xs switch parameters, you must create web services to respond to incoming requests. These are defined using the CREATE SERVICE statement.

For more information, see "Creating web services" on page 222 and "CREATE SERVICE statement" [*ASA SQL Reference*, page 343].

Creating web services

You create web sites by creating services within databases. Services define which URIs are valid and what they do. A single database can define multiple services.

The facilities available through your Adaptive Anywhere web server are defined in databases. Services, created and stored in databases, define which URIs can be used and what they do. A single database can define multiple services. It is possible to define services in different databases so that they appear to be part of a single web site.

Three statements are provided that permit you to create, alter, and delete web services:

- ♦ CREATE SERVICE
- ♦ ALTER SERVICE
- ♦ DELETE SERVICE

The general syntax of the CREATE SERVICE statement is as follows:

CREATE SERVICE service-name T	YPE service-type-string [attributes] [AS
statement]	

Service names	As service names will form part of the URI used to access them, they are
	flexible in terms of what characters they can contain. In addition to the
	standard alpha-numeric characters, the following characters are permitted:
	"-", "_", "", "!", "*", "", "(", and ")".

In addition, service names may contain the slash, "/", but some restrictions apply because this character is a standard URI delimiter and affects how Adaptive Server Anywhere interprets your URIs. It cannot be the first character of a service name. In addition, service names cannot contain two consecutive slashes.

Service types The following service types supported:

- ♦ RAW The result set is sent to the client without any further formatting. You can produce formatted documents, in formats such as HTML or XML, by generating the required tags explicitly within your procedure.
- **HTML** The result set of a statement or procedure are automatically formatted into an HTML document that contains a table.
- XML The result set is assumed to be in XML format. If it is not already so, it is automatically converted to XML RAW format.

	• SOAP The request must be a valid Simple Object Access Protocol, or SOAP, request. The result set is automatically formatted as a SOAP response. For more information about the SOAP standards, see www.w3.org/TR/SOAP.
	• DISH A Deduce Ideal SOAP Handler, or DISH, service acts as a proxy for one or more SOAP services. In use, it acts as a container that holds and provides access to a number of soap services. A Web Services Description Language (WSDL) file is automatically generated for each of the included SOAP services. The included SOAP services are identified by a common prefix, which must be specified in the USING clause.
	Of the types HTML, XML, and RAW, RAW gives you the most control over the output. However, it does require that you do more work as you must explicitly output all the necessary tags. XML format can be tuned to some extent using an assortment of built-in procedures.
	For more information, see "CREATE SERVICE statement" [ASA SQL Reference, page 343].
Statements	The statement is the command, usually a stored procedure, that is called when someone access the service. If you define a statement, this is the only statement that can be run through this service. The statement is mandatory for SOAP services, and ignored for DISH services. The default is NULL, which means no statement.
	You can create services that do not include statements. The statement is taken from the URI. Services configured in this way can be useful when you are testing a service, or want a general way of accessing information. To do so, either omit the statement entirely or use the phrase AS NULL in place of the statement.
	Services without statements are a serious security risk, as they permit web clients to execute arbitrary commands. When creating such services, you must enable authorization, which forces all clients to provide a valid username and password. Even so, only services that define statements should be run in a production system.
Attributes	The following attributes are available. In general, all are optional. However, some are interdependent.
	♦ AUTHORIZATION Authorization controls which users can use the service. The default setting is ON. Authorization must be ON if no statement is provided. In addition, the authorization setting affects how user names are interpreted.
	• SECURE When set to ON, only secure connections are permitted. All connections received on the HTTP port are automatically redirected to

the HTTPS port. The default is OFF, which enables both HTTP and HTTPS requests, provided these ports are enabled using the appropriate switches when the database server is started.

• USER The USER clause controls which database user accounts can be used to process service requests. However, the interpretation of this setting depends on whether authorization is ON or OFF:

When authorization is set to ON, all clients must provide a valid user name and password when they connect. When authorization is ON, the USER option may either be NULL, or can be a list of user names. If it is NULL, any database user can connect make requests. Requests are run using the account and permissions of that user. If a list of users is specified, only those users can run requests. Any other database users are denied permission to use the service.

If authorization is OFF, a statement must be provided. In addition, a user name must be provided. All requests are run using that user's account and permissions. Thus, unless the server is physically isolated, the permissions of the named user account should be minimal.

• URL The URL clause controls the interpretation of URLs. In particular, it determines whether URI paths are accepted and, if so, how they are processed. If the service name ends with the character "/", the URL must be set to OFF.

For more information, see "CREATE SERVICE statement" [ASA SQL Reference, page 343].

Starting a database server that listens for web requests

	When you want a database server to listen for HTTP or HTTPS requests, you must specify which types of web requests it is to listen for on the command line. By default, database servers do not listen for web requests and there is no way for clients to access the services that may be defined in your database.
	You can also specify various properties of the services, such as on which port they are to listen. Doing so permits a the database server to act as a web server. It allows some or all database users to access web services that have been defined within the databases.
	You do not need to open a port for SOAP services. Unlike HTTP and HTTPS services, SOAP services become available the moment you create them. No additional command-line switches are required.
	You must also create web services within the database for clients. For more information, see "Creating web services" on page 222.
	You use the -xs command-line switch to enable web services. Optional additional parameters to this switch let you customize access to each type of service.
	The general syntax of the switch is as follows
	-xs protocol [(option=value,)]
Protocols	The following protocol values are available:
	• http Listen for HTTP connections.
	• https Listen for HTTPS connections.
	• all Start listening for web requests on both HTTP and HTTPS ports.
	• none Do not listen for web service requests. This setting is the default.
Options	The following are some of the options that are available:
	• ServerPort [PORT] The port on which to listen for web requests. By default, Adaptive Server Anywhere listens on port 80 for HTTP requests and on port 443 for secure HTTP (HTTPS) requests.
	For example, if you already have a web server running on port 80, you could use the following options to start the a database server that listens for web requests on port 8080:

dbeng9 web.db -xs http(port=8080)

◆ **DatabaseName [DBN]** Specifies the name of a database to use when processing web requests, or uses the REQUIRED or AUTO keyword to specify whether database names are required as part of the URI.

If this parameter is set to REQUIRED, the URI must specify the database name.

If this parameter is set to AUTO, the URI may specify a database name, but does not need to do so. If the URI contains no database name, the default database on the server is used to process web requests.

If this parameter is set to the name of a database, that database is used to process all web requests. The URI must not contain a database name.

- LocalOnly [LOCAL] When set to YES, this parameter causes the database server to reject all connections from clients running on the same machine. The default value is NO, which means accept from clients no matter where they are located.
- LogFile [LOG] The name of the file to which the database server is to write information about web requests.
- LogFormat [LF] Controls the format of messages written to the log file and which fields appear in them. If they appear in the string, current values are substituted for the codes, such as @T, when each message is written.

The default value is @T - @W - @I - @P - "@M @U @V" - @R - @L - @E, which produces messages such as the following:

06/15 01:30:08.114 - 0.686 - 127.0.0.1 - 80 - "GET /web/show_table HTTP/1.1" - 200 OK - 55133 -

• LogOptions [LOPT] Allows you specify keyword and error numbers that control which messages, or types of messages, are written to the log file.

For a complete list of the available options and detailed information about them, see "Network communications parameters" on page 201.

How URLs are interpreted

	The URIs used to access web services follow the pattern familiar to you from browsing the web. Users browsing through a database server need not be aware that their requests are not being handled by a traditional stand-alone web server.	
	Although they don't look unusual, URIs are necessarily interpreted a bit differently when sent to an Adaptive Server Anywhere database server. In addition, they are interpreted differently depending on which options you specified when you started the database server. The general syntax of the URL is as follows:	
	{ http https } ://host [:port] [/dbn] /service-name [parameters] [?query]	
Host and port	Like all standard http requests, the start of the URL contains the host name or IP number and, optionally, a port number. The IP address or host name, and port, should be the one on which your server is listening. The address will be the address of a network card in the machine running Adaptive Server Anywhere. The port number will the port number you specified using the -xs option when you started the database server. If you did not specify a port number, the default port number for that type of service is used. For example, the server listens by default on port 80 for HTTP requests.	
	For more information, see "Network communications parameters" on page 201.	
Database name	The next token, between the slashes, is usually the name of a database. This database must be running on the server and must contain web services.	
	The database name can be omitted only if the database server is running only one database, or if the database name was specified using the dbn parameter to the -xs switch.	
Service name	The next portion of the URL is the service name. This service must exist in the specified database. The service name may extend beyond the next slash character because web service names can contain slash characters. Adaptive Server Anywhere matches the remainder of the URL with the defined services.	
	If the URL provides no service name, the database server looks for a service named root. If the named service, or the root service, are not defined, the server returns a 404 Not Found error.	
Parameters	Depending on the type of the target service, parameters are supplied can be supplied in different ways. Parameters to HTML, XML, and RAW services can be passed in any of the following ways:	

	• appended to the URL	
	• supplied as an explicit parameters list	
	 supplied as POST data in a POST request 	
	Parameters to SOAP and DISH services must be included as part of a standard SOAP request. Values supplied in other ways are ignored.	
	To access parameter values, parameters must be given names. These host variable names, prefixed with a colon (:) may be included in the statement that forms part of the web service definition.	
	For example, a statement that calls the procedure place_order may require product identification number and size parameters:	
	call place_order(:product, :quantity)	
Query	The query is a list of name = value pairs separated by the & character. This list is introduced by a question mark. GET requests are formatted in this manner. If present, the named variables are defined and assigned the corresponding values.	
	If you have URL PATH set to ON or ELEMENTS, additional variables are defined. However, the two are otherwise independent. You can define variables using a query, by setting PATH to ON or ELEMENTS, or both.	
	For more information about variables, see "Working with variables" on page 232.	

Procedures that handle web requests

Generally, it is easiest to write a procedure that handles the requests sent to a particular service. In general, your procedure should return a web page. Optionally your procedure can accept arguments, passed as part of the URI, to customize its output.

The following example, however, is much simpler. It demonstrates how simple a service can be. This web service simply returns the phrase "Hello world!".

CREATE SERVICE hello TYPE 'RAW' AUTHORIZATION OFF USER DBA AS select 'Hello world!'

Start a server with the -xs switch to enable handling of web requests, then request the URL http://localhost/hello from any web browser. The words Hello world will appear in an otherwise plain page.

HTML pages The above page is displayed in your browser in plain text. This happens because the default HTTP Content-Type is text/plain. To create a more normal web page, formatted in HTML, you must do two things:

- Set the http Content-Type header field to text/html so that the browsers will expect HTML.
- Include HTML tags in the output.

You can write tags to the output in two ways. One way is to use the phrase TYPE HTML in the CREATE SERVICE statement. Doing so asks Adaptive Server Anywhere to add HTML tags for you. This can work if quite well if, for example, you are returning a table.

The other way is to use TYPE RAW and write out all the necessary tags yourself. This second method provides the most control over the output. Note that specifying type RAW does not necessarily mean the output is not in HTML or XML format. It only tells Adaptive Server Anywhere that it can pass the return value directly to the client without adding tags itself.

The following procedure generates a fancier version of Hello world. For convenience the body of the work is done in the following procedure, which formats the web page.

The built-in procedure sa_set_http_header is used to set the HTTP header type, so browsers will correctly interpret the result. If you omit this statement, your browser will display all the HTML codes verbatim, rather than use them to format the document.

```
CREATE PROCEDURE hello pretty world ()
  RESULT (html_doc long varchar)
   BEGIN
     call dbo.sa_set_http_header( 'Content-Type', 'text/html'
        );
      select '<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML
         4.01//EN">\n'
      // '<html>\n'
         || '<head>\n'
      // ' <title>Hello Pretty World</title>\n'
      || '</head>\n'
          || '<body>\n'
          // ' <h1>Hello Pretty World!</h1>\n'
      || ' (If you see the tags in your browser, check that\
        n′
      1 1
                the Content-Type header is set to text/html.)\n'
          || '</body>\n'
          // '</html>';
   END
```

The following statement creates a service that uses this procedure. The statement calls the above procedure, which generates the Pretty World web page.

```
CREATE SERVICE hello_pretty_world TYPE 'RAW'
AUTHORIZATION OFF USER DBA
AS call hello_pretty_world()
```

Once you have created the procedure and the service, you are ready to access the web page. Ensure that your database server was started with the -xs switch, which causes it to listen for web requests, then browse to the URL http://localhost/hello_pretty_world.

You see the results formatted in a simple HTML page, with the title Hello Pretty World. You can make the web page as fancy as you wish by including more content, using more tags, using style sheets, or including scripts that run in the browser. In all cases, you create the necessary services to handle the browser's requests.

For more information about built-in stored procedures, see "System and catalog stored procedures" [*ASA SQL Reference*, page 707].

Root services When no service name is included a URI, Adaptive Server Anywhere looks for a web service named root. The role of root pages is analogous to the role of index.html pages in many traditional web servers.

Root services are handy for creating home pages because they can handle URL requests that contain only the address of your web site. For example, the following procedure and service implement a simple web page that is displayed when you browse to the URL http://localhost.

```
CREATE PROCEDURE home page
    RESULT (html_doc long varchar)
    BEGIN
        call dbo.sa_set_http_header( 'Content-Type', 'text/html'
           );
        select '<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML
           4.01//EN">\n'
        // '<html>\n'
           || '<head>\n'
        || ' ' <title>My Home Page</title>\n'
        // '</head>\n'
            || '<body>\n'
            || ' <hl>My home on the web<hl>\n'
        // ' Thank you for visiting my web site!\n'
            || '</body>\n'
            // '</html>';
    END
 CREATE SERVICE root TYPE 'RAW'
       AUTHORIZATION OFF USER DBA
       AS call home_page()
You can access this web page by browsing to the URL http://localhost, as
```

long as you do not specify that database names are mandatory when you start the database server.

Examples For example, the request in the Quick Start example includes no database name and no parameters. In this request, "/tables" identifies the service.

```
http://localhost:80/tables
```

More extensive examples are included in the *Samples/HTTP* subdirectory of your SQL Anywhere Studio installation.

Working with variables

Variables in HTTP requests come from one of two sources. First, the URI may include a query, which includes various name-value pairs. Get requests are formatted in this manner.

The second way is through the URI path. Setting the URL PATH to either ON or ELEMENTS causes the portion of the path, following the service name, to be interpreted as variable values. This option allows URLs to appear to be requesting a file in a particular directory, as would be the case on a traditional file-based web site, rather than something stored inside a database.

For example, the URL http://localhost/gallery/sunset.jpg appears to request the file sunset.jpg from a directory named gallery, but instead asks the gallery service to retrieve a picture from a database table.

Parameter passed in HTTP requests depends on the setting of URL PATH.

- **OFF** No path parameters are permitted after the service name.
- **ON** All path elements after the service name are assigned to the variable url.
- **ELEMENTS** The remainder of the URI path is to be split at the slash characters into a list of up to 10 elements. These values are assigned the the variables url1, url2, url3, ..., url10. If there are fewer than 10 values, the remaining variables are set to NULL. More than ten variables causes an error.

Apart from the location in which they are defined, their is no difference between variables. You access and use all HTTP variables the same way. For example, the values of variables such as url1 are accessed in the same way as paramters that appear as part of a query, such as **?picture=sunset.jpg**.

Accessing parameters There are main ways to access variables. The first is to mention variables in the statement of the service declaration. For example, the following statement passes the value of multiple variables to the show_table stored procedure:

CREATE SERVICE show_table TYPE 'RAW' AUTHORIZATION ON AS CALL show_table(:user_name, :table_name, :limit, :start)

The other way is to use the built in functions next_http_variable and http_variable within the stored procedure that handles the request. If you do not know which variables are defined, you can use the next_http_variable to find out. The http_variable function returns the variable values.

The next_http_variable function allows you to iterate through the names of the defined variables. The first time you call it, you pass in the NULL value. This returns the name of one variable. Calling it subsequent times, each time passing in the name of the prevous variable, returns the next variable name. When the name of the last variable is passed to this function, it returns NULL.

Iterating through the variable names in this manner guarantees that each variable name will be returned exactly once. However, the order that the values are returned may not be related to how they appear in the request. The order may not be consistent. For example, if you iterate through the names a second time, they may be returned in a different order.

To get the value of each variable, use the http_variable function. The first parameter is the name of the variable. Additional parameters are optional. In the case that multiple values were supplied for a variable, the function returns the first value if supplied with only one parameter. Supplying an integer as the second parameter allows you to retrieve additional values.

The third parameter allows you to retrieve variable header-field values from multi-part requests. Supply the name of a header field to retrieve its value. For example, the following lines of SQL retrieve three variable values, then retrieve the header-field values of the image variable.

```
SET v_id = http_variable( 'id' );
SET v_title = http_variable( 'title' );
SET v_descr = http_variable( 'descr' );
SET v_name = http_variable( 'image', NULL, 'Content-Disposition'
        );
SET v_type = http_variable( 'image', NULL, 'Content-Type' );
SET v_image = http_variable( 'image', NULL, '@BINARY' );
```

Automatic character-set conversion

By default, character set conversion is performed automatically on outgoing result sets of type text. Result sets of other types, such as binary objects, are not translated. The character set of the request is converted to the database character set, and the result set is converted from the database character set to the client character set, as required, except on binary columns in the result set. When the request lists multiple character sets that it can handle, the engine takes the first suitable one from the list.

Character-set conversion can be enabled or disabled by setting the HTTP option CharsetConversion. The allowed values are ON and OFF. The default value is ON. The following statement turns automatic character-set conversion off:

```
call dbo.sa_set_http_option( 'CharsetConversion', 'OFF' )
```

For more information about built-in stored procedures, see "System and catalog stored procedures" [*ASA SQL Reference*, page 707].

Errors

When web service requests fail, the database server generates standard errors that appear in your browser. These errors are assigned numbers consistent with the protocol standards.

The following are some of the more typical errors that you may encounter:

Number	Name	Description
400	Bad Request	The HTTP request is incomplete or mal- formed.
401	Authorization Re- quired	Authorization is required to use the service, but a valid user name and password were not supplied.
403	Forbidden	You do not have permission to access the database.
404	Not Found	The named database is not running on the server, or the named web service does not exist.
408	Request Timeout	The maximum connection idle time was exceeded while receiving the request.
413	Entity Too Large	The request exceeds the maximum permitted size.
414	URI Too Large	The length of the URI exceeds the maximum allowed length.
500	Internal Server Error	An internal error occurred. The request could not be processed.
501	Not Implemented	The HTTP request method is not GET, HEAD, or POST.
503	Service Unavail- able	The number of connections exceeds the allowed maximum.
Part II

WORKING WITH DATABASE FILES

This section describes how to install, use, and backup database files. It also describes how to use international language and character sets.

CHAPTER 8

File Locations and Installation Settings

About this chapter	This chapter describes the installation and operating s Adaptive Server Anywhere. Depending on the operat settings may be stored as environment variables, initia registry settings.	system settings used by ing system, these alization file entries, or
Contents	Торіс:	page
	Installation directory structure	240
	How Adaptive Server Anywhere locates files	242
	Environment variables	244
	Registry and INI files	249

Installation directory structure

When you install Adaptive Server Anywhere, several directories may be created. Some of the files in these directories are essential, and others are not. This section describes the directory structure.

Adaptive Server Anywhere software, whether you receive it as a product or bundled as part of another product, is installed under a single installation directory. The tools provided with the Adaptive Server Anywhere product, however, are installed in other directories. This section describes only the installation directory structure for Adaptive Server Anywhere itself.

The Adaptive Server Anywhere installation directory itself holds several items, including the following:

- **The sample database** The sample database is held in the file *asademo.db*.
- **Read Me First** A Read Me First file named *readme.txt* holds late-breaking information.

For platforms other than Novell NetWare and Windows CE, there are several directories under the installation directory:

• **Executable directories** There is a separate directory for each operating system, which holds executables, dynamic link libraries, and help files.

On Windows except Windows CE, these files are installed in the *win32* or *win64* directory. If you are using UNIX, they are installed in the *bin* directory. On NetWare, the executables are stored in the installation directory itself.

You will not have all these directories on your machine; you will have only the ones required for the operating system version you installed.

- ♦ Java directory Java base classes are stored in this directory.
- **ProcDebug directory** The stored procedure debugger is stored in this directory.
- ◆ Scripts directory The scripts directory contains SQL scripts that are used by the database administration utilities and as examples. With the exception the *custom.SQL* script, *do not edit these scripts*. If the scripts directory is not present, the administration utilities will not work.
- **Samples directories** This holds separate directories for a variety of samples.
- h directory The h directory contains header files for ESQL and ODBC database development.

The Adaptive Server Anywhere installation directory

Novell NetWare file locations	On Novell NetWare, all files are installed to a single directory on the server. Throughout this documentation, when reference is made to files in subdirectories of the installation directory, the file on NetWare is in the installation directory itself.
Windows CE file locations	On Windows CE, all files are installed to the installation directory, and no subdirectories are created. The exception is that all DLLs are installed into the \ <i>Windows</i> directory.

How Adaptive Server Anywhere locates files

The client library and the database server need to locate files for two main purposes:

- DLLs and initialization files are required to run Adaptive Server Anywhere. If an incorrect DLL is located, there is the possibility of version mismatch errors.
- Some files are specified in SQL statements and need to be located at run time, such as INSTALL or LOAD TABLE.

Examples of SQL statements that use file names include the following:

- INSTALL JAVA statement The name of the file that holds Java classes.
- LOAD TABLE and UNLOAD TABLE statements The name of the file from which data should be loaded or to which the data should be unloaded.
- **CREATE DATABASE statement** A file name is needed for this statement and similar statements that can create files (such as CREATE WRITEFILE).

In some cases, Adaptive Server Anywhere uses a simple algorithm to locate files. In other cases, a more extensive search is carried out.

Simple file searching In many SQL statements (such as LOAD TABLE, or CREATE DATABASE), the file name is interpreted as relative to the current working directory of the database server.

Also, when a database server is started and a database file name (**DatabaseFile (DBF**) parameter) is supplied, the path is interpreted as relative to the current working directory.

Extensive file searching Adaptive Server Anywhere programs, including the database server and administration utilities, carry out a more extensive search for required files, such as DLLs or shared libraries. In these cases, Adaptive Server Anywhere programs look for files in the following order:

- 1. **Executable directory** Holds the program executable file.
- 2. **Related directories** Holds directories with the following paths relative to the program executable directory:
 - Parent of the executable directory.
 - A child of the parent directory named *scripts*. The UNIX server does not search in this location.

- 3. **Current working directory** When a program is started, it has a current working directory (the directory from which it is started). This directory is searched for required files.
- 4. Location registry entry When installing onto Windows, Adaptive Server Anywhere adds a LOCATION registry entry. The indicated directory is searched, followed by:
 - ♦ A child named scripts
 - A child with the operating system name (*win32*, *win64*, and so on)
- 5. **System specific directories** This includes directories where common operating system files are held, such as the *Windows* directory and the *Windows**system* directory on Windows operating systems.
- 6. **CLASSPATH directories** For Java files, directories listed in the CLASSPATH environment variable are searched to locate files.
- 7. **PATH directories** Directories in the system path and the user's path are searched to locate files.

Environment variables

Adaptive Server Anywhere uses a set of environment variables to store various types of information. These environment variables are listed in this section. Not all variables need to be set in all circumstances.

Setting environment variables

The way you set an environment variable depends on the operating system you are using.

To set an environment variable (Windows NT)

- 1. Right-click My Computer and choose Properties from the popup menu.
- 2. Click the Environment tab. If the environment variable does not already exist, type variable and its value in the spaces provided, and click Set.

If the variable does exist, select it from the list of System Variables or User Variables, and make any modifications in the Value field. Click Set to make the setting.

✤ To set an environment variable (Windows 2000)

- 1. Right-click My Computer and choose Properties from the popup menu.
- 2. Click the Advanced tab.
- 3. Click the Environment Variables button. The Environment Variables dialog opens. If the environment variable does not already exist, click New and type the variable name and its value in the spaces provided, and then click OK.

If the variable does exist, select it from the list of System Variables or User Variables, click Edit and make any modifications in the Variable Value field. Click OK to make the setting.

To set an environment variable (UNIX)

1. In one of your startup files (*.cshrc*, *.shrc*, *.login*), add a line that sets the variable.

In some shells (such as sh, bash, or ksh) the line is as follows:

export VARIABLE=value

In other shells (such as csh, or tsch) the line is as follows:

setenv VARIABLE value

ASANY9 environment variable

Syntax	ASANY9=directory-name
Default	C:\Program Files\Sybase\SQL Anywhere 9
Description	On installation, the ASANY9 environment variable is set to your SQL Anywhere directory.
	This environment variable must be set for several reasons. Among the list, samples require this environment variable in order to locate SQL Anywhere applications.

ASANYSH9 environment variable

Syntax	ASANYSH9=directory-name
Default	None.
Description	Interactive SQL, Sybase Central, the debugger, and the Console utility use this variable to determine the location of the shared components directory. ASANYSH9 is set to the location of the shared directory. You set the shared directory location during installation. The default shared components directory is <i>C</i> :\ <i>Program Files</i> \ <i>Sybase</i> \ <i>Shared</i> .

ASLANG environment variable

Syntax	ASLANG=language_code
Description	<i>Language_code</i> is the two-letter combination representing a language. For example, setting ASLANG=DE sets the default language to German.
	The first of the following values set determines the default language.
	ASLANG environment variable
	• registry (Windows only) as set by the installer or dblang.exe
	• query the operating system
	If no language information is set, English is the default.
	This environment variable is not supported on CE or NetWare
ASCHARSET envir	onment variable
Syntax	ASCHARSET=charset
Description	<i>Charset</i> is a character set name. For example, setting ASCHARSET=cp1252 sets the default character set to cp1252.

The first of the following values set determines the default character set.

- ♦ ASCHARSET environment variable
- query the operating system

If no character set information is specified, use iso_1 for Unix, or cp850 otherwise.

This environment variable is not supported on CE or NetWare

ASTMP environment variable

Syntax	ASTMP=directory-name
Default	None.
Description	The database server checks the value of the ASTMP environment variable to determine where to hold the temporary file. If the ASTMP environment variable does not exist, then the first of the TMP, TMPDIR, and TEMP environment variables to exist is used on Windows. If the ASTMP environment variable does not exist on UNIX, <i>/tmp</i> is used.
	In many circumstances, ASTMP is not needed. It can be of use in security-conscious environments when running the database server as a service to enable you to hold the temporary file in a directory that cannot be accessed by other programs.
LD_LIBRARY_PAT	H environment variable [UNIX]

Description The LD_LIBRARY_PATH environment variable is used on UNIX only. It is modified by the installation program to include the directories where Adaptive Server Anywhere libraries are located.

LD_LIBRARY_PATH=installation_path/lib

The libraries are located in the *lib* subdirectory of the installation directory (for example, */opt/SYBSasa9/lib*).

PATH environment variable

Syntax	PATH=installation_path
Description	The PATH environment variable is modified by the installation program to include the directories where Adaptive Server Anywhere executables are located.

The executables are located in a subdirectory of the installation directory.

Syntax

In addition, if you are using other Sybase applications, the *SYBASE\bin* and *SYBASE\dll* directories are added to your path.

On UNIX, each user must have the directory holding the executables (*/opt/SYBSasa9/bin*) added to their path.

SQLCONNECT environment variable

Syntax SQLCONNECT=parameter#value; ...

Description

The SQLCONNECT environment variable is optional, and is not set by the installation program.

SQLCONNECT specifies connection parameters that are used by several of the database administration utilities when connecting to a database server. This string is a list of parameter settings, of the form **parameter**=*value*, delimited by semicolons.

The number sign "#" is an alternative to the equal sign, and should be used if you are setting the connection parameters string in the SQLCONNECT environment variable. Using "=" inside an environment variable setting is a syntax error. The = sign is allowed only in Windows NT/2000/XP.

For a description of the connection parameters, see "Connection parameters" on page 70.

SQLLOCALE environment variable

The SQLLOCALE environment variable is no longer supported. It has been replaced by the ASLANG and ASCHARSET environment variables.

SQLPATH environment variable

Syntax	SQLPATH=path;
Description	The SQLPATH environment variable is optional, and is not set by the installation program.
	Interactive SQL searches along SQLPATH for command files and Help files before searching the system path.

SQLREMOTE environment variable

Syntax	SQLREMOTE=path
Description	The SQLREMOTE environment variable is optional, and is not set by the
	installation program.

Addresses for the FILE message link in SQL Remote are subdirectories of the SQLREMOTE environment variable. This variable should point to a shared directory.

On 32-bit Windows operating systems, an alternative to setting the SQLREMOTE environment variable is to set the *SQL Remote\Directory* registry entry to the proper root directory.

SYBASE environment variable

Syntax	SYBASE=path
Description	The SYBASE variable marks the home directory for installation of some
	Sybase applications, including Adaptive Server Enterprise and utilities such as DSE dit. You need this variable only if you are using Adaptive Server
	Anywhere together with other members of the Adaptive Server family.

TEMP environment variable

ASTMP=path
TMP=path
TMPDIR=path
TEMP=path
The database server creates a temporary file for various operations such as sorting and performing unions. Temporary files are placed in the directory specified by the ASTMP, TMP, TMPDIR, or TEMP environment variable. Adaptive Server Anywhere takes the first one of these that it finds, in that order.
If none of the environment variables is defined, temporary files are placed in the current working directory of the server.
On UNIX, only the ASTMP variable is used. If the ASTMP variable does not exist, files are placed in the /tmp/.SQLAnywhere directory.
On Windows CE, you can specify which directory you want to use as the server's temporary directory, in the registry.
For more information about setting the temporary directory value, see "Registry settings on Windows CE" on page 250

Registry and INI files

On Windows operating systems (except Windows CE), Adaptive Server Anywhere uses several registry settings. On UNIX, and NetWare, these settings are held in initialization files instead.

The software makes these settings for you, and in general operation you should not need to access the registry. The settings are provided here for those people who make modifications to their operating environment.

Current user and local machine settings

	Some operating systems, such as Windows NT, hold two levels of system settings. Some settings are specific to an individual user and are used only when that user is logged on; these settings are called current user settings. Some settings are global to the machine, and are available no matter which user is logged on; these are called local machine settings. You must have administrator permissions on your machine to make local machine settings.
	Adaptive Server Anywhere permits both current user and local machine settings. On Windows NT, for example, these are held in the HKEY_CURRENT_USER registry and the HKEY_LOCAL_MACHINE registry, respectively.
Current user takes precedence	If a setting is made in both the current user and local machine registries, the current user setting takes precedence over the local machine setting.
When local machine settings are needed	If you are running an Adaptive Server Anywhere program as a service , you should ensure that the settings are made at the <i>local machine</i> level.
	Services can continue to run under a special account when you log off a machine as long as you do not shut the machine down entirely. They can be made independent of individual accounts, and therefore need access to local machine settings.
	In addition to Adaptive Server Anywhere programs, some Web servers run as services. You must set local machine settings in order for PowerDynamo to work with such a Web server.
	In general, the use of local machine settings is recommended.

Registry structure

On Windows (except for Windows CE), you can access the registry directly with the registry editor. The Adaptive Server Anywhere registry entries are held in either the HKEY_CURRENT_USER or HKEY_LOCAL_MACHINE registries, in the following location:

```
Software
Sybase
Adaptive Server Anywhere
9.0
Sybase Central
4.0
Profiles
Providers
```

Registry settings on installation

The installation program makes the following registry settings in the Sybase registry:

• Location In the Adaptive Server Anywhere 9.0 registry, this entry holds the installation directory location. For example:

Location "c:\Program Files\Sybase\SQL Anywhere 9"

◆ Language In the Adaptive Server Anywhere 9.0 registry, this entry holds a two-letter code indicating the current language for messages and errors. For example:

Language "EN"

The default setting is English (EN). The installation program sets this entry only if the software is installed for a language other than English.

Registry settings on Windows CE

You can specify which directory you want to use as the server's temporary directory on Windows CE by setting the

HKEY_CURRENT_USER\Software\Sybase\Adaptive Server Anywhere\9.0\TempFolder

value in the registry, where *TempFolder* is the name of the temporary directory you want to use. The server will either:

- use the specified directory if it exists, or
- attempt to create the specified directory if it does not already exist, as long as the parent directory already exists.

If the specified directory does not exist and cannot be created, the server will:

- use the \Temp directory if it exists, or
- attempt to create a \Temp directory if it does not already exist.

If the \Temp directory does not exist and cannot be created, the server will:

• use the current directory.

CHAPTER 9

Working with Database Files

About this Chapter	This chapter describes how to create, and wo files.	rk with database and associated
Contents	Торіс:	page
	Overview of database files	254
	Using additional dbspaces	256
	Working with write files	260
	Using the utility database	262

Overview of database files

Basic database files

Each database has the following files associated with it.

• **The database file** This file holds the database information. It typically has the extension *.db*.

For information on creating databases, see "Working with databases" [*ASA SQL User's Guide*, page 27].

• **The transaction log** This file holds a record of the changes made to the database, and is necessary for recovery and replication. It typically has the extension *.log.*

For information on the transaction log, see "Backup and Data Recovery" on page 337.

• The temporary file The database server uses the temporary file to hold information needed during a database session. The database server disregards this file once the database shuts down—even if the server remains running. The file has a server-generated name with the extension *.tmp*.

The temporary file is held in a location determined by an environment variable.

The following environment variables are checked, in order:

- ASTMP
- TMP
- TMPDIR
- TEMP

If none of these is defined, Adaptive Server Anywhere places its temporary file in the current directory on Windows operating systems, or in the */tmp/.SQLAnywhere* directory on UNIX.

The server creates, maintains, and removes the temporary file. You only need to ensure that there is enough free space available for the temporary file.

Additional files Other files can also become part of a database system, including:

♦ Additional database files You can spread your data over several separate files. These additional files are called dbspaces.

For information on dbspaces, see "CREATE DBSPACE statement" [*ASA SQL Reference*, page 297].

 Transaction log mirror files For additional security, you can create a mirror copy of the transaction log. This file typically has the extension .mlg.

For information on mirrored transaction logs, see "Transaction log mirrors" on page 344.

- Write files If the database file you are working with is designated read-only (for example, because it is distributed on CD-ROM), you can use an additional write file to hold changes you make to the data. However, note that you can only use write files with read-only database files, not with a server running in read-only mode. Running the server in read-only mode allows no changes to the database file whatsoever.
- **Compressed database files** You can compress a database file. The resulting file is read only, but can be used in conjunction with a write file. Compressed database files are used in place of the actual database file.

For information on compression, see "The Compression utility" on page 468.

Using additional dbspaces

This section describes how to use additional database files, known as dbspaces.

Typically needed for large databases

For most databases, a single database file is sufficient. However, for users of large databases, additional database files are often necessary. Additional database files are also convenient tools for clustering related information in separate files.

When you initialize a database, it contains one database file. This first database file is called the **main file**. All database objects and all data are placed, by default, in the main file.

Each database file has a maximum allowable size of 256M database pages. For example, a database file created with a database page size of 4 kb can grow to a maximum size of one terabyte (256M*4 kb). However, in practice, the maximum file size allowed by the physical file system in which the file is created affects the maximum allowable size significantly.

While many commonly-employed file systems restrict file size to a maximum of 2 Gb, some, such as the Windows NT/2000/XP file system, allow you to exploit the full database file size. In scenarios where the amount of data placed in the database exceeds the maximum file size, it is necessary to divide the data into more than one database file. As well, you may wish to create multiple dbspaces for reasons other than size limitations, for example to cluster related objects.

Splitting existingIf you wish to split existing database objects among several dbspaces, youdatabasesneed to unload your database and modify the generated command file for
rebuilding the database. To do so, add IN clauses to specify the dbspace for
each table you do not wish to place in the main file.

For more information, see "UNLOAD TABLE statement" [*ASA SQL Reference*, page 590] and "Setting properties for database objects" [*ASA SQL User's Guide*, page 32].

Creating a dbspace

You create a new database file, or **dbspace**, either from Sybase Central, or using the CREATE DBSPACE statement. The database file for a new dbspace may be on the same disk drive as the main file or on another disk drive. You must have DBA authority to create dbspaces.

For each database, you can create up to twelve dbspaces in addition to the

	main dbspace.
Placing tables in dbspaces	A newly created dbspace is empty. When you create a new table you can place it in a specific dbspace with an IN clause in the CREATE TABLE statement. If you don't specify an IN clause, the table appears in the main dbspace.
	Each table is entirely contained in the dbspace it is created in. By default, indexes appear in the same dbspace as their table, but you can place them in a separate dbspace by supplying an IN clause.
	 To create a dbspace (Sybase Central)
	1. Connect to the database.
	2. Open the Dbspaces folder for that database.
	3. Click the New Dbspace button on the toolbar.
	4. Follow the instructions in the wizard.
	The new dbspace then appears in the Dbspaces folder.
	To create a dbspace (SQL)
	1. Connect to the database.
	2. Execute a CREATE DBSPACE statement.
Examples	The following command creates a new dbspace called library in the file <i>library.db</i> in the same directory as the main file:
	CREATE DBSPACE library AS 'library.db'
	The following command creates a table LibraryBooks and places it in the library dbspace.
	CREATE TABLE LibraryBooks (title char(100), author char(50), isbn char(30)) IN library
	The also
	• "CREATE DBSPACE statement" [ASA SQL Reference, page 297]
	• "Creating tables" [ASA SQL User's Guide, page 37]

• "CREATE INDEX statement" [ASA SQL Reference, page 319]

Deleting a dbspace

You can delete a dbspace using either Sybase Central or Interactive SQL. Before you can delete a space, you must delete all tables that use the space. You must have DBA authority to delete a dbspace.

To delete a dbspace (Sybase Central)

- 1. Open the Dbspaces folder.
- 2. Right-click the desired dbspace and choose Delete from the popup menu.

To delete a dbspace (SQL)

- 1. Connect to a database.
- 2. Execute a DROP DBSPACE statement.
- See also
- "Deleting tables" [ASA SQL User's Guide, page 41]
- "DROP statement" [ASA SQL Reference, page 408]

Pre-allocating space for database files

Adaptive Server Anywhere automatically grows database files as needed. Rapidly changing database files can lead to excessive file fragmentation on the disk, resulting in potential performance problems. Unless you are working with a database with a high rate of change, you do not need to worry about explicitly allocating space for database files. If you are working with a database with a high rate of change, you may pre-allocate disk space for dbspaces or for transaction logs using either Sybase Central or the ALTER DBSPACE statement.

You must have DBA authority to alter the properties of a database file.

Performance Tip

Running a disk defragmentation utility after pre-allocating disk space helps ensure that the database file is not fragmented over many disjoint areas of the disk drive. Performance can suffer if there is excessive fragmentation of database files.

	To pre-allocate space (Sybase Central)
	1. Open the Dbspaces folder.
	2. Right-click the desired dbspace and choose Pre-allocate Space from the popup menu.
	3. Enter the number of pages to add to the dbspace and click OK.
	To pre-allocate space (SQL)
	1. Connect to a database.
	2. Execute an ALTER DBSPACE statement.
Example	Increase the size of the SYSTEM dbspace by 200 pages.
	ALTER DBSPACE system ADD 200
Example	Increase the size of the SYSTEM dbspace by 400 megabytes.
	ALTER DBSPACE system ADD 400 MB
	The also
	• "Creating a dbspace" on page 256

• "ALTER DBSPACE statement" [ASA SQL Reference, page 229]

Working with write files

If you have a read-only database file (for example, if you distribute a database on a CD-ROM), you can use a write file to make local changes to the database.

You create a write file using the Create Write File utility or using the CREATE WRITEFILE statement. In this section, the examples use the utility.

For a description of the CREATE WRITEFILE statement, see "CREATE WRITEFILE statement" [*ASA SQL Reference*, page 384].

For more information about opening a database as read-only to prevent local changes to the database, see "-r server option" on page 157.

* To use a write file

1. Create the write file for your database.

For example, to create a write file for the demo database, execute the following command in a directory containing a copy of the demo database file *asademo.db*:

dbwrite -c asademo.db

This command creates a write file named *asademo.wrt*, with a transaction log named *asademo.wlg*.

2. Start a database server and load the write file. By default, the server locates files with the extension *.wrt* first, so the following command starts the personal server running the demo database write file:

dbeng9 asademo

Messages on the server window indicate which file starts.

- 3. Connect to the database using Interactive SQL. You can use the user ID **DBA** and the password **SQL**, as the demo database is the default.
- 4. Execute queries as usual. For example, the following query lists the contents of the department table.

SELECT * FROM department

The data for the department table is obtained from the database file *asademo.db*.

5. Try inserting a row. The following statement inserts a row into the department table:

```
INSERT
                               INTO department (dept_id, dept_name)
                               VALUES (202, 'Eastern Sales')
                             If you committed this change, it would be written to the asademo.wlg
                             transaction log, and when the database checkpoints, the changes are
                             written to the asademo.wrt write file.
                             If you now query the department table, the results come from the write
                             file and the database file.
                          6. Try deleting a row. Set the WAIT_FOR_COMMIT option to avoid
                             referential integrity complaints here:
                               SET TEMPORARY OPTION wait_for_commit = 'on' ;
                               DELETE
                               FROM department
                               WHERE dept_id = 100
                             If you committed this change, the deletion would be marked in the write
                             file. No changes occur to the database file.
                          For some purposes, it is useful to use a write file with a shared database. If,
                          for example, you have a read-only database on a network server, each user
                          could have their own write file. In this way, they could add local information,
                          which would be stored on their own machine, without affecting the shared
                          database. This approach can be useful for application development also.
Deleting a write file
                          You can use the dberase utility to delete a write file and its associated
                          transaction log.
```

Using the utility database

The **utility database** is a phantom database with no physical representation. The utility database has no database file, and therefore it cannot contain data.

The utility database feature allows you to execute database file administration statements such as CREATE DATABASE, or ALTER WRITEFILE without first connecting to an existing physical database.

For example, executing the following statement after having connected to the utility database creates a database named *new.db* in the directory *C*:*temp*.

```
CREATE DATABASE 'C:\\temp\\new.db'
```

For more information on the syntax of these statements, see "CREATE DATABASE statement" [ASA SQL Reference, page 292].

You can also retrieve values of connection properties and database properties using the utility database.

For example, executing the following statement against the utility database returns the default collation sequence, which will be used when creating database:

SELECT property('DefaultCollation')

For a list of database properties and connection properties, see "Database properties" on page 647.

Allowed statements for the utility database the utility database:

- ♦ CREATE DATABASE
- ♦ CREATE WRITEFILE
- ♦ ALTER WRITEFILE
- ♦ CREATE COMPRESSED DATABASE
- ♦ CREATE EXPANDED DATABASE
- ♦ CREATE DECRYPTED FILE
- ♦ DROP DATABASE
- RESTORE DATABASE
- ♦ START DATABASE
- ♦ STOP DATABASE

- ♦ STOP ENGINE
- SELECT (with no FROM or WHERE clauses)

Connecting to the utility database

You can start the utility database on a database server by specifying **utility_db** as the database name when connecting to the server. The user ID and password requirements are different for the personal server and the network server.

For the personal database server, there are no security restrictions for connecting to the utility database. It is assumed that anybody who can connect to the personal database server can access the file system directly, and so no attempt is made to screen users based on passwords.

For more information, see "Utility database passwords" on page 264.

To connect to the utility database on the personal server (Interactive SQL)

1. Start a database server with the following command:

dbeng9.exe -n TestEng

- 2. Start Interactive SQL.
- 3. In the Connect dialog, enter **DBA** as the user ID, and enter any non-blank password. The password itself is not checked, but it must be non-empty.
- 4. On the Database tab, enter **utility_db** as the database name.
- 5. Click OK to connect.

Interactive SQL connects to the utility database on the personal server named **TestEng**, without loading a real database.

For the network server, there are security restrictions on connections. A password is held in the file *util_db.ini* in the same directory as the database server executable.

To connect to the utility database on the network server (Interactive SQL)

1. Add a password to the file *util_db.ini* in the same directory as the database server executable. For example, the following *util_db.ini* file has the password ASA.

[UTILITY_DB] PWD=ASA

2. Start a database server with the following command:

dbsrv9.exe -n TestEng

- 3. Start Interactive SQL.
- 4. In the Connect dialog, enter **DBA** as the user ID, and enter the password held in the file *util_db.ini*.
- 5. On the Database tab, enter **utility_db** as the database name.
- 6. Click OK to connect.

Interactive SQL connects to the utility database on the network server named **TestEng**, without loading a real database.

For more information, see "Connecting to a Database" on page 37.

Utility database server security

There are two aspects to utility database server security:

- Who can connect to the utility database? This is controlled by the use of passwords.
- Who can execute file administration statements? This is controlled by database server options.

Utility database passwords

The personal server and the network server have different security models for connections.

For the personal server, you must specify the user ID **DBA**. You must also specify a password, but it can be any password. Since the personal server is for single machine use, security restrictions (for example passwords) are unnecessary.

For the network server, you must specify the user ID **DBA**, and the password that is held in a file named *util_db.ini*, stored in the same directory as the

database server executable file. As this directory is on the server, you can control access to the file, and thereby control who has access to the password.

The util_db.ini file The util_db.ini file has the following contents:

[UTILITY_DB] PWD=password

Use of the **utility_db** security level relies on the physical security of the computer hosting the database server, since the *util_db.ini* file can be easily read using a text editor.

Permission to execute file administration statements

A level of security is provided for the ability to execute certain administration tasks. The -gu database server option controls who can execute the file administration statements.

There are four levels of permission for the use of file administration statements. These levels include: **all**, **none**, **DBA**, and **utility_db**. The **utility_db** level permits only a person with authority to connect to the utility database to use the file administration statements.

-gu option	Effect	Applies to
All	Anyone can execute file administration statements	Any database includ- ing utility database
None	No one can execute file administration statements	Any database includ- ing utility database
Dba	Only DBA-authority users can execute file administration statements	Any database includ- ing utility database
Utility_db	Only the user who can connect to utility database can execute file administration statements	Only the utility database

For more information on the database server -gu option, see "-gu server option" on page 150.

Examples

◆ To prevent the use of the file administration statements, start the database server using the **none** permission level of the -gu option. The following command starts a database server and names it **TestSrv**. It loads the sample database, but prevents anyone from using that server to create or delete a database, or execute any other file administration statement

regardless of their resource creation rights, or whether or not they can load and connect to the utility database.

```
dbsrv9.exe -n TestSrv -gu none asademo.db
```

• To permit only the users knowing the utility database password to execute file administration statements, start the server at the command prompt with the following command.

```
dbsrv9 -n TestSrv -gu utility_db
```

Assuming the utility database password has been set during installation to **asa**, the following command starts the Interactive SQL utility as a client application, connects to the server named **TestSrv**, loads the utility database, and connects the user.

```
dbisql -c "uid=DBA;pwd=asa;dbn=utility_db;eng=TestSrv"
```

Having executed the above statement successfully, the user connects to the utility database, and can execute file administration statements.

CHAPTER 10

Automating Tasks Using Schedules and Events

About this chapter	This chapter describes how to use scheduling and Adaptive Server Anywhere to automate database tasks.	l event handling features of administration and other
Contents	Торіс:	page
	Introduction	268
	Understanding schedules	270
	Understanding events	272
	Understanding event handlers	276
	Schedule and event internals	278
	Scheduling and event handling tasks	280

Introduction

	Many database administration tasks are best carried out systematically. For example, a regular backup procedure is an important part of proper database administration procedures.		
	You can automate routine tasks in Adaptive Server Anywhere by adding an event to a database, and providing a schedule for the event. Whenever one of the times in the schedule passes, the database server executes a sequence of actions called an event handler .		
	Database administration also requires taking action when certain conditions occur. For example, it may be appropriate to e-mail a notification to a system administrator when a disk containing the transaction log is filling up so that the administrator can handle the situation. These tasks too can be automated by defining event handlers for one of a set of system events .		
Chapter contents	This chapter contains the following material:		
	• An introduction to scheduling and	event handling (this section).	
	 Concepts and background information to help you design and use schedules and event handlers: "Understanding schedules" on page 270. 		
	• "Understanding events" on page 272.		
	 A discussion of techniques for developing event handlers: "Developing event handlers" on page 276. 		
	 Internals information: "Schedule and event internals" on page 278. 		
	 Step by step instructions for how to carry out automation tasks. "Scheduling and event handling tasks" on page 280. 		
Questions and answers			
	To answer the question	Consider reading	

What is a schedule?	"Understanding schedules" on page 270.
What is a system event?	"Understanding events" on page 272
What is an event handler?	"Understanding event handlers" on page 276
How do I debug event handlers?	"Developing event handlers" on page 276

To answer the question	Consider reading
How does the database server use schedules to trigger event handlers?	"How the database server checks for scheduled times" on page 278
How can I schedule regular backups?	For an example, see "Understanding schedules" on page 270.
What kind of system events can the database server use to trigger event handlers?	"Understanding events" on page 272. "CREATE EVENT statement" [ASA SQL Reference, page 304].
What connection do event handlers get executed on?	"How event handlers are executed" on page 279.
How do event handlers get information about what triggered them?	"Developing event handlers" on page 276
	"EVENT_PARAMETER function [System]" [ASA SQL Reference, page 128]

Understanding schedules

set of preset times. The scheduling information and the event handler are both stored in the database itself.	a
Although this is not usually necessary, you can define complex schedules associating more than one schedule with a named event. For example, a retail outlet might want an event to occur once per hour during hours of operation, where the hours of operation vary based on the day of the week You can achieve the same effect by defining multiple events, each with its own schedule, and by calling a common stored procedure.	by c.
The following examples give some ideas for scheduled actions that may buseful.	e
When scheduling events, you can use either full-length English day name (Monday, Tuesday, and so on) or the abbreviated forms of the day (Mon, Tue, and so on). Note that you must use the full-length English day name you want the day names to be recognized by a server running in a languag other than English.	s s if ge
For more information, see "CREATE EVENT statement" [ASA SQL Reference, page 304].	
Carry out an incremental backup daily at 1:00 am:	
create event IncrementalBackup schedule start time '1:00 AM' every 24 hours handler begin backup database directory 'c:\\backup' transaction log only transaction log rename match end	
Summarize orders at the end of each business day:	

Examples

Defining schedules

Schedule definitions have several components to them, to permit flexibility:

- Name Each schedule definition has a name. You can assign more than one schedule to a particular event, which can be useful in designing complex schedules.
- **Start time** You can define a start time for the event, which is the time that it is first executed.
- **Range** As an alternative to a start time, you can specify a range of times for which the event is active. The event occurs between the start and end time specified. Frequency is determined by the specified recurrence.
- ♦ Recurrence Each schedule can have a recurrence. The event is triggered on a frequency that can be given in hours, minutes, or seconds on a set of days that can be specified as days of the week or days of the month. Recurring events include an EVERY or ON clause.

Understanding events

The database server tracks several kinds of system events. Event handlers are triggered when the system event is checked by the database server, and satisfies a provided **trigger condition**.

By defining event handlers to execute when a chosen system event occurs and satisfies a trigger condition that you define, you can improve the security and safety of your data, and help to ease administration.

For more information on the available system events, see "Choosing a system event" on page 272.

For more information on trigger conditions, see "Defining trigger conditions for events" on page 273.

Choosing a system event

Adaptive Server Anywhere tracks several system events. Each system event provides a hook on which you can hang a set of actions. The database server tracks the events for you, and executes the actions (as defined in the event handler) when needed.

The available system events include the following:

- **Backup** You can use the **BackupEnd** event type to take action at the end of a backup.
- **DatabaseStart** The database is started.
- Connection events When a connection is made (Connect) or when a connection attempt fails (ConnectFailed). You may want to use these events for security purposes.
- **Disconnect** You can use the disconnect event to take action when a user or application disconnects.
- Free disk space Tracks the available disk space on the device holding the database file (DBDiskSpace), the log file (LogDiskSpace), or temporary file (TempDiskSpace). This system event is not available on the following operating systems:
 - Windows 95 before OSR2
 - Windows CE

You may want to use disk space events to alert administrators in case of a disk space shortage.
◆ File size The file reaches a specified size. This can be used for the database file (GrowDB), the transaction log (GrowLog), or the temporary file (GrowTemp).

You may want to use file size events to track unusual actions on the database, or monitor bulk operations.

- SQL errors When an error is triggered, you can use the **RAISERROR** event type to take actions.
- Idle time The database server has been idle for a specified time. You
 may want to use this event type to carry out routine maintenance
 operations at quiet times.

Defining trigger conditions for events

Examples

Each event definition has a system event associated with it. It also has one or more trigger conditions. The event handler is triggered when the trigger conditions for the system event are satisfied.

The trigger conditions are included in the WHERE clause of the CREATE EVENT statement, and can be combined using the AND keyword. Each trigger condition is of the following form:

event_condition(

condition-name) comparison-operator value

The *condition-name* argument is one of a set of preset strings, which are appropriate for different event types. For example, you can use **DBSize** (the database file size in Megabytes) to build a trigger condition suitable for the **GrowDB** system event. The database server does not check that the condition-name matches the event type: it is your responsibility to ensure that the condition is meaningful in the context of the event type.

• Limit the transaction log size to 10 Mb:

```
create event LogLimit
type GrowLog
where event_condition( 'LogSize' ) > 10
handler
begin
    backup database
    directory 'c:\\logs'
    transaction log only
    transaction log rename match
end
```

 Notify an administrator when free disk space on the device containing the database file falls below 10%, but do not execute the handler more than once every five minutes (300 seconds):

```
create event LowDBSpace
type DBDiskSpace
where event_condition( 'DBFreePercent' ) < 10
and event_condition( 'Interval' ) >= 300
handler
begin
    call xp_sendmail( recipient='DBAdmin',
        subject='Low disk space',
        "message"='Database free disk space '
        || event_parameter( 'DBFreeSpace' ) );
end
```

• Notify an administrator of a possible attempt to break into the database:

```
create event SecurityCheck
type ConnectFailed
handler
begin
   declare num_failures int;
   declare mins int;
   insert into FailedConnections( log time )
   values ( current timestamp );
   select count( * ) into num_failures
   from FailedConnections
   where log_time >= dateadd( minute, -5,
      current timestamp );
   if( num failures >= 3 ) then
      select datediff( minute, last_notification,
         current timestamp ) into mins
      from Notification;
      if (mins > 30) then
         update Notification
         set last_notification = current timestamp;
         call xp_sendmail( recipient='DBAdmin',
                     subject='Security Check',
                  "message"=
   'over 3 failed connections in last 5 minutes' )
      end if
   end if
end
```

 Run a process when the server has been idle for ten minutes. Do not execute more frequently than once per hour:

```
create event Soak
type ServerIdle
where event_condition( 'IdleTime' ) >= 600
and event_condition( 'Interval' ) >= 3600
handler
begin
    message ' Insert your code here ... '
end
```

Understanding event handlers

Event handlers execute on a separate connection from the action that triggered the event, and so do not interact with client applications. They execute with the permissions of the creator of the event.

Developing event handlers

	Event handlers, whether for scheduled events or for system event handling, contain compound statements, and are similar in many ways to stored procedures. You can add loops, conditional execution, and so on, and you can use the Adaptive Server Anywhere debugger to debug event handlers.
Context information for event handlers	One difference between event handlers and stored procedures is that event handlers do not take any arguments. Certain information about the context in which an event was triggered is available through the event_parameter function, which supplies information about the connection that caused an event to be triggered (connection ID, user ID), as well as the event name and the number of times it has been executed.
	For more information, see "EVENT_PARAMETER function [System]" [ASA SQL Reference, page 128].
Testing event handlers	During development, you want event handlers to be triggered at convenient times. You can use the TRIGGER EVENT statement to explicitly cause an event to execute, even when the trigger condition or scheduled time has not occurred. However, TRIGGER EVENT does not cause disabled event handlers to be executed.
	For more information, see "TRIGGER EVENT statement" [ASA SQL Reference, page 583].
	While it is not good practice to develop event handlers on a production database, you can disable event handlers from Sybase Central or explicitly using the ALTER EVENT statement.
	It can be useful to use a single set of actions to handle multiple events. For example, you may want to take a notification action if disk space is limited on any of the devices holding the database or log files. To do this, create a stored procedure and call it in the body of each event handler.
Debugging event handlers	Debugging event handlers is very similar to debugging stored procedures. The event handlers appear in the procedures list.
	One difference is that, because each event handler runs on its own connection, you must be sure to select All connections before setting a breakpoint in an event handler.

You can also determine how many instances of a particular event handler are currently active using the NumActive event parameter. This function is useful if you want to limit an event handler so that only one instance executes at any given time.

For more information about the NumActive event parameter, see "EVENT_PARAMETER function [System]" [ASA SQL Reference, page 128].

For more information about step-by-step instructions, see "Debugging an event handler" on page 282.

Schedule and event internals

This section describes how the database server processes schedules and event definitions.

How the database server checks for events

Events are classified according to their **event type**, as specified directly in the CREATE EVENT statement or using Sybase Central. There are two kinds of event types:

Active event types Some event types are the result of action by the database server itself. These active event types include growing database files, or the start and end of different database actions (BackupEnd and so on) or RAISERROR.

When the database server takes the action, it checks to see whether the trigger conditions defined in the WHERE clause are satisfied, and if so, triggers any events defined for that event type.

 Polled event types Some event types are not triggered solely by database actions. The free disk space types (DBDiskSpace and so on) as well as the IdleTime types are of this kind.

For these types of events, the database server polls every thirty seconds, starting approximately thirty seconds after the database server is started.

For the **IdleTime** event type, the database server checks whether the server has been idle for the entire thirty seconds. If no requests have started and none are currently active, it adds the idle check interval time in seconds to the idle time total; otherwise, the idle time total is reset to 0. The value for **IdleTime** is therefore always a multiple of thirty seconds. When **IdleTime** is greater than the interval specified in the trigger condition, event handlers associated with **IdleTime** are fired.

How the database server checks for scheduled times

The calculation of scheduled event times is done when the database server starts, and each time a scheduled event handler completes.

The calculation of the next scheduled time is based on the increment specified in the schedule definition, with the increment being added to the previous start time. If the event handler takes longer to execute than the specified increment, so that the next time is earlier than the current time, the database server increments until the next scheduled time is in the future.

An event handler that takes sixty-five minutes to execute and is requested to run every hour between 9:00 and 5:00 will run every two hours, at 9:00, 11:00, 1:00, and so on.

To run a process such that it operates between 9:00 and 5:00 and delays for some period before the next execution, you could define a handler to loop until its completion time has passed, with a sleep instruction, perhaps using xp_cmdshell, between each iteration.

If you are running a database server intermittently, and it is not running at a scheduled time, the event handler does not run at startup. Instead, the next scheduled time is computed at startup. If, for example, you schedule a backup to take place every night at one o'clock, but regularly shut down the database server at the end of each work day, the backup never takes place.

How event handlers are executed

When an event handler is triggered, a temporary internal connection is made on which the event handler is executed. The handler is *not* executed on the connection that caused the handler to be triggered, and consequently statements such as MESSAGE .. TO CLIENT, which interact with the client application, are not meaningful within event handlers.

The temporary connection on which the handler is executed does not count towards the connection limit for licensing purposes.

Event creation requires DBA authority, and events execute with the permissions of their creator. If you wish event handlers to execute with non-DBA authority, you can call a procedure from within the handler, as stored procedures run with the permissions of their creator.

Any event errors are logged to the server console.

Scheduling and event handling tasks

This section collects together instructions for tasks related to automating tasks with schedules and events.

Adding a schedule or event to a database

Schedules and events are handled in a similar fashion, both from Sybase Central and in SQL.

For more information, see "Understanding schedules" on page 270, and "Understanding events" on page 272.

To add a schedule or event to a database (Sybase Central)

- 1. Connect to the database as a user with DBA authority.
- 2. Open the Events folder for your database.
- 3. click the New Event button on the toolbar. The Event Creation wizard appears.
- 4. Follow the instructions in the wizard.

The wizard contains many options, depending on the schedule or event you wish to create. These are explained in detail in other tasks.

To add a schedule or event to a database (SQL)

- 1. Connect to the database as a user with DBA authority.
- 2. Execute a CREATE EVENT statement.

The CREATE EVENT statement contains many options, depending on the schedule or event you wish to create. These are explained in detail in other tasks.

For more information, see "CREATE EVENT statement" [ASA SQL Reference, page 304].

Adding a manually-triggered event to a database

If you create an event handler without a schedule or system event to trigger it, it is executed only when manually triggered.

To add a manually-triggered event to a database (Sybase Central)

- 1. Connect to the database as a user with DBA authority.
- 2. Open the Events folder for your database.
- 3. Click the Add Event event button on the toolbar. The Event Creation wizard appears.
- 4. Type a name for the event, and click Next.
- 5. Select Manually, and click Next.
- 6. Type the SQL statements for your event handler, and click Next.
- 7. Select Enable this event, and select Execute at all Locations, and click Next.
- 8. Type a comment describing the event, and click Finish to add the event to the database.

If you wish to accept the default values for all remaining options, you can click Finish at an earlier stage in the wizard.

* To add a manually-triggered event to a database (SQL)

- 1. Connect to the database as a user with DBA authority.
- 2. Execute a CREATE EVENT statement with no schedule or WHERE clause. The restricted syntax of the CREATE EVENT is as follows:

CREATE EVENT event-name HANDLER BEGIN ... event handler END

If you are developing event handlers, you can add schedules or system events to control the triggering of an event later, either using Sybase Central or the ALTER EVENT statement.

- See also:
- For information on triggering events, see "Triggering an event handler" on page 281.
- For information on altering events, see "ALTER EVENT statement" [ASA SQL Reference, page 231].

Triggering an event handler

Any event handler can be triggered manually, in addition to those occasions when it executes because of a schedule or system event. Triggering events manually can be useful during development of event handlers, and also, for certain events, in production environments. For example, you may have a monthly sales report scheduled, but from time to time you may want to obtain a sales report for a reason other than the end of the month.

For more information on developing event handlers, see "Developing event handlers" on page 276.

To trigger an event handler (Sybase Central)

- 1. Connect to the database as a user with DBA authority.
- 2. Open the Events folder for your database.
- 3. Right-click the event you wish to trigger and choose Trigger from the popup menu. The Trigger Event dialog appears.

The event must be enabled before you can select Trigger from the popup menu. You can enable the event on the General tab of the Event property sheet.

4. Supply any parameters the event handler requires, in the following form:

parameter=value;parameter=value

Click OK to trigger the event handler.

To trigger an event handler (SQL)

- 1. Connect to the database as a user with DBA authority.
- 2. Execute the TRIGGER EVENT statement, supplying the name of the event. For example:

TRIGGER EVENT sales_report_event

For more information, see "TRIGGER EVENT statement" [ASA SQL Reference, page 583].

Debugging an event handler

Debugging is a regular part of any software development. Event handlers can be debugged during the development process.

For more information on developing event handlers, see "Developing event handlers" on page 276.

For more information on using the debugger, see "Debugging Logic in the Database" [ASA SQL User's Guide, page 673].

To debug an event handler

1. Start the Adaptive Server Anywhere debugger.

From the Start menu, choose Programs \succ Sybase SQL Anywhere 9 \succ Adaptive Server Anywhere 9 \succ Debug Database Objects.

- 2. In the Connections window, double-click All Connections.
- 3. In the Procedures window, double-click the event you wish to debug. The event definition appears in the Source window.
- 4. In the Source window, set a breakpoint.
- 5. From Interactive SQL or another application, trigger the event handler using the TRIGGER EVENT statement.
- 6. The execution stops at the breakpoint you have set. You can now use the debugger features to trace execution, local variables, and so on.

CHAPTER 11

International Languages and Character Sets

About this chapter	This chapter describes how to configure your Adaptive Sen installation to handle international language issues.	ver Anywhere
Contents	Торіс:	page
	Introduction to international languages and character sets	286
	Understanding character sets in software	289
	Understanding locales	301
	Understanding collations	307
	Understanding character set translation	317
	Collation internals	320
	International language and character set tasks	325

Introduction to international languages and character sets

This section provides an introduction to the issues you may face when working in an environment that uses more than one character set, or when using languages other than English.

When you create a database, you specify a collating sequence, or collation, for the database to use. A collation is a combination of a character set and a sort order for characters in the database.

Localized Versions of Adaptive Server Anywhere

Adaptive Server Anywhere is available in the following fully localized (packages, software, documentation, books) languages:

- ♦ English
- ♦ German
- French
- ♦ Japanese
- ♦ Simplified Chinese

As well, Adaptive Server Anywhere is available in the following deployment (software only) languages:

- ♦ Spanish
- Portuguese (Brazilian)
- Polish
- ♦ Italian
- Russian
- Ukrainian
- ♦ Korean
- Traditional Chinese

Adaptive Server Anywhere international features

Adaptive Server Anywhere provides two sets of features that are of particular interest when setting up databases for languages.

 Collations You can choose from a wide selection of supplied collations when you create a database. By creating your database with the proper collation, you ensure proper sorting of data.

Whenever the database compares strings, sorts strings, or carries out other string operations such as case conversion, it does so using the collation sequence. The database carries out sorting and string comparison when statements such as the following are executed:

- Queries with an ORDER BY clause.
- Expressions that use string functions, such as LOCATE, SIMILAR, SOUNDEX.
- Conditions using the LIKE keyword.
- Queries that use index lookups on character data.

The database also uses collations to identify valid or unique identifiers (column names and so on).

 Character set translation You can set up Adaptive Server Anywhere to convert data between the character set encoding on your server and client systems, thus maintaining the integrity of your data even in mixed character set environments.

Character set translation is provided between client and server, as well as by the ODBC driver. The Adaptive Server Anywhere ODBC driver provides OEM to ANSI character set translation and Unicode support.

Using the default collation

If you use the default actions when creating a database, the Initialization utility infers a collation from the character set used by the operating system on the machine on which you create the database.

For more information on how to find the default collation in your environment, see "Finding the default collation" on page 325.

If all the machines in your environment share the same character set, then this choice of collation ensures that all the character data in the database and in client applications is represented in the same manner. As long as the collation provides a proper character set and sort order for your data, using this default setting is a simple way of ensuring that characters are represented consistently throughout the system.

If it is not possible to set up your system in this default manner, you need to decide which collation to use in your database, and whether to use character set translation to ensure that data is exchanged consistently between the pieces of your database system. This chapter provides the information you need to make and implement these decisions.

Character set questions and answers

The following table identifies where you can find answers to questions.

To answer the question	Consider reading
How do I set up my computing envi- ronment to treat character sets prop- erly?	"Configuring your character set environment" on page 325
How do I decide which collation to use for my database?	"Understanding collations" on page 307
How are characters represented in software, and Adaptive Server Any-where in particular?	"Understanding character sets in software" on page 289
What collations does Adaptive Server Anywhere provide?	"Choosing collations" on page 307
How do I ensure that error and in- formational messages sent from the database server to client applications are sent in the proper language and character set for my application?	"Character translation for database messages" on page 317
I have a different character set on client machines from that in use in the database. How can I get characters to be exchanged properly between client and server?	"Starting a database server us- ing character set translation" on page 329
What character sets can I use for connection strings?	"Connection strings and character sets" on page 318
How do I create a collation that is different from the supplied ones?	"Creating a database with a custom collation" on page 331
How do I change the collation se- quence of an existing database?	"Changing a database from one collation to another" on page 332.
How do I create a database for Win- dows CE?	"Creating databases for Windows CE" on page 312

Understanding character sets in software

This section provides general information about software issues related to international languages and character sets.

Pieces in the character set puzzle

There are several distinct aspects to character storage and display by computer software:

- Each piece of software works with a character set. A **character set** is a set of symbols, including letters, digits, spaces, and other symbols.
- To handle these characters, each piece of software employs a character set encoding, in which each character is mapped onto one or more bytes of information, typically represented as hexadecimal numbers. This encoding is also called a code page.
- Database servers, which sort characters (for example, list names alphabetically), use a collation. A **collation** is a combination of a character encoding (a map between characters and hexadecimal numbers) and a **sort order** for the characters. There may be more than one sort order for each character set; for example, a case sensitive order and a case insensitive order, or two languages may sort characters in a different order.
- Characters are printed or displayed on a screen using a **font**, which is a mapping between characters in the character set and its appearance. Fonts are handled by the operating system.
- Operating systems also use a **keyboard mapping** to map keys or key combinations on the keyboard to characters in the character set.

Language issues in client/server computing

Database users working at client applications may see or access strings from the following sources:

• Data in the database Strings and other text data are stored in the database. The database server processes these strings when responding to requests.

For example, the database server may be asked to supply all the last names beginning with a letter ordered less than N in a table. This request requires string comparisons to be carried out, and assumes a character set ordering. The database server receives strings from client applications as streams of bytes. It associates these bytes with characters according to the database character set. If the data is held in an indexed column, the index is sorted according to the sort order of the collation.

- Database server software messages Applications can cause database errors to be generated. For example, an application may submit a query that references a column that does not exist. In this case, the database server returns a warning or error message. This message is held in a **language resource library**, which is a DLL or shared library called by Adaptive Server Anywhere.
- Client application The client application interface displays text, and internally the client application may process text.
- Client software messages The client library uses the same language library as the database server to provide messages to the client application.
- **Operating system** The client operating system has text displayed on its interface, and may also process text.

For a satisfactory working environment, all these sources of text must work together. Loosely speaking, they must all be working in the user's language and/or character set.

Code pages

	Many languages have few enough characters to be represented in a single-byte character set. In such a character set, each character is represented by a single byte: a two-digit hexadecimal number.
	At most, 256 characters can be represented in a single byte. No single-byte character set can hold all of the characters used internationally, including accented characters. This problem was addressed by the development of a set of code pages, each of which describes a set of characters appropriate for one or more national languages. For example, code page 869 contains the Greek character set, and code page 850 contains an international character set suitable for representing many characters in a variety of languages.
Upper and lower pages	With few exceptions, characters 0 to 127 are the same for all the single-byte code pages. The mapping for this range of characters is called the ASCII character set. It includes the English language alphabet in upper and lower case, as well as common punctuation symbols and the digits. This range is often called the seven-bit range (because only seven bits are needed to represent the numbers up to 127) or the lower page. The characters from

128 to 255 are called **extended characters**, or **upper** code-page characters, and vary from code page to code page.

Problems with code page compatibility are rare if the only characters used are from the English alphabet, as these are represented in the ASCII portion of each code page (0 to 127). However, if other characters are used, as is generally the case in any non-English environment, there can be problems if the database and the application use different code pages.

ExampleSuppose a database holding French language strings uses code page 850,
and the client operating system uses code page 437. The character À (upper
case A grave) is held in the database as character \xB7 (decimal value 183).
In code page 437, character \xB7 is a graphical character. The client
application receives this byte and the operating system displays it on the
screen, the user sees a graphical character instead of an A grave.

ANSI and OEM code pages in Windows

For PC users, the issue is complicated because there are at least two code pages in use on most PCs. Character-mode applications (those using the console or command prompt window) in Windows 95/98/Me and Windows NT/200/XP, use code pages taken from the IBM set. These are called **OEM code pages** (Original Equipment Manufacturer) for historical reasons.

Windows operating systems do not require the line drawing characters that were held in the extended characters of the OEM code pages, so they use a different set of code pages. These pages are based on the ANSI standard and are therefore commonly called **ANSI code pages**.

Adaptive Server Anywhere supports collations based on both OEM and ANSI code pages.

Example

Consider the following situation:

- A PC is running the Windows 95 operating system with ANSI code page 1252.
- The code page for character-mode applications is OEM code page 437.
- Text is held in a database created using the collation corresponding to OEM code page 850.

An upper case A grave in the database is stored as character 183. This value appears as a graphical character in a character-mode application. The same character appears as a dot in a Windows application.

For more information about choosing a single-byte collation for your database, see "Understanding collations" on page 307.

Supported code pages

The following table lists code pages supported in DBISQL. For more information, see "Opening Interactive SQL using the dbisql command-line utility" on page 492or "Interactive SQL utility options" on page 494.

Code Page	Description
1252	Windows Latin-1
037	USA, Canada (Bilingual, French), Netherlands, Por- tugal, Brazil, Australia
273	IBM Austria, Germany
277	IBM Denmark, Norway
278	IBM Finland, Sweden
280	IBM Italy
284	IBM Catalan/Spain, Spanish Latin America
285	IBM United Kingdom, Ireland
297	IBM France
420	IBM Arabic
424	IBM Hebrew
437	MS-DOS United States, Australia, New Zealand, South Africa
500	EBCDIC 500V1
737	PC Greek
775	PC Baltic
838	IBM Thailand extended SBCS
850	MS-DOS Latin-1
852	MS-DOS Latin-2
855	IBM Cyrillic
856	IBM Hebrew
857	IBM Turkish
858	Variant of Cp850 with Euro character

Code Page	Description
860	MS-DOS Portuguese
861	MS-DOS Icelandic
862	PC Hebrew
863	MS-DOS Canadian French
864	PC Arabic
865	MS-DOS Nordic
866	MS-DOS Russian
868	MS-DOS Pakistan
869	IBM Modern Greek
870	IBM Multilingual Latin-2
871	IBM Iceland
874	IBM Thai
875	IBM Greek
918	IBM Pakistan (Urdu)
921	IBM Latvia, Lithuania (AIX, DOS)
922	IBM Estonia (AIX, DOS)
930	Japanese Katakana-Kanji mixed with 4370 UDC, superset of 5026
933	Korean Mixed with 1880 UDC, superset of 5029
935	Simplified Chinese Host mixed with 1880 UDC, superset of 5031
937	Traditional Chinese Host mixed with 6204 UDC, superset of 5033
939	Japanese Latin Kanji mixed with 4370 UDC, superset of 5035
942	IBM OS/2 Japanese, superset of Cp932
942	C Variant of Cp942
943	IBM OS/2 Japanese, superset of Cp932 and Shift-JIS
943	C Variant of Cp943

Code Page	Description
948	OS/2 Chinese (Taiwan) superset of 938
949	PC Korean
949	C Variant of Cp949
950	PC Chinese (Hong Kong, Taiwan)
964	AIX Chinese (Taiwan)
970	AIX Korean
1006	IBM AIX Pakistan (Urdu)
1025	IBM Multilingual Cyrillic: Bulgaria, Bosnia, Herze- govinia, Macedonia (FYR)
1026	IBM Latin-5, Turkey
1046	IBM Arabic - Windows
1097	IBM Iran (Farsi)/Persian
1098	IBM Iran (Farsi)/Persian (PC)
1112	IBM Latvia, Lithuania
1122	IBM Estonia
1123	IBM Ukraine
1124	IBM AIX Ukraine
1140	Variant of Cp037 with Euro character
1141	Variant of Cp273 with Euro character
1142	Variant of Cp277 with Euro character
1143	Variant of Cp278 with Euro character
1144	Variant of Cp280 with Euro character
1145	Variant of Cp284 with Euro character
1146	Variant of Cp285 with Euro character
1147	Variant of Cp297 with Euro character
1148	Variant of Cp500 with Euro character
1149	Variant of Cp871 with Euro character
1250	Windows Eastern European

Code Page	Description
1251	Windows Cyrillic
1253	Windows Greek
1254	Windows Turkish
1255	Windows Hebrew
1256	Windows Arabic
1257	Windows Baltic
1258	Windows Vietnamese
1381	IBM OS/2, DOS People's Republic of China (PRC)
1383	IBM AIX People's Republic of China (PRC)
33722	IBM-eucJP - Japanese (superset of 5050)
ASCII	American Standard Code for Information Interchange
ISO8859_1	ISO 8859-1, Latin alphabet No. 1
UnicodeBig	Sixteen-bit Unicode Transformation Format, big- endian byte order, with byte-order mark
UnicodeBigUn- marked	Sixteen-bit Unicode Transformation Format, big- endian byte order
UnicodeLittle	Sixteen-bit Unicode Transformation Format, little- endian byte order, with byte-order mark
UnicodeLittleUn- marked	Sixteen-bit Unicode Transformation Format, little- endian byte order
UTF8	Eight-bit Unicode Transformation Format
UTF-16	Sixteen-bit Unicode Transformation Format, byte order specified by a mandatory initial byte-order mark
Big5	Big5, Traditional Chinese
Big5_HKSCS	Big5 with Hong Kong extensions, Traditional Chinese
Big5_Solaris	Big5 with seven additional Hanzi ideograph character mappings for the Solaris zh_TW.BIG5 locale
EUC_CN	GB2312, EUC encoding, Simplified Chinese
EUC_JP	JIS X 0201, 0208, 0212, EUC encoding, Japanese

Code Page	Description
EUC_KR	KS C 5601, EUC encoding, Korean
EUC_TW	CNS11643 (Plane 1-3), EUC encoding, Traditional Chinese
GB18030	Simplified Chinese, PRC standard
GBK	GBK, Simplified Chinese
ISCII91	ISCII91 encoding of Indic scripts
ISO2022CN	ISO 2022 CN, Chinese (conversion to Unicode only)
ISO2022CN_CNS	CNS 11643 in ISO 2022 CN form, Traditional Chinese (conversion from Unicode only)
ISO2022CN_GB	GB 2312 in ISO 2022 CN form, Simplified Chinese (conversion from Unicode only)
ISO2022JP	JIS X 0201, 0208 in ISO 2022 form, Japanese
ISO2022KR	ISO 2022 KR, Korean
ISO8859_2	ISO 8859-2, Latin alphabet No. 2
ISO8859_3	ISO 8859-3, Latin alphabet No. 3
ISO8859_4	ISO 8859-4, Latin alphabet No. 4
ISO8859_5	ISO 8859-5, Latin/Cyrillic alphabet
ISO8859_6	ISO 8859-6, Latin/Arabic alphabet
ISO8859_7	ISO 8859-7, Latin/Greek alphabet
ISO8859_8	ISO 8859-8, Latin/Hebrew alphabet
ISO8859_9	ISO 8859-9, Latin alphabet No. 5
ISO8859_13	ISO 8859-13, Latin alphabet No. 7
ISO8859_15_FDIS	ISO 8859-15, Latin alphabet No. 9
JIS0201	JIS X 0201, Japanese
JIS0208	JIS X 0208, Japanese
JIS0212	JIS X 0208, Japanese
JISAutoDetect	Detects and converts from Shift-JIS, EUC-JP, ISO 2022 JP (conversion to Unicode only)
Johab	Johab, Korean

Code Page	Description
KOI8_R	KOI8-R, Russian
MS874	Windows Thai
MS932	Windows Japanese
MS936	Windows Simplified Chinese
MS949	Windows Korean
MS950	Windows Traditional Chinese
MacArabic	Macintosh Arabic
MacCentralEurope	Macintosh Latin-2
MacCroatian	Macintosh Croatian
MacCyrillic	Macintosh Cyrillic
MacDingbat	Macintosh Dingbat
MacGreek	Macintosh Greek
MacHebrew	Macintosh Hebrew
MacIceland	Macintosh Iceland
MacRoman	Macintosh Roman
MacRomania	Macintosh Romania
MacSymbol	Macintosh Symbol
MacThai	Macintosh Thai
MacTurkish	Macintosh Turkish
MacUkraine	Macintosh Ukraine
SJIS	Shift-JIS, Japanese
TIS620	TIS620, Thai

Multibyte character sets

Some languages, such as Japanese and Chinese, have many more than 256 characters. These characters cannot all be represented using a single byte, but can be represented in multibyte character sets. In addition, some character sets use the much larger number of characters available in a multibyte representation to represent characters from many languages in a

	single, more comprehensive, character set.
	Multibyte character sets are of two types. Some are variable width , in which some characters are single-byte characters, others are double-byte, and so on. Other sets are fixed width , in which all characters in the set have the same number of bytes. Adaptive Server Anywhere supports only variable-width character sets.
	For more information on the multibyte character sets, see "Using multibyte collations" on page 315.
Example	As an example, characters in the Shift-JIS character set are either one or two bytes in length. If the value of the first byte is in the range of hexadecimal values from $x81$ to $x9F$ or from $xE0$ to xEF (decimal values 129-159 or 224-239) the character is a two-byte character and the subsequent byte (called a follow byte) completes the character. A follow byte is any byte(s) other than the first byte.
	If the first byte is outside this range, the character is a single-byte character and the next byte is the first byte of the following character.
	♦ The properties of any Shift-JIS character can be read from its first byte also. Characters with a first byte in the range \x09 to \x0D, or \x20 are space characters.
	• Characters in the ranges \x41 to \x5A, \x61 to \x7A, \x81 to \x9F or \xE0 to \xEF are considered to be alphabetic (letters).
	• Characters in the range \x30 to \x39 are digits.
Sorting characters	s using collations
	The database collation sequence includes the notion of alphabetic ordering of letters, and extends it to include all characters in the character set, including digits and space characters.
Associating more than one character with each sort position	More than one character can be associated with each sort position. This is useful if you wish, for example, to treat an accented character the same as the character without an accent.
	Two characters with the same sort position are considered identical in all ways by the database. Therefore, if a collation assigned the characters a and e to the same sort position, then a query with the following search condition:
	WHERE coll = 'want'
	is satisfied by a row for which col1 contains the entry went .
	At each sort position, lowercase and uppercase forms of a character can be indicated. For case sensitive databases, the lowercase and uppercase

characters are not treated as equivalent. For case insensitive databases, the lowercase and uppercase versions of the character are considered equivalent.

Тір

Any code that selects a default collation for a German system should select 1252LATIN1, *not* 1252DEU. 1252DEU differentiates between characters with and without an umlaut, while 1252LATIN1 does not. 1252LATIN1 considers Muller and Müller equal, but 1252DEU does not consider them equal. Because 1252DEU views characters with umlauts as separate characters, it has the following alphabetic ordering: ob, öa.

First-byte collation orderings for multibyte character sets

A sorting order for characters in a multibyte character set can be specified only for the first byte. Characters that have the same first byte are sorted according to the hexadecimal value of the following bytes.

International aspects of case sensitivity

Adaptive Server Anywhere is always **case preserving** and **case insensitive** for identifiers, such as table names and column names. This means that the names are stored in the case in which they are created, but any access to the identifiers is done in a case insensitive manner.

For example, the names of the system tables are held in upper case (SYSDOMAIN, SYSTABLE, and so on), but access is case insensitive, so that the two following statements are equivalent:

```
SELECT *
FROM systable
SELECT *
FROM SYSTABLE
```

The equivalence of upper and lower case characters is enforced in the collation. There are some collations where particular care may be needed when assuming case insensitivity of identifiers.

Turkish Users with Turkish databases (1254TRK, 920TRK, 857TRK, etc.) may see unexpected errors. These errors are usually related to a database object whose name contains the letter I or i. In the Turkish language and collations, the letters I and i are not equivalent characters. When specifying the name of an object, the proper letters must be used. A simple way to avoid such problems is to specify object names in their correct case.

For example,

SELECT * FROM SYSCOLLATION

would be accepted, however,

SELECT * FROM syscollation

would yield the error, Table 'syscollation' not found.

Example

In the Turkish 857TRK collation, the lower case **i** does not have the character **I** as its upper case equivalent. Therefore, despite the case insensitivity of identifiers, the following two statements are *not* equivalent in this collation:

SELECT * FROM sysdomain

SELECT * FROM SYSDOMAIN

Understanding locales

Both the database server and the client library recognize their language and character set environment using a **locale definition**.

Introduction to locales

The application locale, or client locale, is used by the client library when making requests to the database server, to determine the character set in which results should be returned. If character set translation is enabled (the default), the database server compares its own locale with the application locale to determine whether character set translation is needed. Different databases on a server may have different locale definitions.

The locale consists of the following components:

◆ Language The language is a two-character string using the ISO-639 standard values: DE for German, FR for French, and so on. Both the database server and the client have language values for their locale.

The database server uses the locale language to determine the following behavior:

- Which language library to load.
- The language is used together with the character set to determine which collation to use when creating databases, if no collation is explicitly specified.

The client library uses the locale language to determine the following behavior:

- Which language library to load.
- Which language to request from the database.
 For more information, see "Understanding the locale language" on page 301.
- **Character set** The character set is the code page in use. The client and server both have character set values, and they may differ. If they differ, character set translation may be required to enable interoperability.

For machines that use both OEM and ANSI code pages, the ANSI code page is the value used here.

For more information, see "Understanding the locale character set" on page 303.

Understanding the locale language

The locale language is an indicator of the language being used by the user of the client application, or expected to be used by users of the database server.

	 For more information about how to find locale settings, see "Determining locale information" on page 326.
	The client library determines the language component of the locale as follows:
	1. On Windows, it checks the Adaptive Server Anywhere language registry entry, as described in "Registry settings on installation" on page 250.
	2. On other operating systems, or if the registry setting is not present, it checks the operating system language setting.
	The database server determines the language component of the locale as follows:
	 It checks the ASLANG environment variable, if it exists. For more information, see .
	2. On Windows, it checks the Adaptive Server Anywhere language registry entry, as described in "Registry settings on installation" on page 250.
	3. Queries the operating system.
	4. If the language cannot be determined by the above settings, it defaults to English.
Language label values	The following table displays the valid language label values, together with the equivalent ISO 639 labels:

Language label	Alternative label	ISO_639 language code
chinese	simpchin	ZH
danish	N/A	DA
french	N/A	FR
german	N/A	DE
italian	N/A	IT
japanese	N/A	JA
korean	N/A	КО
norwegian	norweg	NO
polish	N/A	PL
portuguese	portugue	PT
russian	N/A	RU

Language label	Alternative label	ISO_639 language code
spanish	N/A	ES
swedish	N/A	SV
tchinese	tradchin	TW
ukrainian	N/A	UK
us_english	english	EN

Understanding the locale character set

Both application and server locale definitions have a character set. The application uses its character set when requesting character strings from the server. If character set translation is enabled (the default), the database server compares its character set with that of the application to determine whether character set translation is needed.

For more information about how to find locale settings, see "Determining locale information" on page 326.

The client library determines the character set as follows:

1. If the connection string specifies a character set, it is used.

For more information, see "CharSet connection parameter [CS]" on page 177.

- 2. Open Client applications check the *locales.dat* file in the Sybase *locales* directory.
- 3. Character set information from the operating system is used to determine the locale:
 - On Windows operating systems, use the GetACP system call. This returns the ANSI character set, not the OEM character set.
 - On UNIX, default to ISO8859-1.
 - On other platforms, use code page 1252.

The database server determines the character set for a connection as follows:

1. The character set specified by the client is used if it is supported.

For more information, see "CharSet connection parameter [CS]" on page 177.

2. The database's character set is used if the client specifies a character set that is not supported.

When a new database is created, the database server determines the character set for the new database as follows.

- 1. A collation is specified in the CREATE DATABASE statement or on with the *dbinit* utility.
- 2. The ASCHARSET environment variable is used if it exists.
- 3. Character set information from the operating system is used to determine the locale.
 - On Windows operating systems, use the GetACP system call. This returns the ANSI character set, not the OEM character set.
 - On UNIX, default to ISO8859-1.
 - On other platforms, use code page 1252.

The locale character set and language are used to determine which collation to use when creating a database if none is explicitly specified.

Character set labels The following table displays the valid character set label values, together with the equivalent IANA labels and a description:

Character set label	IANA label	Description
big5	<n a=""></n>	Traditional Chinese (cf. CP950)
cp437	<n a=""></n>	IBM CP437 - U.S. code set
cp850	<n a=""></n>	IBM CP850 - European code set
cp852	<n a=""></n>	PC Eastern Europe
cp855	<n a=""></n>	IBM PC Cyrillic
cp856	<n a=""></n>	Alternate Hebrew
cp857	<n a=""></n>	IBM PC Turkish
cp860	<n a=""></n>	PC Portuguese
cp861	<n a=""></n>	PC Icelandic
cp862	<n a=""></n>	PC Hebrew
cp863	<n a=""></n>	IBM PC Canadian French code page
cp864	<n a=""></n>	PC Arabic
cp865	<n a=""></n>	PC Nordic
cp866	<n a=""></n>	PC Russian

Character set label	IANA label	Description
cp869	<n a=""></n>	IBM PC Greek
cp874	<n a=""></n>	Microsoft Thai SB code page
cp932	windows-31j	Microsoft CP932 = Win31J- DBCS
cp936		Simplified Chinese
cp949	<n a=""></n>	Korean
cp950	<n a=""></n>	PC (MS) Traditional Chinese
cp1250	<n a=""></n>	MS Windows Eastern European
cp1251	<n a=""></n>	MS Windows Cyrillic
cp1252	<n a=""></n>	MS Windows US (ANSI)
cp1253	<n a=""></n>	MS Windows Greek
cp1254	<n a=""></n>	MS Windows Turkish
cp1255	<n a=""></n>	MS Windows Hebrew
cp1256	<n a=""></n>	MS Windows Arabic
cp1257	<n a=""></n>	MS Windows Baltic
cp1258	<n a=""></n>	MS Windows Vietnamese
deckanji	<n a=""></n>	DEC UNIX JIS encoding
euccns	<n a=""></n>	EUC CNS encoding: Traditional Chinese with extensions
eucgb	<n a=""></n>	EUC GB encoding = Simplified Chinese
eucjis	euc-jp	Sun EUC JIS encoding
eucksc	<n a=""></n>	EUC KSC Korean encoding (cf. CP949)
greek8	<n a=""></n>	HP Greek-8
iso_1	iso_8859-1:1987	ISO 8859-1 Latin-1
iso15	<n a=""></n>	ISO 8859-15 Latin1 with Euro, etc.

Character set	IANA label	Description
iso88592	iso_8859-2:1987	ISO 8859-2 Latin-2 Eastern Europe
iso88595	iso_8859-5:1988	ISO 8859-5 Latin/Cyrillic
iso88596	iso_8859-6:1987	ISO 8859-6 Latin/Arabic
iso88597	iso_8859-7:1987	ISO 8859-7 Latin/Greek
iso88598	iso_8859-8:1988	ISO 8859-8 Latin/Hebrew
iso88599	iso_8859-9:1989	ISO 8859-9 Latin-5 Turkish
koi8	<n a=""></n>	KOI-8 Cyrillic
mac	macintosh	Standard Mac coding
mac_cyr	<n a=""></n>	Macintosh Cyrillic
mac_ee	<n a=""></n>	Macintosh Eastern European
macgrk2	<n a=""></n>	Macintosh Greek
macturk	<n a=""></n>	Macintosh Turkish
roman8	hp-rpman8	HP Roman-8
sjis	shift_jis	Shift JIS (no extensions)
tis620	<n a=""></n>	TIS-620 Thai standard
turkish8	<n a=""></n>	HP Turkish-8
utf8	utf-8	UTF-8 treated as a character set

Understanding collations

This section describes the supplied collations, and provides suggestions as to which collations to use under certain circumstances.

For more information about how to create a database with a specific collation, see "Creating a database with a named collation" on page 328 and "The Initialization utility" on page 485.

For information on changing a database from one collation to another, see "Changing a database from one collation to another" on page 332.

Choosing collations

When you create a database, Adaptive Server Anywhere chooses a default collation based on operating system language and character set settings. In most cases, the default collation is a suitable choice, but you can also explicitly choose a collation to match your needs from the wide selection of supplied collations.

In some cases, Adaptive Server Anywhere supports more than one collation for a particular language. Different collations vary in the characters they include and in the way they sort special characters, including ligatures (characters consisting of two or more characters joined together) and accented characters. You should choose a collation that uses a character set and sort order that are appropriate for the data in your database.

For more information about sorting of data and international features, see "Adaptive Server Anywhere international features" on page 286.

Another consideration when you choose a collation is whether to use an ANSI or an OEM collation. OEM collations are suited to character-mode applications (those using the console or command prompt window) in Windows 95/98/Me and Windows NT/2000/XP. ANSI collations are designed for Windows environments, and are recommended in the majority of cases.

For more information about the differences between and ANSI and OEM collations, see "ANSI or OEM?" on page 312 and "ANSI and OEM code pages in Windows" on page 291.

Last resort collation

If Adaptive Server Anywhere cannot locate a specific collation based on your operating system's character set and language, the Adaptive Server Anywhere registry setting, or the ASCHARSET environment variable, ISO_BINENG becomes the default, regardless of which operating system you are using.

If this is not suitable for your requirements, you can rebuild the database and choose a different collation.

For more information about changing collations, see "Changing a database from one collation to another" on page 332.

For information about creating a database with a specific collation, see "Creating a database with a named collation" on page 328.

Supplied and recommended collations

The following collations are supplied with Adaptive Server Anywhere. You can obtain the list of main collations by typing the following command at the command prompt:

dbinit -l

Collation label	Description
932JPN	Code Page 932, Japanese Shift-JIS with Microsoft extensions
936ZHO	Code Page 936, Simplified Chinese, PRC GBK 2312-80 8-bit encoding
949KOR	Code Page 949, Korean KS C 5601-1987 encoding, Wansung
950TWN	Code Page 950, Traditional Chinese, Big 5 Encoding
1250LATIN2	Code Page 1250, Windows Latin 2, Central/Eastern European
1250POL	Code Page 1250, Windows Latin 2, Polish
1251CYR	Code Page 1251, Cyrillic
1252LATIN1	Code Page 1252, Windows Latin 1, Western
1253ELL	Code Page 1253, Windows Greek, ISO8859-7 with extensions
Collation label	Description
-----------------	---
1254TRK	Code Page 1254, Windows Latin 5, Turkish, ISO 8859-9 with extensions
EUC_JAPAN	Japanese EUC JIS X 0208-1990 and JIS X 0212-1990 Encoding
EUC_CHINA	Simplified Chinese GB 2312-80 Encoding
EUC_TAIWAN	Taiwanese Big 5 Encoding
EUC_KOREA	Korean KS C 5601-1992 Encoding, Johab
ISO_1	ISO8859-1, Latin 1, Western
ISO_BINENG	Binary ordering, English ISO/ASCII 7-bit letter case mappings
ISO1LATIN1	ISO8859-1, ISO Latin 1, Western, Latin 1 Ordering
ISO9LATIN1	ISO8859-15, ISO Latin 9, Western, Latin 1 Ordering
UTF8	Unicode Transformation Format-8, 8-bit multibyte encoding for Unicode

The following collations are recommended because they provide the most likely match to the character set being used by the application: they have an appropriate sort order and support for the symbols and accented characters required for each specific language. Some languages do not have a recommended UNIX collation.

Language	Windows collations	UNIX collations
Western European (including English, French, German, Ital- ian, Portuguese, and Spanish)	1252LATIN1	ISO9LATIN1 , ISO1LATIN1
Eastern European (including Croatian, Czech, Hungarian, Ro- manian, Serbian, Slo- vakian, and Slovenian)	1250LATIN2	
Japanese	932JPN	EUC_JAPAN
Korean	949KOR	EUC_KOREA

Language	Windows collations	UNIX collations
Polish	1250POL	
Russian and Ukrainian	1251CYR	
Simplified Chinese	936ZHO	EUC_CHINA
Traditional Chinese	950TWN	EUC_TAIWAN
Turkish	1254TRK	
Greek	1253ELL	

Alternate collations

Alternate collations are available for compatibility with older versions of Adaptive Server Anywhere, or for special purposes. You can obtain a list of alternate collations by typing the following command at the command prompt:

dbinit -l+

Specifying the -1+ option lists the following collations, in addition to the list of recommended collations:

Collation label	Туре	Description
437LATIN1	OEM	Code Page 437, Latin 1, Western
437ESP	OEM	Code Page 437, Spanish
437SVE	OEM	Code Page 437, Swedish/Finnish
819CYR	ANSI	Code Page 819, Cyrillic
819DAN	ANSI	Code Page 819, Danish
819ELL	ANSI	Code Page 819, Greek
819ESP	ANSI	Code Page 819, Spanish
819ISL	ANSI	Code Page 819, Icelandic
819LATIN1	ANSI	Code Page 819, Latin 1, Western
819LATIN2	ANSI	Code Page 819, Latin 2, Central/Eastern European
819NOR	ANSI	Code Page 819, Norwegian
819RUS	ANSI	Code Page 819, Russian

Collation label	Туре	Description
819SVE	ANSI	Code Page 819, Swedish/Finnish
819TRK	ANSI	Code Page 819, Turkish
850CYR	OEM	Code Page 850, Cyrillic, Western
850DAN	OEM	Code Page 850, Danish
850ELL	OEM	Code Page 850, Greek
850ESP	OEM	Code Page 850, Spanish
850ISL	OEM	Code Page 850, Icelandic
850LATIN1	OEM	Code Page 850, Latin 1, Western
850LATIN2	OEM	Code Page 850, Latin 2, Central/Eastern European
850NOR	OEM	Code Page 850, Norwegian
850RUS	OEM	Code Page 850, Russian
850SVE	OEM	Code Page 850, Swedish/Finnish
850TRK	OEM	Code Page 850, Turkish
852LATIN2	OEM	Code Page 852, Latin 2, Central/Eastern European
852CYR	OEM	Code Page 852, Cyrillic
852POL	OEM	Code Page 852, Polish
855CYR	OEM	Code Page 855, Cyrillic
856HEB	OEM	Code Page 856, Hebrew
857TRK	OEM	Code Page 857, Turkish
860LATIN1	OEM	Code Page 860, Latin 1, Western
861ISL	OEM	Code Page 861, Icelandic
862HEB	OEM	Code Page 862, Hebrew
863LATIN1	OEM	Code Page 863, Latin 1, Western
865NOR	OEM	Code Page 865, Norwegian
866RUS	OEM	Code Page 866, Russian
869ELL	OEM	Code Page 869, Greek

Collation label	Туре	Description
920TRK	ANSI	Code Page 920, Turkish, ISO-8859-9
1252DEU	ANSI	Code Page 1251, Windows Specialty
		German, Umlaut chars not equal

Creating databases for Windows CE

Windows CE is a Unicode-based operating system. The Adaptive Server Anywhere ODBC driver supports either ASCII (8-bit) or Unicode strings, and carries out character set translation as needed. If developing Embedded SQL applications, you can use Windows API functions to get the Unicode versions of strings from the database.

When creating databases for Windows CE, you should use a collation based on the same single- or multibyte character set that Windows would use for the language of interest. For example, if you are using English, French, or German, use the 1252Latin1 collation. If you are using Japanese, use the 932JPN collation, and if you are using Korean, use the 949KOR collation.

ANSI or OEM?

Adaptive Server Anywhere collations are based on code pages that are designated as either ANSI or OEM. In most cases, using an ANSI code page is recommended. If you use OEM, choose a code page that matches the OEM code pages on your users' client machines.

You should not use a separate translation driver under any circumstance. Translation drivers interfere with the server's character set translation. Using a separate translation driver will likely result in data corruption.

For Interactive SQL and Sybase Central, the jConnect driver or the iAnywhere JDBC driver (depending on which method you use) handles character set translation.

Notes on ANSI collations

The ISO_1 collation ISO_1 is provided for compatibility with the Adaptive Server Enterprise default ISO_1 collation. The differences are as follows:

- ♦ The lower case letter sharp s (\xDF) sorts with the lower case s in Adaptive Server Anywhere, but after ss in Adaptive Server Enterprise.
- ♦ The ligatures corresponding to AE and ae (\xC6 and \xE6) sort after A and a respectively in Adaptive Server Anywhere, but after AE and ae in Adaptive Server Enterprise.

The 1252LATIN1 collation	This collation includes the euro currency symbol and several other characters (Z-with-caron and z-with-caron). For single-byte Windows operating systems, this is the recommended collation in most cases. This collation is recommended for Windows users using English or Western European languages.
	Windows NT service patch 4 changes the default character set in many locales to a new 1252 character set on which 1252LATIN1 is based. If you have this service patch, you should use this collation.
	The euro symbol sorts with the other currency symbols.
The ISO1LATIN1 collation	This collation is the same as ISO_1, but with sorting for values in the range A0-BF. For compatibility with Adaptive Server Enterprise, the ISO_1 collation has no characters for 0xA0-0xBF. However the ISO Latin 1 character set on which it is based does have characters in these positions. The ISO1LATIN1 collation reflects the characters in these positions.
	If you are not concerned with Adaptive Server Enterprise compatibility, ISO1LATIN1 is generally recommended instead of ISO_1.
	ISO1LATIN1 is recommended for Unix users using English or Western European languages.
The ISO9LATIN1 collation	This collation is the same as ISO1LATIN1, but it includes the euro currency symbol and the other new characters included in the 1252 LATIN1 collation.
	If your machine uses the ISO Latin 9 character set, then you should use this collation.

Notes on OEM collations

The following table displays the built-in collations that correspond to OEM code pages. The table and the corresponding collations were derived from several manuals from IBM concerning National Language Support, subject to the restrictions mentioned above. (This table represents the best information available at the time of writing.)

Country	Language	Primary Code Page	Primary Colla- tion	Secondary Code Page	Secondary Col- lation
Argentina	Spanish	850	850ESP	437	437ESP
Australia	English	437	437LATIN1	850	850LATIN1
Austria	German	850	850LATIN1	437	437LATIN1
Belgium	Belgian Dutch	850	850LATIN1	437	437LATIN1
Belgium	Belgian French	850	850LATIN1	437	437LATIN1

Country	Language	Primary Code Page	Primary Colla- tion	Secondary Code Page	Secondary Col- lation
Belarus	Belarussian	855	855CYR		
Brazil	Portuguese	850	850LATIN1	437	437LATIN1
Bulgaria	Bulgarian	855	855CYR	850	850CYR
Canada	Cdn French	850	850LATIN1	863	863LATIN1
Canada	English	437	437LATIN1	850	850LATIN1
Croatia	Croatian	852	852LATIN2	850	850LATIN2
Czech Republic	Czech	852	852LATIN2	850	850LATIN2
Denmark	Danish	850	850DAN		
Finland	Finnish	850	850SVE	437	437SVE
France	French	850	850LATIN1	437	437LATIN1
Germany	German	850	850LATIN1	437	437LATIN1
Greece	Greek	869	869ELL	850	850ELL
Hungary	Hungarian	852	852LATIN2	850	850LATIN2
Iceland	Icelandic	850	850ISL	861	861ISL
Ireland	English	850	850LATIN1	437	437LATIN1
Israel	Hebrew	862	862HEB	856	856HEB
Italy	Italian	850	850LATIN1	437	437LATIN1
Mexico	Spanish	850	850ESP	437	437ESP
Nether-lands	Dutch	850	850LATIN1	437	437LATIN1
New Zealand	English	437	437LATIN1	850	850LATIN1
Norway	Norwegian	865	865NOR	850	850NOR
Peru	Spanish	850	850ESP	437	437ESP
Poland	Polish	852	852LATIN2	850	850LATIN2
Portugal	Portuguese	850	850LATIN1	860	860LATIN1
Romania	Romanian	852	852LATIN2	850	850LATIN2
Russia	Russian	866	866RUS	850	850RUS
S. Africa	Afrikaans	437	437LATIN1	850	850LATIN1

Country	Language	Primary Code Page	Primary Colla- tion	Secondary Code Page	Secondary Col- lation
S. Africa	English	437	437LATIN1	850	850LATIN1
Slovak Repub- lic	Slovakian	852	852LATIN2	850	850LATIN2
Slovenia	Slovenian	852	852LATIN2	850	850LATIN2
Spain	Spanish	850	850ESP	437	437ESP
Sweden	Swedish	850	850SVE	437	437SVE
Switzerland	French	850	850LATIN1	437	437LATIN1
Switzerland	German	850	850LATIN1	437	437LATIN1
Switzerland	Italian	850	850LATIN1	437	437LATIN1
Turkey	Turkish	857	857TRK	850	850TRK
UK	English	850	850LATIN1	437	437LATIN1
USA	English	437	437LATIN1	850	850LATIN1
Venezuela	Spanish	850	850ESP	437	437ESP
Yugoslavia	Macedon-ian	852	852LATIN2	850	850LATIN2
Yugoslavia	Serbian Cyrillic	855	855CYR	852	852CYR
Yugoslavia	Serbian Latin	852	852LATIN2	850	850LATIN2

Using multibyte collations

This section describes how multibyte character sets are handled. The description applies to the supported collations and to any multibyte custom collations you may create.

Adaptive Server Anywhere provides collations using several multibyte character sets.

For more information, see "Understanding collations" on page 307.

Adaptive Server Anywhere supports variable width 8-bit character sets. In these sets, some characters are represented by one byte, and some by more than one, to a maximum of four bytes. The value of the first byte in any character indicates the number of bytes used for that character, and also indicates whether the character is a space character, a digit, or an alphabetic (alpha) character.

For the UTF8 collation, UTF-8 characters are represented by one to four bytes. For other multibyte collations, one or two bytes are used. For all provided multibyte collations, characters comprising two or more bytes are considered to be "alphabetic", such that they can be used in identifiers without requiring double quotes.

Adaptive Server Anywhere does not support 16-bit or 32-bit character sets such as UTF-16 or UTF-32.

All client libraries other than Embedded SQL are Unicode-enabled, using the UTF-16 encoding. Translation occurs between the client and the server.

Understanding character set translation

Adaptive Server Anywhere can carry out character set translation among character sets that represent the same characters, but at different positions in the character set or code page. There needs to be a degree of compatibility between the character sets for this to be possible. For example, character set translation is possible between EUC-JIS and Shift-JIS character sets, but not between EUC-JIS and OEM code page 850.

This section describes how Adaptive Server Anywhere carries out character set translation. This information is provided for advanced users, such as those who may be deploying applications or databases in a multi-character-set environment.

Character translation for database messages

Error and other messages from the database software are held in a language resource library. Localized versions of this library are provided with localized versions of Adaptive Server Anywhere.

Client application users may see messages from the database as well as data from the database. Some database messages, which are strings from the language library, may include placeholders that are filled by characters from the database. For example, if you execute a query with a column that does not exist, the returned error message is:

Column column-name not found

where column-name is filled in from the database.

To present these kinds of information to the client application in a consistent manner, even if the database is in a different character set from the language library, the database server automatically translates the characters of the messages so that they match the character set used in the database collation.

To use character translation for database messages, ensure that the collation for your database is compatible with the character set used on your computer, and with the character set used in the Adaptive Server Anywhere language resource library. The language resource library differs among different localized versions of Adaptive Server Anywhere. You must check that the characters of interest to you exist in each character set.

Messages are always translated into the database collation character set, regardless of whether character set translation is turned on or off.



A further character set translation is carried out if character set is turned on (the default) for the database server, and if the client character set is different from that used in the database collation.

Connection strings and character sets

Connection strings present a special case for character set translation. The connection string is parsed by the client library, in order to locate or start a database server. This parsing is done with no knowledge of the server character set or language.

The interface library parses the connection string as follows:

- It is broken down into its keyword = value components. This can be done independently of the character set, as long as you do not use the curly braces {} around CommLinks (LINKS) connection parameters. Instead, use the recommended parentheses (). Curly braces are valid follow bytes (bytes other than the first byte) in some multibyte character sets.
- 2. The server is located. The server name is interpreted according to the character set of the client machine. In the case of Windows operating systems, the ANSI character set is used. Extended chars can be used unless they cause character set conversion issues between the client and server machines.

For maximum compatibility among different machines, you should use server names built from alphabetic or numeric ASCII characters 0 to 127 (or 33 to 126) and the underscore, using no punctuation characters. Server names are truncated at 40 characters.

- 3. The **DatabaseName (DBN)** or **DatabaseFile (DBF)** connection parameters are interpreted in the database server character set.
- 4. Once the database is located, the remaining connection parameters are

interpreted according to its character set.

Avoiding character-set translation

There is a performance cost associated with character set translation. If you can set up an environment such that no character set translation is required, then you do not have to pay this cost, and your setup is simpler to maintain.

If you work with a single-byte character set and are concerned only with seven-bit ASCII characters (values 0 through 127), then you do not need character set translation. Even if the code pages are different in the database and on the client operating system, they are compatible over this range of characters. Many English-language installations meet these requirements. In Adaptive Server Anywhere 9.0 and higher, character set translation is turned on by default. You can turn it off using the -ct- option.

For more information, see "Turning off character set translation on a database server" on page 329.

If you do require the use of extended characters, there are other steps you may be able to take:

• If the code page on your client machine operating system matches that used in the database, no character set translation is needed for data in the database.

For example, in many environments it is appropriate to use the 1252LATIN1 collation in your database, which corresponds to the ANSI code page in many single-byte environments.

• If you are able to use a version of Adaptive Server Anywhere built for your language, and if you use the code page on your operating system, no character set translation is needed for database messages. The character set used in the Adaptive Server Anywhere message strings is as follows:

Language	Character set
English	cp1252
French	cp1252
German	cp1252
Japanese	cp932 (Shift-JIS)

Also, recall that client/server character set translation takes place by default. You can turn it off using the -ct- option.

For more information, see "-ct server option" on page 137.

Collation internals

This section describes internal technical details of collations, including the file format of collation files.

This section is of particular use if you want to create a database using a custom collation. For information on the steps involved, see "Creating a custom collation" on page 330, and "Creating a database with a custom collation" on page 331.

You can create a database using a collation different from the supplied collations. This section describes how to build databases using such custom collations.

In building multibyte custom collations, you can specify which ranges of values for the first byte signify single- and double-byte (or more) characters, and which specify space, alpha, and digit characters. However, all first bytes of value less than x40 must be single-byte characters, and no follow bytes may have values less than x40. This restriction is satisfied by all supported encodings.

Collation files may include the following elements:

- Comment lines, which are ignored by the database.
- A title line.
- ♦ A Collation section.
- An Encodings section.
- A Properties section.

Comment lines

In the collation file, spaces are generally ignored. Comment lines start with either the percent sign (%) or two dashes (-).

The title line

The first non-comment line must be of the form:

Collation collation_label (collation_name) (ase_charset) (ase_so_sensitive) (ase_so_insensitive) (java_charset)

Descriptions of arguments

Argument	Description
Collation	A required keyword.
collation_label	The collation label, which appears in the system tables as SYS.SYSCOLLATION.collation_label and SYS SYSINFO.default_collation. The label must contain no more than 10 characters.
collation_name	A text description of the collation, usually describing the character set and ordering of the collation.
ase_charset	The ASE name of a character set matching the character set of this collation. This name is used to make character set mappings when server character set translation is enabled (the default).
ase_so_sensitive	The name of an Open Client or ASE case sensitive collation matching this collation.
ase_so insensitive	The name of an Open Client or ASE case insensitive collation matching this collation.
java_charset	The name of a Java character set matching the character set of this collation. This name is used when converting character data between the Java virtual machine and the database.

For example, the 932JPN collation file contains the following collation line, with label 932JPN and name (Code Page 932, Japanese Shift-JIS with Microsoft extensions):

```
Collation 932JPN (Code Page 932, Japanese Shift-JIS with
Microsoft extensions) (cp932) (bin_cp932) (bin_cp932)
(SJIS)
```

The Collation section

After the title line, each non-comment line describes one position in the collation. The ordering of the lines determines the sort ordering used by the database, and determines the result of comparisons. Characters on lines appearing higher in the file (closer to the beginning) sort before characters that appear later.

The form of each line in the sequence is:

[sort-position] : character [[, character] . . .]

or

[sort-position] : character [lowercase uppercase]

Descriptions of arguments

sort-position

Argument	Description
sort-position	Optional. Specifies the position at which the characters on that line will sort. Smaller numbers represent a lesser value, so will sort closer to the beginning of the sorted set. Typically, the sort-position is omitted, and the char- acters sort immediately following the characters from the previous sort position.
character	The character whose sort-position is being specified.
lowercase	Optional. Specifies the lowercase equivalent of the char- acter. If not specified, the character has no lowercase equivalent.
uppercase	Optional. Specifies the uppercase equivalent of the char- acter. If not specified, the character has no uppercase equivalent.

Multiple characters may appear on one line, separated by commas (,). In this case, these characters are sorted and compared as if they were the same character.

Specifying character and Each character and sort position is specified in one of the following ways:

Specification	Description
\dnnn	Decimal number, using digits 0-9 (such as \d001)
\xhh	Hexadecimal number, using digits 0-9 and letters a-f or A-F (such as $\Begin{aligned} xB4 \end{aligned}$
'c'	Any character in place of c (such as ',')
c	Any character other than quote ([*]), back-slash (\), colon (:) or comma (,). These characters must use one of the previous forms.

The following are some sample lines for a collation:

```
% Sort some special characters at the beginning:
                            : / /
                            : \xF2
                            : \xee
                           : \xF0
                           : -
                           : ','
                            : ;
                            : ':'
                            : 1
                            % Sort some letters in alphabetical order
                            : A a A
                            : a a A
                            : B b B
                            : b b B
                            % Sort some E's from code page 850,
                            % including some accented extended characters:
                            : e e E, \x82 \x82 \x90, \x8A \x8A \xD4
                            : E e E, x90 x82 x90, xD4 x8A xD4
Other syntax notes
                         For databases using case insensitive sorting and comparison (no -c specified
                         in the dbinit command), the lower case and upper case mappings are used to
                         find the lower case and upper case characters that are sorted together.
                         For multibyte character sets, the first byte of a character is listed in the
                         collation sequence, and all characters with the same first byte are sorted
                         together, and ordered according to the value of the following bytes. For
                         example, the following is part of the Shift-JIS collation file:
                            :
                                \xfb
                            :
                               \xfc
```

: \xfd

In this collation, all characters with first byte xfc come after all characters with first byte xfb and before all characters with first byte xfd. The two-byte character xfc x01 would be ordered before the two-byte character xfc x02.

Any characters omitted from the collation are added to the end of the collation. The tool that processes the collation file issues a warning.

The Encodings section

The encodings section is optional, and follows the collation sequence. It is not useful for single-byte character sets.

The encodings section lists which characters are lead-bytes, for multibyte character sets, and what are valid follow-bytes.

For example, the Shift-JIS encodings section is as follows:

```
Encodings:
[\x00-\x80,\xa0-\xdf,\xf0-\xff]
[\x81-\x9f,\xe0-\xef][\x00-\xff]
```

The first line following the section title lists valid single-byte characters. The square brackets enclose a comma-separated list of ranges. Each range is listed as a hyphen-separated pair of values. In the Shift-JIS collation, values x00 to x80 are valid single-byte characters, but x81 is not a valid single-byte character.

The second line following the section title lists valid multibyte characters. Any combination of one byte from the second line followed by one byte from the first is a valid character. Therefore x81x00 is a valid double-byte character, but xd0 x00 is not.

The Properties section

The properties section is optional, and follows the encodings section.

If a properties section is supplied, an encodings section must be supplied also.

The properties section lists values for the first-byte of each character that represent alphabetic characters, digits, or spaces.

The Shift-JIS properties section is as follows:

```
Properties:
space: [\x09-\x0d,\x20]
digit: [\x30-\x39]
alpha: [\x41-\x5a,\x61-\x7a,\x81-\x9f,\xe0-\xef]
```

This indicates that characters with first bytes x09 to x0d, as well as x20, are to be treated as space characters, digits are found in the range x30 to x39 inclusive, and alphabetic characters in the four ranges x41-x5a, x61-x7a, x81-x9f, and xe0-xef.

International language and character set tasks

This section groups together the tasks associated with international language and character set issues.

Finding the default collation

If you do not explicitly specify a collation when creating a database, a default collation is used. The default collation depends on the operating system you are working on.

To find the default collation for your machine

- 1. Start Interactive SQL. Connect to the sample database.
- 2. Enter the following query:

```
SELECT PROPERTY( 'DefaultCollation' )
```

The default collation is returned.

For more information about this collation, see "Choosing collations" on page 307.

Configuring your character set environment

This section describes how to set up your computing environment so that character set issues are handled properly. If you set your locale environments properly, then you do not need to explicitly choose collations for your databases, and you can turn off character set translation between client and server.

For more information about turning off character set translation, see "Turning off character set translation on a database server" on page 329.

To configure your character set environment

1. Determine the default locale of each computing platform in your environment. The default locale is the character set and language of each computer. On Windows operating systems, the character set is the ANSI code page.

For more information about how to find locale information, see "Determining locale information" on page 326.

- 2. Decide whether the locale settings are appropriate for your environment.
 - For more information, see "Understanding collations" on page 307.

3. If the default settings are inappropriate, decide on a character set, language, and database collation that matches your data and avoids character set translation.

For more information, see "Avoiding character-set translation" on page 319.

- 4. Set locales on each of the machines in the environment to these values.
 - For more information, see "Setting locales" on page 327.
- 5. Create your database using the default collation. If the default collation does not match your needs, create a database using a named collation.

For more information, see "Creating a database with a named collation" on page 328, and "Changing a database from one collation to another" on page 332.

When choosing the collation for your database,

- Choose a collation that uses a character set and sort order appropriate for the data in the database. It is often the case that there are several alternative collations that meet this requirement, including some that are OEM collations and some that are ANSI collations.
- There is a performance cost, as well as extra complexity in system configuration, when you use character set translation. Choose a collation that avoids the need for character set translation.

You can avoid character set translation by using a collation sequence in the database that matches the character set in use on your client machine operating system. In the case of Windows operating systems on the client machine, choose the ANSI character set. Character set translation is enabled by default for database servers that are version 8.0 or higher of Adaptive Server Anywhere. You can turn off character set translation using the -ct- option.

For more information, see "Avoiding character-set translation" on page 319.

Determining locale information

You can determine locale information using system functions.

For more information, see "System functions" [ASA SQL Reference, page 92].

*	To determine the locale of a database server
	1. Start Interactive SQL, and connect to a database server.
	2. Execute the following statement to determine the database server character set:
	<pre>SELECT PROPERTY('CharSet')</pre>
	The query returns one of the supported character sets listed in "Character set labels" on page 304.
	3. Execute the following statement to determine the database server language:
	SELECT PROPERTY('Language')
	The query returns one of the supported languages listed in "Language label values" on page 302.
	4. Execute the following statement to determine the database server default collation:
	SELECT PROPERTY('DefaultCollation')
	The query returns one of the collations listed in "Choosing collations" on page 307.
Notes	To obtain client locale information, connect to a database server running on your current machine.
	To obtain the character set for an individual database, execute the following statement:
	SELECT DB_PROPERTY ('CharSet')
Setting locales	

You can use the default locale on your operating system, or explicitly set a locale for use by the Adaptive Server Anywhere components on your machine.

* To set the Adaptive Server Anywhere locale

1. If the default locale is appropriate for your needs, you do not need to take any action.

For more information about how to find out the default locale of your operating system, see "Determining locale information" on page 326.

2. If you need to change the locale, you can set either or both of the ASLANG and ASCHARSET environment variables:

ASCHARSET=charset;ASLANG=language_code

where *charset* is a character set label from the list in "Character set labels" on page 304, and *language_code* is a language label from the list in "Language label values" on page 302

For more information on how to set environment variables on different operating systems, see "Setting environment variables" on page 244.

Creating a database with a named collation

You may specify the collation for each database when you create the database. The default collation is inferred from the code page and sort order of the database server's computer's operating system.

To specify a database collation when creating a database (Sybase Central)

1. You can use the Create Database wizard in Sybase Central to create a database. In the wizard, there is a page where you choose a collation from a list.



To specify a database collation when creating a database (command prompt)

1. List the recommended collation sequences:

dbinit -l

The first column of the list is the collation label, which you supply when creating the database.

```
437LATIN1Code Page 437, Latin 1, Western437ESPCode Page 437, Spanish437SVECode Page 437, Swedish/Finnish819CYRCode Page 819, Cyrillic819DANCode Page 819, Danish...
```

 Create a database using the *dbinit* utility, specifying a collation sequence using the -z option. The following command creates a database with a Greek collation.

dbinit -z 1253ELL mydb.db

To specify a database collation when creating a database (SQL)

1. You can use the CREATE DATABASE statement to create a database. The following statement creates a database with a Greek collation:

CREATE DATABASE 'mydb.db' COLLATION '1253ELL'

Starting a database server using character set translation

Character set translation takes place if the client and server locales are different. Character set translation is turned on by default for database servers that are Adaptive Server Anywhere version 8.0 or higher. For database servers that are version 7.x or earlier, you must explicitly turn on character set conversion in the database server command.

To enable character set translation on a database server

1. Start the database server using the -ct+ option. For example:

dbsrv9 -ct+ asademo.db

Turning off character set translation on a database server

Character set translation is turned on by default. If you do not require character set translation, you can disable it in the database server command using the -ct- option.

To disable character set translation on a database server

1. Start the database server using the -ct- option. For example:

```
dbsrv9 -ct- asademo.db
```

For more information, see "-ct server option" on page 137.

Creating a custom collation

If none of the supplied collations meet your needs, you can modify a supplied collation to create a **custom collation**. You can then use this custom collation when creating a database.

For more information about supplied collations, see "Choosing collations" on page 307.

To create a custom collation

1. **Decide on a starting collation** You should choose a collation as close as possible to the one you want to create as a starting point for your custom collation.

For a listing of recommended collations, see "Understanding collations" on page 307. Alternatively, run *dbinit* with the -1 (lower case L) option:

dbinit -l

2. **Create a custom collation file** You do this using the Collation utility. The output is a collation file.

For example, the following statement extracts the 1252LATIN1 collation into a file named *mycustomcol.col*:

dbcollat -z 1252LATIN1 mycustomcol.col

- 3. Edit the custom collation file Open the collation file (in this case *mycustomcol.col*) in a text editor.
- 4. **Change the collation label** Make the changes you wish in the custom collation file to define the collation label. For example, if you wanted to make a custom version of the 1252LATIN1 collation, you could change the Collation line from:

```
Collation 1252LATIN1 (Code Page 1252, Windows Latin 1,
Western) (cp1252) (dictionary_iso_1) (nocase_iso_1)
(Cp1252)
```

to

```
Collation MyOrdering (Code Page 1252, My Company Ordering)
(cp1252) (dictionary_iso_1) (nocase_iso_1) (Cp1252)
```

The other entries on this line relate the Adaptive Server Anywhere collation label to the names that Java and the Sybase TDS interface give to the same collation information. If you are not using these interfaces you do not need to alter these entries.

The Collation line takes the following form:

```
Collation collation_label (collation_name) (ase_charset)
(ase_so_sensitive) (ase_so_insensitive) (java_
charset)
```

where character set label (*ase_charset*) and the two sort-order labels (*ase_so_sensitive* and *ase_so_insensitive*) state which Open Client character set and sort order is the closest to the current collation. The *java_charset* label is the closest character set known to Java.

You should edit this line to provide a new label. The label you need to change is the *collation_label*: in the example in step two, it is 1252LATIN1.

5. **Change the collation definition** Make the changes you wish in the custom collation file to define your new collation.

For more information about the collation file contents and format, see "Collation internals" on page 320.

6. **Convert the file to a SQL script** You do this using the *dbcollat* utility using the *-d* option.

For example, the following command creates the *mycustom.SQL* file from the *mycustomcol.col* collation file:

dbcollat -d mycustomcol.col mycustom.SQL

7. Add the SQL script to the script in your installation The script used when creating databases is held in the scripts subdirectory of your Adaptive Server Anywhere installation directory. Append the contents of *mycustom.SQL* to the end of *custom.SQL*.

The new collation is now in place, and can be used when creating a database.

Creating a database with a custom collation

If none of the supplied collations meet your needs, you can create a database using a custom collation. The custom collation is used in indexes and any string comparisons.

To create a database with a custom collation

1. Create a custom collation.

You must have a custom collation in place to use when creating a database.

For more information about how to create custom collations, see "Creating a custom collation" on page 330.

2. Create the new database.

Use the Initialization utility, specifying the name of your custom collation.

For example, the following command creates a database named *temp.db* using the custom collation sequence **newcol**.

dbinit -z newcol temp.db

You can also use the Initialization utility from Sybase Central.

Changing a database from one collation to another

Changing your database from one collation to another may be a good idea for any number of reasons. It can be especially useful, for example, when:

- you want to avoid character set translation across your setup.
- the characters in your database don't match the collation in your database. Using the same character set defined in your database is especially important for sorting purposes.
- you want to use a different character set. You may, for example, want to move from an OEM character set to a Windows character set.

Simply modifying the collation in an existing database is not permitted, since it would invalidate all the indexes for that database. In order to change the collation for a database, you must rebuild the database. Rebuilding a database creates a new database with new settings (including collation settings), using the old database's data.

When you change the collation for a database, there are two main scenarios to consider. The difference between the two lies in whether the character set of the data needs to be changed.

Example 1 In the first example, only the collation needs to be changed. The data should not change character sets. To resolve the collation issue, you need to rebuild the database with new collation settings using the old data.

In an English environment, consider an old database using the 850LATIN1 collation. If the database contains data inserted from a Windows 'windowed'

	application, it is likely that the data is actually from the CP1252 character set, which does not match CP850 used by the 850LATIN1 collation. This situation will often be discovered when an ORDER BY clause seems to sort accented characters incorrectly. To correct this problem, you would create a new database using the 1252LATIN1 collation, and move the data from the old database to the new database without translation, since the data is already in the character set (CP1252) that matches the new database's collation.
	The simplest way to ensure that translation does not occur is to start the server with the $-ct-$ option. Then, rebuild the database normally.
	For more information about rebuilding a database, see "Rebuilding databases" [ASA SQL User's Guide, page 539].
	For more information about specifying collations when creating databases, see "Creating a database with a named collation" on page 328.
Example 2	In the second situation, both the collation and the character set need to be changed. To resolve the collation and character set issues, you need to rebuild the database with the new collation settings, and change the character set of the data.
	Suppose that the 850LATIN1 database had been used properly such that it contains characters from the CP850 character set. However, you want to update both the collation and the character set, perhaps to avoid character set translation. You would create a new database using 1252LATIN1, and move the data from the old database to the new database with translation, thus converting the CP850 characters to CP1252.
	The translation of the database data from one character set to another occurs using the client-server translation feature of the server, which translates the character data during the communication between the client application and the server. The database's collation determines the character set for the database server side of the communication. The locale of the operating system determines the client's default character set, however, the client's character set can be overridden by the CharSet (CS) connection parameter.
	For more information about the CharSet (CS) connection parameter, see "CharSet connection parameter [CS]" on page 177.
	Since character set translation takes place during the communication between the client application and the server, an external unload or reload is necessary. An internal unload and reload does not do character set translation. Similarly, if character set translation occurs in both the unload and the reload steps, you perform the translation and then immediately undo the translation and still end up where you began. Character set translation can occur in either the unload or the reload steps, but not in both.

To convert a database from one collation to another, and translate the data's character set (using translation on reload)

1. Unload the data from the source database.

You can use the Unload utility to produce a *reload.SQL* file and a set of data files in the character set of the source database. Since we do not want any translation during this phase, ensure that character set translation is not enabled on the server running the source database. For servers before version 8, ensure that -ct is not specified. If you are using a server that is version 8 or higher, ensure that -ct- (no character set translation) is specified when the server is started.

If the unload/reload is occurring on a single machine, use the -ix option to do an internal unload and an external reload. If the unload/reload occurs across machines, use the Unload utility with the -xx option to force an external unload and an external reload.

Remember that an "external" unload or reload means that an application (dbunload and DBISQL) opens a cursor on the database and either reads or writes the data to disk. Character set translation occurs. An "internal" unload or reload means that an UNLOAD TABLE or LOAD TABLE is used so that the server reads or writes the data itself. Character set translation does *not* occur.

If you want to unload data from specific tables, use the -t option, or the Interactive SQL OUTPUT statement.

For more information on the Unload utility, see "The Unload utility" on page 533.

2. Create a target database with the appropriate collation using the Initialization utility including the -z option to specify the collation sequence for the target database.

The database should be created and reloaded using the version of the server and tools corresponding to the server that they will use to run it.

For more information on specifying collations when creating databases, see "Creating a database with a named collation" on page 328.

3. Start the target database using the server running with character set translation. You can also specify the -z option.

The -z option, while not required, allows for verification of the character sets that will be used, and that translation will occur. The server window will show character set translation settings for each connection. However, using the -z server option can cause slower performance during the reload.

For more information, see "Starting a database server using character set translation" on page 329.

4. If you generated a *reload.SQL* file in step 1, you can run the file in a command shell using a command such as the following.

dbisql -c "uid=dba;pwd=sql;charset=utf8" -codepage 950
 reload.sql

You must supply the appropriate connection parameters for your database, as well as the code page (950 in the example above) that was used to generate the *reload.SQL* file.

If the data files and *reload.sql* contain UTF-8 data, use *dbisqlc* instead of *dbisql*.

For more information about the -codepage option, see "The Interactive SQL utility" on page 492.

If you exported only table data in step 1, you can import the data using the INPUT statement in Interactive SQL.

For more information about the INPUT statement, see "INPUT statement [Interactive SQL]" [ASA SQL Reference, page 472].

CHAPTER 12

Backup and Data Recovery

About this chapter	This chapter describes how to protect your data against operating system crashes, file corruption, disk failures, and total machine failure.		
	The chapter describes how to make backups of your database, how to restore data from a backup, and how to run your server so that performance and data protection concerns are addressed.		
Contents	Торіс:	page	
	Introduction to backup and recovery	338	
	Understanding backups	343	
	Designing backup procedures	346	
	Configuring your database for data protection	355	
	Backup and recovery internals	359	
	Backup and recovery tasks	367	

Introduction to backup and recovery

A **backup** is a copy of the information in a database, held in some physically separate location from your database. If the database becomes unavailable, perhaps because of damage to a disk drive, you can **restore** it from the backup. Depending on the nature of the damage, it is often possible to restore from backups all committed changes to the database up to the time it became unavailable.

Restoring databases from a backup is one aspect of database **recovery**. The other aspect is recovery from operating system or database server crashes, and improper shutdowns. The database server checks on database startup whether the database was shut down cleanly at the end of the previous session. If it was not, the server executes an automatic recovery process to restore information. This mechanism recovers all changes up to the most recently committed transaction.

Chapter contents This chapter contains the following material:

- An introduction to backup and recovery (this section).
- Concepts and background information to help you design and use an appropriate backup strategy:

For more information, see "Understanding backups" on page 343.

• Information to help you decide the type and frequency of backups you use, and the way you run your database server so that your data is well protected:

For more information, see "Designing backup procedures" on page 346 and "Configuring your database for data protection" on page 355.

• Information for advanced users, describing Adaptive Server Anywhere internal operations related to backup and recovery:

For more information, see "Backup and recovery internals" on page 359.

- Step by step instructions for how to carry out backup and recovery tasks.
 - For more information, see "Backup and recovery tasks" on page 367

Questions and answers

To answer the question	Consider reading
What is a backup?	"Introduction to backup and recov-
	erv" on page 338

To answer the question	Consider reading
What is recovery?	"Introduction to backup and recovery" on page 338
What is a transaction log?	"The transaction log" on page 343
What are media and system failure?	"Protecting your data against fail- ure" on page 340
From what kinds of failure do backups protect my data?	"Protecting your data against fail- ure" on page 340
What tools are available for backups?	"Ways of making backups" on page 341
What types of backup are available?	"Types of backup" on page 346
What type of backup should I use?	"Designing backup procedures" on page 346
If my database file or transaction log becomes corrupt, what data may be lost?	"Protecting your data against media failure" on page 344
How are backups executed?	"Understanding backups" on page 343
How often do I carry out backups?	"Scheduling backups" on page 347
Can I schedule automatic backups?	"Scheduling backups" on page 347
My database is involved in replica- tion. How does this affect my backup strategy?	"A backup scheme for databases involved in replication" on page 350 "Backup methods for remote databases in replication installa- tions" on page 352
How can I backup to tape?	"Backing up a database directly to tape" on page 377
How do I plan a backup schedule?	"Designing a backup and recovery plan" on page 353
Can I automate backups?	"Automating Tasks Using Schedules and Events" on page 267
How can I be sure that my database file is not corrupt?	"Ensuring your database is valid" on page 354 "Validating a database" on page 369

To answer the question	Consider reading
How can I be sure that my transaction log is not corrupt?	"Validating the transaction log on database startup" on page 365 "Validating a transaction log" on page 370
How can I run my database for maxi- mum protection against failures?	"Configuring your database for data protection" on page 355
How can I ensure high availability and machine redundancy?	"Protecting against total machine failure" on page 356 "Making a live backup" on page 378
How do I carry out a backup?	"Making a full backup" on page 367
How do I restore data from backups when a failure occurs?	"Recovering from media failure on the database file" on page 378 "Recovering from media failure on an unmirrored transaction log" on page 379 "Recovering from media failure on a mirrored transaction log" on page 380

Protecting your data against failure

If your database has become unusable, you have experienced a database **failure**. Adaptive Server Anywhere provides protection against the following categories of failure:

Media failure The database file and/or the transaction log become unusable. This may occur because the file system or the device storing the database file becomes unusable, or it may be because of file corruption.

For example:

- The disk drive holding the database file or the transaction log file becomes unusable.
- The database file or the transaction log file becomes corrupted. This can happen because of hardware problems or software problems.

Backups protect your data against media failure.

For more information, see "Understanding backups" on page 343.

System failure A system failure occurs when the computer or operating system goes down while there are partially completed transactions. This

could occur when the computer is inappropriately turned off or rebooted, when another application causes the operating system to crash, or because of a power failure.

For example:

 The computer or operating system becomes temporarily unavailable while there are partially completed transactions, perhaps because of a power failure or operating system crash, or because the computer is inappropriately rebooted.

After a system failure occurs, the database server recovers automatically when you next start the database. The results of each transaction committed before the system error are intact. All changes by transactions that were not committed before the system failure are canceled.

For more information about the recovery mechanism, see "Backup and recovery internals" on page 359.

It is possible to recover uncommitted changes manually. For information, see "Recovering uncommitted operations" on page 382.

Ways of making backups

There are several distinct ways of making backups. This section introduces each of the major approaches, but does not address any issues of appropriate options.

You can make backups in the following ways:

- Sybase Central You can use the Backup Database wizard in Sybase Central to make a backup. You can access the wizard by selecting a database and choosing Backup Database from the File menu (or from the popup menu).
- utility The *dbbackup* command-line utility makes backups. For example, executing the following command at the command prompt makes backup copies of the database and transaction log in the directory *c:\backup* on the client machine:

dbbackup -c "connection-string" c:\backup

 SQL Statement You can use a SQL statement to make the database server execute a backup operation. For example, the following statement places backup copies of the database file and transaction log into the directory c:\backup on the server machine.

BACKUP DATABASE DIRECTORY 'c:\\backup' Notes

• Offline backup The above examples are all online backups, executed against a running database. You can make offline backups by copying the database files when the database is not running.

You must have DBA authority or REMOTE DBA authority to make backups of a database.

Understanding backups

To understand what files you need to back up, and how you restore databases from backups, you need to understand how the changes made to the database are stored on disk.

The database file

When a database is shut down, the database file holds a complete and current copy of all the data in the database. When a database is running, however, the database file is generally not current.

The only time a database file is guaranteed to hold a complete and current copy of all data is when a checkpoint takes place. At a checkpoint, all the contents of the database cache are written out to the disk.

The database server checkpoints a database under the following conditions:

- As part of the database shutdown operations
- When the amount of time since the last checkpoint exceeds the database option CHECKPOINT_TIME
- When the estimated time to do a recovery operation exceeds the database option RECOVERY_TIME
- When the database server is idle long enough to write all dirty pages
- When a connection issues a CHECKPOINT statement
- When the database server is running without a transaction log and a transaction is committed

Between checkpoints, you need both the database file and another file, called the transaction log, to ensure that you have a complete copy of all committed transactions.

For more information about checkpoints, see "Checkpoints and the checkpoint log" on page 360, and "How the database server decides when to checkpoint" on page 363.

The transaction log

The **transaction log** is a separate file from the database file. It stores all changes to the database. Inserts, updates, deletes, commits, rollbacks, and database schema changes are all logged. The transaction log is also called the **forward log** or the **redo log**.

	The transaction log is a key component of backup and recovery, and is also essential for data replication using SQL Remote or the Replication Agent.
	By default, all databases use transaction logs. Using a transaction log is optional, but you should always use a transaction log unless you have a specific reason not to. Running a database with a transaction log provides much greater protection against failure, better performance, and the ability to replicate data.
	For more information on how to use a transaction log to protect against media failure, see "Protecting against media failure on the database file" on page 355.
When changes are forced to disk	Like the database file, the transaction log is organized into pages : fixed size areas of memory. When a change is recorded in the transaction log, it is made to a page in memory. The change is forced to disk when the earlier of the following happens:
	• The page is full.
	• A COMMIT is executed.
	In this way, completed transactions are guaranteed to be stored on disk, while performance is improved by avoiding a write to the disk on every operation.
	Configuration options are available to allow advanced users to tune the precise behavior of the transaction log. For more information, see "COOPERATIVE_COMMITS option [database]" on page 586, and "DELAYED_COMMITS option [database]" on page 590.
Transaction log mirrors	A transaction log mirror is an identical copy of the transaction log, maintained at the same time as the transaction log. If a database has a mirrored transaction log, every database change is written to both the transaction log and the transaction log mirror. By default, databases do not have transaction log mirrors.
	A transaction log mirror provides extra protection for critical data. It enables complete data recovery in the case of media failure on the transaction log. A mirrored transaction log also enables a database server to carry out automatic validation of the transaction log on database startup.
	For more information, see "Protecting against media failure on the transaction log" on page 355.

Protecting your data against media failure

Backups protect your data against media failure as part of the data protection mechanisms Adaptive Server Anywhere provides.
For an overview of data protection mechanisms, see "Protecting your data against failure" on page 340.

The practical aspects of recovery from media failure depend on whether the media failure is on the database file or the transaction log file.

Media failure on the database file If your database file is not usable, but your transaction log is still usable, you can recover all committed changes to the database as long as you have a proper backup procedure in place. All information since the last backed up copy of the database file is held in backed up transaction logs, or in the online transaction log.

For more information on how to configure your database system, see "Protecting against media failure on the database file" on page 355.

Media failure on the transaction log file Unless you use a mirrored transaction log, you cannot recover information entered between the last database checkpoint and a media failure on the transaction log. For this reason, it is recommended that you use a mirrored transaction log in setups such as SQL Remote consolidated databases, where loss of the transaction log can lead to loss of key information, or the breakdown of a replication system.

For more information, see "Protecting against media failure on the transaction log" on page 355.

Designing backup procedures

When you make a backup, you have a set of choices to make about how to manage transaction logs. The choices you make depend on a set of factors including the following:

• Is the database involved in replication?

In this chapter, replication means SQL Remote replication, or MobiLink synchronization where *dbmlsync.exe* is running, or a database using the Replication Agent. Each of these replication methods requires access to the transaction log, and potentially to old transaction logs.

 How fast is the transaction log file growing relative to your available disk space? If the transaction log is growing quickly, you may not be able to afford to keep transaction logs available.

Types of backup

This section assumes that you are familiar with basic concepts related to backups.

For more information about concepts related to backups, see "Introduction to backup and recovery" on page 338, and "Understanding backups" on page 343.

Backups can be categorized in several ways:

Full backup and incremental backup A full backup is a backup of both the database file and of the transaction log. An incremental backup is a backup of the transaction log only. Typically, full backups are interspersed with several incremental backups.

For more information on making backups, see "Making a full backup" on page 367, and "Making an incremental backup" on page 368.

Server-side backup and client-side backup You can execute an online backup from a client machine using the Backup utility. To execute a server side backup, you execute the BACKUP statement; the database server then carries out the backup.

You can easily build server side backup into applications because it is a SQL statement. Also, server-side backup is generally faster because the data does not have to be transported across the client/server communications system.

Instructions for server-side and client-side backups are given together for each backup procedure.

Archive backup and image backup An archive backup copies the database file and the transaction log into a single archive file, typically on a tape drive. An image backup makes a copy of the database file and/or the transaction log, each as separate files. You can only carry out archive backups as server-side backups, and you can only make full backups.

You should use an archive backup if you are backing up directly to tape. Otherwise, an image backup has more flexibility for transaction log file management.

Archive backups are supported on Windows NT/2000/XP and UNIX platforms only. On Windows CE, only image backups are permitted.

For more information on archive backups, see "Backing up a database directly to tape" on page 377.

Online and offline backup Backing up a running database provides a snapshot of a consistent database, even though other users are modifying the database. An offline backup consists simply of copying the files. You should only carry out an offline backup when the database is not running, and when the database server was shut down properly.

The information in this chapter focuses on online backups.

• Live backup A live backup is a *continuous* backup of the database that helps protect against total machine failure.

For more information on when to use live backups, see "Protecting against total machine failure" on page 356.

For more information on how to make a live backup, see "Making a live backup" on page 378.

Scheduling backups

Most backup schedules involve periodic full backups interspersed with incremental backups of the transaction log. There is no simple rule for deciding how often to make backups of your data. The frequency with which you make backups depends on the importance of your data, how often it changes, and other factors.

Most backup strategies involve occasional full backups, interspersed by several incremental backups. A common starting point for backups is to carry out a weekly full backup, with daily incremental backups of the transaction log. Both full and incremental backups can be carried out online (while the database is running) or offline, on the server side or the client side. Archive backups are always full backups. The kinds of failure against which a backup schedule protects you depends not only on how often you make backups, but also on how you operate your database server.

For more information, see "Configuring your database for data protection" on page 355.

You should always keep more than one full backup. If you make a backup on top of a previous backup, a media failure in the middle of the backup leaves you with no backup at all. You should also keep some of your full backups offsite to protect against fire, flood, earthquake, theft, or vandalism.

You can use the event scheduling features of Adaptive Server Anywhere to perform online backups automatically at scheduled times.

For more information on scheduling operations such as backups, see "Automating Tasks Using Schedules and Events" on page 267.

A backup scheme for when disk space is plentiful

If disk space is not a problem on your production machine (where the database server is running) then you do not need to worry about choosing special options to manage the transaction log file. In this case, you can use a simple form of backup that makes copies of the database file and transaction log, and leaves the transaction log in place. All backups leave the database file in place.

A full backup of this kind is illustrated in the figure below. In an incremental backup, only the transaction log is backed up.



For more information on how to carry out backups of this type, see "Making a backup, continuing to use the original transaction log" on page 371.

A backup scheme for databases not involved in replication

In many circumstances, disk space limitations make it impractical to let the transaction log grow indefinitely. In this case, you can choose to delete the contents of the transaction log when the backup is complete, freeing the disk space. You should not choose this option if the database is involved in replication because replication requires access to the transaction log.

A full backup, which truncates the log file, is illustrated in the figure below. In an incremental backup, only the transaction log is backed up.



Deleting the transaction log after each incremental backup makes recovery from a media failure on the database file a more complex task as there may then be several different transaction logs since the last full backup. Each transaction log needs to be applied in sequence to bring the database up to date.

You can use this kind of backup at a database that is operating as a MobiLink consolidated database because MobiLink does not rely on the transaction log. If you are running SQL Remote or the MobiLink *dbmlsync.exe* application, you must use a scheme suitable for preserving old transaction logs, as described in the following section.

For more information on how to carry out a backup of this type, see "Making a backup, deleting the original transaction log" on page 372.

A backup scheme for databases involved in replication

If your database is part of a SQL Remote installation, the Message Agent needs access to old transactions. If it is a consolidated database, it holds the master copy of the entire SQL Remote installation, and thorough backup procedures are essential.

If your database is a primary site in a Replication Server installation, the Replication Agent requires access to old transactions. However, disk space limitations often make it impractical to let the transaction log grow indefinitely.

If your database is participating in a MobiLink setup, using the

dbmlsync.exe executable, the same considerations apply. However, if your database is a MobiLink consolidated database, you do not need old transaction logs and can use a scheme for databases not involved in replication, as described in the previous section.

In these cases, you can choose backup options to rename and restart the transaction log. This kind of backup prevents open-ended growth of the transaction log, while maintaining information about the old transactions for the Message Agent and the Replication Agent.



This kind of backup is illustrated in the figure below.

For more information on how to carry out a backup of this kind, see "Making a backup, renaming the original transaction log" on page 373.

Offline transaction logs In addition to backing up the transaction log, the backup operation renames the online transaction log to a filename of the form *YYMMDDxx.log*. This file is no longer used by the database server, but is available for the Message Agent and the Replication Agent. It is called an **offline** transaction log. A new online transaction log is started with the same name as the old online transaction log.

There is no Year 2000 issue with the two-digit year in the *YYMMDDxx.log* filenames. The names are used for distinguishability only, not for ordering. For example, the renamed log file from the first backup on December 10, 2000, is named *001210AA.log*. The first two digits indicate the year, the second two digits indicate the month, the third two digits indicate the day of the month, and the final two characters distinguish among different backups

made on the same day.

The Message Agent and the Replication Agent can use the offline copies to provide the old transactions as needed. If you set the DELETE_OLD_LOGS database option to ON, then the Message Agent and Replication Agent delete the offline files when they are no longer needed, saving disk space.

Backup methods for remote databases in replication installations

Backup procedures are not as crucial at remote databases as at the consolidated database. You may choose to rely on replication to the consolidated database as a data backup method. In the event of a media failure, the remote database would have to be re-extracted from the consolidated database, and any operations that have not been replicated would be lost. (You could use the Log Translation utility to attempt to recover lost operations.)

Even if you do choose to rely on replication to protect remote database data, backups may still need to be done periodically at remote databases to prevent the transaction log from growing too large. You should use the same option (rename and restart the log) as at the consolidated database, running the Message Agent so that it has access to the renamed log files. If you set the DELETE_OLD_LOGS option to ON at the remote database, the old log files will be deleted automatically by the Message Agent when they are no longer needed.

Automatic transaction logYou can use the -x Message Agent option to eliminate the need to rename
the transaction log on the remote computer when the database server is shut
down. The -x option renames the transaction log after it has been scanned
for outgoing messages.

Backup a database to a tape drive

All the types of backup described above are image backups. The backup copy of each file is also a file. To make backups to tape using an image backup, you have to take each backup copy and put it on tape using a disk backup utility.

You can carry out direct backup to a tape drive using an archive backup. Archive backups are always full backups. An archive backup makes copies of both the database file and the transaction log, but these copies are placed into a single file.

Solution with the second statement of the statement of the statement. For statement, see "Backing up a database directly to tape" on page 377, and "BACKUP statement" [*ASA SQL Reference*, page 263].

Solution You can restore the backup using the RESTORE statement. For information, see "Restoring an archive backup" on page 381, and "RESTORE DATABASE statement" [ASA SQL Reference, page 525].

Designing a backup and recovery plan

For reliable protection of your data, you should develop and implement a backup schedule. You should also ensure that you have a set of tested recovery instructions.

Typical schedules call for occasional full backups interspersed with several incremental backups. The frequency of each depends on the nature of the data that you are protecting.

If you use internal backups, you can use the scheduling features in Adaptive Server Anywhere to automate the task. Once you specify a schedule, the backups are carried out automatically by the database server.

For more information on automating backups, see "Automating Tasks Using Schedules and Events" on page 267.

The length of time your organization can function without access to the data in your database imposes a maximum recovery time, and you should develop and test a backup and recovery plan that meets this requirement.

You should verify that you have the protection you need against media failure on the database file and on the transaction log file. If you are running in a replication environment, you should consider using a mirrored transaction log.

For more information about media failure, see "Protecting your data" against media failure" on page 344.

External factors such as available hardware, the size of database files, recovery medium, disk space, and unexpected errors can affect your recovery time. When planning a backup strategy, you should allow additional recovery time for miscellaneous tasks that must be performed, such as entering recovery commands or retrieving and loading tapes.

> Adding more files into the recovery scenario increases the places where recovery can fail. As the backup and recovery strategy develops, you should consider checking your recovery plan.

For more information about how to implement a backup and recovery plan, see "Implementing a backup and recovery plan" on page 367.

Factors that affect recovery time

Ensuring your database is valid

Database file corruption may not be apparent until applications try to access the affected part of the database. As part of your data protection plan, you should periodically check that your database has no errors. You can do this by **validating** the database. This task requires DBA authority.

Database validation includes a scan of every row in every table and a look-up of each row in each index on the table. Validation requires exclusive access to each table in turn. For this reason, it is best to validate when there is no other activity on the database. Database validation does not validate data, continued row references, or foreign key relationships unless you perform a full validation using the -f option.

Backup copies of the database and transaction log must not be changed in any way. If there were no transactions in progress during the backup, you can check the validity of the backup database using read-only mode. However, if transactions were in progress, the database server must carry out recovery on the database when you start it. Recovery modifies the backup copy, which is not desirable.

If you can be sure that no transactions are in progress when the backup is being made, the database server does not need to carry out the recovery steps. In this case, you can carry out a validity check on the backup using the read-only database option.

Тір

Using the BACKUP statement with the WAIT BEFORE START clause ensures that no transactions are in progress when you make a backup.

If a base table in the database file is corrupt, you should treat the situation as a media failure, and recover from your previous backup. If an index is corrupt, you may want to unload the database without indexes, and reload.

For more information, see "Validating a database" on page 369, and "Validating a transaction log" on page 370.

For more information on read-only databases, see "-r server option" on page 157.

Configuring your database for data protection

There are several ways in which you can configure your database and the database server to provide protection against media failure while maintaining performance.

Protecting against media failure on the database file

When you create a database, by default the transaction log is put on the same device and in the same directory as the database. This arrangement does not protect against all kinds of media failure, and you should consider placing the transaction log in another location for production use.

For comprehensive protection against media failure, you should keep the transaction log on a different device from the database file. Some machines with two or more hard drives have only one physical disk drive with several logical drives or partitions: if you want reliable protection against media failure, make sure that you have a machine with at least two storage devices.

Placing the transaction log on a separate device can also result in improved performance by eliminating the need for disk head movement between the transaction log and the main database file.

You should not place the transaction log on a network directory. Reading and writing pages over a network gives poor performance and may result in file corruption.

For more information on creating databases, see "Creating a database" [*ASA SQL User's Guide*, page 27].

For more information on how to change the location of a transaction log, see "Changing the location of a transaction log" on page 383.

Protecting against media failure on the transaction log

It is recommended that you use a transaction log mirror when running high-volume or extremely critical applications. For example, at a consolidated database in a SQL Remote setup, replication relies on the transaction log, and if the transaction log is damaged or becomes corrupt, data replication can fail.

If you are using a mirrored transaction log, and an error occurs while trying to write to one of the logs (for example, if the disk is full), the database server stops. The purpose of a transaction log mirror is to ensure complete recoverability in the case of media failure on either log device; this purpose would be lost if the server continued with a single log.

Where to store the transaction log mirror	There is a performance penalty for using a mirrored log as each database log write operation must be carried out twice. The performance penalty depends on the nature and volume of database traffic and on the physical configuration of the database and logs.
	A transaction log mirror should be kept on a separate device from the transaction log. This improves performance. Also, if either device fails, the other copy of the log keeps the data safe for recovery.
Alternatives to a transaction log mirror	Alternatives to a mirrored transaction log are to use a disk controller that provides hardware mirroring, or operating-system level software mirroring, as provided by Windows NT/2000/XP and NetWare. Generally, hardware mirroring is more expensive, but provides better performance.
For more information	Live backups provide additional protection that has some similarities to transaction log mirroring. For more information, see "Differences between live backups and transaction log mirrors" on page 356.
	For information on creating a database with a mirrored transaction log, see "The Initialization utility" on page 485.
	For information on changing an existing database to use a mirrored

Protecting against total machine failure

You can use a **live backup** to provide a redundant copy of the transaction log that is available for restart of your system on a secondary machine in case the machine running the database server becomes unusable.

transaction log, see "The Transaction Log utility" on page 527.

A live backup runs continuously, terminating only if the server shuts down. If you suffer a system failure, the backed up transaction log can be used for a rapid restart of the system. However, depending on the load that the server is processing, the live backup may lag behind and may not contain all committed transactions.

For more information on making a live backup, see "Making a live backup" on page 378.

For information on restarting database using a live backup, see "Recovering from a live backup" on page 381.

Differences between live backups and transaction log mirrors

Both a live backup and a transaction log mirror appear to provide a secondary copy of the transaction log. However, there are several differences between using a live backup and using a transaction log mirror:

In general, a live backup is made to a different machine Running a transaction log mirror on a separate machine is not recommended. It can lead to performance and data corruption problems, and stops the database server if the connection between the machines goes down.

By running the Backup utility on a separate machine, the database server does not do the writing of the backed up log file, and the data transfer is done by the Adaptive Server Anywhere client/server communications system. Therefore, performance impact is decreased and reliability is greater.

- ◆ A live backup provides protection against a machine becoming unusable Even if a transaction log mirror is kept on a separate device, it does not provide immediate recovery if the whole machine becomes unusable. You could consider an arrangement where two machines share access to a set of disks.
- ◆ A live backup may lag behind the database server A mirrored transaction log contains all the information required for complete recovery of committed transactions. Depending on the load that the server is processing, the live backup may lag behind and may not contain all the committed transactions.

Live backups and regular The live backup of the transaction log is always the same length or shorter backups The live backup of the transaction log. When a live backup is running, and another backup restarts the transaction log (**dbbackup -r** or **dbbackup -x**), the live backup automatically truncates the live backup log and restarts the live backup at the beginning of the new transaction log.

For more information about how to make a live backup, see "Making a live backup" on page 378.

Controlling transaction log size

The size of the transaction log can determine what kind of backup is right for you, and can also affect recovery times.

You can control how fast the transaction log file grows by ensuring that all your tables have compact primary keys. If you carry out updates or deletes on tables that do not have a primary key or a unique index not allowing NULL, the entire contents of the affected rows are entered in the transaction log. If a primary key is defined, the database server needs to store only the primary key column values to uniquely identify a row. If the table contains many columns or wide columns, the transaction log pages fill up much faster if no primary key is defined. In addition to taking up disk space, this extra writing of data affects performance. If a primary key does not exist, the engine looks for a UNIQUE NOT NULL index on the table (or a UNIQUE constraint). A UNIQUE index that allows NULL is not sufficient.

Backup and recovery internals

This section describes the internal mechanisms used during backup and during automatic recovery mechanism from system failures.

Backup internals

When you issue a backup instruction, the database may be in use by many people. If you later need to use your backup to restore your database, you need to know what information has been backed up, and what has not.

The database server carries out a backup as follows:

- 1. Issue a checkpoint. Further checkpoints are disallowed until the backup is complete. While the backup is taking place, any pages modified by other connections are saved before modification in the temporary file, instead of the database file, so that the backup image is made as of the checkpoint.
- 2. Make a backup of the database file, if the backup instruction is for a full backup.
- 3. Make a backup of the transaction log.

The backup includes all operations recorded in the transaction log before the final page of the log is read. This may include instructions issued after the backup instruction was issued.

The backup copy of the transaction log is generally smaller than the online transaction log. The database server allocates space to the online transaction logs in multiples of 64K, so the transaction log file size generally includes empty pages. However, only the non-empty pages are backed up.

4. On a database created with version 8.0 or later of Adaptive Server Anywhere, if the backup instruction requires the transaction log to be truncated or renamed, uncommitted transactions are carried forward to the new transaction log.

On a database created with version 7.x or earlier of Adaptive Server Anywhere, if the backup instruction requires the transaction log to be truncated or renamed, then the database server waits until there are no uncommitted transactions before truncating or renaming the log file. If the database is busy, this wait may be significant.

For more information about renaming and truncating the transaction log, see "Designing backup procedures" on page 346.

5. Mark the backup image of the database to indicate that recovery is needed. This causes any operations that happened since the start of the

backup to be applied. It also causes operations that were incomplete at the checkpoint to be undone if they were not committed.

Restrictions during backup and recovery

The database server prevents the following operations from being executed while a backup is in progress:

- Another backup, with the exception of a live backup.
- A checkpoint, other than the one issued by the backup instruction itself.
- Any statement that causes a checkpoint. This includes data definition statements as well as the LOAD TABLE and TRUNCATE TABLE statements.

During recovery, including restoring backups, no action is permitted by other users of the database.

Checkpoints and the checkpoint log

The database file is composed of pages: fixed size portions of hard disk. The checkpoint log is located at the end of the database file. Pages are added to the checkpoint log as necessary during a session, and the entire checkpoint log is deleted at the end of the session.

Before any page is updated (made **dirty**), the database server carries out the following operations:

- It reads the page into memory, where it is held in the database cache.
- It makes a copy of the original page. These copied pages are the **checkpoint log**.



Changes made to the page are applied to the copy in the cache. For performance reasons they are not written immediately to the database file on disk.



When the cache is full, the changed page may get written out to disk. The copy in the checkpoint log remains unchanged.



A **checkpoint** is a point at which all dirty pages are written to disk. Following a checkpoint, the contents of the checkpoint log are deleted. The empty checkpoint log pages remain in the checkpoint log within a given session and can be reused for new checkpoint log data. As the checkpoint log increases in size, so does the database file.

At a checkpoint, all the data in the database is held on disk in the database file. The information in the database file matches that in the transaction log. The checkpoint represents a known consistent state of the database on disk. During recovery, the database is first recovered to the most recent checkpoint, and then changes since that checkpoint are applied.

The entire checkpoint log, including all empty checkpoint log pages, is deleted at the end of each session. Deleting the checkpoint log causes the database to shrink in size.

For more information, see "How the database server decides when to checkpoint" on page 363.

Transactions and the rollback log

As changes are made to the contents of a database, a **rollback log** is kept for the purpose of canceling changes if a transaction is rolled back or if a transaction is uncommitted when a system failure occurs. There is a separate rollback log for each connection. When a transaction is committed or rolled back, the rollback log contents for that connection are deleted. The rollback logs are stored in the database, and rollback log pages are copied into the checkpoint log along with other pages that are changed. The rollback log is also called the **undo log**.

For more information about transaction processing, see "Using Transactions and Isolation Levels" [*ASA SQL User's Guide*, page 99].

The automatic recovery process

When a database is shut down during normal operation, the database server carries out a checkpoint so that all the information in the database is held in the database file. This is a **clean** shutdown.

Each time you start a database, the database server checks whether the last shutdown was clean or the result of a system failure. If the database was not shut down cleanly, it automatically takes the following steps to recover from a system failure:

1. **Recover to the most recent checkpoint** All pages are restored to their state at the most recent checkpoint by copying the checkpoint log pages over the changes made since the checkpoint.



- 2. **Apply changes made since the checkpoint** Changes made between the checkpoint and the system failure, which are held in the transaction log, are applied.
- 3. **Rollback uncommitted transactions** Any uncommitted transactions are rolled back, using the rollback logs.

How the database server decides when to checkpoint

The priority of writing dirty pages to the disk increases as the time and the amount of work since the last checkpoint grows. The priority is determined by the following factors:

 Checkpoint Urgency The time that has elapsed since the last checkpoint, as a percentage of the checkpoint time setting of the database. The server -gc option controls the maximum desired time, in minutes, between checkpoints. You can also set the desired time using the CHECKPOINT_TIME option.

- For more information, see "-gc server option" on page 144.
- Recovery Urgency A heuristic to estimate the amount of time required to recover the database if it fails right now. The server -gr option controls the maximum desired time, in minutes, for recovery in the event of system failure. You can also set the desired time using the RECOVERY_TIME option.
 - For more information, see "-gr server option" on page 149.

The checkpoint and recovery urgencies are important only if the server does not have enough idle time to write dirty pages.

Frequent checkpoints make recovery quicker, but also create work for the server writing out dirty pages.

There are two database options that allow you to control the frequency of checkpoints. CHECKPOINT_TIME controls the maximum desired time between checkpoints and RECOVERY_TIME controls the maximum desired time for recovery in the event of system failure.

The writing of dirty pages to disk is carried out by a task within the server called the **idle I/O task**. This task shares processing time with other database tasks.

There is a threshold for the number of dirty pages, below which writing of database pages does not take place.

When the database is busy, the urgency is low, and the cache only has a few dirty pages, the idle I/O task runs at a very low priority and no writing of dirty pages takes place.

Once the urgency exceeds 30%, the priority of the idle I/O task increases. At intervals, the priority is increased again. As the urgency becomes high, the engine shifts its primary focus to writing dirty pages until the number gets below the threshold again. However, the engine only writes out pages during the idle I/O task if the number of dirty pages is greater than the threshold.

If, because of other activity in the database, the number of dirty pages falls to zero, and if the urgency is 50% or more, then a checkpoint takes place automatically since it is a convenient time.

Both the checkpoint urgency and recovery urgency values increase in value until the checkpoint occurs, at which point they drop to zero. They do not decrease otherwise.

Validating the transaction log on database startup

When a database using a transaction log mirror starts up, the database server carries out a series of checks and automatic recovery operations to confirm that the transaction log and its mirror are not corrupted, and to correct some problems if corruption is detected.

On startup, the server checks that the transaction log and its mirror are identical by carrying out a full comparison of the two files; if they are identical, the database starts as usual. The comparison of log and mirror adds to database startup time.

If the database stopped because of a system failure, it is possible that some operations were written into the transaction log but not into the mirror. If the server finds that the transaction log and the mirror are identical up to the end of the shorter of the two files, the remainder of the longer file is copied into the shorter file. This produces an identical log and mirror. After this automatic recovery step, the server starts as usual.

If the check finds that the log and the mirror are different in the body of the shorter of the two, one of the two files is corrupt. In this case, the database does not start, and an error message is generated saying that the transaction log or its mirror is invalid.

Improving performance when validating databases

The VALIDATE TABLE statement can be slow when used on large databases running on servers with a cache size too small to contain the table and its largest index. It is often the case that all pages in the table are read at least once for each index. As well, if full compares are required for index lookups, the number of page reads can be proportional to the number of rows (not pages) in the table.

If you want to reduce the time taken to validate, you can use the WITH EXPRESS CHECK option with the VALIDATE TABLE statement, or the -fx option with the *dbvalid* utility. Depending on the size of your database, the size of your cache, and the type of validation you require, these two features can significantly reduce the time taken to perform validation.

Express validation causes each row of the table to be read and all columns evaluated, as is done with traditional validation using the WITH DATA CHECK clause. Each index is completely scanned once, and checks are done to ensure that the rows referenced in the index exist in the table. The Express check option also does checks on the validity of individual index pages. The number of rows in the table must match the number of entries in the index. The Express option saves time because it does not perform individual index lookups for each row.

Because the Express check feature does not perform individual lookups, it is possible (though unlikely) for some form of index corruption to go unnoticed by the Express validation feature. If index corruption should occur, data can be recovered by unloading and rebuilding the database since validation has confirmed that all of the data can be read.

Express validation is only supported for databases created with ASA 8.0 or later.

For more information about the Express check option, see the "VALIDATE TABLE statement" [*ASA SQL Reference*, page 603], "Validating a database using the dbvalid command-line utility" on page 548, and the "sa_validate system procedure" [*ASA SQL Reference*, page 745].

Backup and recovery tasks

This section collects together instructions for tasks related to backup and recovery.

Implementing a backup and recovery plan

Regular backups and tested recovery commands are part of a comprehensive backup and recovery plan.

For more information, see "Designing a backup and recovery plan" on page 353.

* To implement a backup and recovery plan

- 1. Create and verify your backup and recovery commands.
- 2. Measure the time it takes to execute backup and recovery commands.
- 3. Document the backup commands and create written procedures outlining where your backups are kept and identify any naming convention used, as well as the kind of backups performed.
- 4. Set up your backup procedures on the production server.
- 5. Monitor backup procedures to avoid unexpected errors. Make sure any changes in the process are reflected in your documentation.

For more information about carrying out backups, see "Making a full backup" on page 367, and "Making an incremental backup" on page 368.

Making a full backup

A full backup is a backup of the database file and the transaction log file.

For more information about the difference between a full backup and an incremental backup, see "Types of backup" on page 346.

To make a full backup (overview)

- 1. Ensure that you have DBA authority on the database.
- 2. Perform a validity check on your database to ensure that it is not corrupt. You can use the Validation utility or the sa_validate stored procedure.
 - For more information, see "Validating a database" on page 369.
- 3. Make a backup of your database file and transaction log.

For information on how to carry out the backup operation, see the following:

- "Making a backup, continuing to use the original transaction log" on page 371.
- "Making a backup, deleting the original transaction log" on page 372.
- "Making a backup, renaming the original transaction log" on page 373.

Validity checking requires exclusive access to entire tables on your database. For more information and alternative approaches, see "Ensuring your database is valid" on page 354.

If you validate your backup copy of the database, make sure that you do so in read-only mode. Start the database server with the -r option to use read-only mode.

Making an incremental backup

An incremental backup is a backup of the transaction log file only. Typically, you should make several incremental backups between each full backup.

For more information about the difference between a full backup and an incremental backup, see "Types of backup" on page 346.

To make an incremental backup (overview)

- 1. Ensure that you have DBA authority on the database.
- 2. Make a backup of your transaction log only, not your database file.

For more information about how to carry out the backup operation, see the following:

- "Making a backup, continuing to use the original transaction log" on page 371.
- "Making a backup, deleting the original transaction log" on page 372.
- "Making a backup, renaming the original transaction log" on page 373.

The backup copies of the database file and transaction log file have the same names as the online versions of these files. For example, if you make a backup of the sample database, the backup copies are called *asademo.db* and *asademo.log*. When you repeat the backup statement, choose a new backup directory to avoid overwriting the backup copies.

For more information about how to make a repeatable incremental backup command by renaming the backup copy of the transaction log, see

Notes

Notes

"Renaming the backup copy of the transaction log during backup" on page 376.

Validating a database

Validating a database is a key part of the backup operation. For information, see "Ensuring your database is valid" on page 354.

For an overview of the backup operation, see "Making a full backup" on page 367.

- To check the validity of an entire database (Sybase Central)
 - 1. Connect to the database as a user with DBA authority.
 - 2. Right-click the database and choose Validate Database from the popup menu.
 - 3. Follow the instructions in the wizard.

A message box indicates whether the database is valid or not.

To check the validity of an entire database (SQL)

- 1. Connect to the database as a user with DBA authority.
- 2. Execute the sa_validate stored procedure:

call sa_validate

The procedure returns a single column, named Messages. If all tables are valid, the column contains No errors detected.

For more information, see "sa_validate system procedure" [ASA SQL Reference, page 745].

To check the validity of an entire database (Command line)

- 1. Connect to the database as a user with DBA authority.
- 2. Run the *dbvalid* utility:

```
dbvalid -c "connection_string"
```

For more information, see "The Validation utility" on page 547.

Notes

If you are checking the validity of a backup copy, you should run the database in read-only mode so that it is not modified in any way. You can only do this when there were no transactions in progress during the backup.

For information on running databases in read-only mode, see "-r server option" on page 157.

Validating a single table

You can check the validity of a single table either from Sybase Central or using a SQL statement. You must have DBA authority or be the owner of a table to check its validity.

* To check the validity of a table (Sybase Central)

- 1. Open the Tables folder.
- 2. Right-click the table and choose Validate from the popup menu.

A message box indicates whether the table is valid or not.

To check the validity of a table (SQL)

1. Execute the VALIDATE TABLE statement:

VALIDATE TABLE table_name

Notes

If errors are reported, you can drop all of the indexes and keys on a table and recreate them. Any foreign keys to the table will also need to be recreated. Another solution to errors reported by VALIDATE TABLE is to unload and reload your entire database. You should use the -u option of *dbunload* so that it does not try to use a possibly corrupt index to order the data.

Validating a transaction log

The process for validating a transaction log file depends on whether it is in use on a production database (online) or is not in use (offline, or a backup copy).

To validate an online transaction log

1. Run your database with a mirrored transaction log. The database server automatically validates the transaction log each time the database is started.

To validate an offline or backed up transaction log

1. Run the Log Translation utility (*dbtran*) against the log file. If the Log Translation utility can successfully read the log file, it is valid.

Making a backup, continuing to use the original transaction log

This task describes the simplest kind of backup, which leaves the transaction log untouched.

For more information about when to use this type of backup, see "A backup scheme for when disk space is plentiful" on page 348.

To make a backup, continuing to use the original transaction log (Sybase Central)

- 1. Start Sybase Central. Connect to the database as a user with DBA authority.
- 2. Right-click the database and choose Create Backup Images from the popup menu. The Create Backup Images wizard appears.
- 3. Click Next on the introductory page of the wizard.
- 4. Select the database that you want to back up.
- 5. On the next page, enter the name of a directory to hold the backup copies, and choose whether to carry out a complete backup (all database files) or an incremental backup (transaction log file only).
- 6. On the next page, select the option Continue To Use The Same Transaction Log.
- 7. Click Finish to start the backup.

The procedure describes a client-side backup. There are more options available for this kind of backup.

If you choose a server-side backup, and the server is running on a different machine from Sybase Central, you cannot use the Browse button to locate a directory in which to place the backups. The Browse button browses the client machine, while the backup directory is relative to the server.

To make a backup, continuing to use the original transaction log (SQL)

1. If you are using the BACKUP statement, use the following clauses only:

```
BACKUP DATABASE
DIRECTORY directory_name
[ TRANSACTION LOG ONLY ]
```

Include the TRANSACTION LOG ONLY clause if you are making an incremental backup.

To make a backup, continuing to use the original transaction log (Command line)

1. If you are using the *dbbackup* utility, use the following syntax:

dbbackup -c "connection_string" [-t] backup_directory

Include the -t option only if you are making an incremental backup.

Making a backup, deleting the original transaction log

If your database is not involved in replication, and if you have limited disk space on your online machine, you can delete the contents of the online transaction log (**truncate** the log) when you make a backup. In this case, you need to use every backup copy made since the last full backup during recovery from media failure on the database file.

For more information about when to use this type of backup, see "A backup scheme for databases not involved in replication" on page 349.

To make a backup, deleting the transaction log (Sybase Central)

- 1. Start Sybase Central. Connect to the database as a user with DBA authority.
- 2. Right-click the database and choose Create Backup Images from the popup menu. The Create Backup Images wizard appears.
- 3. Click Next on the introductory page of the wizard.
- 4. Select the database that you want to back up.
- 5. On the next page, enter the name of a directory to hold the backup copies, and choose whether to carry out a complete backup (all database files) or an incremental backup (transaction log file only).
- 6. On the next page, select the option Truncate The Transaction Log.
- 7. Click Finish to start the backup.

To make a backup, deleting the transaction log (SQL)

1. Use the BACKUP statement with the following clauses:

```
BACKUP DATABASE
DIRECTORY backup_directory
[ TRANSACTION LOG ONLY ]
TRANSACTION LOG TRUNCATE
```

Include the TRANSACTION LOG ONLY clause only if you are making an incremental backup.

The backup copies of the transaction log and database file are placed in *backup_directory*. If you enter a path, it is relative to the working directory of the database server, not your client application.

To make a backup, deleting the transaction log (Command line)

1. From the command prompt, enter the following command:

dbbackup -c "connection_string" -x [-t] backup_directory

Include the -t option only if you are making an incremental backup.

The backup copies of the transaction log and database file are placed in *backup_directory*. If you enter a path, it is relative to the directory from which you run the command.

Before the online transaction log can be erased, all outstanding transactions must terminate. If there are outstanding transactions, your backup cannot complete. For information, see "Backup internals" on page 359.

You can determine which connection has an outstanding transaction using the sa_conn_info system procedure. If necessary, you can disconnect the user with a DROP CONNECTION statement.

For more information, see "Determining which connection has an outstanding transaction" on page 375.

Making a backup, renaming the original transaction log

Notes

This set of backup options is typically used for databases involved in replication. In addition to making backup copies of the database file and transaction log, the transaction log at backup time is renamed to an offline log, and a new transaction log is started with the same name as the log in use at backup time.

For more information about when to use this set of backup options, see "A backup scheme for databases involved in replication" on page 350.

To make a backup, renaming the transaction log (Sybase Central)

- 1. Start Sybase Central. Connect to the database as a user with DBA authority.
- 2. Right-click the database and choose Create Backup Images from the popup menu. The Create Backup Images wizard appears.
- 3. Click Next on the introductory page of the wizard.
- 4. Select the database that you want to back up.
- 5. On the next page, enter the name of a directory to hold the backup copies, and choose whether to carry out a complete backup (all database files) or an incremental backup (transaction log file only).
- 6. On the next page, select the option Rename The Transaction Log.
- 7. Click Finish to start the backup.

To make a backup, renaming the transaction log (SQL)

1. Use the BACKUP statement with the following clauses:

```
BACKUP DATABASE
DIRECTORY backup_directory
[ TRANSACTION LOG ONLY ]
TRANSACTION LOG RENAME
```

Include the TRANSACTION LOG ONLY clause only if you are making an incremental backup.

The backup copies of the transaction log and database file are placed in *backup_directory*. If you enter a path, it is relative to the working directory of the database server, not your client application.

To make a backup, renaming the transaction log (Command line)

1. From the command prompt, enter the following command. You must enter the command on a single line:

dbbackup -c "connection_string" -r [-t] backup_directory

Include the -t option if you are making an incremental backup.

The backup copies of the transaction log and database file are placed in *backup_directory*. If you enter a path, it is relative to the directory from which you run the command.

Notes Before the online transaction log can be renamed, all outstanding transactions must terminate. If there are outstanding transactions, your backup will not complete. For information, see "Backup internals" on page 359.

You can determine which connection has an outstanding transaction using the sa_conn_info system procedure. If necessary, you can disconnect the user with a DROP CONNECTION statement.

For more information, see "Determining which connection has an outstanding transaction" on page 375.

Determining which connection has an outstanding transaction

If you are carrying out a backup that renames or deletes the transaction log on a database created with version 8.0 or later of Adaptive Server Anywhere, incomplete transactions are carried forward to the new transaction log.

If you are carrying out a backup that renames or deletes the transaction log on a database created with version 7.x or earlier of Adaptive Server Anywhere, however, and if there are outstanding transactions, the backup must wait until those transactions are complete before it can complete.

You can use a system procedure to determine which user has outstanding transactions. If there are not too many connections, you can also use the Console utility to determine which connection has outstanding connections.

To determine which connection has an outstanding transaction (SQL)

- 1. Connect to the database from Interactive SQL or another application that can call stored procedures.
- 2. Execute the sa_conn_info system procedure:

call sa_conn_info

3. Inspect the **UncmtOps** column to see which connection has uncommitted operations.

For more information, see "sa_conn_info system procedure" [ASA SQL Reference, page 710].

To determine which connection has an outstanding transaction (Console utility)

1. Connect to the database from the Console utility.

For example, the following command connects to the default database using user ID DBA and password SQL:

dbconsole -c "uid=DBA;pwd=SQL"

For more information, see "The Console utility" on page 471.

2. Double-click each connection, and inspect the Uncommitted Ops entry to see which users have uncommitted operations. If necessary, you can disconnect the user to enable the backup to finish.

Renaming the backup copy of the transaction log during backup

By default, the backup copy of the transaction log file has the same name as the online file. For each backup operation, you must assign a different name or location for the backup copy, or you must move the backup copy before the next backup is done.

You can make a repeatable incremental backup command by renaming the backup copy of the transaction log.

To rename the backup copy of the transaction log (SQL)

1. Use the MATCH keyword in the BACKUP statement. For example, the following statement makes an incremental backup of the asademo database to the directory *c:\backup*. The backup copy of the transaction log is called *YYMMDDxx.log*, where *YYMMDD* is the date and *xx* is a counter, starting from AA.

BACKUP DATABASE DIRECTORY 'C:\\backup' TRANSACTION LOG ONLY TRANSACTION LOG RENAME MATCH

To rename the backup copy of the transaction log (Command line)

1. Supply the -n option to *dbbackup*. For example, the following command makes an incremental backup of the sample database, renaming the backup copy of the transaction log.

dbbackup -c "uid=DBA;pwd=SQL;dbn=asademo" -r -t -n c:\backup

Notes The backup copy of the transaction log is named *YYMMDDxx.log*, where *YY* is the year, *MM* is the month, *DD* is the day of the month, and xx runs

from AA to ZZ, incrementing if there is more than one backup per day. There is no Year 2000 issue with the two-digit year in the *YYMMDDxx.log* filenames. The names are used for distinguishability only, not for ordering.

Backing up a database directly to tape

An archive backup makes a copy of the database file and transaction log file in a single archive destination. Only server-side, full backups can be made in this manner. When you make an archive backup in Sybase Central, you have the option of backing up the database directly to tape or to disk.

For more information, see "Types of backup" on page 346.

To make an archive backup to tape (Sybase Central)

- 1. Start Sybase Central. Connect to the database as a user with DBA authority.
- 2. Right-click the database and choose Backup Database from the popup menu. The Backup Database wizard appears.
- 3. Click Next on the introductory page of the wizard.
- 4. Select the database that you want to back up to tape.
- 5. On the second page of the wizard, select the On Tape To The Following Device option and type the tape drive in the text box below.
- 6. Click Finish to start the backup.

To make an archive backup to tape (SQL)

1. Use the BACKUP statement, with the following clauses:

```
BACKUP DATABASE
TO archive_root
[ ATTENDED { ON | OFF } ]
[ WITH COMMENT comment string ]
```

If you set the ATTENDED option to OFF, the backup fails if it runs out of tape or disk space. If ATTENDED is set to ON, you are prompted to take an action, such as replacing the tape, when there is no more space on the backup archive device.

Notes The BACKUP statement makes an entry in the text file *backup.syb*, in the same directory as the server executable.

For more information about restoring from an archive backup, see "Restoring an archive backup" on page 381. Example The following statement makes a backup to the first tape drive on a Windows NT machine:

```
BACKUP DATABASE
TO '\\\.\\tape0'
ATTENDED OFF
WITH COMMENT 'May 6 backup'
```

The following statement makes an archive file on disk named *c:\backup\archive.1*.

```
BACKUP DATABASE
TO 'c:\\backup\\archive'
```

Making a live backup

You carry out a live backup of the transaction log using the *dbbackup* utility with the -1 option.

For more information about live backups, see "Protecting against total machine failure" on page 356.

To make a live backup

- 1. Set up a secondary machine from which you can run the database if the online machine fails. For example, ensure that you have Adaptive Server Anywhere installed on the secondary machine.
- 2. Periodically, carry out a full backup to the secondary machine.
- 3. Run a live backup of the transaction log to the secondary machine.

dbbackup -l path\filename.log -c "connection_string"

You should normally run the *dbbackup* utility from the secondary machine.

If the primary machine becomes unusable, you can restart your database using the secondary machine. The database file and the transaction log hold the information needed to restart.

Recovering from media failure on the database file

The recovery process depends on whether you leave the transaction log untouched on incremental backup in your backup process. If your backup operation deletes or renames the transaction log, you may have to apply changes from several transaction logs. If your backup operation leaves the transaction log untouched, you need to use only the online transaction log in recovery.

For more information about the backup types discussed here, see "Designing backup procedures" on page 346.

To recover from media failure on the database file

- 1. Make an extra backup copy of the current transaction log. The database file is gone, and the only record of changes since the last backup is in the transaction log.
- 2. Create a recovery directory to hold the files you use during recovery.
- 3. Copy the database file from the last full backup to the recovery directory.
- 4. Apply the transactions held in the backed up transaction logs to the recovery database.

For each log file, in chronological order,

- Copy the log file into the recovery directory.
- Start the database server with the apply transaction log (-a) option, to apply the transaction log:

dbeng9 db_name.db -a log_name.log

The database server shuts down automatically once the transactions have been applied.

5. Copy the online transaction log into the recovery directory. Apply the transactions from the online transaction log to the recovery database.

dbeng9 db_name.db -a db_name.log

- 6. Perform validity checks on the recovery database.
- 7. Make a post-recovery backup.
- 8. Move the database file to the production directory.
- 9. Allow user access to the production database.

Recovering from media failure on an unmirrored transaction log

If your database is a primary site in a Replication Server installation, or a consolidated database in a SQL Remote installation, you should use a mirrored transaction log, or hardware equivalent.

For more information, see "Protecting against media failure on the transaction log" on page 355.

To recover from media failure on an unmirrored transaction log (partial recovery)

- 1. Make an extra backup copy of the database file immediately. With the transaction log gone, the only record of the changes between the last backup and the most recent checkpoint is in the database file.
- 2. Delete or rename the transaction log file.
- 3. Restart the database with the -f option.

dbeng9 asademo.db -f

Caution

This command should only be used when the database is not participating in a SQL Remote or Replication Server replication system. If your database is a consolidated database in a SQL Remote replication system, you may have to re-extract the remote databases.

Without the -f option, the server reports the lack of a transaction log as an error. With the option, the server restores the database to the most recent checkpoint and then rolls back any transactions that were not committed at the time of the checkpoint. A new transaction log is then created.

Recovering from media failure on a mirrored transaction log

To recover from media failure on a mirrored transaction log

- 1. Make an extra copy of the backup of your database file taken at the time the transaction log was started.
- 2. Identify which of the two files is corrupt. Run the Log Translation utility on the transaction log and on its mirror to see which one generates an error message. The Log Translation utility is accessible from Sybase Central or as the *dbtran* utility.

The following command line translates a transaction log named *asademo.log*, placing the translated output into *asademo.SQL*:

dbtran asademo.log

The Log Translation utility properly translates the intact file, and reports an error while translating the corrupt file.

- 3. Copy the correct file over the corrupt file so that you have two identical files again.
- 4. Restart the server.
Recovering from a live backup

A live backup is made to a separate machine from the primary machine that is running your production database. To restart a database from a live backup, you must have Adaptive Server Anywhere installed on the secondary machine.

For more information about live backups, see "Protecting against total machine failure" on page 356.

To restart a database using a live backup

1. Start the database server on the secondary machine with the apply transaction log (-a) option to apply the transaction log and bring the database up to date:

dbeng9 asademo.db -a filename.log

The database server shuts down automatically once the transaction log is applied.

2. Start the database server in the normal way, allowing user access. Any new activity is appended to the current transaction log.

Restoring an archive backup

If you use an archive backup (typically to tape), you use the RESTORE statement to recover your data.

For more information about making archive backups, see "Backing up a database directly to tape" on page 377.

To restore a database from an archive backup (Sybase Central)

- 1. In Sybase Central, connect to a database with DBA authority.
- 2. In the left pane, select the Adaptive Server Anywhere plug-in, and click the Utilities tab in the right pane.
- 3. In the right pane, double-click Restore Database.
- 4. Follow the instructions in the wizard.

	To restore a database from an archive backup (Interactive SQL)
	1. Start a personal database server. Use a command such as the following, which starts a server named restore:
	dbeng9 -n restore
	2. Start Interactive SQL. On the Identification tab of the Connect dialog, enter a user ID of DBA and a password of SQL . Leave all other fields on this tab blank.
	3. Click the Database tab and enter a database name of utility_db . Leave all other fields on this tab blank.
	4. Click OK to connect.
	5. Execute the RESTORE statement, specifying the archive root. At this time, you can choose to restore an archived database to its original location (default), or to a different machine with different device names using the RENAME clause.
	For more information, see "RESTORE DATABASE statement" [ASA SQL Reference, page 525].
Example	The following statement restores a database from a tape archive to the database file <i>c:\newdb\newdb.db</i> .
	<pre>RESTORE DATABASE 'c:\\newdb\\newdb.db' FROM '\\\.\\tape0'</pre>
	The following statement restores a database from an archive backup in file <i>c:\backup\archive.1</i> to the database file <i>c:\newdb\newdb.db</i> . The transaction log name and location are specified in the database.
	RESTORE DATABASE 'c:\\newdb\\newdb.db'

```
RESTORE DATABASE 'c:\\newdb\\newdb.db
FROM 'c:\\backup\\archive'
```

For more information, see "RESTORE DATABASE statement" [ASA SQL Reference, page 525].

Recovering uncommitted operations

When recovering from media failure on the database file, the transaction log is intact. Recovery reapplies all committed transactions to the database. In some circumstances, you may wish to find information about transactions that were incomplete at the time of the failure.

To recover uncommitted operations from a transaction log (Sybase Central)

- 1. Do one of the following:
 - If you are connected to a database, in the left pane, select the Adaptive Server Anywhere plug-in, and then click the Utilities tab in the right pane. In the right pane, double-click Translate Log File.
 - ◆ If you are not connected to a database, click Tools ➤ Adaptive Server Anywhere 9 ➤ Translate Log File.
- 2. Follow the instructions in the wizard.
- 3. Edit the translated log (SQL command file) in a text editor and identify the instructions you need.

To recover uncommitted operations from a transaction log (Command line)

1. Run *dbtran* to convert the transaction log into a SQL command file, using the -a option to include uncommitted transactions. For example, the following command uses *dbtran* to convert a transaction log:

dbtran -a sample.log changes.SQL

2. Edit the translated log (SQL command file) in a text editor and identify the instructions you need.

For more information on the Log Translation utility, see "The Log Translation utility" on page 508.

Note The transaction log may or may not contain changes right up to the point where a failure occurred. It does contain any changes made before the end of the most recently committed transaction that made changes to the database.

Changing the location of a transaction log

The database must not be running when you change the location of a transaction log.

For more information about how to choose the location of a transaction log, see "Protecting against media failure on the database file" on page 355.

To change the location of a transaction log (Sybase Central)

- 1. Do one of the following:
 - If you are already connected to the database associated with the log file, select the Adaptive Server Anywhere plug-in in the left pane and then open the Utilities tab in the right pane. In the right pane, double-click Change Log File Settings.
 - ◆ If you are not connected to the database, click Tools ➤ Adaptive Server Anywhere 9 ➤ Change Log File Settings.
- 2. Follow the instructions in the wizard.

To start a transaction log mirror for an existing database (Command line)

- 1. Ensure that the database is not running.
- 2. Enter the following command at the command prompt:

dblog -t new-log-file database-file

For more information about *dblog* options, see "Transaction log utility options" on page 529.

Creating a database with a transaction log mirror

You can choose to maintain a transaction log mirror when you create a database. This option is available either from the CREATE DATABASE statement, from Sybase Central, or from the *dbinit* utility.

For more information about why you may wish to use a transaction log mirror, see "Protecting against media failure on the transaction log" on page 355.

To create a database that uses a transaction log mirror (Sybase Central)

- 1. Do one of the following:
 - If you are connected to a database, in the left pane, select the Adaptive Server Anywhere plug-in and then open the Utilities tab in the right pane. In the right pane, double-click Create Database.
 - ◆ If you are not connected to a database, click Tools ➤ Adaptive Server Anywhere 9 ➤ Create Database.
- 2. Follow the instructions in the wizard.

To create a database that uses a transaction log mirror (SQL)

1. Use the CREATE DATABASE statement, with the TRANSACTION LOG clause.

For more information, see "CREATE DATABASE statement" [ASA SQL Reference, page 292].

To create a database that uses a transaction log mirror (Command line)

1. Use the *dbinit* command with the -m option. For example, the following command (which should be entered on one line) initializes a database named *company.db*, with a transaction log kept on a different device and a mirror on a third device.

dbinit -t d:\log_dir\company.log -m
e:\mirr_dir\company.mlg c:\db_dir\company.db

For more information about initialization options, see "Initialization utility options" on page 487.

Starting a transaction log mirror for an existing database

Using the Transaction Log utility, you can choose to maintain a transaction log mirror for an existing database any time the database is not running. This option is available from either Sybase Central or the *dblog* utility.

For more information about why you may wish to use a transaction log mirror, see "Protecting against media failure on the transaction log" on page 355.

To start a transaction log mirror for an existing database (Sybase Central)

1. Do one of the following:

- If you are already connected to the database associated with the log mirror, in the left pane, select the Adaptive Server Anywhere plug-in, and then open the Utilities tab in the right pane. In the right pane, double-click Change Log File Settings.
- ◆ If you are not connected to the database, click Tools ➤ Adaptive Server Anywhere 9 ➤ Change Log File Settings.
- 2. Follow the instructions in the wizard.

To start a transaction log mirror for an existing database (Command line)

- 1. Ensure that the database is not running.
- 2. Enter the following command at the command prompt:

```
dblog -m mirror-file database-file
```

```
For more information about dblog options, see "Transaction log utility options" on page 529.
```

You can also use the *dblog* utility and Sybase Central to stop a database from using a transaction log mirror.

Part III

PERMISSIONS, SECURITY, AND REPLICATION

This section describes how to use user IDs and permissions to maintain a secure database. It also describes how to replicate data with the Replication Server.

CHAPTER 13

Managing User IDs and Permissions

About this chapter	Each user of a database must have a name they type when connecting to the database, called a user ID. This chapter describes how to manage user IDs.			
Contents	Торіс:	page		
	Database permissions overview	390		
	Setting user and group options	393		
	Managing individual user IDs and permissions	394		
	Managing connected users	405		
	Managing groups	406		
	Database object names and prefixes	413		
	Using views and procedures for extra security	415		
	Changing Ownership on Nested Objects	418		
	How user permissions are assessed	420		
	Managing the resources connections use	421		
	Users and permissions in the system tables	422		

Database permissions overview

	Proper management of user IDs and permissions lets users of a database carry out their jobs effectively, while maintaining the security and privacy of information within the database.			
	You use SQL statements for assigning user IDs to new users of a database, granting and revoking permissions for database users, and finding out the current permissions of users.			
	Database permissions are assigned to user IDs. Throughout this chapter, the term user is used as a synonym for user ID. Remember, however, that you grant and revoke permissions for each user ID.			
Setting up individual user IDs	Even if there are no security concerns regarding a multi-user database, there are good reasons for setting up an individual user ID for each user. The administrative overhead is very low if a group with the appropriate permissions is set up. This chapter discusses groups of users.			
	You may want to use individual user IDs, since:			
	• The log translation utility can selectively extract the changes made by individual users from a transaction log. This is very useful when troubleshooting or piecing together what happened if data is incorrect.			
	• Sybase Central displays much more useful information with individual user IDs as you can tell which connections are which users.			
	• Row locking messages (with the BLOCKING option set to OFF) are more informative.			
DBA authority ove	rview			
	When you create a database, you also create a single usable user ID. This first user ID is DBA , and the password is initially SQL . The DBA user ID automatically has DBA authority within the database. This level of permission enables DBA users to carry out any activity in the database. They can create tables, change table structures, create new user IDs, revoke permissions from users, and so on.			
Users with DBA authority	A user with DBA authority becomes the database administrator . In this chapter, references made to the database administrator, or DBA , include <i>any user or users with DBA authority</i> .			
	Although DBA authority may be granted or transferred to other user IDs, this chapter assumes that the DBA user ID is the database administrator, and that the abbreviation <i>DBA</i> means both the DBA user ID and any user ID with DBA authority.			

Adding new users The DBA has the authority to add new users to the database. As the DBA adds users, they are also granted permissions to carry out tasks on the database. Some users may need to simply look at the database information using SQL queries, others may need to add information to the database, and others may need to modify the structure of the database itself. Although some of the responsibilities of the DBA may be handed over to other user IDs, the DBA is responsible for the overall management of the database by virtue of the DBA authority.

The DBA has authority to create database objects and assign ownership of these objects to other user IDs.

RESOURCE authority overview

RESOURCE authority is the permission to create database objects, such as tables, views, stored procedures, and triggers. RESOURCE authority may be granted only by the DBA.

To create a trigger, a user needs both RESOURCE authority and ALTER permissions on the table to which the trigger applies.

Ownership permissions overview

	The creator of a database object becomes the owner of that object. Ownership of a database object carries with it permissions to carry out actions on that object. These permissions are not assigned to users in the same way that other permissions in this chapter are assigned.		
Owners	A user who creates a new object within the database is called the owner of that object, and automatically has permission to carry out any operation on that object. The owner of a table may modify the structure of that table, for instance, or may grant permissions to other database users to update the information within the table.		
	The DBA has permission to modify any component within the database, and so could delete a table created by another user. The DBA has all the permissions regarding database objects that the owners of each object have. As well, the DBA can also create database objects for other users. In this case the owner of an object is not the user ID that executed the CREATE statement. A use for this ability is discussed in "Groups without passwords" on page 410. Despite this possibility, this chapter refers interchangeably to the owner and creator of database objects as the same person.		

Table and views permissions overview

There are several distinct permissions you may grant to user IDs concerning

tables and views:

Permission	Description
ALTER	Permission to alter the structure of a table or create a trigger on a table
DELETE	Permission to delete rows from a table or view
INSERT	Permission to insert rows into a table or view
REFER- ENCES	Permission to create indexes on a table, and to create foreign keys that reference a table
SELECT	Permission to look at information in a table or view
UPDATE	Permission to update rows in a table or view. This may be granted on a set of columns in a table only
ALL	All the above permissions

Group permissions overview

Setting permissions individually for each user of a database can be a time-consuming and error-prone process. For most databases, permission management based on groups, rather than on individual user IDs, is a much more efficient approach.

You can assign permissions to a group in exactly the same way as to an individual user. You can then assign membership in appropriate groups to each new user of the database, and they gain a set of permissions by virtue of their group membership.

ExampleFor example, you may create groups for different departments in a company
database (sales, marketing, and so on) and assign these groups permissions.
Each salesperson becomes a member of the sales group, and automatically
gains access to the appropriate areas of the database.

Each user ID can be a member of multiple groups, and they inherit all permissions from each of the groups.

Setting user and group options

In Sybase Central, configurable options for users and groups are located in the User Options and Group Options dialogs (the same dialog as for setting database options). In Interactive SQL, you can specify an option in a SET OPTION statement.

* To set options for a user or group (Sybase Central)

- 1. Connect to a database and open the Users & Groups folder.
- 2. Right-click the desired user or group and choose Options from the popup menu.
- 3. Edit the desired values.

* To set the options for a user or group (SQL)

- 1. Specify the desired properties within a SET OPTION statement.
- See also
- "Setting properties for database objects" [ASA SQL User's Guide, page 32]
- "Database Options" on page 555

Managing individual user IDs and permissions

This section describes how to create new users and grant permissions to them. For most databases, the bulk of permission management should be carried out using **groups**, rather than by assigning permissions to individual users one at a time. However, as a group is simply a user ID with special properties, read and understand this section before moving on to the discussion of managing groups.

Creating new users

In both Sybase Central and Interactive SQL, you can create new users. In Sybase Central, you can manage users or groups in the Users & Groups folder. In Interactive SQL, you can add a new user using the GRANT CONNECT statement. For both tools, you need DBA authority to create new users.

All new users are automatically added to the PUBLIC group. Once you have created a new user, you can:

- add it to other groups
- set its permissions on tables, views, and procedures
- set it as the publisher or as a remote user of the database

Initial permissions for
new usersBy default, new users are not assigned any permissions beyond connecting
to the database and viewing the system tables. To access tables in the
database, they need to be assigned permissions.

The DBA can set the permissions granted automatically to new users by assigning permissions to the special PUBLIC user group, as discussed in "Special groups" on page 411.

To create a new user (Sybase Central)

- 1. Open the Users & Groups folder.
- 2. Click the New User button on the toolbar.
- 3. Follow the instructions in the wizard.

To create a new user (SQL)

- 1. Connect to a database with DBA authority.
- 2. Execute a GRANT CONNECT TO statement.

Example Add a new user to the database with the user ID of M_Haneef and a password of welcome.

GRANT CONNECT TO M_Haneef IDENTIFIED BY welcome

See also

• "GRANT statement" [ASA SQL Reference, page 456]

Changing a password

Changing a user's password	Using the GRANT statement, you can change your password or that of another user if you have DBA authority. For example, the following command changes the password for user ID M_Haneef to new_password:		
	GRANT CONNECT TO M_Haneef IDENTIFIED BY new_password		
Changing the DBA password	The default password for the DBA user ID for all databases is SQL . You should change this password to prevent unauthorized access to your database. The following command changes the password for user ID DBA to new_password :		

GRANT CONNECT TO DBA IDENTIFIED BY new_password

Granting DBA and RESOURCE authority

You can grant DBA and RESOURCE authority in the same manner.

* To grant RESOURCE permissions to a user ID

- 1. Connect to the database as a user with DBA authority.
- 2. Type and execute the SQL statement:

GRANT RESOURCE TO userid

For DBA authority, the appropriate SQL statement is:

GRANT DBA TO userid

Only the DBA may grant DBA or RESOURCE authority to database users.

Notes

- DBA authority is very powerful, since anyone with this authority has the ability to carry out any action on the database and as well as access to all the information in the database. It is wise to grant DBA authority to only a few people.
- Consider giving users with DBA authority two user IDs, one with DBA authority and one without, so that they connect as DBA only when necessary.
- RESOURCE authority allows the user to create new database objects, such as tables, views, indexes, procedures, or triggers.

Granting permissions on tables

You can assign a set of permissions on individual tables and grant users combinations of these permissions to define their access to a table.

You can use either Sybase Central or Interactive SQL to set permissions. In Interactive SQL, you can use the following SQL statements to grant permissions on tables.

- The ALTER permission allows a user to alter the structure of a table or to create triggers on a table. The REFERENCES permission allows a user to create indexes on a table, and to create foreign keys. These permissions grant the authority to modify the database schema, and so will not be assigned to most users. These permissions do not apply to views.
- The DELETE, INSERT, and UPDATE permissions grant the authority to modify the data in a table. Of these, the UPDATE permission may be restricted to a set of columns in the table or view.
- The SELECT permission grants authority to look at data in a table, but does not give permission to alter it.
- ALL permission grants all the above permissions.

* To grant permissions on tables or columns (Sybase Central)

- 1. Connect to the database.
- 2. Open the Tables folder for that database.
- 3. Right-click a table and choose Properties from the popup menu.
- 4. On the Permissions tab of the Table property sheet, configure the permissions for the table:
 - Click Grant to select users or groups to which to grant full permissions.

- Click in the fields beside the user or group to set specific permissions. Permissions are indicated by a check mark, and grant options are indicated by a check mark with two '+' signs.
- Select a user and click the button beside References, Select, or Update to set that type of permission on individual columns.
- Select a user or group in the list and click Revoke to revoke all permissions.

Tips

Legend for the columns on the Permissions page: A=Alter, D=Delete, I=Insert, R=Reference, S=Select, U=Update.

You can also assign permissions from the user/group property sheet. To assign permissions to many users and groups at once, use the table's property sheet. To assign permissions to many tables at once, use the user's property sheet.

To grant permissions on tables or columns (SQL)

- 1. Connect to the database with DBA authority or as the owner of the table.
- 2. Execute a GRANT statement to assign the permission.

For more information, see "GRANT statement" [*ASA SQL Reference*, page 456].

Example 1 All table permissions are granted in a very similar fashion. You can grant permission to M_Haneef to delete rows from the table named sample_table as follows:

- 1. Connect to the database as a user with DBA authority, or as the owner of sample_table.
- 2. Type and execute the following SQL statement:

```
GRANT DELETE
ON sample_table
TO M_Haneef
```

Example 2 You can grant permission to M_Haneef to update the column_1 and column_2 columns only in the table named sample_table as follows:

- 1. Connect to the database as a user with DBA authority, or as the owner of sample_table.
- 2. Type and execute the following SQL statement:

```
GRANT UPDATE (column_1, column_2)
ON sample_table
TO M_Haneef
```

Table view permissions are limited in that they apply to all the data in a table (except for the UPDATE permission which may be restricted). You can fine-tune user permissions by creating procedures that carry out actions on tables, and then granting users the permission to execute the procedure.

See also

♦ "GRANT statement" [ASA SQL Reference, page 456]

Granting permissions on views

Setting permissions on views is similar to setting them on tables; for more information about the SQL statements involved, see "Granting permissions on tables" on page 396.

A user may perform an operation through a view if one or more of the following are true:

- The appropriate permission(s) on the view for the operation has been granted to the user by a DBA.
- The user has the appropriate permission(s) on all the base table(s) for the operation.
- The user was granted appropriate permission(s) for the operation on the view by a non-DBA user. This user must be either the owner of the view or have WITH GRANT OPTION of the appropriate permission(s) on the view. The owner of the view must be either:
 - a DBA.
 - a non-DBA, but also the owner of all the base table(s) referred to by the view.
 - a non-DBA, and not the owner of some or all of the base table(s) referred to by the view. However, the view owner has SELECT permission WITH GRANT OPTION on the base table(s) not owned and any other required permission(s) WITH GRANT OPTION on the base table(s) not owned for the operation.

Instead of the owner having permission(s) WITH GRANT OPTION on the base table(s), permission(s) may have been granted to PUBLIC. This includes SELECT permission on system tables.

UPDATE permissions can be granted only on an entire view. Unlike tables, UPDATE permissions cannot be granted on individual columns within a view.

* To grant permissions on views (Sybase Central)

- 1. Connect to the database.
- 2. Open the Views folder for that database.
- 3. Right-click a view and choose Properties from the popup menu.
- 4. On the Permissions tab of the View property sheet, configure the permissions for the view:
 - Click Grant to select users or groups to which to grant full permissions.
 - Click in the fields beside the user or group to set specific permissions. Permissions are indicated by a check mark, and grant options are indicated by a check mark with two '+' signs.
 - Select a user or group in the list and click Revoke to revoke all permissions.

Тір

You can also assign permissions from the User or Group property sheet. To assign permissions to many users and groups at once, use the view's property sheet. To assign permissions to many views at once, use the User's or Group's property sheet.

Behavior change

There was a behavior change with Version 5 of the software concerning the permission requirements. Previously, permissions on the underlying tables were required to grant permissions on views.

See also

• "GRANT statement" [ASA SQL Reference, page 456]

Granting users the right to grant permissions

You can assign each of the table and view permissions described in "Granting permissions on tables" on page 396 with the WITH GRANT OPTION. This option gives the right to pass on the permission to other users. In the context of groups, you can read about this feature in section "Permissions of groups" on page 409.

In Sybase Central, you can specify a grant option by displaying the property sheet of a user, group, or table, clicking the Permissions tab, and double-clicking in the fields provided so that a check mark with two '+' signs appears.

Example	You can grant permission to M_Haneef to delete rows from the table named sample_table, and the right to pass on this permission to other users, as follows:		
	1. Connect to the database as a user with DBA authority, or as the owner of sample_table:		
	2. Type and execute the SQL statement:		

```
GRANT DELETE ON sample_table
TO M_Haneef
WITH GRANT OPTION
```

For more information, see "GRANT statement" [ASA SQL Reference, page 456]

Granting permissions on procedures

The DBA or the owner of the procedure (the user ID that created the procedure) may grant permission to execute stored procedures. The EXECUTE permission is the only permission that may be granted on a procedure. This permission executes (or calls) the procedure.

The method for granting permissions to execute a procedure is similar to that for granting permissions on tables and views, discussed in "Granting permissions on tables" on page 396. However, the WITH GRANT option clause of the GRANT statement does not apply to the granting of permissions on procedures.

You can use either Sybase Central or Interactive SQL to set permissions.

To grant permissions on procedures (Sybase Central)

- 1. Connect to the database.
- 2. Open the Procedures & Functions folder for that database.
- 3. Right-click a procedure and choose Properties from the popup menu.
- 4. On the Permissions tab of the Procedure property sheet, configure the permissions for the procedure:
 - Click Grant to select users or groups to which to grant full permissions.
 - Click beside users in the Execute column to toggle between granting or not granting permission.
 - Select a user or group in the list and click Revoke to revoke all permissions.

	\mathbf{n}
-	~

You can also assign permissions from the User or Group property sheet. To assign permissions to many users and groups at once, use the procedure's property sheet. To assign permissions to many views at once, use the User's or Group's property sheet.

To grant permissions on procedures (SQL)

- 1. Connect to the database with DBA authority or as the owner of the procedure.
- 2. Execute a GRANT EXECUTE ON statement.

Example You can grant M_Haneef permission to execute a procedure named my_procedure, as follows:

- 1. Connect to the database as a user with DBA authority or as owner of my_procedure procedure.
- 2. Execute the SQL statement:
 - GRANT EXECUTE ON my_procedure TO M_Haneef

Execution permissions of Procedures execute with the permissions of their owner. Any procedure that updates information on a table will execute successfully only if the owner of the procedure has UPDATE permissions on the table.

As long as the procedure owner has the proper permissions, the procedure executes successfully when called by any user assigned permission to execute it, whether or not they have permissions on the underlying table. You can use procedures to allow users to carry out well-defined activities on a table, without having any general permissions on the table.

I See also

• "GRANT statement" [ASA SQL Reference, page 456]

Execution permissions of triggers

The server executes triggers in response to a user action. Triggers do not require permissions to be executed. When a trigger executes, it does so with the permissions of the creator of the table with which they are associated.

For more information on trigger permissions, see "Trigger execution permissions" [ASA SQL User's Guide, page 629].

Granting and revoking remote permissions

	In Sybase Central, you can manage the remote permissions of both users and groups. Remote permissions allow normal users and groups to become remote users in a SQL Remote replication setup in order to exchange replication messages with the publishing database.
Granting remote permissions	You cannot grant remote permissions to a user until you define at least one message type in the database.
	To grant remote permissions to a group, you must explicitly grant remote permissions to each user in the group. You cannot grant remote permissions to a group itself.
Revoking remote permissions	Revoking remote permissions reverts a remote user to a normal user. Revoking these permissions also automatically unsubscribes that user from all publications.
*	To grant remote permissions to users (Sybase Central)
	1. Connect to a database.
	2. Open the Users & Groups folder.
	3. Right-click the desired user and choose Change to Remote User from the popup menu.
	4. In the resulting dialog, enter the desired values.
	Once you have granted remote permissions to a user, you can subscribe it to publications:
*	To revoke remote permissions from remote users
	1. Open either the Users & Groups folder or the SQL Remote Users folder.
	2. Right-click the desired remote user and choose Revoke Remote from the popup menu.
	For more information, see "SQL Remote Concepts" [SQL Remote User's Guide, page 7].
Revoking user per	missions

Any user's permissions are a combination of those that have been granted and those that have been revoked. By revoking and granting permissions, you can manage the pattern of user permissions on a database. The REVOKE statement is the exact converse of the GRANT statement. To disallow M_Haneef from executing my_procedure, the command is:

```
REVOKE EXECUTE
ON my_procedure
FROM M_Haneef
```

The DBA or the owner of the procedure must issue this command.

Permission to delete rows from sample_table may be revoked by issuing the command:

```
REVOKE DELETE
ON sample_table
FROM M_Haneef
```

For more information on using Sybase Central to grant or revoke permission, see the following sections:

- "Granting permissions on tables" on page 396
- "Granting permissions on views" on page 398
- "Granting permissions on procedures" on page 400

Deleting users from the database

You can delete a user from the database using both Sybase Central and Interactive SQL. The user being removed cannot be connected to the database during this procedure.

Deleting a user also deletes all database objects (such as tables) that they own.

Only the DBA can delete a user.

To delete a user from the database (Sybase Central)

- 1. Open the Users & Groups folder.
- 2. Right-click the desired user and choose Delete from the popup menu.

Tips

You cannot delete users when you select them within a group container.

	To delete a user from the database (SQL)
	1. Connect to a database.
	2. Execute a REVOKE CONNECT FROM statement.
Example	Remove the user M_Haneef from the database.
	REVOKE CONNECT FROM M_Haneef
	See also
	• "REVOKE statement" [ASA SQL Reference, page 530]
	 "Revoking user permissions" on page 402

• "Deleting groups from the database" on page 412

Managing connected users

If you are working with Sybase Central, you can keep track of all users connected to the database. You can view properties of these connected users, and you can disconnect them if you want.

* To display a list of all users connected to a database

- 1. In Sybase Central, connect to a database.
- 2. Click the database in the left pane, and click the Connected Users tab in the right pane.

This tab displays all other users currently connected to a given database (including yourself), regardless of the client that they used to connect (Sybase Central, Interactive SQL, a custom client application, etc.).

✤ To inspect the properties of a user's connection to a database

- 1. In Sybase Central, connect to a database.
- 2. Click the database in the left pane, and click the Connected Users tab in the right pane.
- 3. Right-click the desired user and choose Properties from the popup menu.
- 4. Inspect the desired properties.

* To disconnect users from a database

- 1. In Sybase Central, connect to a database.
- 2. Click the database in the left pane, and click the Connected Users tab in the right pane.
- 3. Right-click the desired user and choose Disconnect from the popup menu.

Managing groups

	Once you understand how to manage permissions for individual users (as described in the previous section), working with groups is straightforward. A user ID identifies a group, just like it does a single user. However, a group user ID has the permission to have members.		
DBA, RESOURCE, and GROUP permissions	When you grant permissions to a group or revoke permissions from a group for tables, views, and procedures, all members of the group inherit those changes. The DBA, RESOURCE, and GROUP permissions are not inherited: you must assign them individually to each of the individual user IDs requiring them.		
	A group is simply a user ID with special permissions. You grant permissions to a group and revoke permissions from a group in exactly the same manner as any other user, using the commands described in "Managing individual user IDs and permissions" on page 394.		
	You can construct a hierarchy of groups, where each group inherits permissions from its parent group. That means that a group can also be a member of a group. As well, each user ID may belong to more than one group, so the user-to-group relationship is many-to-many.		
	The ability to create a group without a password enables you to prevent anybody from signing on using the group user ID.		
	For more information about this security feature, see "Groups without passwords" on page 410.		
	For more information on altering database object properties, see "Setting properties for database objects" [ASA SQL User's Guide, page 32].		
	For more information about granting remote permissions for groups, see "Granting and revoking remote permissions" on page 402.		
Creating groups			
	You can create a new group in both Sybase Central and Interactive SQL. You need DBA authority to create a new group.		

The GROUP permission, which gives the user ID the ability to have members, is not inherited by members of a group. Otherwise, every user ID would automatically be a group as a consequence of its membership in the special PUBLIC group.

*	To create a	new	group	(Sybase	Central)
---	-------------	-----	-------	---------	----------

- 1. Open the Users & Groups folder.
- 2. Click the New Group button on the toolbar.
- 3. Follow the instructions in the wizard.

To create a new group (SQL)

- 1. Connect to a database.
- 2. Execute a GRANT GROUP TO statement. If the user ID you cite in this statement has not already been created, you need to create it first.

Create the user ID personnel.

GRANT CONNECT TO personnel IDENTIFIED BY group_password

Make the user ID personnel a group.

GRANT GROUP TO personnel

See also

- "GRANT statement" [ASA SQL Reference, page 456]
- "Creating new users" on page 394

Granting group membership to existing users or groups

You can add existing users to groups or add groups to other groups in both Sybase Central and Interactive SQL. In Sybase Central, you can control group membership on the property sheets of users or groups. In Interactive SQL, you can make a user a member of a group with the GRANT statement.

When you assign a user membership in a group, they inherit all the permissions on tables, views, and procedures associated with that group.

Only the DBA can grant membership in a group.

Example

*	To add a user or group to another group (Sybase Central)
	1. Open the Users & Groups folder.
	2. Right-click the user/group that you want to add to another group and choose Properties from the popup menu.
	3. Click the Membership tab of the property sheet.
	4. Click Join to open the Join Groups dialog.
	5. Select the desired group and click OK. The user or group associated with the property sheet is then added to this desired group.
•	To add a user or group to another group (SQL)
	1. Connect to a database.
	Execute a GRANT MEMBERSHIP IN GROUP statement, specifying the desired group and the users involved.
Example	Grant the user M_Haneef membership in the personnel group:
	GRANT MEMBERSHIP IN GROUP personnel TO M_Haneef
	See also

- "GRANT statement" [ASA SQL Reference, page 456]
- "Creating new users" on page 394

Revoking group membership

You can remove users or groups from a group in both Sybase Central and Interactive SQL.

Removing a user or group from a group does *not* delete them from the database (or from other groups). To do this, you must delete the user/group itself.

Only the DBA can revoke membership in a group.

To remove a user or group from another group (Sybase Central)

- 1. Open the Users & Groups folder.
- 2. Right-click the desired user/group and choose Properties from the popup menu.
- 3. Click the Membership tab of the property sheet.
- 4. Select the desired group to leave and click Leave. The user or group associated with the property sheet is then removed from this desired group.

Тір

You can perform this action by opening the Users & Groups folder, rightclicking the user or group that is currently a member of another group, and choosing Leave.

To remove a user or group from another group (SQL)

- 1. Connect to a database.
- 2. Execute a REVOKE MEMBERSHIP IN GROUP statement, specifying the desired group and the users involved.

Remove the user M_Haneef from the personnel group:

REVOKE MEMBERSHIP IN GROUP personnel FROM M_Haneef

See also

- "REVOKE statement" [ASA SQL Reference, page 530]
- "Creating new users" on page 394
- "Deleting users from the database" on page 403
- "Deleting groups from the database" on page 412

Permissions of groups

Example

You may grant permissions to groups in exactly the same way as to any other user ID. Permissions on tables, views, and procedures are inherited by members of the group, including other groups and their members. Some complexities to group permissions exists, that database administrators need to keep in mind. NotesMembers of a group do not inherit the DBA, RESOURCE, and GROUP
permissions. Even if the personnel user ID has RESOURCE permissions,
the members of personnel do not have RESOURCE permissions.

Ownership of database objects is associated with a single user ID and is not inherited by group members. If the user ID personnel creates a table, then the personnel user ID is the owner of that table and has the authority to make any changes to the table, as well as to grant privileges concerning the table to other users. Other user IDs who are members of personnel are not the owners of this table, and do not have these rights. Only granted permissions are inherited. For example, if the DBA or the personnel user ID explicitly grants SELECT authority to the personnel user ID, all group members do have select access to the table.

Referring to tables owned by groups

Groups are used for finding tables and procedures in the database. For example, the query

SELECT * FROM SYSGROUPS

always finds the table SYSGROUPS, because all users belong to the PUBLIC group, and PUBLIC belongs to the SYS group which owns the SYSGROUPS table. (The SYSGROUPS table contains a list of *group_name*, *member_name* pairs representing the group memberships in your database.)

If a table named employees is owned by the user ID personnel, and if **M_Haneef** is a member of the personnel group, then **M_Haneef** can refer to the employees table simply as employees in SQL statements. Users who are not members of the personnel group need to use the qualified name personnel.employees.

Creating a group to own A good practice to follow that allows everyone to access the tables without qualifying names, is to create a group whose only purpose is to own the tables. Do not grant any permissions to this group, but make all users members of the group. You can then create permission groups and grant users membership in these permission groups as warranted.

For an example, see the section "Database object names and prefixes" on page 413.

Groups without passwords

Users connected to a group's user ID have certain permissions. This user ID can grant and revoke membership in the group. Also, this user would have ownership permissions over any tables in the database created in the name of

the group's user ID.

It is possible to set up a database so that only the DBA handles groups and their database objects, rather than permitting other user IDs to make changes to group membership. You can do this by disallowing connection as the group's user ID when creating the group. To do this, type the GRANT CONNECT statement without a password. Thus:

GRANT CONNECT TO personnel

creates a user ID personnel. This user ID can be granted group permissions, and other user IDs can be granted membership in the group, inheriting any permissions that have been given to personnel. However, nobody can connect to the database using the personnel user ID, because it has no valid password.

The user ID personnel can be an owner of database objects, even though no user can connect to the database using this user ID. The CREATE TABLE statement, CREATE PROCEDURE statement, and CREATE VIEW statement all allow the owner of the object to be specified as a user other than that executing the statement. Only the DBA can carry out this assignment of ownership.

Special groups

	When you create a database, the SYS and PUBLIC groups are also automatically created. Neither of these groups has passwords, so it is not possible to connect to the database as either SYS or as PUBLIC. However, the two groups serve important functions in the database.
The SYS group	The SYS group owns the system tables and views for the database, which contain the full description of database structure, including all database objects and all user IDs.
	For more information about the system tables and views, together with a description of access to the tables, see the chapters "System Tables" [<i>ASA SQL Reference</i> , page 611], and also "System Views" [<i>ASA SQL Reference</i> , page 695].
The PUBLIC group	The PUBLIC group has CONNECT permissions to the database and SELECT permission on the system tables. As well, the PUBLIC group is a member of the SYS group, and has read access for some of the system tables and views, so any user of the database can find out information about the database schema. If you wish to restrict this access, you can REVOKE PUBLIC's membership in the SYS group.
	Any new user ID is automatically a member of the PUBLIC group and

inherits any permissions specifically granted to that group by the DBA. You can also REVOKE membership in PUBLIC for users if you wish.

Deleting groups from the database

You can delete a group from the database using both Sybase Central and Interactive SQL.

Deleting users or groups from the database is different from *removing* them from other groups. Deleting a group from the database does *not* delete its members from the database, although they lose membership in the deleted group.

Only the DBA can delete a group.

To delete a group from the database (Sybase Central)

- 1. Open the Users & Groups folder.
- 2. Right-click the desired group and choose Delete from the popup menu.

To delete a group from the database (SQL)

- 1. Connect to a database.
- 2. Execute a REVOKE CONNECT FROM statement.

Example

Remove the group personnel from the database.

REVOKE CONNECT FROM personnel

- See also
- "REVOKE statement" [ASA SQL Reference, page 530]
- "Revoking user permissions" on page 402
- "Deleting users from the database" on page 403

Database object names and prefixes

The name of every database object is an identifier. The rules for valid identifiers appear in "Identifiers" [*ASA SQL Reference*, page 7].

In queries and sample SQL statements throughout this guide, database objects from the sample database are generally referred to using their simple name. For example:

SELECT * FROM employee

Tables, procedures, and views all have an owner. The owner of the tables in the sample database is the user ID **DBA**. In some circumstances, you must prefix the object name with the owner user ID, as in the following statement.

```
SELECT *
FROM "DBA".employee
```

The employee table reference is said to be **qualified**. In other circumstances it is sufficient to give the object name. This section describes when you need to use the owner prefix to identify tables, views and procedures, and when you do not.

When referring to a database object, you require a prefix unless:

- You are the owner of the database object.
- The database object is owned by a group ID of which you are a member.

Consider the following example of a corporate database. The user ID company created all the tables, and since the this user ID belongs to the database administrator, it therefore has DBA authority.

```
GRANT CONNECT TO company
IDENTIFIED BY secret;
GRANT DBA TO company;
```

Example

The company user ID created the tables in the database.

CONNECT USER company IDENTIFIED BY secret; CREATE TABLE company.Customers (...); CREATE TABLE company.Products (...); CREATE TABLE company.Orders (...); CREATE TABLE company.Invoices (...); CREATE TABLE company.Employees (...); CREATE TABLE company.Salaries (...);

Not everybody in the company should have access to all information. Consider two user IDs in the sales department, Joe and Sally, who should

413

have access to the Customers, Products and Orders tables. To do this, you create a Sales group.

```
GRANT CONNECT TO Sally IDENTIFIED BY xxxxx;
GRANT CONNECT TO Joe IDENTIFIED BY xxxxx;
GRANT CONNECT TO Sales IDENTIFIED BY xxxxx;
GRANT GROUP TO Sales;
GRANT ALL ON Customers TO Sales;
GRANT ALL ON Orders TO Sales;
GRANT SELECT ON Products TO Sales;
GRANT MEMBERSHIP IN GROUP Sales TO Sally;
GRANT MEMBERSHIP IN GROUP Sales TO Joe;
```

Now Joe and Sally have permission to use these tables, but they still have to qualify their table references because the table owner is company, and Sally and Joe are not members of the company group:

SELECT * FROM company.customers

To rectify the situation, make the Sales group a member of the company group.

```
GRANT GROUP TO company;
GRANT MEMBERSHIP IN GROUP company TO Sales;
```

Now Joe and Sally, being members of the Sales group, are indirectly members of the company group, and can reference their tables without qualifiers. The following command now works:

SELECT * FROM Customers

Joe and Sally do not have any extra permissions because of their membership in the company group. The company group has not been explicitly granted any table permissions. (The company user ID has implicit permission to look at tables like Salaries because it created the tables and has DBA authority.) Thus, Joe and Sally still get an error executing either of these commands:

```
SELECT *
FROM Salaries;
SELECT *
FROM company.Salaries
```

In either case, Joe and Sally do not have permission to look at the Salaries table.

Note

Using views and procedures for extra security

For databases that require a high level of security, defining permissions directly on tables has limitations. Any permission granted to a user on a table applies to the whole table. There are many cases when users' permissions need to be shaped more precisely than on a table-by-table basis. For example:

- It is not desirable to give access to personal or sensitive information stored in an employee table to users who need access to other parts of the table.
- You may wish to give sales representatives update permissions on a table containing descriptions of their sales calls, but limit such permissions to their own calls.

In these cases, you can use views and stored procedures to tailor permissions to suit the needs of your organization. This section describes some of the uses of views and procedures for permission management.

For more information about how to create views, see "Working with views" [ASA SQL User's Guide, page 50].

For more information about view permissions, see "Granting permissions on views" on page 398.

Using views for tailored security

	Views are computed tables that contain a selection of rows and columns from base tables. Views are useful for security when it is appropriate to give a user access to just one portion of a table. The portion can be defined in terms of rows or in terms of columns. For example, you may wish to disallow a group of users from seeing the salary column of an employee table, or you may wish to limit a user to see only the rows of a table they have created.
Example	The Sales manager needs access to information in the database concerning employees in the department. However, there is no reason for the manager to have access to information about employees in other departments.
	This example describes how to create a user ID for the sales manager, create views that provides the information she needs, and grants the appropriate permissions to the sales manager user ID.
	1. Create the new user ID using the GRANT statement. While logged in as a user ID with DBA authority, enter the following:

```
CONNECT "DBA"
IDENTIFIED by SQL ;
GRANT CONNECT
TO SalesManager
IDENTIFIED BY sales
```

2. Define a view which only looks at sales employees as follows:

```
CREATE VIEW emp_sales AS
SELECT emp_id, emp_fname, emp_lname
FROM "DBA".employee
WHERE dept_id = 200
```

The table should therefore be identified as DBA.employee, with the owner of the table explicitly identified, for the SalesManager user ID to be able to use the view. Otherwise, when SalesManager uses the view, the SELECT statement refers to a table that user ID does not recognize.

3. Give SalesManager permission to look at the view:

```
GRANT SELECT
ON emp_sales
TO SalesManager
```

You use exactly the same command to grant permission on views and on tables.

The next example creates a view which allows the Sales Manager to look at a summary of sales orders. This view requires information from more than one table for its definition:

1. Create the view.

```
CREATE VIEW order_summary AS
SELECT order_date, region, sales_rep, company_name
FROM "DBA".sales_order
KEY JOIN "DBA".customer
```

2. Grant permission for the Sales Manager to examine this view.

```
GRANT SELECT
ON order_summary
TO SalesManager
```

3. To check that the process has worked properly, connect to the SalesManager user ID and look at the views you created:

```
CONNECT SalesManager
IDENTIFIED BY sales ;
SELECT *
FROM "DBA".emp_sales ;
SELECT *
FROM "DBA".order_summary ;
```

Example 2
No permissions have been granted to the Sales Manager to look at the underlying tables. The following commands produce permission errors.

```
SELECT * FROM "DBA".employee ;
SELECT * FROM "DBA".sales_order
```

Other permissions on
viewsThe previous example shows how to use views to tailor SELECT
permissions. You can grant INSERT, DELETE, and UPDATE permissions
on views in the same way.

For more information about allowing data modification on views, see "Using views" [ASA SQL User's Guide, page 52].

Using procedures for tailored security

While views restrict access on the basis of data, procedures restrict the actions a user may take. As described in "Granting permissions on procedures" on page 400, a user may have EXECUTE permission on a procedure without having any permissions on the table or tables on which the procedure acts.

Strict security For strict security, you can disallow all access to the underlying tables, and grant permissions to users or groups of users to execute certain stored procedures. This approach strictly defines the manner in which data in the database can be modified.

Changing Ownership on Nested Objects

Views and procedures can access underlying objects that are owned by different users. For example, if usera, userb, userc, and userd were four different users, userd.viewd could be based on userc.viewc, which could be based on userb.viewb, which could be based on usera.table. Similarly for procedures, userd.procd could call userc.procc, which could call userb.procb, which could insert into usera.tablea.

The following Discretionary Access Control (DAC) rules apply to nested views and tables:

- To create a view, the user must have SELECT permission on all of the base objects (for example tables and views) in the view.
- To access a view, the view owner must have been granted the appropriate permission on the underlying tables or views with the GRANT OPTION and the user must have been granted the appropriate permission on the view.
- Updating with a WHERE clause requires both SELECT and UPDATE permission.
- If a user owns the tables in a view definition, the user can access the tables through a view, even if the user is not the owner of the view and has not been granted access on the view.

The following DAC rules apply to nested procedures:

- A user does not require any permissions on the underlying objects (for example tables, views or procedures) to create a procedure.
- For a procedure to execute, the owner of the procedure needs the appropriate permissions on the objects that the procedure references.
- Even if a user owns all the tables referenced by a procedure, the user will not be able to execute the procedure to access the tables unless the user has been granted EXECUTE permission on the procedure.

Following are some examples that describe this behavior.

Example 1: User1 creates table1, and user2 creates view2 on table1

- User1 can always access table1, since user1 is the owner.
- User1 can always access table1 through view2, since user1 is the owner of the underlying table. This is true even if user2 does not grant permission on view2 to user1.

- User2 can access table1 directly or through view2 if user1 grants permission on table1 to user2.
- User3 can access table1 if user1 grants permission on table1 to user3
- User3 can access table1 through view2 if user1 grants permission on table1 to user2 with grant option AND user2 grants permission on view2 to user3.

Example 2: User2 creates procedure2 that accesses table1

• User1 can access table1 through procedure2 if user2 grants EXECUTE permission on procedure2 to user1. Note that this is different from the case of view2, where user1 did not need permission on view2.

Example 3: User1 creates table1, user2 creates table2, and user3 creates view3 joining table1 and table2

- User3 can access table1 and table2 through view3 if user1 grants permission on table1 to user3 AND user2 grants permission on table2 to user3.
- If user3 has permission on table1 but not on table2, then user3 cannot use view3, even to access the subset of columns belonging to table1.
- User1 or user2 can use view3 if (a) user1 grants permission with grant option on table1 to user3, (b) user2 grants permission with grant option on table2 to user3, AND (c) user3 grants permission on view3 to that user.

How user permissions are assessed

Groups do introduce complexities in the permissions of individual users. Suppose user M_Haneef has SELECT and UPDATE permissions on a specific table individually, but is also a member of two groups. Suppose one of these groups has no access to the table at all, and one has only SELECT access. What are the permissions in effect for this user?

Adaptive Server Anywhere decides whether a user ID has permission to carry out a specific action in the following manner:

- 1. If the user ID has DBA authority, the user ID can carry out any action in the database.
- 2. Otherwise, permission depends on the permissions assigned to the individual user. If the user ID has been granted permission to carry out the action, then the action proceeds.
- 3. If no individual settings have been made for that user, permission depends on the permissions of each of the groups to which the member belongs. If any of these groups has permission to carry out the action, the user ID has permission by virtue of membership in that group, and the action proceeds.

This approach minimizes problems associated with the order in which permissions are set.

Managing the resources connections use

	Building a set of users and groups allows you to manage permissions on a database. Another aspect of database security and management is to limit the resources an individual user can use.
	For example, you may wish to prevent a single connection from taking too much of the available memory or CPU resources, so you can avoid a having a connection slow down other users of the database.
	Adaptive Server Anywhere provides a set of database options that the DBA can use to control resources. These options are called resource governors .
Setting options	You can set database options using the SET OPTION statement, with the following syntax:
	SET [TEMPORARY] OPTION [userid. PUBLIC.]option-name = [option-value]
	For more information about options, see "Database Options" on page 555. For information on the SET OPTION statement, see "SET OPTION statement" [<i>ASA SQL Reference</i> , page 556].
Resources that can be	You can use the following options to manage resources:
managed	◆ JAVA_HEAP_SIZE Sets the maximum size (in bytes) of the part of the memory allocated to Java applications on a per connection basis.
	• MAX_CURSOR_COUNT Limits the number of cursors for a connection.
	• MAX_STATEMENT_COUNT Limits the number of prepared statements for a connection.
	• BACKGROUND_PRIORITY Limits the impact requests on the current connection have on the performance of other connections.
	Database option settings are not inherited through the group structure.

Users and permissions in the system tables

The database system tables and system views stores information about the current users of a database and about their permissions.

For more information about each of these tables, see "System Tables" [ASA SQL Reference, page 611].

The special user ID SYS owns the system tables. You cannot connect to the SYS user ID.

The DBA has SELECT access to all system tables, just as to any other tables in the database. The access of other users to some of the tables is limited. For example, only the DBA has access to the SYS.SYSUSERPERM table, which contains all information about the permissions of users of the database, as well as the encrypted passwords of each user ID. However, SYS.SYSUSERPERMS is a view containing all information in SYS.SYSUSERPERM except for the password, and by default all users have SELECT access to this view. You can fully modify all permissions and group memberships set up in a new database for SYS, PUBLIC, and DBA.

The following table summarizes the system tables containing information about user IDs, groups, and permissions. The user ID SYS owns all tables and views, and so their qualified names are SYS.SYSUSERPERM and so on.

Appropriate SELECT queries on these tables generates all the user ID and permission information stored in the database.

Table	Default	Contents
SYSUSERPERM	DBA only	Database-level permissions and pass- word for each user ID
SYSGROUP	PUBLIC	One row for each member of each group
SYS- TABLEPERM	PUBLIC	All permissions on table given by the GRANT command s
SYSCOLPERM	PUBLIC	All columns with UPDATE permission given by the GRANT command
SYSDUMMY	PUBLIC	Dummy table, can be used to find the current user ID
SYSPROCPERM	PUBLIC	Each row holds one user granted per- mission to use one procedure

Views	Default	Contents
SYSUSERAUTH	DBA only	All information in SYSUSER- PERM except for user numbers
SYSUSERPERMS	PUBLIC	All information in SYSUSER- PERM except for passwords
SYSUSERLIST	PUBLIC	All information in SYSUSER- AUTH except for passwords
SYSGROUPS	PUBLIC	Information from SYSGROUP in a more readable format
SYSTABAUTH	PUBLIC	Information from SYS- TABLEPERM in a more readable format
SYSCOLAUTH	PUBLIC	Information from SYSCOLPERM in a more readable format
SYSPROCAUTH	PUBLIC	Information from SYSPROCPERM in a more read- able format

The following table summarizes the system views containing information about user IDs, groups, and permissions.

In addition to these, there are tables and views that contain information about each object in the database.

CHAPTER 14

Replicating Data with Replication Server

About this chapter	This chapter describes how you can use Replication Se between an Adaptive Server Anywhere database and o databases in the replication system may be Adaptive Se databases or other kinds of database.	erver to replicate data ther databases. Other erver Anywhere	
Before you begin	Replication Server administrators who are setting up A Anywhere to take part in their Replication Server insta chapter especially useful. You should have knowledge documentation, and familiarity with the Replication Se chapter does not describe Replication Server itself.	Replication Server administrators who are setting up Adaptive Server Anywhere to take part in their Replication Server installation will find this chapter especially useful. You should have knowledge of Replication Server documentation, and familiarity with the Replication Server product. This chapter does not describe Replication Server itself.	
	and administration, see your Replication Server docum	ientation.	
Contents	Торіс:	page	
	Introduction to replication	426	
	Tutorial: Replicate data using Replication Server	429	
	Configuring databases for Replication Server	439	
	Using the LTM	442	

Introduction to replication

Data replication is the sharing of data among physically distinct databases.
Changes made to shared data at any one database are copied precisely to the
other databases in the replication system.Data replication brings some key benefits to database users.Data availabilityReplication makes data available locally, rather than through potentially
expensive, less reliable and slower connections to a single, central database.
Since data is accessible locally, you are always have access to data, even in
the event of a long-distance network connection failure.Response timeReplication improves response times for data requests for two reasons. First,
retrieval rates are faster since requests are processed on a local server
without accessing some wide area network. Second, competition for
processor time decreases, since local processing offloads work from a
central database server.

Sybase replication technologies

Sybase provides the following replication technologies for Adaptive Server Anywhere:

- MobiLink MobiLink is designed for two-way replication involving a consolidated database and large numbers of remote databases, typically including many mobile databases.
- SQL Remote Like MobiLink, SQL Remote is designed for two-way replication involving a consolidated database and large numbers of remote databases, typically including many mobile databases. Administration and resource requirements at the remote sites are minimal.
- **Replication Server** Replication Server is designed for replication among relatively small numbers of data servers, with a typical time lag between primary data and replicate data of a few seconds, and generally with an administrator at each site.

Each replication technology has its own documentation. This chapter describes how to use Adaptive Server Anywhere with Replication Server.

For information about SQL Remote, see the book SQL Remote User's Guide. For information about MobiLink, see the book MobiLink Synchronization User's Guide.

Replicate sites and primary sites

In a Replication Server installation, the data to be shared among databases is arranged in **replication subscriptions**.

For each replication definition, there is a **primary site**, where changes to the data in the replication occur. The sites that receive the data in the replication are called **replicate sites**.

Replicate site components

You can use Adaptive Server Anywhere as a replicate site with no additional components.

The following diagram illustrates the components required for Adaptive Server Anywhere to participate in a Replication Server installation as a replicate site.



- Replication Server receives data changes from primary site servers.
- Replication Server connects to Adaptive Server Anywhere to apply the changes.
- Adaptive Server Anywhere makes the changes to the database.

Asynchronous procedure The Replication Server can use asynchronous procedure calls (APC) at replicate sites to alter data at a primary site database. If you are using APCs, the above diagram does not apply. Instead, the requirements are the same as for a primary site.

Primary site components

To use an Adaptive Server Anywhere database as a primary site, you need to

use the Log Transfer Manager (LTM), or Replication Agent. The LTM supports Replication Server version 10.0 and greater.

The following diagram illustrates the components required for Adaptive Server Anywhere to participate in a Replication Server installation as a primary site. The arrows in the diagram represent data flow.



- The Adaptive Server Anywhere database server manages the database.
- The Adaptive Server Anywhere Log Transfer Manager connects to the database. It scans the transaction log to pick up changes to the data, and sends them to Replication Server.
- Replication Server sends the changes to replicate site databases.

Tutorial: Replicate data using Replication Server

	This section provides a step-by-step tutorial describing how to replicate data from a primary database to a replicate database. Both databases in the tutorial are Adaptive Server Anywhere databases.
Replication Server assumed	This section assumes you have a running Replication Server. For more information about how to install or configure Replication Server, see the Replication Server documentation.
What is in the tutorial	This tutorial describes how to replicate only tables. For information about replicating procedures, see "Preparing procedures and functions for replication" on page 444.
	The tutorial uses a simple example of a (very) primitive office news system: a single table with an ID column holding an integer, a column holding the user ID of the author of the news item, and a column holding the text of the news item. The id column and the author column make up the primary key.
	Before you work through the tutorial, create a directory (for example, <i>c:\tutorial</i>) to hold the files you create in the tutorial.

Lesson 1: Set up the Adaptive Server Anywhere databases

This section describes how to create and set up the Adaptive Server Anywhere databases for replication.

You can create a database using Sybase Central or the *dbinit* utility. For this tutorial, we use the *dbinit* utility.

Create the primary site database

1. Enter the following command from the tutorial directory you created (for example *c:\tutorial*).

dbinit primedb

This creates a database file *primedb.db* in the current directory.

Create the replicate site database

1. Enter the following command from the tutorial directory you created (for example *c*:*tutorial*).

dbinit repdb

This creates a database file *repdb.db* in the current directory.

What's next? Next, you have to start database servers running on these databases.

Lesson 2: Start the database servers

You need to run the primary site database server, with the primary database loaded.

Start the primary site database server

- 1. Change to the tutorial directory.
- 2. Enter the following command to start a network database server running the **primedb** database. You should be using the TCP/IP network communication protocol on the default communications port (2638):

dbsrv9 -x tcpip primedb.db

Start the replicate site database server

- 1. Change to the tutorial directory.
- 2. Enter the following command to start a network database server running the **repdb** database, but on a different port:

dbsrv9 -x tcpip(port=2639) -n REPSV repdb.db

What's next? Next, you have to make entries for each of the Adaptive Server Anywhere servers in an interfaces file, so Replication Server can communicate with these database servers.

Lesson 3: Set up the Open Servers in your system

You need to add a set of Open Servers to the list of Open Servers in your system.

Adding Open Servers Open Servers are defined in your interfaces file (*SQL.ini*) using the *DSEdit* utility. For NetWare and UNIX users, the interfaces file is named *interfaces*, and the utility is named *sybinit*.

For full instructions on how to add definitions to your interfaces file, see "Configuring Open Servers" on page 114.

Required Open Servers For each Open Server definition you must provide a **name** and an **address**. Do not alter the other attributes of the definition. You need to add an Open Server entry for each of the following:

- The primary database Create an entry named PRIMEDB with address as follows:
 - Protocol NLWNSCK

- Network address localhost,2638
- The replicate database Create an entry named REPDB with address as follows:
 - Protocol NLWNSCK
 - Network address localhost, 2639
- The LTM at the primary database This is necessary so you can shut down the LTM properly. Create an entry named PRIMELTM with address as follows:
 - Protocol NLWNSCK
 - Network address localhost,2640
- Your Replication Server This tutorial assumes you already have the Replication Server Open Server defined.
- What's next? Next, confirm that the Open Servers are configured properly.

Lesson 4: Confirm the Open Servers are configured properly

You can confirm that each Open Server is available by selecting ServerObject \succ Ping Server from the DSEdit utility.

Alternatively, you can confirm that each Open Server is configured properly by connecting to the database using an Open Client application such as the *isql* utility.

To start *isql* running on the primary site database, type

isql -U DBA -P SQL -S PRIMEDB

The Open Client *isql* utility is not the same as the Adaptive Server Anywhere Interactive SQL utility.

Lesson 5: Add Replication Server information to the primary database

You need to add Replication Server tables and procedures to the primary site database for the database to participate in a Replication Server installation. You also need to create two user IDs for use by Replication Server. The SQL command file *rssetup.sql* comes with Adaptive Server Anywhere, and carries out these tasks.

The *rssetup.sql* command file must be run on the Adaptive Server Anywhere server from the Interactive SQL utility.

*	Run the rssetup script
	1. From Interactive SQL, connect to the Adaptive Server Anywhere database as user ID DBA using password SQL .
	2. Run the <i>rssetup</i> script using the following command:
	read "path\rssetup.sql"
	where path is your Adaptive Server Anywhere installation directory.
	You can alternatively use File \succ Run Script, and browse to the file.
Actions carried out by	The <i>rssetup.sql</i> command file carries out the following functions:
rssetup.sql	• Creates a user named dbmaint , with password dbmaint and with DBA permissions. This is the maintenance user name and password required by Replication Server to connect to the primary site database.
	 Creates a user named sa, with password sysadmin and with DBA permissions. This is the user ID used by Replication Server when materializing data.
	◆ Adds sa and dbmaint to a group named rs_systabgroup .
Passwords and user IDs	While the hard-wired user IDs (dbmaint and sa) and passwords are useful for test and tutorial purposes, you should change the password and perhaps also the user IDs when running databases that require security. Users with DBA permissions have full authority in an Adaptive Server Anywhere database.
	The user ID sa and its password must match that of the system administrator account on the Replication Server. Adaptive Server Anywhere does not currently accept A NULL password.
Permissions	The <i>rssetup.sql</i> script carries out a number of operations, including some permissions management. The permissions changes made by <i>rssetup.sql</i> are outlined here. You do not have to make these changes yourself.
	For replication, ensure that the dbmaint and sa users can access this table without explicitly specifying the owner. To do this, the table owner user ID must have group membership permissions, and the dbmaint and sa users must be members of the table owner group. To grant group permissions, you must have DBA authority.
	For example, if user DBA owns the table, you should grant group permissions to the DBA user ID:
	GRANT GROUP

TO DBA

You should then grant the **dbmaint** and **sa** users membership in the DBA group. To grant group membership, you must either have DBA authority or be the group ID.

```
GRANT MEMBERSHIP
IN GROUP "DBA"
TO dbmaint ;
GRANT MEMBERSHIP
IN GROUP "DBA"
TO sa ;
```

Lesson 6: Create the table for the primary database

In this section, you create a single table in the primary site database, using *isql*. First, make sure you are connected to the primary site database:

isql -U DBA -P SQL -S PRIMEDB

Next, create a table in the database:

```
CREATE TABLE news (
ID int,
AUTHOR char( 40 ) DEFAULT CURRENT USER,
TEXT char( 255 ),
PRIMARY KEY ( ID, AUTHOR )
)
go
```

Identifier case sensitivity

In Adaptive Server Anywhere, all identifiers are case insensitive. In Adaptive Server Enterprise, identifiers are case sensitive by default. Even in Adaptive Server Anywhere, ensure the case of your identifiers matches in all parts of the SQL statement to ensure compatibility with Adaptive Server Enterprise.

In Adaptive Server Anywhere, the database determines case sensitivity. For example, passwords are case insensitive in case insensitive databases, and case sensitive in case sensitive databases. Extended characters used in passwords are case sensitive regardless of database sensitivity. User IDs, being identifiers, are case insensitive in all Adaptive Server Anywhere databases.

For **news** to act as part of a replication primary site, you must set the REPLICATE flag to ON for the table using an ALTER TABLE statement:

```
ALTER TABLE news
REPLICATE ON
go
```

This is equivalent to running the **sp_setreplicate** or **sp_setreptable** procedure on the table in Adaptive Server Enterprise. You cannot set REPLICATE ON in a CREATE TABLE statement.

Lesson 7: Add Replication Server information to the replicate database

You should run the *rssetup.sql* command file on the replicate database in exactly the same manner as it ran on the primary database. Also ensure that the **dbmaint** and **sa** users can access this table without explicitly specifying the table owner.

These tasks are the same as those carried out on the primary database. For a complete explanation, see "Lesson 5: Add Replication Server information to the primary database" on page 431.

Lesson 8: Create the tables for the replicate database

The replicate site database needs to have tables to hold the data it receives. Now is a good time to create these tables. As long as the database elements are in place, no extra statements are necessary for them to act as a replicate site in a Replication Server installation. In particular, you do not need to set the REPLICATE flag to ON, which is necessary only at the primary site.

Replication Server allows replication between tables and columns with different names. As a simple example, however, create a table in the replicate database identical in definition to that in the primary database (except for the REPLICATE flag, which is not set to ON in the replicate database). The table creation statement for this is:

```
CREATE TABLE news (
ID int,
AUTHOR char( 40 ) DEFAULT CURRENT USER,
TEXT char( 255 ),
PRIMARY KEY ( ID, AUTHOR )
)
GO
```

For the tutorial, the CREATE TABLE statement must be *exactly* the same as that at the primary site.

You must ensure that the users **dbmaint** and **sa** can access this table without specifying the owner name. Also, these user IDs must have SELECT and UPDATE permissions on the table.

Lesson 9: Set up Replication Server

You need to carry out the following tasks on the Replication Server:

- Create a connection for the primary site data server.
- Create a connection for the replicate site data server.
- Create a replication definition.
- Create a subscription to the replication.

This section describes each of these tasks. It also describes starting the Adaptive Server Anywhere LTM.

Create a connection for the primary site

Using *isql*, connect to Replication Server and create a connection to the primary site Adaptive Server Anywhere database.

The following command creates a connection to the **primedb** database on the **PRIMEDB** Open Server.

```
create connection to PRIMEDB.primedb
set error class rs_sqlserver_error_class
set function string class rs_sqlserver_function_class
set username dbmaint
set password dbmaint
with log transfer on
go
```

If you have changed the **dbmaint** user ID and password in the *rssetup.sql* command file, make sure you replace the **dbmaint** username and password in this command.

Replication Server does not actually use the **primedb** database name; instead, the database name is read from the command line of the **PRIMEDB** Open Server. You must, however, include a database name in the CREATE CONNECTION statement to conform to the syntax.

For a full description of the create connection statement, see the chapter "Replication Server Commands" in the *Replication Server Reference Manual.*

Create a connection for the replicate site.

Using *isql*, connect to Replication Server and create a connection to the replicate site Adaptive Server Anywhere database.

The following command creates a connection to the **repdb** database on the **REPDB** Open Server.

```
create connection to REPDB.repdb
set error class rs_sqlserver_error_class
set function string class rs_sqlserver_function_class
set username dbmaint
set password dbmaint
go
```

This statement differs from the primary site server statement in that there is no **with log transfer on** clause in this statement.

If you have changed the **dbmaint** user ID and password in the *rssetup.sql* command file, make sure you replace the **dbmaint** username and password in this command.

Create a replication definition

Using *isql*, connect to Replication Server and create a replication definition. The following statement creates a replication definition for the news table on the **primedb** database:

```
create replication definition news
with primary at PRIMEDB.primedb
( id int, author char(40), text char(255) )
primary key ( id, author )
go
```

For a full description of the CREATE REPLICATION DEFINITION statement, see your *Replication Server Reference Manual*.

If you set the **qualify_table_owner** option to on in the LTM configuration file, you must specify the table owner in the statement, for all replicating tables.

Configure and start the Adaptive Server Anywhere LTM

For replication to take place, the Adaptive Server Anywhere LTM must be running against the primary site server. Before you start the Adaptive Server Anywhere LTM, make sure it is properly configured by editing an LTM configuration file.

Below is a sample configuration file for the **primedb** database. If you are following the examples, you should make a copy of this file as *primeltm.cfg*:

```
#
# Configuration file for 'PRIMELTM'
#
SQL_server=PRIMEDB
SQL_database=primedb
SQL_user=sa
SQL_pw=sysadmin
RS_source_ds=PRIMEDB
RS source db=primedb
RS=your_rep_server_name_here
RS user=sa
RS pw=sysadmin
LTM_admin_user=DBA
LTM_admin_pw=SQL
LTM_charset=cp850
scan_retry=2
APC user=sa
APC_pw=sysadmin
SQL_log_files=C:\TUTORIAL
```

If you have changed the user ID and password in the *rssetup.sql* command file from **sa** and **sysadmin**, you should use the new user ID and password in this configuration.

To start the Adaptive Server Anywhere LTM running on the primary site server, enter the following command:

dbltm -S PRIMELTM -C primeltm.cfg

The connection information is in *primeltm.cfg*. In this command, PRIMELTM is the server name of the LTM.

You can find usage information about the Adaptive Server Anywhere LTM by typing the following statement:

dbltm -?

You can run the Adaptive Server Anywhere LTM as a Windows service. For information on running programs as services, see "Running the server outside the current session" on page 21.

Create a subscription for your replication

Using *isql*, connect to Replication Server and create a subscription for the replication.

The following statement creates a subscription for the news replication defined in "Create a replication definition" on page 436 with replicate site as the **repdb** database.

```
create subscription NEWS_SUBSCRIPTION
for news
with replicate at REPDB.repdb
go
```

You have now completed your installation. Try replicating data to confirm that the setup is working properly.

Lesson 10: Enter data at the primary site for replication

You can now replicate data from the primary database to the replicate database. As an example, connect to the primary database using the *isql* utility, and enter a row in the **news** table.

```
insert news (id, text)
values (1, 'Test news item.' )
commit
go
```

The Adaptive Server Anywhere LTM sends only committed changes to the Replication Server. The data change is replicated next time the LTM polls the transaction log.

Tutorial complete You have now completed the tutorial.

Configuring databases for Replication Server

Each Adaptive Server Anywhere database that participates in a Replication Server installation needs to be configured before it can do so. Configuring the database involves the following tasks:

- Selecting a secure user ID for the maintenance user and the name used by Replication Server when materializing data.
- Setting up the database for Replication Server.
- Configuring the language and character set, where necessary.

Configuring the LTM Each primary site Adaptive Server Anywhere database requires an LTM to send data to Replication Server. Each primary or replicate site Adaptive Server Anywhere database requires an Open Server definition so that Replication Server can connect to the database.

For information on configuring the LTM, see "Configuring the LTM" on page 445.

Setting up the database for Replication Server

	Once you have created your Adaptive Server Anywhere database, and created the necessary tables and so on within the database, you can make database ready for use with Replication Server. You do this using a setup script supplied with the Adaptive Server Anywhere Replication Agent product. The script is named <i>rssetup.sql</i> .
When you need to run the setup script	You need to run the setup script at any Adaptive Server Anywhere database that is taking part in a Replication Server installation, whether as a primary or a replicate site.
What the setup script does	The setup script creates user IDs required by Replication Server when connecting to the database. It also creates a set of stored procedures and tables used by Replication Server. The tables begin with the characters rs _, and the procedures begin with the characters sp Procedures include some that are important for character set and language configuration.

Prepare to run the setup script

Replication Server uses a special data server maintenance user login name
for each local database containing replicated tables. This allows Replication
Server to maintain and update the replicated tables in the database.The maintenance userThe setup script creates a maintenance user with name dbmaint and
password dbmaint. The maintenance user has DBA permissions in the

Adaptive Server Anywhere database, which allows it full control over the database. For security reasons, you should change the maintenance user ID and password.

* To change the maintenance user ID and password

- 1. Open the *rssetup.sql* setup script in a text editor. The script is held in the *scripts* subdirectory of your Adaptive Server Anywhere installation directory.
- 2. Change all occurrences of the dbmaint user ID to the new maintenance user ID of your choice.
- 3. Change the **dbmaint** password to the new maintenance user password of your choice. The password occurs in the following place at the top of the setup script file:

GRANT CONNECT TO dbmaint IDENTIFIED BY dbmaint

The materialization userWhen Replication Server connects to a database to materialize the initial
copy of the data in the replication, it does so using the Replication Server
system administrator account.

The Adaptive Server Anywhere database must have a user ID and password that match the Replication Server system administrator user ID and password. Adaptive Server Anywhere does not accept a NULL password.

The setup script assumes a user ID of sa and a password of **sysadmin** for the Replication Server administrator. You should change this to match the actual name and password.

To change the system administrator user ID and password

- 1. Open the *rssetup.sql* setup script in a text editor.
- 2. Change all occurrences of the **sa** user ID to match the Replication Server system administrator user ID.
- 3. Change the **sa** password to match the Replication Server system administrator password. The password has the initial setting of **sysadmin**.

Run the setup script

Once you have modified the setup script to match the user IDs and passwords appropriately, you can run the setup script to create the maintenance and system administrator users in the Adaptive Server Anywhere database.

To run the setup script

- 1. Start the Adaptive Server Anywhere database on an Adaptive Server Anywhere database engine or server.
- 2. Start the Interactive SQL utility, and connect to the database as a user with DBA authority. When you create an Adaptive Server Anywhere database, it has a user with user ID **DBA** and password **SQL**, which has DBA authority.
- 3. Run the script by entering the following command in the SQL Statements pane:

read path\rssetup.sql

where *path* is the path to the setup script.

Character set and language issues

Upon creation, each Adaptive Server Anywhere database is assigned a specific collation (character set and sort order). Replication Server uses a different set of identifiers for character sets and sort orders.

Set the character set and language parameters in the LTM configuration file. If you are unsure of the characters set label to specify, you can do the following to determine the character set of the server:

* To determine the character set

1. Execute the following command:

exec sp_serverinfo csname

The language is one of the language labels listed in "Language label values" on page 302.

Using the LTM

Since the Adaptive Server Anywhere LTM relies on information in the Adaptive Server Anywhere transaction log, take care not to delete or damage the log without storing backups (for example, using a transaction log mirror).

For more information about transaction log management, see the section "Transaction log and backup management" on page 449.

You cannot substitute an Adaptive Server Anywhere LTM for an Adaptive Server Enterprise LTM since the transaction logs have different formats.

The Adaptive Server Anywhere LTM supports replication of inserts, updates, and deletes, as well as replication of Transact-SQL-dialect stored procedure calls.

The Adaptive Server Enterprise LTM sends data changes to the Replication Server before they are committed. The Replication Server holds the changes until a COMMIT statement arrives. By contrast, the Adaptive Server Anywhere LTM sends only committed changes to Replication Server. For long transactions, this may lead to some added delay in replication, since all changes have to go through the Replication Server before distribution.

Configuring tables for replication

Adaptive Server Anywhere does not support the **sp_setreplicate** system procedure. Instead, a table is identified as a primary data source using the ALTER TABLE statement with a single clause:

ALTER TABLE table-name SET REPLICATE ON

Setting REPLICATE ON places extra information into the transaction log.
Whenever an UPDATE, INSERT, or DELETE action occurs on the table.
The Adaptive Server Anywhere Replication Agent uses this extra information to submit the full pre-image of the row, where required, to Replication Server for replication.

Even if only some of the data in the table needs to be replicated, all changes to the table are submitted to Replication Server. It is Replication Server's responsibility to distinguish the data to be replicated from that which is not.

When you update, insert, or delete a row, the pre-image of the row is the contents of the row before the action, and the post-image is the contents of the row after the action. For INSERTS, only the post-image is submitted (the pre-image is empty). For DELETES, the post-image is empty and only the pre-image is submitted. For UPDATES, both the pre-image and the updated

The effects of setting REPLICATE ON for a table values are submitted.

The following data types are supported for replication:

	Data type	Description (Open Client/Open Server type)
	Exact integer data types	int, smallint, tinyint
	Exact decimal data types	decimal, numeric
	Approximate numeric data types	float (8-byte), real
	Money data types	money, smallmoney
	Character data types	char(n), varchar(n), text
	Date and time data types	datetime, smalldatetime
	Binary data types	binary(n), varbinary(n), image
	Bit data types	bit
Notes	Adaptive Server Anywher However, non-null long vareplicated to a replicate si If a primary table has colur replicate the data if you cr supported data type. For e could define the column a	re supports data of zero length that is not NULL. archar and long binary data of zero length is te as NULL. umns with unsupported data types, you can reate a replication definition using a compatible example, to replicate a DOUBLE column, you s FLOAT in the replication definition.
Side effects of setting REPLICATE ON for a table	There can be a replication performance hit for heavily updated tables. You could consider using replicated procedures if you experience performance problems that may be related to replication traffic, since replicated procedures send only the call to the procedure instead of each individual action.	
	Since setting REPLICATE log, this log grows faster t	E ON sends extra information to the transaction han for a non-replicating database.
Minimal column replication definitions	The Adaptive Server Any replicate minimal column Server.	where LTM supports the Replication Server s feature. This feature is enabled at Replication
	For more information Replication Server docum	on replicate minimal columns, see your entation.

Preparing procedures and functions for replication

You can use stored procedures to modify the data in tables. Updates, inserts, and deletes execute from within the procedure.

Replication Server can replicate procedures as long as they satisfy certain conditions. The first statement in a procedure must carry out an update for the procedure to be replicated. See your Replication Server documentation for a full description of how Replication Server replicates procedures.

Adaptive Server Anywhere supports two dialects for stored procedures: the Watcom-SQL dialect, based on the draft ISO/ANSI standard, and the Transact-SQL dialect. You can use either dialect in writing stored procedures for replication.

Function APC formatThe Adaptive Server Anywhere LTM supports the Replication Serverfunction APC format. To make use of these functions, set the configuration
parameter rep_func to on (the default is off).

The LTM interprets all replicated APCs as either table APCs or function APCs. A single Adaptive Server Anywhere database cannot combine function APCs with other table APCs.

For more information about replicate functions, see your Replication Server documentation.

SQL Statements for controlling procedure replication

A procedure can be configured to act as a replication source using the ALTER PROCEDURE statement.

The following statement makes the procedure **MyProc** act as a replication source.

```
ALTER PROCEDURE MyProc
REPLICATE ON
```

The following statement prevents the procedure **MyProc** from acting as a replication source.

ALTER PROCEDURE MyProc REPLICATE OFF

These statements have the same effect as running **sp_setreplicate** or **sp_setrepproc 'table'** on the procedure in Adaptive Server Enterprise. The **sp_setrepproc 'function'** syntax is not supported.

When a procedure is used as a replication data source, calling the procedure sends extra information to the transaction log.

The effects of setting REPLICATE ON for a procedure

Asynchronous procedures

	A procedures called at a replicate site database to update data at a primary site database is an asynchronous procedure . The procedure carries out no action at the replicate site, but rather, the call to the procedure is replicated to the primary site, where a procedure of the same name executes. This is called an asynchronous procedure call (APC). The changes made by the APC are then replicated from the primary to the replicate database in the usual manner.
	For information about APCs, see your Replication Server documentation.
The APC_user and APC support	Support for APCs in Adaptive Server Anywhere is different from that in Adaptive Server Enterprise. In Adaptive Server Enterprise, each APC executes using the user ID and password of the user who called the procedure at the replicate site. In Adaptive Server Anywhere, however, the transaction log does not store the password, and so it is not available at the primary site. To work around this difference, the LTM configuration file holds a single user ID with associated password, and this user ID (the APC_user) executes the procedure at the primary site. The APC_user must, therefore, have appropriate permissions at the primary site for each APC that may be called.
Configuring the LT	"M

Э

	You control LTM behavior by modifying the LTM configuration file , which is a plain text file created and edited using a text editor. The LTM configuration file contains information the LTM needs, such as the Adaptive Server Anywhere server it transfers a log from, the Replication Server it transfers the log to. You need a valid configuration file to run the LTM.
Creating a configuration file	You must create a configuration file, using a text editor, before you can run the LTM. The -C LTM command specifies the name of the configuration file to use, and has a default of <i>dbltm.cfg</i> .
Configuration file format	The LTM configuration file shares the same format as the Replication Server configuration file described in your <i>Replication Server Administration Guide</i> . In summary:
	• The configuration file contains one entry per line.
	• An entry consists of a parameter, followed by the = character, followed by the value:

Entry=value

	• Lines beginning with a # character are comments ignored by the LTM.
	• The configuration file cannot contain leading blanks.
	• Entries are case sensitive.
	For the full list of available configuration file parameters, see "The LTM configuration file" on page 503.
Example configuration file	• The following is a sample Adaptive Server Anywhere LTM configuration file.
	<pre># This is a comment line # Names are case sensitive. SQL_user=sa SQL_pw=sysadmin SQL_server=PRIMESV SQL_database=primedb RS_source_ds=PRIMESV RS_source_db=primedb RS=MY_REPSERVER RS_user=sa RS_pw=sysadmin LTM_admin_user=DBA LTM_admin_pw=SQL LTM_charset=cp850 scan retry=2</pre>

SQL_log_files=e:\logs\backup APC_user=sa APC_pw=sysadmin

Replicating transactions in batches

Effects of buffering transactions	The LTM allows buffering of replication commands to Replication Server. Buffering the replication commands and sending them in batches results in fewer messages being sent, and can significantly increase overall throughput, especially on high volume installations.
How batch mode works	By default, the LTM buffers transactions. The buffer flushes (the transactions sent to Replication Server when the buffer:
	• Reaches maximum number of commands The batch_ltl_sz parameter sets the maximum number of LTL (log transfer language) commands stored in the buffer before it flushes. The default setting is 200.
	• Reaches maximum memory used The batch_ltl_mem parameter sets the maximum memory that the buffer can occupy before flushes. The default setting is 256K.

• Completes transaction log processing If there are no more entries in the transaction log to process (that is, the LTM is up to date with all committed transactions), then the buffer flushes.

Turning off buffering You can turn off buffering of transactions by setting the **batch_ltl_cmds** parameter to **off**:

batch_ltl_cmds=off

Language and character set issues

Language and character set issues are an important consideration in many replication sites. Each database and server in the system uses a specific collation (character set and sorting order) for storing and ordering strings. Adaptive Server Anywhere character set support is carried out in a different manner to character set support in Adaptive Server Enterprise and other Open Client/Open Server based applications.

This section describes how to configure the Adaptive Server Anywhere LTM such that data in an Adaptive Server Anywhere database can be shared with Replication Server and hence with other databases.

The LTM automatically uses the default Open Client/Open Server language, sort order, and character set. You can override these defaults by adding entries to the LTM configuration file.

Open Client/Open Server collations

Adaptive Server Enterprise, Replication Server, and other Open Client/Open Server applications share a common means of managing character sets.

For information on Open Client/Open Server character set support, see the chapter "Configuring Character Sets, Sort Orders, and Languages" in the Adaptive Server Enterprise System Administration Guide. For more information about character set issues in Replication Server, see the chapter "International Replication Design Considerations" in the *Replication Server Design Guide*.

This section provides a brief overview of Open Client/Open Server character set support.

Internationalization files Files that support data processing in a particular language are called **internationalization files**. Several types of internationalization files come with Adaptive Server Enterprise and other Open Client/Open Server applications.

There is a directory named *charsets* under your Sybase directory. Charsets has a set of subdirectories, including one for each character set available to

you. Each character set contains a set of files, as described in the following table

File	Description
Charset.loc	Character set definition files that define the lexical properties of each character such as alphanumeric, punctuation, operand, upper or lower case.
*.srt	Defines the sort order for alphanumeric and special characters.
*.xlt	Terminal-specific character translation files for use with utili-

Character set settings in the LTM configuration file

Three settings in the LTM configuration file refer to character set issues:

- ♦ LTM_charset The character set for the LTM to use. You can specify any Sybase-supported character set.
- LTM_language The language used by the LTM to print its messages to the error log and to its clients. You can choose any language to which the LTM has been localized, as long as it is compatible with the LTM character set.

The Adaptive Server Anywhere LTM has been localized to several languages.

Notes Character set In an Open Client/Open Server environment, an LTM should use the same character set as the data server and Replication Server attached to it.

Adaptive Server Anywhere character sets are specified differently than Open Client/Open Server character sets. Consequently, the requirement is that the Adaptive Server Anywhere character set be compatible with the LTM character set.

Language The *locales.dat* file in the *locales* subdirectory of the Sybase release directory contains valid map settings. However, the LTM output messages in the user interface are currently available in those languages to which the LTM has been localized.

Sort order All sort orders in your replication system should be the same. You can find the default entry for your platform in the *locales.dat* file in the *locales* subdirectory of the Sybase release directory.

Example • The following settings are valid for a Japanese installation:

LTM_charset=SJIS LTM_language=Japanese

Transaction log and backup management

	One of the differences between the Adaptive Server Enterprise LTM and the Adaptive Server Anywhere LTM is that while the Adaptive Server Enterprise LTM depends on a temporary recovery database for access to old transactions, the Adaptive Server Anywhere LTM depends on access to old transaction logs. No temporary recovery database exists for the Adaptive Server Anywhere LTM.
	Replication depends on access to operations in the transaction log, and for Adaptive Server Anywhere primary site databases, sometimes access to old transaction logs. This section describes how to set up backup procedures at an Adaptive Server Anywhere primary site to ensure proper access to old transaction logs.
Consequences of lost transaction logs	Good backup practices at Adaptive Server Anywhere primary database sites are crucial. A lost transaction log could mean rematerializing replicate site databases. At primary database sites, a transaction log mirror is recommended. For information on transaction log mirrors and other backup procedure information, see the Adaptive Server Anywhere <i>User's Guide</i> .
	The LTM configuration file contains a directory entry, which points to the directory where backed up transaction logs are kept. This section describes how to set up a backup procedure to ensure that such a directory stays in proper shape.
Backup utility options	With the Backup utility, you have the option of renaming the transaction log on backup and restart. For the <i>dbbackup</i> utility, this is the $-r$ option. It is recommended that you use this option when backing up the consolidated database and remote database transaction logs.
	For example, consider a database named <i>primedb.db</i> , in directory <i>c:\prime</i> , with a transaction log in directory <i>d:\primelog\primedb.log</i> . Backing up this transaction log to a directory <i>e:\primebak</i> using the rename and restart option carries out the following tasks:
	1. Backs up the transaction log, creating a backup file <i>e:\primebak\primedb.log</i> .
	2. Renames the existing transaction log to <i>d</i> :\ <i>primelog</i> \ <i>YYMMDDxx.log</i> , where xx are sequential characters ranging from AA to ZZ .
	3. Starts a new transaction log, as <i>d</i> :\ <i>primelog</i> \ <i>primedb</i> .log.
	After several backups, the directory <i>d</i> :\ <i>primelog</i> will contain a set of sequential transaction logs. The log directory should not contain any

transaction logs other than the sequence of logs generated by this backup procedure.

Using the DELETE_OLD_LOGS option

The DELETE_OLD_LOGS Adaptive Server Anywhere database option's default is OFF. If you set the default to ON, then the LTM automatically deletes the old transaction logs when Replication Server no longer needs access to the transactions. This option can help to manage disk space in replication setups.

You set the DELETE_OLD_LOGS option for the PUBLIC group:

SET OPTION PUBLIC.DELETE_OLD_LOGS = 'ON'

For more information, see "DELETE_OLD_LOGS option [replication]" on page 590.

The Unload utility and replication

If a database participates in replication, care must be taken when unloading and reloading in order to avoid needing to re-materialize the database. Replication is based on the transaction log, and unloading and reloading a database deletes the old transaction log. For this reason, good backup practices are especially important when participating in replication.

Replicating an entire database

Adaptive Server Anywhere provides a short cut for replicating an entire database, so you don't have to set each table in the database as a replicated table.

You can set a PUBLIC database option called REPLICATE_ALL using the SET OPTION statement. You can designate a whole database for replication using the following command:

SET OPTION PUBLIC.Replicate_all='ON'

You require DBA authority to change this and other PUBLIC option settings. You must restart the database for the new setting to take effect. The REPLICATE_ALL option has no effect on procedures.

For more information, see "REPLICATE_ALL option [replication]" on page 621.

Stopping the LTM

Example

You can shut down the LTM from the user interface in Windows NT/2000/XP, or in other circumstances by issuing a command.

- To stop the LTM in Windows NT/2000/XP, when the LTM is not running as a service
 - 1. Click SHUTDOWN on the user interface.

To stop the LTM by issuing a command

- 1. Connect to the LTM from *isql* using the LTM_admin_user login name and password in the LTM configuration file. The user ID and password are case sensitive.
- 2. Stop the LTM using the SHUTDOWN statement.

• The following statements connect *isql* to the LTM PRIMELTM, and shut it down:

```
isql -SPRIMELTM -UDBA -PSQL
l> shutdown
2> go
```
Part IV

UTILITIES, OPTIONS, AND PROPERTIES

This section describes database administration utilities, options, properties, and limitations.

CHAPTER 15

Database Administration Utilities

About this chapter	Adaptive Server Anywhere includes a set of utility programs for backing up databases and performing other database administration tasks. This chapter provides reference information for each of the database administration utilities.	
Contents	Торіс:	page
	Administration utilities overview	457
	The Backup utility	458

The Backup utility	458
The Collation utility	462
The File Hiding utility	466
The Compression utility	468
The Console utility	471
The Data Source utility	472
The Erase utility	478
The Histogram utility	481
The Information utility	483
The Initialization utility	485
The Interactive SQL utility	492
The Language utility	497
The License utility	499
The Log Transfer Manager	501
The Log Translation utility	508
The Ping utility	514
The Rebuild utility	517
The Server Location utility	518
The Service Creation utility	519

Торіс:	page
The Spawn utility	523
The Stop utility	525
The Transaction Log utility	527
The Uncompression utility	531
The Unload utility	533
The Upgrade utility	542
The Validation utility	547
The Write File utility	551

Administration utilities overview

This chapter presents reference information on the programs and database administration utilities that are part of Adaptive Server Anywhere. Each of the utilities can be accessed from one or more of Sybase Central, Interactive SQL, or at the command prompt.

For comprehensive documentation on Sybase Central, see the Sybase Central online Help. For an introduction to the Sybase Central database administration tool, see "Tutorial: Managing Databases with Sybase Central" [*Introducing SQL Anywhere Studio*, page 47].

The administration utilities use a set of system environment variables. These variables are described in "Environment variables" on page 244.

Database file administration statements A set of SQL statements are available that carry out some of the tasks that the administration utilities carry out. These statements are listed in "SQL Statements" [*ASA SQL Reference*, page 213].

The Backup utility

You can use the Backup utility to back up running databases, database files, transaction logs, and write files.

You can access the Backup utility in the following ways:

- From Sybase Central, using the Create Backup Images wizard.
- At the command prompt, using the *dbbackup* command. This is useful for incorporating backup procedures into batch or command files.

The Backup utility makes a backup copy of all the files for a single database. A simple database consists of two files: the main database file and the transaction log. More complicated databases can store tables in multiple files, with each file as a separate dbspace. All backup filenames are the same as the database filenames.

Running the Backup utility on a running database is equivalent to copying the database files when the database is not running. Thus, you can back up the database while other applications or users are using it.

Exit codes are 0 (success) or non-zero (failure).

For a description of suggested backup procedures, see "Backup and Data Recovery" on page 337.

For more information about the backup utility, see "BACKUP statement" [*ASA SQL Reference*, page 263].

Backing up a database using the Create Backup Images wizard

To back up a database file or a running database

- 1. In the left pane, select the Adaptive Server Anywhere plug-in.
- 2. In the right pane, click the Utilities tab.
- 3. In the right pane, double-click Create Backup Images.

The Create Backup Images wizard appears.

4. Follow the instructions in the wizard.

For full information on backing up a database from Sybase Central, see "Backup and Data Recovery" on page 337. For more information about options, see "Backup utility options" on page 460

Тір

Syntax

Description

You can also access the Create Backup Images Database wizard from within Sybase Central using any of the following methods:

choosing Tools \triangleright Adaptive Server Anywhere 9 \triangleright Create Backup Images.

Selecting a database in the left pane, and choosing Create Backup Images from the File menu.

Right-clicking a database, and choosing Create Backup Images from the popup menu.

Backing up a database using the dbbackup command-line utility

Outlan	Description
Option	Description
-c "keyword=value; "	Supply database connection parameters
-d	Only back up the main database file
–l file	Live backup of the transaction log to a file
-n	Change the naming convention for the backup transaction log
–o filename	Log output messages to a file
- q	Quiet mode-do not print messages
- r	Rename and start a new transaction log
t	Only back up the transaction log
- W	Only back up the write file
- x	Delete and restart the transaction log
-X0	Delete and restart the transaction log without making a backup
- y	Replace files without confirmation
If none of the options -d, -	-t, or -w are used, all database files are backed
For more information about the Backup utility options, see "Backup utility options" on page 460.	
This utility accepts @filen	ame parameters. For more information, see

dbbackup [options] directory

"@filename server option" on page 131.

up.

Backup utility options

Directory The directory to which the backup files are copied. If the directory does not exist, it is created. However, the parent directory must exist.

Connection parameters (-c) For a description of the connection parameters, see "Connection parameters" on page 70. If the connection parameters are not specified, connection parameters from the SQLCONNECT environment variable are used, if set. The **user ID** must have DBA authority or REMOTE DBA authority.

For example, the following command backs up the **asademo** database running on the server **sample_server**, connecting as user ID **DBA** with password **SQL**, into the *asabackup* directory:

Backup main database only (-d) Back up the main database files only, without backing up the transaction log file or a write file, if one exists.

Live backup (-I lower-case L) This option is provided to enable a secondary system to be brought up rapidly in the event of a server crash. A live backup does not terminate, but continues running while the server runs. It runs until the primary server crashes. At that point, it is shut down, but the backed up log file is intact and can be used to bring a secondary system up quickly.

Change backup transaction log naming convention (–n) This option is used in conjunction with -r. It changes the naming convention of the backup transaction log file to *yymmddxx.log*, where xx are sequential letters ranging from AA to ZZ and *yymmdd* represents the current year, month and day.

The backup copy of the transaction log file is stored in the directory specified in the command, and with the *yymmddxx.log* naming convention. This allows backups of multiple versions of the transaction log file to be kept in the same backup directory.

The two-digit year notation does not cause any year 2000 problems. The names are used solely for identification, not for ordering.

You can also use the both the -x option and the -n option to rename the log copy. For example

DBBACKUP -x -n

Log output messages to file (-o) Write output messages to the named

file.

Operate quietly (-q) Do not display output messages. This option is available only when you run this utility from the command prompt.

Rename and start new transaction log (–r) This option forces a checkpoint and the following three steps to occur:

- The current working transaction log file is copied and saved to the directory specified in the command.
- 2. The current transaction log remains in its current directory, but is renamed using the format *yymmddxx.log*, where xx are sequential characters starting at AA and running through to ZZ, and *yymmdd* represents the current year, month and day. This file is then no longer the current transaction log.
- 3. A new transaction log file is generated that contains no transactions. It is given the name of the file that was previously considered the current transaction log, and is used by the database server as the current transaction log.

Back up the transaction log file only (-t) This can be used as an incremental backup since the transaction log can be applied to the most recently backed up copy of the database file(s).

Back up the database write file only (–w) For a description of database write files, see "The Write File utility" on page 551.

Delete and restart the transaction log (–x) Back up the existing transaction log, then delete the original log and start a new transaction log. This option causes the backup to wait for a point when all transactions from all connections are committed.

Delete and restart the transaction log without a backup (–xo) Delete the current transaction log and start a new one. This operation does not carry out a backup; its purpose is to free up disk space in non-replication environments.

Operate without confirming actions (–y) Choosing this option creates the backup directory or the replacement of a previous backup file in the directory without confirmation.

The Collation utility

With the Collation utility, you can extract a collation (sorting sequence) into a file suitable for creating a database using a custom collation.

The file that is produced can be modified and used with Sybase Central or the -z option of *dbinit* to create a new database with a custom collation.

Exit codes are 0 (success) or non-zero (failure).

You must change the label on the following line in the collation file. If you do not, the custom collation will conflict with the original collation on which it is based.

Collation label (name)

For more information about custom collating sequences, see "Creating a database with a custom collation" on page 331.

You can access the Collation utility in the following ways:

- From Sybase Central, using the Create Custom Collation wizard.
- At the command prompt, using the *dbcollat* command.

Extracting a collation using the Create Custom Collation wizard

To use the Create Custom Collation wizard

- 1. In the left pane, select the Adaptive Server Anywhere plug-in.
- 2. In the right pane, click the Utilities tab.
- 3. In the right pane, double-click Create Custom Collation.

The Create Custom Collation wizard appears.

4. Follow the instructions in the wizard.

Тір

You can also access the Create Custom Collation wizard from within Sybase Central using any of the following methods:

choosing Tools \succ Adaptive Server Anywhere 9 \succ Create Custom Collation.

Selecting a database in the left pane, and choosing Create Custom Collation from the File menu.

Right-clicking a database, and choosing Create Custom Collation from the popup menu.

Extracting a collation using the dbcollat command-line utility

ī.

Options	Description
-c "keyword=value; "	Supply database connection parameters
-d collation-file	Convert the definition file to INSERT statements with collation mapping placed in <i>output-file</i>
-е	Include empty mappings
–o filename	Log output messages to a file
- q	Quiet mode-do not print messages
-x	Use hex for extended characters (7F-FF)
- y	Replace the file without confirmation
-z col-seq	Specify a collating sequence label

Syntax

dbcollat [options] output-file

This utility accepts @filename parameters. For more information, see "@filename server option" on page 131.

For more information about the Collation utility options, see "Collation utility options" on page 463.

Collation utility options

Connection parameters (–c) For a description of the connection parameters, see "Connection parameters" on page 70. If the connection parameters are not specified, connection parameters from the SQLCONNECT environment variable are used, if set.

For example, the following command extracts a collation file from the **asademo** database that is running on the **sample_server** server, and connects as user ID **DBA** with password **SQL**:

dbcollat -c "eng=sample_server;dbn=asademo;uid=DBA;pwd=SQL" c:\
 sample\col

Convert the collation definition file to INSERT statements that describe the collation (-d) When a database is created, the collation is inserted into the SYS.SYSCOLLATION system table. A mapping from the collation to character sets and Sybase TDS collations is also inserted into the SYS.SYSCOLLATIONMAPPINGS system table. This collation is selected

from the set of provided collations in the *collseqs.sql* file or from the custom collations in the *custom.sql* file in the *scripts* subdirectory of your Adaptive Server Anywhere installation directory.

For more information about the SYSCOLLATIONMAPPINGS system table, see "SYSCOLLATIONMAPPINGS system table" [*ASA SQL Reference*, page 625].

Custom collations are added to the *custom.sql* script. The -d option converts the collation definition file that you provide into INSERT statements that can be copied into *custom.sql*.

For example, you can use the -d option with the *dbcollat* command as follows:

dbcollat -d coll-defn-file custom-file

The *coll-defn-file* is read and parsed as a collation definition. Output is written to *custom-file*. The *custom-file* contents must be added to *custom.sql*.

For more information about creating a custom collation using the -d option, see "Creating a custom collation" on page 330.

Include empty mappings (-e) Normally, collations don't specify the actual value that a character is to sort to. Instead, each line of the collation sorts to one position higher than the previous line. However, older collations have gaps between some sort positions. Normally, the Collation utility skips the gaps and writes the next line with an explicit sort-position. This option causes the Collation utility to write empty mappings (consisting of just a colon (:)) for each line in the gap.

Log output messages to file (-o) Write output messages to the named file.

Operate quietly (-q) Do not display output messages. This option is available only when you run this utility from the command prompt.

Use hexadecimal for extended characters [7F to FF] (-x) Extended single-byte characters (whose value is greater than hex 7F) may or may not appear correctly on your screen, depending on whether the code page on your computer is the same as the code page of the collation that you are extracting. This option causes the Collation utility to write any character to hex 7F or above as a two-digit hexadecimal number, in the form xdd. For example:

x80, xFE

Without the -x option, only characters from x00 to x1F, x7F and xFF are

written in hexadecimal form.

Operate without confirming actions (–y) Choosing this option automatically replaces an existing collation file without prompting for confirmation.

Specify a collating sequence label (–z) Specify the label of the collation to be extracted. The names of the recommended collation sequences can be found by executing the following command:

```
dbinit -l
```

If the -z option is specified with one of the available collation labels, then *dbcollat* does not connect to a database. Otherwise, it connects to a database and extracts the collation of that database. If the collation label does not match the collation label of the database, an error is returned.

The File Hiding utility

With the File Hiding [dbfhide] utility you can add simple encryption to configuration files to hide the contents of the file.

You can access the File Hiding utility:

• At the command prompt, using the *dbfhide* command. This is useful for incorporating into batch or configuration files.

Hiding the contents of files using the dbfhide command-line utility

Syntax	dbfhide original-configuration-file encrypted-configuration-file
Description	Configuration files (also called command files) are used by some utilities to contain command-line options. These options may contain a password. You can use the File Hiding utility to add simple encryption to configuration files, and thereby obfuscate the contents of the file. The original file will not be modified.
	The utilities that use configuration files include:
	♦ dbsrv9
	♦ dbmlsrv9
	♦ dbmlsync
	♦ dbltm
	♦ dbremote
	♦ ssremote
	♦ ssqueue
	This utility accepts @filename parameters. For more information, see "@filename server option" on page 131.
Example	Create a configuration file that starts the personal database server and the sample database. It should set a cache of 10 Mb, and name this instance of the personal server <i>Elora</i> . The configuration file would be written as follows:
	<pre># Configuration file for server Elora -n Elora -c 10M path\asademo.db</pre>
	(Note that lines beginning with # are treated as comments.)

Name the file *sample.txt*. If you wanted to start the database using this configuration file, your command line would be:

```
dbeng9 @sample.txt
```

Now, add simple encryption to the configuration.

dbfhide sample.txt encrypted_sample.txt

Use the encrypted_sample.txt file to start a database.

```
dbsrv9 @encrypted_sample.txt
```

For more information about Encryption, see "Keeping Your Data Secure" [*SQL Anywhere Studio Security Guide*, page 3]. For more information about using configuration files, see "Using configuration files" on page 10.

The Compression utility

With the Compression utility you can compress a database file. The Compression utility reads the given database file and creates a compressed database file. Compressed databases are usually 40 to 60 per cent of their original size. The database server cannot update compressed database files: they must be used as read-only files, in conjunction with write files.

The Compression utility does not compress files other than the main database file. The database to be compressed must not be running.

Exit codes are 0 (success) or non-zero (failure).

You can access the Compression utility in the following ways:

- From Sybase Central, using the Compress Database wizard.
- At the command prompt, using the *dbshrink* command. This is useful for incorporating into batch or command files.

Compressing a database using the Compress Database wizard

To compress a database file

- 1. In the left pane, select the Adaptive Server Anywhere plug-in.
- 2. In the right pane, click the Utilities tab.
- In the right pane, double-click Compress Database. The Compress Database wizard appears.
- 4. Follow the instructions in the wizard.

Тір

You can also access the Compress Database wizard from within Sybase Central by right-clicking a database, and choosing Compress Database from the popup menu.

choosing Tools \succ Adaptive Server Anywhere $9 \succ$ Compress Database.

Selecting a database in the left pane, and choosing Compress Database from the File menu.

Right-clicking a database, and choosing Compress Database from the popup menu.

Compressing a database using the dbshrink command-line utility

Syntax

dbshrink [options] database-file [compressed-database-file]

	Option	Description	
	– ek key	Specify encryption key	
	-ep	Prompt for encryption key	
	–o filename	Log output messages to a file	
	- q	Quiet mode-do not print messages	
	- y	Replace an existing output file without confirma- tion	
Description	tion The <i>dbshrink</i> utility reads <i>database-file</i> and creates a compress file. The compressed filename defaults to the same name as the database file, but with an extension of <i>.cdb</i> . The output filenam extension) must not be same as the input filename (with extensi		
	For more information about the Compression utility options, see "Compression utility options" on page 469.		
	This utility accepts ("@filename server o	@filename parameters. For more information, see ption" on page 131.	

i.

Compression utility options

Specify encryption key (-ek) This option allows you to specify the encryption key for strongly encrypted databases directly in the command. If you have a strongly encrypted database, you must provide the encryption key to use the database or transaction log in any way. For strongly encrypted databases, you must specify either -ek or -ep, but not both. The command will fail if you do not specify a key for a strongly encrypted database.

Prompt for encryption key (–ep) This option allows you to specify that you want to be prompted for the encryption key. This option causes a dialog box to appear, in which you enter the encryption key. It provides an extra measure of security by never allowing the encryption key to be seen in clear text. For strongly encrypted databases, you must specify either –ek or –ep, but not both. The command will fail if you do not specify a key for a strongly encrypted database.

Log output messages to file (-o) Write output messages to the named file.

Operate quietly (-q) Do not display output messages. This option is available only when you run this utility from the command prompt.

Operate without confirming actions (–y) Choosing this option automatically replaces existing compressed database file(s) without

requesting confirmation.

The Console utility

The Console utility provides administration and monitoring facilities for database server connections. The Console utility is available on all operating systems.

Monitoring connections using the dbconsole command-line utility

Syntax	dbconsole [options]		
	Option	Description	
	-c "keyword=value;"	Database connection parameters	
Description	Start the server console u	tility.	
The server console utility allows you to monitor the server machine. You can use it to track who is logged on to a data elsewhere on your network. You can also display both the client statistics on your local client screen, disconnect users the database server. The Console utility can display inform connections.		allows you to monitor the server from a client to track who is logged on to a database server rk. You can also display both the server and the ocal client screen, disconnect users, and configure Console utility can display information for multiple	
	Exit codes are 0 (success) or non-zero (failure).		
*	To disconnect a user	from a database	
	1. Connect to the database from the Console utility.		
	2. Double-click the connection to display the Connection Detail window.		
	3. Click Disconnect to disconnect the specified connection from the database.		
	This utility does not acce see "@filename server op	ept @filename parameters. For more information, potion" on page 131.	
Console utility option	IS		
	-c "keyword=value; …	" Specify connection parameters.	
For a description of connection parameters, see "Connectio parameters" on page 70.		connection parameters, see "Connection	

The Data Source utility

The Data Source utility is a cross-platform alternative to the ODBC Administrator for creating, deleting, describing, and listing Adaptive Server Anywhere ODBC data sources.

Managing ODBC data sources using the dbdsn command-line utility

Syntax

dbdsn [modifier-options]
{ -l[u | s] [qq]
| -d[u | s] dsn
| -g[u | s] dsn
| -w[u | s] dsn [details-options;...]
| -cl[qq] }

Parameters

Major option	Description
-l[u s] [qq]	List either all Adaptive Server Anywhere users or all Adaptive Server Anywhere system data sources. You can also list both Adaptive Server Anywhere user and Adaptive Server Anywhere system data sources. User data sources is the default. Using -qq with this option lists the DSNs without any banner or titles.
$-\mathbf{d}[\mathbf{u} \mid \mathbf{s}] dsn$	Delete the named Adaptive Server Anywhere user or Adaptive Server Anywhere system data source. User data sources is the default.
$-\mathbf{g}[\mathbf{u} \mid \mathbf{s}] dsn$	List (get) details about the named Adaptive Server Anywhere user or Adaptive Server Any- where system data source. User data sources is the default.
-w[u s] dsn [details- options]	Create (write) a user or system data source definition. User data sources is the default.
-cl[qq]	List available connection parameters. Using -qq with this option lists the available connection parameters without any banner or titles.

Modifier option	Description
-b	Brief. Print connection string for the data source
-q	Quiet. Do not print banner
- v	Verbose. Print connection parameters in tabular form
-y	Delete or overwrite data source without confirmation
Details-option	Description
−cw	Ensure that the DBF parameter (specified using -c) is an absolute filename. If the value of DBF not an absolute filename, the Data Source utilit prepends the current working directory (CWD)
-c "keyword=value;"	Supply database connection parameters
-ec encryption type	Encrypt all network packets
–o filename	Write client message to filename
–p size	Set maximum network packet size
-r	Disable multiple record fetching
-tl seconds	Client liveness timeout period
– x list	List network drivers to run
	Display debugging information
-Z	Display debugging mornation

See also	"Working with ODBC data sources" on page 53
	"Using ODBC data sources on UNIX" on page 63
Description	The Data Source utility is a cross-platform alternative to the ODBC Administrator for creating, deleting, describing, and listing Adaptive Server Anywhere ODBC data sources. On Windows operating systems, the data sources are held in the registry. On UNIX operating systems, the data sources are held in the <i>.odbc.ini</i> file. The utility is useful for batch operations.
	The modifier options can occur before or after the major option specification. The order makes a difference only when a connection parameter value is

specified more than once. In such a case, the last value specified is used.

Exit codes are 0 (success) or non-zero (failure).

This utility accepts @filename parameters. For more information, see "@filename server option" on page 131.

Write a definition of the data source **newdsn**. Do not prompt for confirmation if the data source already exists.

dbdsn -y -x tcpip -w newdsn -c "uid=DBA;pwd=SQL" -v

or, with a different option order,

dbdsn -w newdsn -c "uid=DBA;pwd=SQL" -x tcpip -y

List all known user data sources, one data source name per line:

dbdsn -l

List all known system data sources, one data source name per line:

dbdsn -ls

List all data sources along with their associated connection string:

dbdsn -l -b

Report the connection string for user data source **MyDSN**:

dbdsn -g MyDSN

Report the connection string for system data source MyDSN:

dbdsn -gs MyDSN

Delete the data source **BadDSN**, but first list the connection parameters for **BadDSN** and prompt for confirmation:

dbdsn -d BadDSN -v

Delete the data source **BadDSN** without prompting for confirmation.

dbdsn -d BadDSN -y

Create a data source named NewDSN for the database server MyServer:

dbdsn -w NewDSN -c "uid=DBA;pwd=SQL;eng=MyServer"

If a **NewDSN** already exists, the previous definition is overwritten.

The following example connects to the **sample** database server. The server name **sample** overrides the previous specified value of **MyServer**:

dbdsn -w NewDSN -c "uid=DBA;pwd=SQL;eng=MyServer" sample

Examples

List all connection parameter names and their aliases:

```
dbdsn -cl
```

List all user DSNs (without banner and titles):

```
dbdsn -l -qq
```

List all connection parameter names (without banner and titles):

dbdsn -cl -qq

Specify an absolute filename. When the DSN is created, it will contain *dbf=E:\asa90\my.db*.

```
E:\asa90> dbdsn -w testdsn -cw -c
uid=dba;pwd=sql;eng=asa;dbf=my.db
```

Data Source utility options

When you use the Data Source utility, you can choose from major options, modifier options, and details options.

Major options	List defined data sources (–I) Lists the available Adaptive Server Anywhere ODBC data sources. You can modify the list format using the $-b$ or $-v$ options. You can modify the option using the u (user) or s (system) specifiers. The default specifier is u.
	Delete the named data source (-d) Deletes the named Adaptive Server Anywhere data source. You can modify the option using the u (user) or s (system) specifiers. U is the default specifier. If you supply $-y$, any existing data source is overwritten without confirmation.
	List (get) details of the named data source (–g) List the definition of the named Adaptive Server Anywhere data source. You can modify the format of the output using the $-b$ or $-v$ options. You can modify the option using the u (user) or s (system) specifiers. U is the default specifier.
	Create (write) a data source definition (–w) Creates a new data source, or overwrites one if one of the same name exists. You can modify the option using the u (user) or s (system) specifiers. U is the default specifier. If you supply $-y$, any existing data source is overwritten without confirmation.
	List available connection parameters (–cl) This convenience option lists the connection parameters supported by the <i>dbdsn</i> utility.
Modifier options	Print connection string for the data source (–b) Format the output of the

list as a single line connection string.

Do not print banner (-q) Suppress the informational banner.

Do not print banner or titles (-qq) Suppress both the informational banner and titles. This option can only be used with the -l and the -cl options.

Print connection parameters in tabular form (–v) Format the output of the list over several lines, as a table.

Delete or overwrite data source without confirmation (–y) Automatically delete or overwrite each data source without prompting you for confirmation.

Details options Absolute filenames (-cw) Ensure that the DBF parameter (specified using -c) is an absolute filename. If the value of DBF is not an absolute filename, dbdsn will prepend the current working directory (CWD). This option is useful because some operating systems (Win 9x) do not have CWD information readily available in batch files.

Connection parameters (-c) Specify connection parameters as a connection string.

For more information, see "Connection parameters" on page 70.

The remainder of the details options are supplied for users familiar with the *dbcli6.exe* utility shipped with Adaptive Server Anywhere version 6.

Encrypt network packets (–ec) Encrypt packets sent between the client application and the server.

For more information, see "Encryption connection parameter [ENC]" on page 188.

Log output messages to file (-o) Write output messages to the named file. By default, messages are written to the console.

For more information, see "Logfile connection parameter [LOG]" on page 194.

Operate quietly (-q) Do not display output messages. This option is available only when you run this utility from the command prompt.

Set maximum network packet size (-p) The maximum packet size for network communications, in bytes. The value must be greater than 300, and less than 16000. The default setting is 1460.

For more information, see "CommBufferSize connection parameter [CBSIZE]" on page 178.

Disable multiple-record fetching (-r) By default, when the database

server gets a simple fetch request, the application asks for extra rows. You can disable this behavior using this option.

For more information, see "DisableMultiRowFetch connection parameter [DMRF]" on page 187.

Set client liveness timeout (-tl) Terminates connections when they are no longer intact. The value is in seconds.

The default is server setting, which in turn has a default value of 120 seconds.

For more information, see "LivenessTimeout connection parameter [LTO]" on page 194.

Set communications links (–x) A comma separated list of network drivers to run.

For more information, see "CommLinks connection parameter [LINKS]" on page 179.

Display debugging information (–z) Provide diagnostic information on communications links on startup.

Server name Connect to the named server. Only the first 40 characters are used.

For more information, see "The Database Server" on page 123.

The Erase utility

With the Erase utility, you can erase a database file or write file and its associated transaction log, or you can erase a transaction log file or transaction log mirror file. All database files, write files, and transaction log files are marked read-only to prevent accidental damage to the database and accidental deletion of the database files.

Deleting a database file that references other dbspaces does not automatically delete the dbspace files. If you want to delete the dbspace files on your own, change the files from read-only to writable, and then delete the files individually. As an alternative, you can use the DROP DATABASE statement to erase a database and its associated dbspace files.

If you erase a database file or write file, the associated transaction log and transaction log mirror are also deleted. If you erase a transaction log for a database that also maintains a transaction log mirror, the mirror is not deleted.

The database to be erased must not be running when this utility is used.

Exit codes are 0 (success) or non-zero (failure).

You can access the Erase utility in the following ways:

- From Sybase Central, using the Erase Database wizard.
- At the command prompt, using the *dberase* command. This is useful for incorporating into batch or command files.

For more information, see the "DROP DATABASE statement" [ASA SQL Reference, page 410].

Erasing a database using the Erase Database wizard

To erase a database file

- 1. In the left pane, select the Adaptive Server Anywhere 9 plug-in.
- 2. In the right pane, click the Utilities tab.
- 3. In the right pane, double-click Erase Database

The Erase Database wizard appears.

4. Follow the instructions in the wizard.

For full information on erasing a database from Sybase Central, see "Erasing a database" [ASA SQL User's Guide, page 30].

Тір

You can also access the Erase Database wizard from within Sybase Central using any of the following methods:

choosing Tools \blacktriangleright Adaptive Server Anywhere $9 \triangleright$ Erase Database.

Selecting a server in the left pane, and choosing Erase Database from the File menu.

Right-clicking a server, and choosing Erase Database from the popup menu.

Erasing a database using the dberase command-line utility

Syntax	dberase [options] database-file		
	Option	Description	
	– ek key	Specify encryption key	
	-ер	Prompt for encryption key	
	–o filename	Log output messages to a file	
	- q	Operate quietly—do not print messages	
	-y	Erase files without confirmation	
Description	The <i>database-file</i> may be a database file, write file, or transaction log file. The full filename must be specified, including extension. If a database file o write file is specified, the associated transaction log file (and mirror, if one is maintained) is also erased.		
	For more information about the Erase utility options, see "Erase utility options" on page 479.		
	This utility accepts @filename parameters. For more information, see "@filename server option" on page 131.		
Erase utility options			
	Specify encryption key (–ek) This option allows you to specify the encryption key for strongly encrypted databases directly in the command. If you have a strongly encrypted database, you must provide the encryption key to use the database or transaction log in any way. For strongly encrypted databases, you must specify either –ek or –ep, but not both. The command will fail if you do not specify a key for a strongly encrypted database.		
	Prompt for encryption ke you want to be prompted f	ey (-ep) This option allows you to specify that for the encryption key. This option causes a dialog	

box to appear, in which you enter the encryption key. It provides an extra measure of security by never allowing the encryption key to be seen in clear text. For strongly encrypted databases, you must specify either -ek or -ep, but not both. The command will fail if you do not specify a key for a strongly encrypted database.

Log output messages to file (-o) Write output messages to the named file.

Operate quietly (-q) Do not display output messages. This option is available only when you run this utility from the command prompt.

Operate without confirming actions (–y) Choosing this option automatically deletes each file without prompting you for confirmation.

The Histogram utility

The Histogram utility converts a histogram into a Microsoft Excel chart containing information about the selectivity of predicates. The utility only works with Excel 97 and higher, and only on Windows.

Converting a histogram using the dbhist command-line utility

Syntax

dbhist [options] **-t** table-name [excel-output-name]

Parameters

	Optional options	Description		
	-c options	Connection string.		
	– n colname	Column to associate with the histogram		
	– u owner	The table owner		
	-t table-name	The name of the table to generate histograms for		
	excel-output-name:	The name of the generated Excel file. If no name is specified, Excel prompts you to enter one with a Save As dialog.		
Description	Histograms are stored in retrieved with the sa_ge converts a histogram int about the selectivity of p higher, and only on Win	Histograms are stored in the SYSCOLSTAT system table and can also be retrieved with the sa_get_histogram stored procedure. The Histogram utility converts a histogram into a Microsoft Excel chart containing information about the selectivity of predicates. The utility only works with Excel 97 and higher, and only on Windows.		
	To determine the selecti use the ESTIMATE or F retrieve a histogram from the Histogram utility to	To determine the selectivity of a predicate over a string column, you should use the ESTIMATE or ESTIMATE_SOURCE functions. Attempting to retrieve a histogram from string columns causes both sa_get_histogram and the Histogram utility to generate an error.		
	Exit codes are 0 (succes	Exit codes are 0 (success) or non-zero (failure).		
	For more information "sa_get_histogram system"	For more information about the sa_get_histogram stored procedure, see "sa_get_histogram system procedure" [<i>ASA SQL Reference</i> , page 717].		
	This utility accepts @fil "@filename server optic	This utility accepts @filename parameters. For more information, see "@filename server option" on page 131.		
Example	Assuming that a histogr statement (entered all or column prod_id in the ta saves it as <i>histgram.xls</i> .	am has been created for the column, the following n the same line) generates an Excel chart for the able sales_order_items for database asademo.db, and		

The following statement generates charts for every column with a histogram in the table sales_order, assuming that asademo is already loaded. This statement also attempts to connect using uid=dba, pwd=sql. No output file name is specified, so Excel prompts you to enter one.

dbhist -t sales_order

Histogram utility options

Specify a connection string (–c) Specify connection parameters.

For a description of the connection parameters, see "Connection parameters" on page 70.

Specify a column (-n) Specify the name of the column to associate the histogram with. If you do not specify a column, all columns that have histograms in the table are returned.

Specify an owner (–u) Specify the owner of the table.

Specify a table name (-t) Specify the name of the table to generate histograms for.

The Information utility

Suntay

The Information utility can display information about a database. The utility indicates when the database was created, the name of any transaction log file or log mirror that is maintained, the page size, the version of installed Java classes, and other information. Optionally, it can also provide table usage statistics and details.

• Use the *dbinfo* command to access the Information utility.

Obtaining database information using the dbinfo command-line utility

dhinfo [ontions]

Oymax				
	Option	Description		
	-c "keyword=value; "	Database connection parameters		
	–o filename	Log output messages to a file		
	- q	Operate quietly		
	–u	Output page usage statistics		
Description	The <i>dbinfo</i> utility disp when the database was mirror, the page size, t and label, and other in statistics and details. N SYSCOLLATION tab	The <i>dbinfo</i> utility displays information about a database. It reports the time when the database was created, the name of any transaction log file or log mirror, the page size, the version of installed Java classes, the collation name and label, and other information. Optionally, it can also provide table usage statistics and details. Note that if your database does not have a SYSCOLLATION table, the collation name is not returned.		
	Exit codes are 0 (succe	Exit codes are 0 (success) or non-zero (failure).		
	For more information options" on page 483.	For more information about the options, see "Information utility options" on page 483.		
	This utility accepts @ "@filename server opt	This utility accepts @filename parameters. For more information, see "@filename server option" on page 131.		
Information utility	options			
	Connection parameter	ers (-c) Specify connection parameters.		
	For a description of parameters" on page 7	of the connection parameters, see "Connection 0.		
	Any valid user ID can	run the Information utility, but to obtain page usage		

statistics you need DBA authority.

Log output messages to file (-o) Write output messages to the named file.

Operate quietly (-q) Do not display output messages. This option is available only when you run this utility from the command prompt.

Page usage statistics (–u) Display information about the usage and size of all tables, including system and user-defined tables.

You can only request page usage statistics if no other users are connected to the database.

The Initialization utility

With the Initialization utility, you can initialize (create) a database. A number of database attributes are specified at initialization and cannot be changed later except by unloading, reinitializing, and rebuilding the entire database. These database attributes include:

- Case sensitivity or insensitivity
- Storage as an encrypted file
- Treatment of trailing blanks in comparisons
- Page size
- Collation sequence

In addition, the choice of whether to use a transaction log and a transaction log mirror is made at initialization. This choice can be changed later using the Transaction Log utility.

Exit codes are 0 (success) or non-zero (failure).

You can access the Initialization utility in the following ways:

- From Sybase Central, using the Create Database wizard.
- At the command prompt, using the *dbinit* command. This is useful for incorporating into batch or command files.

For more information about creating a database, see "CREATE DATABASE statement" [*ASA SQL Reference*, page 292].

Creating a database using the Create Database wizard

To create a database

- 1. In the left pane, select the Adaptive Server Anywhere 9 plug-in.
- 2. In the right pane, click the Utilities tab.
- 3. In the right pane, double-click Create Database.

The Create Database wizard appears.

4. Follow the instructions in the wizard.

Tip

You can also access the Create Database wizard from within Sybase Central using any of the following methods:

choosing Tools \succ Adaptive Server Anywhere 9 \succ Create Database.

Selecting a server in the left pane, and choosing Create Database from the File menu.

Right-clicking a server, and choosing Create Database from the popup menu.

For full information on creating a database in Sybase Central, see "Creating a database" [ASA SQL User's Guide, page 27].

Creating a database using the dbinit command-line utility

Syntax

dbinit [options] new-database-file

Option	Description
-b	Blank padding of strings for comparisons and fetching
- c	Case sensitivity for all string comparisons
-е	Encrypt the database using simple encryption
-ea algorithm	Specify which strong encryption algorithm to encrypt your database with: you can choose AES or MDSR
–ek key	Specify encryption key for strong encryption
-ер	Prompt for encryption key for strong encryption
i	Do not install Sybase jConnect support
—j	Do not install runtime Java classes
-ja	Install default runtime Java classes
-jdk version	Install entries for the named version of the Java Development Kit
-k	Omit Watcom SQL compatibility views SYS SYSCOLUMNS and SYS.SYSINDEXES
-l	List the recommended collating sequences
– m file-name	Use a transaction log mirror (default is no mirror)
–n	No transaction log
–o filename	Log output messages to a file

Option	Description
– p page-size	Set page size
- q	Quiet mode-do not print messages
-t log-name	Transaction log filename (default is the database name with . <i>log</i> extension)
–z col-seq	Collation sequence used for comparisons

Description For example, the database *test.db* can be created with 1024 byte pages as follows:

dbinit -p 1024 test.db

÷.

For more information about the Initialization utility options, see "Initialization utility options" on page 487.

This utility accepts @filename parameters. For more information, see "@filename server option" on page 131.

Initialization utility options

Blank padding (-b) Ignore trailing blanks for comparison purposes and pad strings that are fetched into character arrays. For example, the two strings

'Smith' 'Smith

would be treated as equal in a database created with blank-padding.

This option is provided for compatibility with the ISO/ANSI SQL standard, which is to ignore trailing blanks in comparisons. The default is that blanks are significant for comparisons.

For related information, see "ANSINULL option [compatibility]" on page 577.

Case sensitivity for all string comparisons (–c) For databases created with this option, all values are considered to be case sensitive in comparisons and string operations. Identifiers in the database are case insensitive, even in case sensitive databases.

This option is provided for compatibility with the ISO/ANSI SQL standard. The default is that all comparisons are case insensitive.

User ID and password

All databases are created with at least one user ID, **DBA**, with password **SQL**. If you create a case-sensitive database, all passwords are case sensitive. Extended characters used in passwords are case-sensitive regardless of the database sensitivity setting. User IDs, like other identifiers, are case insensitive even in case sensitive databases.

Encrypt the database using simple encryption (-e) Simple encryption makes it more difficult for someone to decipher the data in your database using a disk utility to look at the file. File compaction utilities cannot compress encrypted database files as much as unencrypted ones.

This simple encryption is equivalent to obfuscation and is intended only to keep data hidden in the event of casual direct access of the database file. For greater security, you can use strong encryption by supplying the -ea and -ek options.

Specify encryption algorithm (-ea) This option allows you to choose which strong encryption algorithm to encrypt your database with. You can choose either AES or MDSR. Algorithm names are case insensitive. If you specify the -ea option, you must also specify either -ep or -ek.

Specify encryption key (–ek) This option allows you to create a strongly encrypted database by specifying an encryption key directly in the command. If you specify the –ek option without specifying an algorithm (–ea option), the algorithm used is AES.

Prompt for encryption key (–ep) This option allows you to specify that you want to create a strongly encrypted database by inputting the encryption key in a dialog box. This provides an extra measure of security by never allowing the encryption key to be seen in clear text. If you specify the –ep option without specifying an algorithm (–ea option), the algorithm used is AES.

You must input the encryption key twice to confirm that it was entered correctly. If the keys don't match, the initialization fails.

Do not install Sybase jConnect support (-i) If you wish to use the Sybase jConnect JDBC driver to access system catalog information, you need to install jConnect support. Use this option if you wish to exclude the jConnect system objects. You can still use JDBC, as long as you do not access system information. If you want, you can add Sybase jConnect support at a later time using Sybase Central.

For more information, see "Installing jConnect system objects into a database" [*ASA Programming Guide*, page 111].

Do not install runtime Java classes (-j) If you do not intend to use Java
classes, you can specify either no Java option, or the -j option to avoid including these classes in the database.

You can add Sybase runtime Java classes later using the Upgrade Database wizard, or the ALTER DATABASE statement.

For more information, see "Java-enabling a database" [ASA Programming Guide, page 84].

Install runtime Java classes (-ja) If you wish to use Java in your database, you must install the runtime Java classes. Specifying -ja installs version 1.3 of the JDK into the database.

For example, the following command creates a database that supports JDK 1.3 applications in the database.

dbinit -ja java2.db

The runtime classes add several megabytes to the size of a database, so if you do not intend to use Java classes, you can omit the -ja option to avoid installing them.

You can add the runtime Java classes at a later time using the Upgrade Database wizard, or the ALTER DATABASE statement.

For more information, see "Java-enabling a database" [ASA Programming Guide, page 84].

Install support for the named version of the JDK (-jdk) If you want to install a version of Java other than 1.3 into your database, use the -jdk option followed by the version number. Currently, the only other version of Java supported is 1.1.8

For example, the following command creates a database that supports JDK 1.1.8 applications in the database.

dbinit -jdk 1.1.8 java2.db

The runtime classes add several megabytes to the size of a database, so if you do not intend to use Java classes, you can omit the -jdk option to avoid installing them.

The -jdk option implies the -ja option.

You can add the runtime Java classes at a later time using the Upgrade Database wizard, or the ALTER DATABASE statement.

For more information, see "Java-enabling a database" [ASA Programming Guide, page 84].

Omit Watcom SQL compatibility views (-k) By default, database creation

generates the views SYS.SYSCOLUMNS and SYS.SYSINDEXES for compatibility with system tables that were available in Watcom SQL (versions 4 and earlier of this software). These views will conflict with the Sybase Adaptive Server Enterprise compatibility views dbo.syscolumns and dbo.sysindexes unless the -c case sensitivity option is used.

List the available collating sequences (-I) *dbinit* lists the recommended collating sequences and then stops. No database is created. A list of available collating sequences is automatically presented in the Sybase Central Create Database wizard.

Use a transaction log mirror (-m) A transaction log mirror is an identical copy of a transaction log, usually maintained on a separate device, for greater protection of your data. By default, Adaptive Server Anywhere does not use a mirrored transaction log.

Do not use a transaction log (-n) Creating a database without a transaction log saves disk space. The transaction log is required for data replication and provides extra security for database information in case of media failure or system failure. Databases that do not use transaction logs typically run slower than databases that use transaction logs.

Log output messages to file (-o) Write output messages to the named file.

Page size (-p) The page size for a database can be (in bytes) 1024, 2048, 4096, 8192, 16384, or 32768, with 2048 being the default. Any other value for the size will be changed to the next larger size.

Large databases usually benefit from a larger page size. For example, the number of I/O operations required to scan a table is generally lower, as a whole page is read in at a time. Also, the number of levels in an index may be reduced as more entries can be stored on each page.

However, there are additional memory requirements for large page sizes. Also, the maximum number of rows stored on a page is 255, so that tables with small rows will not fill each page. For example, with 8 kb pages the average row size must be at least 32 bytes to fully use each page. For most applications, 16 kb or 32 kb page sizes are not recommended. You should not use page sizes of 16 kb or 32 kb in production systems unless you can be sure that a large database server cache is always available, and only after you have investigated the tradeoffs of memory and disk space with its performance characteristics.

Operate quietly (-q) Do not display output messages. This option is available only when you run this utility from the command prompt.

Set the transaction log filename (-t) The transaction log is a file where

the database server logs all changes, made by all users, no matter what application system is being used. The transaction log plays a key role in backup and recovery (see "The transaction log" on page 343), and in data replication. If the filename has no path, it is placed in the same directory as the database file. If you run *dbinit* without specifying -t or -n, a transaction log is created with the same filename as the database file, but with extension .log.

Collating sequence (–z) The collation sequence is used for all string comparisons in the database.

If you want to create a custom collation, use the Collation utility to create a file containing the collation. Once you have modified the collation and inserted it into the appropriate scripts, you use the Initialization utility to create the database and specify the new collation.

You must change the collation label in the custom collation file. Otherwise, the Initialization utility prevents the insertion of the new collation, since it conflicts with an existing collation.

For more information on custom collating sequences, see "International Languages and Character Sets" on page 285. For information on the Collation utility, see "The Collation utility" on page 462.

In order to change the collation that an existing database uses, it is necessary to unload the database, create a new database using the appropriate collation, and then reload the database. It may be necessary to translate the data as well.

If -z is not specified, the default collation is used. Normal ASCII (binary) ordering is used for the lower 128 characters, subject to the case sensitivity setting (-c). For the upper 128 characters (also called the extended characters), any character that is an accented form of a letter in the lower 128 are sorted to the same position as the unaccented form. The determination of whether or not an extended character is an accented letter is based upon code page 850 (multilingual code page).

For a list of the available collating sequence labels, see "Understanding collations" on page 307.

The Interactive SQL utility

Interactive SQL provides an interactive environment for database browsing and for sending SQL statements to the database server.

You can access Interactive SQL:

- from Sybase Central, using the Open Interactive SQL option.
- at the command prompt, using the *dbisql* command.

Starting Interactive SQL from Sybase Central

To open Interactive SQL from Sybase Central

- 1. Open the Utilities folder in the left pane.
- 2. Double-click Open Interactive SQL in the right pane.

The Interactive SQL window appears.

Tip

You can also access Interactive SQL by choosing Start > Programs > SQL Anywhere 9 > Adaptive Server Anywhere 9 > Interactive SQL, or from within Sybase Central by choosing any of the following options:

choosing Tools \triangleright Adaptive Server Anywhere 9 \triangleright Open Interactive SQL.

Selecting a database in the left pane, and choosing Open Interactive SQL from the File menu.

Right-clicking a database, and choosing Open Interactive SQL from the popup menu.

Right-clicking a stored procedure, and choosing Execute From Interactive SQL from the popup menu. Interactive SQL opens with the procedure in the SQL Statements pane.

Opening Interactive SQL using the dbisql command-line utility

Syntax

dbisql [options] [dbisql-command | command-file]

Option	Description
-c "keyword=value;"	Supply database connection parameters
-codepage codepage	Specify a codepage to use when reading or writing files

	Option	Description		
	-d delimiter	Use the given string as the command delimiter		
	-d1	Print statements as they are executed [command- prompt mode only]		
	-f fileName	Open (without running) the file called 'fileName'.		
	-datasource dsn- name	Specify an ODBC data source to connect to		
	-host hostname	Specify the hostname or IP address of the machine running a database server		
	-jConnect	Use jConnect to connect to the database		
	–nogui	Run in command-prompt mode		
	-ODBC	Use the iAnywhere JDBC Driver to connect to the database		
	-onerror { continue exit }	Override the ON_ERROR option for all users		
	-port portnumber	Look on the specified port number for the database server		
	-q	Quiet mode-no windows or messages		
	- x	Syntax check only-no commands executed		
Description	Interactive SQL allows also provides feedback for each command, the	you to type SQL commands, or run command files. I about the number of rows affected, the time required execution plan of queries, and any error messages.		
	If <i>dbisql-command</i> is s You can also specify a specified, Interactive So command into a command	If <i>dbisql-command</i> is specified, Interactive SQL executes the command. You can also specify a command file name. If no <i>dbisql-command</i> is specified, Interactive SQL enters interactive mode, where you can type a command into a command window.		
	Exit codes are 0 (succes	Exit codes are 0 (success) or non-zero (failure).		
	Interactive SQL require be set to ON since a nu statements, rely on this automatically sets it ON	Interactive SQL requires that the QUOTED_IDENTIFIER database option be set to ON since a number of database functions, including some statements, rely on this setting to function properly. Interactive SQL automatically sets it ON when connecting to a database.		
	This utility does <i>not</i> ac see "@filename server	This utility does <i>not</i> accept @filename parameters. For more information, see "@filename server option" on page 131.		

Examples The following command, entered at a command prompt, runs the command file *mycom.sql* against the current default server, using the user ID DBA and the password SQL. If there is an error in the command file, the process terminates.

```
dbisql -c "uid=DBA;pwd=SQL" -onerror exit mycom.sql
```

The following command, when entered on a single line at a system prompt, adds a user to the current default database:

```
dbisql -c "uid=DBA;pwd=SQL" grant connect to joe identified by passwd
```

Interactive SQL utility options

-c Specify connection parameters. See "Connection parameters" on page 70 for a description of the connection parameters. If Interactive SQL cannot connect, you are presented with a dialog where you can enter the connection parameters.

-codepage Specify the codepage to use when reading or writing files. The default code page is the default code page for the platform you are running on.

For example, on an English Windows NT machine, Interactive SQL uses the 1252 (ANSI) code page. If you want Interactive SQL to read files created using the 297 (IBM France) code page, specify the following option.

-codepage 297

For a list of supported code pages, see "Supported code pages" on page 292.

-d Specify a command delimiter. Quotation marks around the delimiter are optional, but required when the command shell itself interprets the delimiter in some special way.

Command delimiters are used for all connections in that Interactive SQL session, regardless of the setting stored in the database (for the user, or the PUBLIC setting).

-d1 (The final character is a number 1, not a lower-case L). Interactive SQL echoes all statements it executes to the Command window (STDOUT). This can provide useful feedback for debugging SQL scripts, or when Interactive SQL is processing a long SQL script.

-datasource Specify an ODBC data source to connect to. You do not need to be using the iAnywhere JDBC driver to use this option. However, if the data source to which you are connecting is not configured to use TCP/IP, you

must use the iAnywhere JDBC driver to connect. Adaptive Server Anywhere data sources are configured to use TCP/IP by default.

-f filename Open (but do not run) the file called *filename*. The file name can be enclosed in quotation marks, and MUST be enclosed in quotation marks if the file name contains a blank. If the file does not exist, or if it's really a directory instead of a file, DBISQL prints an error message to the console and then quits. If the file name does not include a full drive and path specification, it is assumed to be relative to the current directory.

-host Specify the hostname or IP address of the computer on which the database server is running. You can use the name **localhost** to represent the current machine.

-jConnect Use the Sybase jConnect JDBC driver to connect to the database. This is the default method, and is recommended in most circumstances.

-nogui Run Interactive SQL in a command-prompt mode, with no windowed user interface. This is useful for batch operations.

In this mode, Interactive SQL sets the program exit code to indicate success or failure. On Windows operating systems, the environment variable ERRORLEVEL is set to the program exit code. The exit codes are as follows:

Program	exit code	Description
0		Success.
1		General failure. At some point, a SQL or Interactive SQL statement did not execute successfully and the user chose to stop executing SQL statements. Alternatively, Interactive SQL noted an internal error.
5		The user terminated interactive SQL. When an error occurs during execution, the user is prompted to ignore it, stop, or exit Interactive SQL. If the user opts to exit, the program returns code 5. Code 5 is also returned if an error occurs and the Interactive SQL option ON_ERROR is set to EXIT.
9		Unable to connect.
255		Bad command. The command contained incomplete or invalid options.
-ODBC	Connect usi	ing the iAnywhere JDBC driver. If you do not specify

this option, Interactive SQL connects using jConnect.

-onerror Controls what happens if an error is encountered while reading statements from a command file. This option overrides the ON_ERROR setting. It is useful when using Interactive SQL in batch operations.

For more information, see "ON_ERROR option [ISQL]" on page 611.

-port Specify the port number on which the database server is running. The default port number for Adaptive Server Anywhere is **2638**.

-q Do not display output messages. This is useful only if you start Interactive SQL with a command or command file.

-x Scan commands but do not execute them. This is useful for checking long command files for syntax errors.

For detailed descriptions of SQL statements and Interactive SQL commands, see "SQL Language Elements" [*ASA SQL Reference*, page 3].

The Language utility

The Language utility reports and changes the registry settings that control the languages used by Adaptive Server Anywhere and Sybase Central.

Managing languages using the dblang command-line utility

Syntax

dblang { options } [language-code]

Option	Description
-d	Change deployment settings only
- q	Operate quietly-do not print messages

Language code	Language
DE	German
EN	English
ES	Spanish
FR	French
IT	Italian
JA	Japanese
КО	Korean
PL	Polish
РТ	Portuguese
RU	Russian
TW	Traditional Chinese
UK	Ukrainian
ZH	Simplified Chinese

Description The utility is installed only when the International Resources Deployment Kit (IRDK) is selected during installation.

Running *dblang* without a language code reports the current settings. These settings are as follows:

♦ Adaptive Server Anywhere A key from HKEY_LOCAL_MACHINE that holds a two-letter language code from the table above.

This setting controls which language resource library is used to deliver informational and error messages from the Adaptive Server Anywhere database server. The language resource library is a DLL with a name of the form *dblgXX9.dll*, where XX is a two-letter language code. Ensure that you have the appropriate language resource library on your machine when you change the settings. • Sybase Central A key from HKEY_LOCAL_MACHINE that holds a two-letter language code from the table above. This setting controls the resources used to display user interface elements for Sybase Central and Interactive SQL. You must have purchased the appropriate localized version of SQL Anywhere Studio for this setting to take effect. Exit codes are 0 (success) or non-zero (failure). This utility does not accept @filename parameters. For more information, see "@filename server option" on page 131. Examples If you selected the International Resources Deployment Kit (IRDK) during installation, the following command displays a dialog box containing the current settings:

dblang



The following command changes the settings to French, and displays a dialog containing the previous and new settings:

dblang FR

Language utility options

Deployment only (-d) Changes the registry setting only for Adaptive Server Anywhere. Do not change the Sybase Central setting.

Operate quietly (-q) Do not display output messages.

The License utility

The license utility adds licensed users to your network database server.

Managing licenses using the dblic command-line utility

Syntax

dblic { options } executable-name user-name company-name

ī

	Option	Description
	–l type	License type. Allowed values are perseat or concurrent
	–o filename	Log output messages to a file
	– p operating-system	Target operating system. Allowed values are NetWare , UNIX , WIN32 , WINNT
	-q	Operate quietly—do not print messages
	–u license-number	Total number of users for license
	Executable-name	The executable to be licensed, with path relative to the current directory. Should be either an Adaptive Server Anywhere network database server or a MobiLink synchronization server.
Description	The License utility adds h must only use this utility to license the number of u command does <i>not</i> grant the current license inform server.	icensed users to your network database server. You in accordance with your Sybase license agreement users to which you are entitled. Running this you license. You can also use this utility to view ation for a server executable without starting the
	Repeated running of the c	<i>Iblic</i> command updates the license information.
	Exit codes are 0 (success)	or non-zero (failure).
	On UNIX, the dbsrv and of the license [dblic] utility of dbsrv and dbmlsrv execut before you use the license	dbmlsrv executables are not writable, and so using with those executables will fail. Make sure the ables are writable (for example, using chmod +w) [dblic] utility.
	This utility accepts @filer "@filename server option	name parameters. For more information, see "on page 131.
Example	The following command, server executable, applies <i>Sys Admin</i> , for company	executed in the same directory as the database a license for 50 concurrent users, in the name of <i>My Co</i> , to a Windows NT network database

server. The command must be entered all on one line:

```
dblic -l concurrent -p WINNT -u 50 dbsrv9.exe "Sys Admin" "My Co"
```

The following message appears on the screen to indicate the success of the license:

Licensed nodes: 50 User: Sys Admin Company: My Co

License utility options

License type (–I) Enter the license type that matches the licensing model described in your software license agreement. Valid entries are **perseat** and **concurrent**.

Log output messages to file (–o) Write output messages to the named file.

Operating system (–p) Enter the operating system for which you are licensed. Allowed values are as follows:

- ◆ **NetWare** Novell NetWare operating system.
- UNIX A UNIX operating system.
- ♦ WIN32 Windows operating systems.
- WINNT This has the same meaning as WIN32.

Operate quietly (–q) Do not display output messages.

License number (–u) The total number of licensed seats or concurrent connections. If you are adding extra licenses, this is the total, not the number of additional licenses.

Executable name Enter the path and file name of the network database server executable (*dbsrv9.exe*) or MobiLink synchronization server (*dbmlsrv9.exe*) you are licensing. The path must be either absolute or relative to the current working directory.

You can view the current license information for a server executable without starting the server by entering only the executable name.

User name A user name for the license. This name appears on the database server window on startup. If there are spaces in the name, enclose it in double quotes.

Company name The company name for the license. This name appears on the database server window on startup. If there are spaces in the name, enclose it in double quotes.

The Log Transfer Manager

The Log Transfer Manager (LTM) is also known as a replication agent.

The LTM is required for any Adaptive Server Anywhere database that participates in a Replication Server installation as a primary site.

Using the dbltm command-line utility

Syntax

dbltm [options]

Parameters

Option	Description
-A	Do not filter updates
-C config_file	Use the named configuration file
–I interface_file	Use the named interface file
$-\mathbf{M}$	Recovery mode
-S LTM_name	Specify an LTM name
–dl	Display log messages on screen
–ek key	Specify encryption key
-ер	Prompt for encryption key
–o filename	Log output messages to a file
-os size	Maximum size of output file
–ot file	Truncate file, and log output messages to file
- q	Run in minimized window
-s	Show Log Transfer Language (LTL) commands
–ud	Run as a daemon [UNIX]
- v	Verbose mode

Description The Adaptive Server Anywhere LTM reads a database transaction log and sends committed changes to Replication Server. The LTM is not required at replicate sites.

The LTM sends messages to Replication Server in a language named Log Transfer Language (LTL).

By default, the LTM uses a log file named *DBLTM.LOG* to hold status and other messages. You can use options to change the name of this file and to change the volume and type of messages that are sent to it.

Exit codes are 0 (success) or non-zero (failure).

This utility accepts @filename parameters. For more information, see "@filename server option" on page 131.

Log Transfer Manager utility options

Use configuration file (–C) Use the configuration file *config_file* to determine the LTM settings. The default configuration file is *dbltm.cfg*.

For a description of the configuration file, see "The LTM configuration file" on page 503.

Use interface file (–I) (Upper case i.) Use the named interfaces file. The interfaces file is the file created by SQLEDIT which holds the connection information for Open Servers. The default interfaces file is *SQL.ini* in the *ini* subdirectory of your *Sybase* directory.

Specify LTM name (–S) Provides the server name for this LTM. The default LTM_name is DBLTM_LTM. The LTM_name must correspond to the Open Server name for the LTM that was entered in SQLEDIT.

Do not filter updates (-A) Do not filter updates. By default, all changes made by the maintenance user are not replicated. If the -A option is set, these changes are replicated. This may be useful in non-hierarchical Replication Server installations, where a database acts as both a replicate site and as a primary site.

Recover mode (–M) This is used to initiate recovery actions. The LTM starts reading logs from the earliest available position. If the offline directory is specified in the configuration file, the LTM reads from the oldest offline log file.

Specify encryption key (-ek) This option allows you to specify the encryption key for strongly encrypted databases directly in the command. If you have a strongly encrypted database, you must provide the encryption key to use the database or transaction log in any way, including offline transaction logs. For strongly encrypted databases, you must specify either -ek or -ep, but not both. The command fails if you do not specify a key for a strongly encrypted database.

Prompt for encryption key (–ep) This option allows you to specify that you want to be prompted for the encryption key. This option causes a dialog box to appear, in which you enter the encryption key. It provides an extra

measure of security by never allowing the encryption key to be seen in clear text. For strongly encrypted databases, you must specify either -ek or -ep, but not both. The command fails if you do not specify a key for a strongly encrypted database.

Display Log Transfer Language messages (-dl) Display Log Transfer Language (LTL) messages in the LTM window or at the command prompt and also in the log file. This should be used only to diagnose problems, and is not recommended in a production environment. It carries a significant performance penalty.

Log output messages to file (-o) Use a log file different from the default (*dbltm.log*). Write output messages from log transfer operations to this file.

Limit size of output file (-os) Specify the maximum size of the output file, in bytes. The minimum value is 10000 (ten thousand). If the log file grows to the point where it would exceed this limit, it is renamed to *yymmddAA.log*, and a new file named *yymmddAB.log* is started. The value of *xx* in *yymmddxx.log* is incremented for each file created on a given day.

Truncate log file and use named log file (-ot) Use a log file different from the default (*dbltm.log*), and truncate the log file (all existing content is deleted) when the LTM starts. Output messages from log transfer operations are sent to this file for later review.

Operate quietly (-q) Minimize the window when the LTM is started.

Log all LTL commands (-s) Log all LTL commands that are generated by the LTM. This should be used only to diagnose problems, and is not recommended in a production environment. It carries a significant performance penalty.

Run as a daemon (–ud) You can run the LTM as a daemon on UNIX operating systems. If you run in this manner, output is logged to the log file.

Operate in verbose mode (–v) Displays messages, other than LTL messages, for debugging purposes.

The LTM configuration file

	The Adaptive Server Anywhere and Adaptive Server Enterprise LTM
	configuration files are very similar. This section describes the entries in the
	Adaptive Server Anywhere LTM configuration file, and the differences from
	the Adaptive Server Enterprise LTM configuration file.
	The configuration file that an LTM uses is specified using the -C option.
LTM configuration file parameters	The following table describes each of the configuration parameters that the

LTM recognizes. Parameters that are used by the Adaptive Server Enterprise LTM but not by the Adaptive Server Anywhere LTM are included in this list, and marked as either ignored (in which case they may be present in the configuration file, but have no effect) or as unsupported (in which case they will cause an error if present in the configuration file).

Parameter	Description
APC_pw	The password for the APC_user login name. This entry is present only in Adaptive Server Anywhere LTM configuration files.
APC_user	A user ID that is used when executing asyn- chronous procedures at the primary site. This user ID must have permissions appropriate for all asynchronous procedures at the primary site. This entry is present only in Adaptive Server Anywhere LTM configuration files.
backup_only	By default, this is off. If set to on, the LTM will replicate only backed-up transactions.
batch_ltl_cmds	Set to on (the default) to use batch mode. Batch mode can increase overall throughput, but may lead to longer response times.
batch_ltl_sz	The number of commands that are saved in the buffer before being sent to Replication Server, when batch_ltl_cmds is on . The default is 200.
batch_ltl_mem	The amount of memory that the buffer can use before its contents are sent to Replication Server, when batch_ltl_cmds is on . The default is 256 kb.
Continuous	By default, this is on. When set to off, the LTM automatically shuts down as soon as all committed data has been replicated.
LTM_admin_pw	The password for the LTM_admin_user login name.
LTM_admin_user	The system administrator LTM login name that is used to log in to the LTM. This parameter is required so that the LTM can check whether a user logging on to the LTM to shut it down has the correct login name.

Parameter	Description
LTM_charset	The Open Client/Open Server character set for the LTM to use.
LTM_language	The Open Client/Open Server language for the LTM to use.
LTM_sortorder	The Open Client/Open Server sort order for the LTM to use to compare user names. You can specify any Adaptive Server Enterprise-supported sort order that is compatible with the LTM's character set. All sort orders in your replication system should be the same.
	The default sort order is a binary sort.
maint_cmds_to_skip	[ignored]
qualify_table_owner	Set to on for the LTM to send LTLs with table names and columns names, as well as table owners to Replication Server. The setting applies to all replicating tables, and the create replication definition statements must match this setting. The default is off .
rep_func	Set to on to use asynchronous procedure calls (APCs). The default is off .
Retry	The number of seconds to wait before retrying a failed connection to an Adaptive Server Anywhere database server or Replication Server. The default is 10 seconds.
RS	The name of the Replication Server to which the LTM is transferring the log.
RS_pw	The password for the RS_user login name.
RS_source_db	The name of the database whose log the LTM transfers to the Replication Server. This name must match the name of the database as defined within the Replication Server connection definitions. Most configurations use the same setting for both RS_Source_db and SQL_database configuration options.

Parameter	Description
RS_source_ds	The name of the server whose log the LTM trans- fers to the Replication Server. This name must match the name of the server as defined within the Replication Server connection definitions. Most configurations use the same setting for both RS Source_ds and SQL_server configuration options.
RS_user	A login name for the LTM to use to log into Replication Server. The login name must have been granted connect source permission in the Replication Server.
scan_retry	The number of seconds that the LTM waits be- tween scans of the transaction log. This parameter is somewhat different in meaning to that of the Adaptive Server Enterprise LTM. The Adaptive Server Anywhere server does not <i>wake up</i> and scan the log when records arrive in the log. For this reason, you may wish to set the <i>scan_retry</i> value to a smaller number then that for an Adap- tive Server Enterprise LTM.
skip_ltl_cmd_err	[ignored]
SQL_database	The primary site database name on the server SQL_server to which the LTM connects. For Adaptive Server Enterprise during recovery, this is the temporary database whose logs the LTM will transfer to Replication Server. The Adaptive Server Anywhere LTM uses the SQL_log_files parameter to locate offline transaction logs.
SQL_log_files	A directory that holds off-line transaction logs. The directory must exist when the LTM starts up. This entry is present only in Adaptive Server Anywhere LTM configuration files.
SQL_pw	The password for the SQL_user user ID.

Parameter	Description
SQL_server	The name of the primary site Adaptive Server Anywhere server to which the LTM connects. For Adaptive Server Enterprise during recovery, this is a data server with a temporary database whose logs the LTM will transfer to Replication Server. The LTM uses the SQL_log_files parameter to locate offline transaction logs.
SQL_user	The login name that the LTM uses to connect to the database specified by RS_source_ds and RS_source_db.

Example configuration	• The following is a sample LTM configuration file.	
file	# This is a comment line	
	# Names are case sensitive.	
	SQL_user=sa	
	SQL_pw=sysadmin	
	SQL_server=PRIMESV	
	SQL_database=primedb	
	RS_source_ds=PRIMEOS	
	RS_source_db=primedb	
	RS=MY_REPSERVER	
	RS_user=sa	
	RS_pw=sysadmin	
	LTM_admin_user=DBA	
	LTM_admin_pw=SQL	
	LTM_charset=cp850	
	scan_retry=2	
	SQL_log_files=e:\logs\backup	
	APC_user=sa	
	APC_pw=sysadmin	

The Log Translation utility

With the Log Translation utility, you can translate a transaction log into a SQL command file.

You can access the Log Translation utility in the following ways:

- From Sybase Central, using the Translate Log File wizard.
- At the command prompt, using the *dbtran* command. This is useful for incorporating into batch or command files.

Translating a transaction log using the Translate Log File wizard

To translate a transaction log into a command file

- 1. In the left pane, select the Adaptive Server Anywhere 9 plug-in.
- 2. In the right pane, click the Utilities tab.
- 3. In the right pane, double-click Translate Log File. The Translate Log File wizard appears.
- 4. Follow the instructions in the wizard.

Тір

You can also access the Translate Log File wizard by choosing Tools ► Adaptive Server Anywhere 9 ► Translate Log File.

You can also access the Translate Log File wizard from within Sybase Central using any of the following methods:

choosing Tools \blacktriangleright Adaptive Server Anywhere 9 \triangleright Translate Log File.

Selecting a database in the left pane, and choosing Translate Log File from the File menu.

Right-clicking a database, and choosing Translate Log File from the popup menu.

Translating a transaction log using the dbtran command-line utility

Syntax	Running against a transaction log.
	dbtran [options] transaction-log [SQL-file]
	Running against a database server.
	dbtran [options]

Option	Description
-a	Include uncommitted transactions
-c "keyword=value; …"	Supply database connection parameters—cannot be used with a transaction log name
-d	Display output in chronological order
– ek key	Specify encryption key
-ер	Prompt for encryption key
- f	Output only since the last checkpoint
-g	Include audit records in output
-ir offset1,offset2	Include only the portion of the log between the two specified offsets
–is source,	Include only rows originating from the specified sources
-it user.table,	Include only operations on specified tables by specifying a comma-delimited list of user.table
–j date/time	Output from the last checkpoint prior to the given time
m	Specify transaction logs directory (requires -n option)
–n filename	Output SQL file, when used against a database server
–o filename	Log output messages to a file
- q	Run quietly, do not print messages
-r	Remove uncommitted transactions (default)
– rsu username,	Override default Replication Server user names
S	Produce ANSI standard SQL UPDATE transac- tions
-sr	Generate SQL Remote comments
-t	Include trigger-generated transactions in output
– u userid,	Translate transactions for listed users only
– x userid,	Exclude transactions for listed users

Option	Description	
-y	Replace file without confirmation	
-Z	Include trigger-generated transactions as com- ments only	
Transaction-log	Log file to be translated—cannot be used together with $-c$ or $-n$	
SQL-file	Output file containing the translated information— for use with <i>transaction-log</i> only	

Description

The *dbtran* utility takes the information in a transaction log and places it as a set of SQL statements and comments into an output file. The utility can be run in the following ways:

◆ Against a database server Run in this way, the utility is a standard client application. It connects to the database server using the connection string specified following the -c option, and places output in a file specified with the -n option. DBA authority is required to run in this way.

The following command translates log information from the server **asademo9** and places the output in a file named *asademo.SQL*.

◆ Against a transaction log file Run in this way, the utility acts directly against a transaction log file. You should protect your transaction log file from general access if you wish to prevent users from having the capability of running this statement.

```
dbtran asademo.log asademo.sql
```

When the *dbtran* utility runs, it displays the earliest log offset in the transaction log. This can be an effective method for determining the order in which multiple log files were generated.

Exit codes are 0 (success) or non-zero (failure).

For more information about the Log translation utility options, see "Log translation utility options" on page 510.

This utility accepts @filename parameters. For more information, see "@filename server option" on page 131.

Log translation utility options

Include uncommitted transactions (-a) The transaction log contains any

changes made before the most recent COMMIT by any transaction. Changes made after the most recent commit are not present in the transaction log.

Connection string (–c) When running the utility against a database server, this parameter specifies the connection string.

DBA authority is required to run dbtran.

For a description of the connection parameters, see "Connection parameters" on page 70.

Display output in chronological order (-d) Transactions are displayed in order from earliest to latest. This feature is provided primarily for use when auditing database activity: the output of this command should not be applied against a database.

Specify encryption key (–ek) This option allows you to specify the encryption key for strongly encrypted databases directly in the command. If you have a strongly encrypted database, you must provide the encryption key to use the database or transaction log in any way.

For strongly encrypted databases, you must specify either -ek or -ep, but not both. The command will fail if you do not specify a key for a strongly encrypted database.

If you are running against a database server (using the -c option), make sure you specify the key using a connection parameter, not using the -ek option. For example, the following command gets the transaction log information about database *enc.db* from engine **sample**, and saves its output in *log.sql*.

```
dbtran -n log.sql -c
eng=sample;dbf=enc.db;uid=dba;pwd=sql;dbkey=mykey
```

Prompt for encryption key (-ep) This option allows you to specify in the command that you want to be prompted for the encryption key. This option causes a dialog box to appear, in which you enter the encryption key. It provides an extra measure of security by never allowing the encryption key to be seen in clear text.

For strongly encrypted databases, you must specify either -ek or -ep, but not both. The command will fail if you do not specify a key for a strongly encrypted database.

If you are running against a database server (using the -c option), make sure you specify the key using a connection parameter, not using the -ep option. For example, the following command gets the transaction log information about database *enc.db* from engine **sample**, and saves its output in *log.sql*.

```
dbtran -n log.sql -c
    eng=sample;dbf=enc.db;uid=dba;pwd=sql;dbkey=mykey
```

Output from last checkpoint only (-f) Only transactions that were completed since the last checkpoint are output.

Include audit information (–g) If the AUDITING database option is turned on, auditing information is added to the transaction log. You can include this information as comments in the output file using this option.

For more information, see "AUDITING option [database]" on page 578.

Include rows from specified sources (-is) Isolate operations on rows that have been modified by operations from one or more of the following sources, specified as a comma-separated list:

- All All rows. This is the default setting.
- **SQLRemote** Include only rows that were modified using SQL Remote. You can also use the short form **SR**.
- RepServer Include only rows that were modified using the Replication Agent (LTM) and Replication Server. You can also use the short form RS.
- ◆ Local Include only rows that are not replicated.

Include offset range (-ir) Isolate a portion of the transaction log between two specified offsets.

Include specified tables (-it) Isolate those operations on the specified, comma-separated list of tables. Each table should be specified as *owner.table*.

Output from the last checkpoint prior to a given date (–j) Only transactions from the most recent checkpoint prior to the given date and/or time are translated. The user-provided argument can be a date, time or date and time enclosed in quotes. If the time is omitted, the time is assumed to be the beginning of the day. If the date is omitted, the current day is assumed. The following is an acceptable format for the date and time: "YY/MMM/DD HH:NN".

Transaction logs directory (-m) Use this option to specify a directory that contains transaction logs. This option must be used in conjunction with the -n option.

Output file (-n) When you run the *dbtran* utility against a database server, use this option to specify the output file that holds the SQL statements.

Log output messages to file (-o) Write output messages to the named file.

Operate quietly (-q) Do not display output messages. This option is available only when you run this utility from the command prompt.

Do not include uncommitted transactions (–r) Remove any transactions that were not committed. This is the default behavior.

Override Replication Server user names (-rsu) By default, the -is option assumes the default Replication Server user names of dbmaint and sa. You can override this assumption using the -rsu option with a comma-separated list of user names.

Generate ANSI standard SQL UPDATE (-s) If the option is not used, and there is no primary key or unique index on a table, the Translation utility generates UPDATE statements with a non-standard FIRST keyword in case of duplicate rows. If the option is used, the FIRST keyword is omitted for compatibility with the SQL standard.

Generate SQL Remote comments (-sr) Place generated comments in the output file describing how SQL Remote distributes operations to remote sites.

Include transactions generated by triggers (-t) By default, actions carried out by triggers are not included in the command file. If the matching trigger is in place in the database, when the command file is run against the database, the trigger will carry out the actions automatically. Trigger actions should be included if the matching trigger does not exist in the database against which the command file is to be run.

Output transactions for listed users only (–u) This option allows you to limit the output from the transaction log to include only specified users.

Output transactions except for listed users (–x) This option allows you to limit the output from the transaction log to exclude specified users.

Operate without confirming actions (–y) Choosing this option automatically replaces existing command file(s) without prompting you for confirmation.

Include transactions generated by triggers as comments only (-z) Transactions that were generated by triggers will be included only as comments in the output file.

The Ping utility

The Ping utility is provided to assist in troubleshooting connection problems.

Troubleshooting connections using the dbping command-line utility

i.

Syntax

dbping [options]

	Option	Description
	-c "keyword=value; "	Database connection parameters
	-d	Make a database connection if the server is found
	–l library	Load specified ODBC driver or library
	–m	Use ODBC driver manager
	–o filename	Log output messages to a file
	– pc property,	Report specified connection properties
	– pd property,	Report specified database properties
	– ps property,	Report specified database server properties
	- q	Operate quietly—do not print messages
	-Z	Display debugging information
Description	The <i>dbping</i> utility is a tool to help debug connection problems. It takes a full or partial connection string and returns a message indicating whether the attempt to locate a server or database, or to connect, was successful. The utility can be used for embedded SQL or ODBC connections. It cannot be used for jConnect (TDS) connections.	
	Exit codes are 0 (success) or non-zero (failure). This utility accepts @filename parameters. For more information, see "@filename server option" on page 131.	
Ping utility options		
	Connection parameters ((-c) For a description of the connection

Connection parameters (–c) For a description of the connection parameters, see "Connection parameters" on page 70. If the connection parameters are not specified, connection parameters from the

SQLCONNECT environment variable are used, if set.

Make database connection (-d) Ping the database, not just the server.

If you do not supply the -d option, then *dbping* reports success if it finds the server specified by the -c option. If you do supply the -d option, then *dbping* reports success only if connects to the server and also connect to a database.

For example, if you have a server named **blair** running the database sample, the following succeeds:

dbping -c "eng=blair;dbn=asademo"

The following command fails, with the message Ping database failed - specified database not found:

dbping -c "eng=blair;dbn=sample"

Load specified library (–I) Specify the library to use (without its file extension). This option avoids the use of the ODBC driver manager, and so is particularly useful on UNIX operating systems.

For example, the following command loads the ODBC driver directly:

dbping -m -c "dsn=ASA 9.0 Sample" -1 dbodbc9

On UNIX, if you wish to use a threaded connection library, you must use the threaded version of the Ping utility, *dbping_r*.

Use ODBC to connect (-m) Establish a connection using ODBC. By default, the utility connects using the embedded SQL interface.

Report connection properties (–pc) Upon connection, display the specified connection properties. Supply the properties in a comma-separated list. You must specify enough connection information to establish a database connection if you use this option.

For a list of connection properties, see "Connection-level properties" on page 647.

For example, the following command displays the DIVIDE_BY_ZERO_ERROR option setting, which is available as a connection property.

dbping -c ... -pc Divide_by_zero_error

Report database properties (–pd) Upon connection, display the specified database properties. Supply the properties in a comma-separated list. You must specify enough connection information to establish a database connection if you use this option.

For a list of database properties, see "Database-level properties" on page 664.

For example, the following command displays the Java version in use by the database:

```
dbping -c ... -pd JDKVersion
```

Report database server properties (–ps) Upon connection, display the specified database server properties. Supply the properties in a comma-separated list.

For a list of database server properties, see "Server-level properties" on page 657.

For example, the following command displays the number of licensed seats for the database server:

dbping -c ... -ps LicenseCount

Log output messages to file (-o) Write output messages to the named file.

Operate quietly (-q) If *dbping* fails, a message is always displayed. If *dbping* succeeds, no message appears if -q is specified.

Display debugging information (–z) This option is available only when an embedded SQL connection is being attempted. That is, it cannot be combined with -m or -1. It displays the network communication protocols used to attempt connection, and other diagnostic messages.

The Rebuild utility

Databases can be rebuilt using the Rebuild batch file, command file, or shell script, which invokes a series of utilities to rebuild a database, or as part of the unload process using the Unload Database wizard in Sybase Central.

For more information about the Unload Database wizard, see "The Unload utility" on page 533.

Rebuilding a database using the rebuild batch or command file

Syntax	rebuild old-database new-database [DBA-password]		
See also	"Unloading a database using the dbunload command-line utility" on page 534		
	"Creating a database using the dbinit command-line utility" on page 486		
	"The Interactive SQL utility" on page 492		
Description	This batch file, command file, or shell script uses <i>dbunload</i> to rebuild <i>old-database</i> into <i>new-database</i> . This is a simple script, but it helps document the rebuilding process, and provides a basis for customization. Both database names should be specified without extensions. An extension of <i>.db</i> is automatically added.		
	You can use <i>dbunload</i> with the -ar option to carry out unloading and reloading without using the rebuild batch file.		
	The <i>DBA-password</i> must be specified if the password to the DBA user ID in the <i>old-database</i> is not the initial password SQL.		
	Rebuild runs the <i>dbunload</i> , <i>dbinit</i> , and Interactive SQL commands with the default options. If you need different options, you will need to run the three commands separately.		
	Exit codes are 0 (success) or non-zero (failure).		
	This utility accepts @filename parameters. For more information, see "@filename server option" on page 131.		

The Server Location utility

The server location utility is provided to assist in diagnosing connection problems by locating database servers on the immediate TCP/IP network.

Locating servers using the dblocate command-line utility

Syntax	dblocate [options]		
	Option	Description	
	–o filename	Log output messages to a file	
	- q	Operate quietly-do not print messages	
Description	The <i>dblocate</i> utility locates any Adaptive Server Anywhere database ser running over TCP/IP on the immediate network. It prints a list of databa servers and their addresses.		
	Depending on your network, the utility may take several seconds before printing its results.		
	Exit codes are 0 (success) or non-zero (failure).		
	This utility accepts @filename parameters. For more information, see "@filename server option" on page 131.		
	The database server can register itself with an LDAP server which keeps track of all servers in an enterprise. This allows both clients and the Server Location utility [dblocate] to find them regardless of whether they are on a WAN or LAN, through firewalls, and without specifying an IP address. LDAP is only used with TCP/IP, and only on network servers.		
	For more information about LDAP, see "Connecting using an LDAP server" on page 97.		
Server location utility	options		

Log output messages to file (-o) Write output messages to the named file.

Operate quietly (-q) Do not display output messages.

The Service Creation utility

The Service Creation utility is a tool used to create, delete, and modify Adaptive Server Anywhere services.

Managing services using the dbsvc command-line utility

Syntax

dbsvc [options] <*svc*>

dbsvc [-q] [-y] -d <*svc*>

dbsvc [-q] -g <svc>

dbsvc [-q] -l

dbsvc [-q] [-y] < creation options> -w < svc> < details>

Option	Description
- a acct	Account name to use (used with -p)
-as	Use local system account
-d	Delete a service
-i	Allow service to interact with desktop
-g	Get details of a service
_ l	List all Adaptive Server Anywhere services
- p	Password for account (used with -a)
- q	Do not print banner
-rg dependency,	Specify group dependencies when creating a service
-rs dependency,	Specify service dependencies when creating a service
–s startup	Startup option (default manual). You must specify automatic, manual, or disabled
- t type	Type of service
uservice_name	starts the service named service_name
–w executable parameters	Create write service
xservice_name	Stops the service named service_name

	Option	Description		
	y	Delete or overwrite service without confirma- tion		
Description	A service runs a database s This utility provides a com Anywhere services on Win same functionality as the S	erver or other application with a set of options. prehensive way of managing Adaptive Server dows. The Service Creation utility provides the ervice Creation wizard in Sybase Central.		
	You must be a member of tube use the Service Creation ut	You must be a member of the Administrators group on the local machine to use the Service Creation utility.		
	Exit codes are 0 (success)	Exit codes are 0 (success) or non-zero (failure).		
	For more information services" on page 23.	about services, see "Understanding Windows		
	This utility accepts @filena "@filename server option"	ame parameters. For more information, see on page 131.		
Examples	Create a personal server se engine with the specified p user:	rvice called myserv , which starts the specified arameters. The engine runs as the LocalSystem		
	dbsvc -as -w myserv win32\dbeng9 Sybase\SQL #	"C:\Program Files\Sybase\SQL Anywhere 9\ 9.exe" -n myeng -c 8m "C:\Program Files\ Anywhere 9\sample.db"		
	Create a network server ser under the local account, an booted:	rvice called mynetworkserv . The server runs d starts automatically when the machine is		
	dbsvc -as -s auto -t n Sybase\SQL Any "C:\Program F	etwork -w mynetworkserv "C:\Program Files\ /where 9\win32\dbsrv9.exe" -x tcpip -c 8m iles\Sybase\SQL Anywhere 9\sample.db"		
	List all details about servic	e myserv :		
	dbsvc -g myserv			
	Delete the service called m	yserv, without prompting for confirmation:		
	dbsvc -y -d myserv			
	Create a service dependent	on the Workstation service and the TDI group:		
	dbsvc -rs Workstation	n -rg TDI -w		
	Create a service called Ada	aptive Server Anywhere - mysyncservice:		

```
dbsvc -as -s manual -t dbmlsync -w mysyncservice "C:\Program
Files\Sybase\SQL Anywhere 9\win32\dbmlsync.exe" -c
"dsn=ultralite 9.0 sample"
```

Service creation utility options

Account name (-a) All services run under a Windows account. If you run under an account you've created, you must name the account with the -a option and supply a password with the -p option.

Use local system account (-as) All services run under a Windows account. Using the -as option, the service will run under a Windows account. No password is required.

Delete a service (-d) Removes the server name from the list of services. If you supply -y, any service is deleted without confirmation.

Allow service to interact with desktop (-i) Displays an icon that you can double-click to display the server window.

Get details of a service (-g) Lists the definition of the service, not including the password.

List all Adaptive Server Anywhere services (–I) Lists the available Adaptive Server Anywhere services.

Password for account (-p) Use this option with the -a option to specify the password for the account the service runs under.

Do not print banner (-q) Suppress the informational banner. The -q option can be used with any of the -d, -g, -1, or -w options.

Set group dependencies (-rg) At least one service from each of the groups in the list must be started before the service being created is allowed to start.

Set service dependencies (–rs) All the services in the list must have started before the service being created is allowed to start.

Startup option (–s) Sets startup behavior for Adaptive Server Anywhere services. You can set startup behavior to Automatic, Manual, or Disabled.

Type of service (-t type) Specifies the executable to run for this service. You can choose from the following types:

Туре	Description
Network	Adaptive Server Anywhere network database server (<i>db-srv9</i>)
Standalone	Adaptive Server Anywhere personal database server (<i>dbeng9</i>)
DBRemote	SQL Remote Message Agent (dbremote)
MobiLink	MobiLink synchronization server (dbmlsrv9)
DBMLSync	MobiLink synchronization client (<i>dbmlsync</i>)

The default setting is **Standalone**. If creating a service, you must specify options for the appropriate executable along with the type.

Start a service (-u service_name) Starts a service named service_name.

Create service (-w) Creates a new service, or overwrites one if one of the same name exists. If you supply -y, any existing service is overwritten without confirmation.

You must supply the full path to the executable that you wish to use as a service, as the account under which the service is running may not have the appropriate SQL Anywhere directory in its path.

You must supply parameters appropriate for the service you are creating. For more information, see the following locations:

- dbsrv9 and dbeng9 "The database server" on page 124.
- dbmlsrv9 "MobiLink Synchronization Server Options" [MobiLink Synchronization Reference, page 3].
- ♦ dbremote "The Message Agent" [SQL Remote User's Guide, page 292].

Stop a service (-x service_name) Stops a service named service_name.

Delete or overwrite service without confirmation (–y) Automatically carries out the action without prompting for confirmation. This option can be used with the -w or -d options.

The Spawn utility

This utility is provided to start a database server in the background.

Running a server in the background using the dbspawn command-line utility

Syntax	dbspawn [-q] serve	dbspawn [-q] server command		
	Option	Description		
	f	Do not check for a running server		
	- p	Report operating system process ID		
	- q	Quiet mode—do not print messages		
Description	scription The <i>dbspawn</i> utility is provided to start a server in the starts the server in the background and returns with ar (success) or non-zero (failure). If the server specified already running, <i>dbspawn</i> reports failure.			
	The <i>dbspawn</i> utility especially when subs is accepting requests	The <i>dbspawn</i> utility is useful for starting a server from a batch file, especially when subsequent commands in the batch file require a server that is accepting requests.		
	If the specified path includes at least one space, and if you are executing the spawn utility from a batch file, you must enclose the path in one set of double quotes. For example,			
	dbspawn dbeng9 "C:\Program Files\Sybase\SQL Anywhere 9\ asademo.db"			
	If the specified path includes at least one space, and if you are executing the spawn utility at the command prompt, you must provide an extra set of quotes, and escape the quotes around the database file. For example,			
	dbspawn dbeng9 "\"C:\Program Files\Sybase\SQL Anywhere 9\ asademo.db\""			
	If the specified path	does not contain spaces, then quotes are not required.		
	For a description of the server commands, see "The database server" on page 124.			
	This utility accepts @ "@filename server o	@filename parameters. For more information, see ption" on page 131.		

Spawn utility options

Do not check for a running server (-f) If a database server is already running, the *dbspawn* command may use it. This option forces *dbspawn* to start a new database server each time it is run.

Report process ID (–p) The operating system process ID of the database server process. For example:

```
dbspawn -p dbeng9 -n newserver
```

reports a message of the following form to the command prompt:

New process id is 306

Operate quietly (-q) Do not display output messages. This option is available only when you run this utility from the command prompt.
The Stop utility

The Stop utility stops a database server. You can use the -d option to stop a specified database.

The Stop utility can only be run at the command prompt. In windowed environments, you can stop a database server by clicking Shutdown on the server window.

The stop [dbstop] utility is not available on NetWare.

Stopping a database server using the dbstop command-line utility

Syntax	dbstop [options]	
	Option	Description
	server-name	Name of a local database server to stop.
	-c "keyword=value;"	Connection parameters
	-d	Stop specified database only.
	–o filename	Log output messages to a file
	- q	Quiet mode-do not print messages
	-X	Do not stop if there are active connections
	- y	Stop without prompting even if there are active connections
Description	Options let you control whether a server is stopped even if there are active connections, and whether to stop a server or only a database.	
	Exit codes are 0 (success) or non-zero (failure).	
	For more information about the Stop utility options, see "Stop utility options" on page 525.	
	This utility accepts @filename parameters. For more information, see "@filename server option" on page 131.	
Stop utility options		
	Server name The name of a database server running on the current machine. The database server must be started so that no permissions are required to shut it down. The personal database server starts in this mode b default. For the network database server, you must supply the -gk all option.	

If you supply a server name, do not supply connection parameters as well.

For more information, see "-gk server option" on page 146.

Connection parameters (-c) When stopping a network server, you must supply a connection string with a user ID that has permissions to stop the server. By default, DBA permission is required on the network server, and all users can shut down a personal server, but the -gk server option can be used to change this.

The behavior of *dbstop* can be controlled if there are active connections on a server. If there are active connections, *dbstop* provides a prompt asking if you wish to shut down the server. If you specify **unconditional=true**, the server is shut down without prompting, even if there are active connections.

If you supply connection parameters, do not supply a server name as well.

For more information, see "Connection parameters" on page 70, "Unconditional connection parameter [UNC]" on page 199, and "-gk server option" on page 146.

Stop database only (-d) Do not stop the database server. Instead, only stop the database specified in the connection string.

Log output messages to file (-o) Write output messages to the named file.

Operate quietly (–q) Do not print a message if the database was not running.

Do not stop if there are active connections (-x) Do not stop the server if there are still active connections to the server.

Stop without prompting (–y) Stop the server even if there are still active connections to the server.

Example 1 You are running the server named **myserver** without a database. To stop the server, specify the utility database as a **DatabaseName** (**DBN**) connection parameter:

dbstop -c "uid=DBA;pwd=SQL;eng=myserver;dbn=utility_db"

Example 2 You are running the server named **myserver** with the database **asademo.db** started. To stop the server and database:

dbstop -c "uid=DBA;pwd=SQL;eng=myserver "

The Transaction Log utility

With the Transaction Log utility, you can display or change the name of the transaction log or transaction log mirror associated with a database. You can also stop a database from maintaining a transaction log or mirror, or start maintaining a transaction log or mirror.

A transaction log mirror is a duplicate copy of a transaction log, maintained by the database in tandem.

The name of the transaction log is first set when the database is initialized. The Transaction Log utility works with database files or with write files. The database server must not be running on that database when the transaction log filename is changed (or an error message appears).

You can access the Transaction Log utility in the following ways:

- From Sybase Central, using the Change Log File Settings wizard.
- At the command prompt, using the *dblog* command.

Managing log files using the Change Log File Settings wizard

To change a transaction log file name

- 1. In the left pane, select the Adaptive Server Anywhere 9 plug-in.
- 2. In the right pane, click the Utilities tab.
- 3. In the right pane, double-click Change Log File Settings. The Change Log File Settings wizard appears.
- 4. Follow the instructions in the wizard.

Тір

You can also access the Change Log File Settings wizard by choosing Tools ➤ Adaptive Server Anywhere 9 ➤ Change Log File Settings.

Managing log files using the dblog command-line utility

Syntax

dblog [options] database-file

Option	Description
–ek key	Specify encryption key
-ер	Prompt for encryption key

Option	Description
- g n	Sets the LTM generation number to <i>n</i>
–il	Ignores the LTM truncation offset stored in the database
_ir	Ignores the SQL Remote truncation offset stored in the database
– m mirror-name	Set transaction log mirror name
-n	No longer use a transaction log or mirror log
–o filename	Log output messages to a file
- q	Quiet mode-do not print messages
-r	No longer use a transaction log mirror
-t log-name	Set the transaction log name
- x n	Set the transaction log current relative offset to n
- z n	Set the transaction log starting offset to <i>n</i>

Description

The dblog utility allows you to display or change the name of the transaction log or transaction log mirror associated with a database. You can also stop a database from maintaining a transaction log or mirror, or start maintaining a transaction log or mirror.

The utility displays additional information about the transaction log, including the following:

- Version number
- Starting offset, for use in replication
- Ending offset, for use in replication
- Page size
- Total number of pages
- Number of empty pages
- Percentage of the log file in use

Exit codes are 0 (success) or non-zero (failure).

For more information about the Transaction log utility options, see "Transaction log utility options" on page 529.

This utility accepts @filename parameters. For more information, see "@filename server option" on page 131.

Transaction log utility options

Specify encryption key (-ek) This option allows you to specify the encryption key for strongly encrypted databases directly in the command. If you have a strongly encrypted database, you must provide the encryption key to use the database or transaction log in any way. For strongly encrypted databases, you must specify either -ek or -ep, but not both. The command will fail if you do not specify a key for a strongly encrypted database.

Prompt for encryption key (-ep) This option allows you to specify that you want to be prompted for the encryption key. This option causes a dialog box to appear, in which you enter the encryption key. It provides an extra measure of security by never allowing the encryption key to be seen in clear text. For strongly encrypted databases, you must specify either -ek or -ep, but not both. The command will fail if you do not specify a key for a strongly encrypted database.

Set the generation number (-g) Use this option if you are using the Log Transfer Manager to participate in a Replication Server installation. It can be used after a backup is restored, to set the generation number. It performs the same function as the following Replication Server function:

dbcc settrunc('ltm', 'gen_id', n)

For information on generation numbers and *dbcc*, see your Replication Server documentation.

Ignore the LTM truncation offset (-il) Use this option if you are using the Log Transfer Manager to participate in a Replication Server installation. It performs the same function as the following Replication Server function:

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

For information on *dbcc*, see your Replication Server documentation.

Ignore the SQL Remote truncation offset (-ir) Use this option if you are using the Log Transfer Manager to participate in a Replication Server installation and a SQL Remote installation. It resets the offset that is kept for the DELETE_OLD_LOGS option, allowing transaction logs to be deleted when they are no longer needed.

Set the name of the transaction log mirror file (–m) This option sets a filename for a new transaction log mirror. If the database is not currently using a transaction log mirror, it starts using one. If the database is already using a transaction log mirror, it changes to using the new file as its

transaction log mirror.

No longer use a transaction log (–n) Stop using a transaction log, and stop using a mirror log. Without a transaction log, the database can no longer participate in data replication or use the transaction log in data recovery.

Log output messages to file (-o) Write output messages to the named file.

Operate quietly (-q) Do not display output messages. This option is available only when you run this utility from the command prompt.

No longer use a transaction log mirror (-r) For databases that maintain a mirrored transaction log, this option changes their behavior to maintain only a single transaction log.

Set the name of the transaction log file (-t) This option sets a filename for a new transaction log. If the database is not currently using a transaction log, it starts using one. If the database is already using a transaction log, it changes to using the new file as its transaction log.

Set the current log offset (–x) For use when reloading a SQL Remote consolidated database. This option resets the current log offset so that the database can take part in replication.

For information on how to use this option, see "Unloading and reloading a database participating in replication" [*SQL Remote User's Guide*, page 258].

Set the starting log offset (-z) For use when reloading a SQL Remote consolidated database. This option resets the starting log offset so that the database can take part in replication.

For information on how to use this option, see "Unloading and reloading a database participating in replication" [*SQL Remote User's Guide*, page 258].

The Uncompression utility

With the Uncompression utility, you can expand a compressed database file created by the Compression utility. The Uncompression utility reads the compressed file and restores the database file to its uncompressed state.

You can access the Uncompression utility in the following ways:

- From Sybase Central, using the Uncompress Database wizard.
- At the command prompt, using the *dbexpand* command. This is useful for incorporating into batch or command files.

Uncompressing a database using the Uncompress Database wizard

To uncompress a compressed database file

- 1. In the left pane, select the Adaptive Server Anywhere 9 plug-in.
- 2. In the right pane, click the Utilities tab.
- 3. In the right pane, double-click Uncompress Database.

The Uncompress Database wizard appears.

4. Follow the instructions in the wizard.

Тір

Syntax

Description

You can also access the Uncompress Database wizard by choosing Tools \succ Adaptive Server Anywhere 9 \succ Uncompress Database.

Uncompressing a database using the dbexpand command-line utility

dbexpand [options] compressed-database-file [database-file]

Option	Description
– ek key	Specify encryption key
—ер	Prompt for encryption key
–o filename	Log output messages to a file
- q	Operate quietly-do not print messages
$-\mathbf{y}$	Erase an existing output file without confirmation
The input filename extensi	ion defaults to <i>cdb</i> . The output filename (with

extension) must not have the same name as the input filename (with

extension).

Exit codes are 0 (success) or non-zero (failure).

For more information about the Uncompression utility options, see "Uncompression utility options" on page 532.

This utility accepts @filename parameters. For more information, see "@filename server option" on page 131.

Uncompression utility options

Specify encryption key (-ek) This option allows you to specify the encryption key for strongly encrypted databases directly in the command. If you have a strongly encrypted database, you must provide the encryption key to use the database or transaction log in any way. For strongly encrypted databases, you must specify either -ek or -ep, but not both. The command will fail if you do not specify a key for a strongly encrypted database.

Prompt for encryption key (-ep) This option allows you to specify that you want to be prompted for the encryption key. This option causes a dialog box to appear, in which you enter the encryption key. It provides an extra measure of security by never allowing the encryption key to be seen in clear text. For strongly encrypted databases, you must specify either -ek or -ep, but not both. The command will fail if you do not specify a key for a strongly encrypted database.

Log output messages to file (-o) Write output messages to the named file.

Operate quietly (-q) Do not display output messages. This option is available only when you run this utility from the command prompt.

Operate without confirming actions (-y) Choosing this option automatically replaces an existing database file without prompting you for confirmation.

The Unload utility

	With the Unload utility, you can unload a database and put a set of data files in a named directory. The Unload utility creates an Interactive SQL command file to rebuild your database. It also unloads all of the data in each of your tables, into files in the specified directory in comma-delimited format. Binary data is properly represented with escape sequences.
	You can also use the Unload utility to directly create a new database from an existing one. This avoids potential security problems with the database contents being written to ordinary disk files.
Accessing the Unload	You can access the Unload utility in the following ways:
utility	• From Sybase Central, using the Unload Database wizard.
	• At the command prompt, using the <i>dbunload</i> command. This is useful for incorporating into batch or command files.
	The Unload utility should be run from a user ID with DBA authority. This is the only way you can be sure of having the necessary privileges to unload all the data. In addition, the <i>reload.sql</i> file should be run from the DBA user ID. (Usually, it will be run on a new database where the only user ID is DBA with password SQL .)
	The database server -gl option controls the permissions required to unload data from the database. For information, see "-gl server option" on page 147.
Objects owned by dbo	The dbo user ID owns a set of Adaptive Server Enterprise-compatible system objects in a database.
	The Unload utility does not unload the objects that were created for the dbo user ID during database creation. Changes made to these objects, such as redefining a system procedure, are lost when the data is unloaded. Any objects that were created by the dbo user ID since the initialization of the database are unloaded by the Unload utility, and so these objects are preserved.
Unloading and replication	There are special considerations for unloading databases involved in replication.
	For information, see "Unloading and reloading a database participating in replication" [<i>SQL Remote User's Guide</i> , page 258].

Unloading a database using the Unload Database wizard utility

To unload a database file or a running database

- 1. In the left pane, select the Adaptive Server Anywhere 9 plug-in.
- 2. In the right pane, click the utilities tab.
- 3. In the right pane, double-click Unload Database.

The Unload Database wizard appears.

- 4. Follow the instructions in the wizard.
 - Right-clicking the database and then clicking Unload Database from the popup menu

Тір

You can also access the Unload Database wizard from within Sybase Central using any of the following methods:

choosing Tools \succ Adaptive Server Anywhere 9 \succ Unload Database.

Selecting a database in the left pane, and choosing Unload Database from the File menu.

Right-clicking a database, and choosing Unload Database from the popup menu.

For full information on unloading a database from Sybase Central, see "Exporting databases" [*ASA SQL User's Guide*, page 536].

Unloading a database using the dbunload command-line utility

Syntax

dbunload [options] [directory]

Option	Description
-ac "keyword=value;	Supply connection parameters for the reload
– an database	Creates a database file with the same settings as the database being unloaded, and automatically reloads it
-ar [directory]	Rebuild and replace database
−c "keyword=value;"	Supply database connection parameters for unload
-d	Unload data only
–e <i>table</i> ,	Do not unload listed tables

Option	Description
-ea algorithm	Specify which strong encryption algorithm to encrypt your database with: you can choose AES or MDSR
– ek key	Specify encryption key
-ер	Prompt for encryption key
–ii	Internal unload, internal reload (default)
-ix	Internal unload, external reload
–j nnn	Repeated unload of view creation statements
–jr	Ignore Java when unloading or reloading
-m	Do not preserve user ids for a replicating database
-n	No data—schema definition only
–o filename	Log output messages to a file
– p char	Escape character for external unloads (default "\")
- q	Quiet mode-no windows or messages
– r reload-file	Specify name and directory of generated reload Interactive SQL command file (default <i>reloadsql</i>)
t <i>table</i> ,	Unload only the listed tables
-u	Unordered data. Do not use indexes to unload data.
- v	Verbose messages
-xi	External unload, internal reload
XX	External unload, external reload
- y	Replace the command file without confirmation

Description The *directory* is the destination directory where the unloaded data is to be placed. The *reload.sql* command file is always relative to the current directory of the user.

In the default mode, or if -ii or -ix is used, the directory used by *dbunload* to hold the data is relative to the database server, not to the current directory

of the user.

For details of how to supply a filename and path in this mode, see "UNLOAD TABLE statement" [ASA SQL Reference, page 590].

If the -xi or -xx is used, the directory is relative to the current directory of the user.

If no list of tables is supplied, the whole database is unloaded. If a list of tables is supplied, only those tables are unloaded.

Unloaded data includes the column list for the LOAD TABLE statements generated in the reload.sql file. Unloading the column list facilitates easier reordering of the columns in a table. Tables can be dropped or recreated, and then repopulated using reload.sql.

If you want to unload a strongly encrypted database, you will be required to provide the encryption KEY. You can use the **EncryptionKey** (**DBKEY**) connection parameter to provide the KEY in the command. Alternatively, if you want to be prompted for the encryption key rather than entering it in plain view, you can use the -ep option as follows:

```
dbunload -c "dbf=enc.db;start=dbeng9 -ep"
```

If you are using dbunload -an to unload a database and reload into a new one, and you want to use the -ek or -ep options to set the encryption password for the new database, keep the following in mind:

- ◆ If the original database is strongly encrypted, you will need to specify the key for the original database using the EncryptionKey (DBKEY) connection parameter in the -c option, not using -ek or -ep.
- Using the -ek and -ep options, it is possible to unload an unencrypted database and reload into a new, strongly encrypted database. When you use -ep and -an, you must confirm the key correctly or the unload fails.
- ♦ If the original database is strongly encrypted, but the -ek and -ep options are not used, then the new database will be encrypted without requiring a key.
- The -ek and -ep options are ignored if -an is not specified.

Exit codes are 0 (success) or non-zero (failure).

For more information about the Unload utility options, see "Unload utility options" on page 537.

There are special considerations for unloading databases involved in replication. For information, see "Unloading and reloading a database participating in replication" [*SQL Remote User's Guide*, page 258].

For more information about encryption, see "-ep server option" on page 143 or the "Encryption Key connection parameter [DBKEY]" on page 190.

This utility accepts @filename parameters. For more information, see "@filename server option" on page 131.

Unload utility options

Connection parameters for reload database (–ac) This option causes the Unload utility to connect to an existing database and reload the data directly into it. You can combine the operation of unloading a database and reloading the results into an existing database using this option.

Typically, you would create a new database using the Initialization utility, and then reload it using this option. This method is useful when you want to change the initialization options, such as page size, or collation. If you are changing collations, the -xx switch should also be used, to ensure that character set translation occurs properly for both the old and new databases.

For example, the following command (which should be entered all on one line) loads a copy of the *asademo.db* database into an existing database file named *newdemo.db*:

dbunload -c "uid=DBA;pwd=SQL;dbf=asademo.db" -ac "uid=DBA;pwd=SQL;dbf=newdemo.db"

If you use this option, no interim copy of the data is created on disk, so you do not specify an unload directory in the command. This provides greater security for your data.

You do not need to specify a directory for this option.

Create a database for reloading (–an) You can combine the operations of unloading a database, creating a new database, and loading the data using this option. This option applies to personal server connections, and network server connections over shared memory.

Typically, you would use this option when you do not want to change the initialization options of your database. The options specified when you created the source database are used to create the new database.

For example, the following command (which should be entered all on one line) creates a new database file named *asacopy.db* and copies the schema and data of *asademo.db* into it:

```
dbunload -c "uid=DBA;pwd=SQL;dbf=asademo.db" -an asacopy.db
```

If you use this option, no interim copy of the data is created on disk, so you

do not specify an unload directory in the command. This provides greater security for your data, but at some cost for performance.

You do not need to specify a directory for this option

Rebuild and replace database (–ar) This option creates a new database with the same settings as the old database, reloads it, and replaces the old database. If you use this option, there can be no other connections to the database, and the database connection must be local, not over a network.

If you specify an optional *directory*, the transaction log offsets are reset for replication purposes, and the transaction log from the old database is moved to the specified directory. The named directory should be the directory that holds the old transaction logs used by the Message Agent and the Replication Agent. The transaction log management is handled only if the database is used in replication: if there is no SQL Remote publisher or LTM check, then the old transaction log is not needed and is deleted instead of being copied to the specified directory.

For more information on transaction log management, see "Backup methods for remote databases in replication installations" on page 352.

Connection parameters for source database (-c) For a description of the connection parameters, see "Connection parameters" on page 70. The user ID should have DBA authority, to ensure that the user has permissions on all the tables in the database.

For example, the following statement unloads the **asademo** database running on the **sample_server** server, connecting as user ID **DBA** with password **SQL**. The data is unloaded into the *c*:\unload directory.

dbunload -c "eng=sample_server;dbn=asademo;uid=DBA;pwd=SQL" c:\ unload

Unload data only (–d) With this option, none of the database definition commands are generated (CREATE TABLE, CREATE INDEX, and so on); *reload.sql* contains statements to reload the data only.

No data output for listed tables (–e) This option is accessible only when you run this utility at the command prompt. If you wish to unload almost all of the tables in the database, the -e option unloads all tables except the specified tables. A *reload.sql* file created with the -e option should not be used to rebuild a database because the file will not include all the database tables.

Specify encryption algorithm (-ea) This option allows you to choose which strong encryption algorithm to encrypt your new database with. You can choose either AES or MDSR. Algorithm names are case insensitive. If

you specify the -ea option, you must also specify -ep or -ek. If you specify -ea without specifying -an, the -ea option is ignored.

Specify encryption key (-ek) This option allows you to specify an encryption key for the new database created if you unload and reload a database (using the -an option). If you create a strongly encrypted database, you must provide the encryption key to use the database or transaction log in any way. If you specify the -ek option without specifying an algorithm (using the -ea option), the algorithm used will be AES. If you specify -ek without specifying -an, the -ek option is ignored.

Prompt for encryption key (-ep) This option prompts you to specify an encryption key for the new database created if you unload and reload your database (using the -an option). It provides an extra measure of security by never allowing the encryption key to be seen in clear text. If you specify the -ep option without specifying an algorithm (-ea option), the algorithm used will be AES. If you specify -ep without specifying -an, the -ep option is ignored. If you specify -ep and -an, you must input the encryption key twice to confirm that it was entered correctly. If the keys don't match, the unload fails.

Internal versus external unloads and reloads

The following options offer combinations of internal and external unloads and reloads: -ii, -ix, -xi and -xx. A significant performance gain can be realized using internal commands (UNLOAD/LOAD) versus external commands (Interactive SQL's INPUT and OUTPUT statement). However, internal commands are executed by the server so that file and directory paths are relative to the location of the database server. Using external commands, file and directory paths are relative to the current directory of the user.

In Sybase Central, you can specify whether to unload relative to the server or client.

For more information on filenames and paths for the Unload utility, see "UNLOAD TABLE statement" [ASA SQL Reference, page 590].

Use internal unload, internal reload (-ii) This option uses the UNLOAD statement to extract data from the database, and uses the LOAD statement in the *reload.sql* file to repopulate the database with data. This is the default.

Use internal unload, external reload (-ix) This option uses the UNLOAD statement to extract data from the database, and uses the Interactive SQL INPUT statement in the *reload.sql* file to repopulate the database with data.

Repeated unload of view creation statements (–j) If your database contains view definitions that are dependent on each other, you can use this

option to avoid failure when reloading the views into a database. This option causes view creation statements to be unloaded multiple times, as specified by the count entered. This count should be small, and should correspond to the number of levels of view dependency.

Unload schema definition only (–n) With this option, none of the data in the database is unloaded; *reload.sql* contains SQL statements to build the structure of the database only.

Log output messages to file (–o) Write output messages to the named file.

Escape character (–p) The default escape character (\) for external unloads (dbunload -x option) can be replaced by another character, using this option. This option is available only when you run this utility from the command prompt.

Operate quietly (-q) Do not display output messages. This option is available only when you run this utility from the command prompt.

Specify reload filename (–r) Modify the name and directory of the generated reload Interactive SQL command file. The default is *reload.sql* in the current directory. The directory is relative to the current directory of the client application, not the server.

Unload only listed tables (-t) Provide a list of tables to be unloaded. By default, all tables are unloaded. Together with the -n option, this allows you to unload a set of table definitions only.

Output unordered data (–u) Normally, the data in each table is ordered by the primary key. Use this option if you are unloading a database with a corrupt index, so that the corrupt index is not used to order the data.

Enable verbose mode (–v) The table name of the table currently being unloaded, and how many rows have been unloaded, appears. This option is available only when you run this utility from the command prompt.

Use external unloading, internal reload (-xi) This option unloads data to the dbunload client, and uses the LOAD statement in the generated reload command file, *reload.sql*, to repopulate the database with data.

Use external unloading, external reload (–xx) This option unloads data to the dbunload client, and uses the InteractiveăSQL INPUT statement in the generated reload command file, *reload.sql*, to repopulate the database with data.

Operate without confirming actions (–y) Choosing this option replaces existing command files without prompting you for confirmation.

Rebuilding a database To unload a database, start the database server with your database, and run the Unload utility with the DBA user ID and password.

To reload a database, create a new database and then run the generated *reload.sql* command file through Interactive SQL.

In Windows 95/98/Me and NT/2000/XP, and UNIX, there is a file (*rebuild.bat*, *rebuild.cmd*, or *rebuild*) that automates the unload and reload process.

The Upgrade utility

While you can run a database created with an older version of Adaptive Server Anywhere on a newer release of Adaptive Server Anywhere, the new features are only available if the database is upgraded.

The Upgrade utility upgrades a database from an older version of Adaptive Server Anywhere to a newer version of Adaptive Server Anywhere, and allows you to take advantage of the newer release's full list of features.

Upgrading a database does not require you to unload and reload your database.

If you wish to use replication on an upgraded database, you must also archive your transaction log and start a new one on the upgraded database.

You can access the Upgrade utility in the following ways:

- From Sybase Central, using the Upgrade Database wizard.
- At the command prompt, using the *dbupgrad* command.

Back up before upgrading

As with any software, it is recommended that you make a backup of your database before upgrading.

Upgrading a database using the Upgrade Database wizard

To upgrade a database

- 1. Connect to the database.
- 2. In the left pane, select the Adaptive Server Anywhere 9 plug-in.
- 3. In the right pane, click the Utilities tab.
- 4. In the right pane, double-click Upgrade Database. The Upgrade Database wizard appears.
- 5. Follow the instructions in the wizard.

Тір

You can also access the Upgrade Database wizard by:

Choosing Tools \succ Adaptive Server Anywhere 9 \succ Upgrade Database.

Right-clicking the database, and choosing Upgrade Database from the popup menu.

Selecting a database in the left pane, and choosing Upgrade Database from the File menu.

Upgrading databases that are too old for the Upgrade utility

To upgrade a database created with a version of Adaptive Server Anywhere that is too old to be upgraded

- 1. Unload the database using the Unload utility.
- 2. Create a database with the name you wish to use for the upgraded version using the Initialization utility.
- 3. Connect to the new database from Interactive SQL as the DBA user ID, and read the *reload.sql* command file to build the new database.

You can also use the *dbunload* utility to directly rebuild.

Upgrading a database using the dbupgrad command-line utility

Syntax

dbupgrad [options]

Option	Description
-c "keyword=value; "	Supply database connection parameters
-I	Do not install Sybase jConnect support
—j	Do not install runtime Java classes
–ja	Add runtime Java classes
-jdk version	Install entries for the named version of the Java Development Kit
–jr	Remove runtime Java classes
–o filename	Log output messages to a file
- q	Quiet mode—no windows or messages

Description	The <i>dbupgrad</i> utility upgrades a database created with earlier versions of the software to enable features from the current version of the software. The earliest version that can be upgraded is Watcom SQL 3.2. While later versions of the database server do run against databases were created with earlier releases of the software, some of the features introduced since the version that created the database are unavailable unless the database is upgraded.
	Exit codes are 0 (success) or non-zero (failure).
	For more information about the Upgrade utility options, see "Upgrade utility options" on page 544.
	This utility accepts @filename parameters. For more information, see "@filename server option" on page 131.
	Not all features made available Features that require a physical reorganization of the database file are not made available by <i>dbupgrad</i> . Such features include index enhancements and changes in data storage. To obtain the benefits of these enhancements, you must unload and reload your database.
	New in SQL Anywhere Studio, page 178].
Java in the database	By default, Java in the database is not included in new databases. Java in the database is removed from the database during an upgrade if:
	 the database is Java-enabled, but Java in the database is not being used, and
	• the Java VM cannot be loaded by the database server, and
	• you did not specify any Java-related options to the upgrade.
Upgrade utility option	ons
	Connection parameters (–c) For a description of the connection parameters, see "Connection parameters" on page 70. The user ID must have DBA authority.

For example, the following command upgrades a database called sample40 to a version 9.0 format, connecting as user **DBA** with password **SQL**:

dbupgrad -c "uid=DBA;pwd=SQL;dbf=c:\wsql40\sample40.db"

The dbupgrad utility must be run by a user with DBA authority.

Do not install Sybase jConnect support (-i) If you wish to use the

Sybase jConnect JDBC driver to access system catalog information, you need to install jConnect support. Use this option if you wish to exclude the jConnect system objects. You can still use JDBC, as long as you do not access system information. If you want, you can add Sybase jConnect support later using Sybase Central.

For more information, see "Installing jConnect system objects into a database" [*ASA Programming Guide*, page 111].

Do not install runtime Java classes (–j) If you do not intend to use Java classes, you can specify either no Java option, or the –j option to avoid including these classes in the database.

The runtime classes add several megabytes to the size of a database. If you do not intend to use Java classes, you can specify the –j option to avoid including these classes in the upgraded database. If you want, you can add Sybase runtime Java classes later using Sybase Central or ALTER DATABASE.

For more information, see "Java-enabling a database" [ASA Programming Guide, page 84].

Install runtime Java classes (-ja) Add Java in the database to the upgraded database. Specifying -ja installs version 1.3 of the JDK into the database.

Java in the database is a separately-licensable component. These classes are installed during upgrade only if you have installed the Java-in-the-database option or if the Java classes were in use in the database being upgraded.

The runtime classes add several megabytes to the size of a database, so if you do not intend to use Java classes, you can omit the -ja option to avoid installing them.

You can add Sybase runtime Java classes later using the Upgrade Wizard, or the ALTER DATABASE statement.

For more information, see "Java-enabling a database" [ASA Programming Guide, page 84].

Remove runtime Java classes (–jr) Remove Java in the database from the upgraded database.

Install support for the named version of the JDK (–jdk) If you want to install a version of Java, use the -jdk option followed by the version number. Valid values are 1.1.8, and 1.3.

For example, the following command creates a database that supports JDK 1.1.8 applications in the database.

```
dbupgrad -jdk 1.1.8 java2.db
```

The -jdk option implies the -ja option.

The runtime classes add several megabytes to the size of a database, so if you do not intend to use Java classes, you can omit the -jdk option to avoid installing them.

You can add Sybase runtime Java classes later using the Upgrade Wizard, or the ALTER DATABASE statement.

For more information, see "Java-enabling a database" [ASA Programming Guide, page 84].

Log output messages to file (-o) Write output messages to the named file.

Operate quietly (-q) Do not display output messages. This option is available only when you run this utility from the command prompt.

The Validation utility

With the Validation utility, you can validate the indexes and keys on some or all of the tables in the database. The Validation utility scans the entire table, and looks up each record in every index and key defined on the table.

This utility can be used in combination with regular backups (see "Backup and Data Recovery" on page 337) to give you confidence in the integrity of the data in your database.

You can access the Validation utility in the following ways:

- From Sybase Central, using the Validate Database wizard
- At the command prompt, using the *dbvalid* command. This is useful for incorporating into batch or command files.

For more information on validating tables, see "VALIDATE TABLE statement" [*ASA SQL Reference*, page 603].

Validating a database using the Validate Database wizard

You can validate a database from Sybase Central, and validate individual tables.

To validate a database

- 1. In the left pane, select the Adaptive Server Anywhere 9 plug-in.
- 2. In the right pane, click the Utilities tab.
- In the right pane, double-click Validate Database. The Validate Database wizard appears.
- 4. Follow the instructions in the wizard.

To validate a running database

- 1. Connect to the database.
- 2. Right-click the database and choose Validate Database from the popup menu.

The Validate Database wizard appears.

3. Follow the instructions in the wizard.

* To validate an individual table

- 1. Connect to the database.
- 2. Locate the table you wish to validate.
- 3. Right-click the table and choose Validate from the popup menu.

Тір

You can also access the Validate Database wizard from within Sybase Central using any of the following methods:

choosing Tools \blacktriangleright Adaptive Server Anywhere $9 \succ$ Validate Database.

Right-clicking the database, and choosing Validate Database from the popup menu.

Selecting a database in the left pane, and choosing Validate Database from the File menu.

Validating a database using the dbvalid command-line utility

Syntax

dbvalid [options] [object-name,...]

Option	Description
object-name	The name of a table or (if -i is used) an index to validate
-c "keyword=value;"	Supply database connection parameters
–o filename	Log output messages to a file
-f	Validate tables with full check
-fd	Validate tables with data check
-fi	Validate tables with index check
-fx	Validate tables with express check
-I	Each object-name is an index
- q	Quiet mode-do not print messages
-t	Each object-name is a table

Description

With the Validation utility, you can validate the indexes and keys on some or all of the tables in the database. This utility scans the entire table, and confirms that each row in the table exists in the appropriate indexes. It is equivalent to running the VALIDATE TABLE statement on each table.

i.

Exit codes are:

Program Exit Code	Description
0	database validated successfully
1	general failure in utility
2	error validating database
7	unable to find database to connect to (database name is wrong)
8	unable to connect to database (user ID/password is wrong
11	unable to find server to connect to (server name is wrong)
12	incorrect encryption key for starting database

For information on specific checks made during validation, see "VALIDATE TABLE statement" [*ASA SQL Reference*, page 603].

For more information about the Validation utility options, see "Validation utility options" on page 549.

This utility accepts @filename parameters. For more information, see "@filename server option" on page 131.

Validation utility options

Connection parameters (-c) For a description of the connection parameters, see "Connection parameters" on page 70. The user ID must have DBA authority or REMOTE DBA authority.

For example, the following validates the sample database, connecting as user **DBA** with password **SQL**:

dbvalid -c "uid=DBA;pwd=SQL;dbf=c:\asa\asademo.db"

Full check for each table (-f) In addition to the default validation checks, carry out both data checks (-fd) and index checks (-fi). This corresponds to the WITH FULL CHECK option on the VALIDATE TABLE statement. Depending on the contents of your database, this option may significantly extend the time required to validate.

Data check for each table (-fd) In addition to the default validation

checks, check that all of each LONG BINARY, LONG VARCHAR, TEXT, or IMAGE data type can be read. This corresponds to the WITH DATA CHECK option on the VALIDATE TABLE statement. Depending on the contents of your database, this option may significantly extend the time required to validate.

Index check for each table (-fi) In addition to the default validation checks, validate each index on the table. This corresponds to the WITH INDEX CHECK option on the VALIDATE TABLE statement. Depending on the contents of your database, this option may significantly extend the time required to validate.

Express check for each table (-fx) In addition to the default and data checks (-fd), check that the number of rows in the table matches the number of entries in the index. This corresponds to the WITH EXPRESS CHECK option on the VALIDATE TABLE STATEMENT. This option does not perform individual index lookups for each row. Using this option can significantly improve performance when validating large databases with a small cache. This option is only available for databases created with version 8.0 or later of Adaptive Server Anywhere.

Validate specified indexes (-i) Instead of validating tables, validate indexes. In this case, for *dbvalid*, each of the *object-name* values supplied represents an index rather than a table, and has a name of the following form:

[[

owner.]table-name.]index-name

Log output messages to file (-o) Write output messages to the named file.

Operate quietly (-q) Do not display output messages. This option is available only when you run this utility from the command prompt.

Validate tables (-t) The list of object-name values represents a list of tables. This is also the default behavior.

The Write File utility

The Write File utility is used to manage database write files. A **write file** is a file associated with a particular database. All changes are written into the write file, leaving the database file unchanged.

Using write files for Write files can be used effectively for testing when you do not wish to modify the production database. They can also be used in network and student environments where read-only access to a database is desired, or when you distribute on CD-ROM a database that you wish users to be able to modify.

Compressed databases If you are using a compressed database, you must use a write file; compressed database files cannot be accessed directly. The write file name is then used in place of the database name when connecting to the database, or when loading a database on the database server command line.

You can access the Write File utility in the following ways:

- From Sybase Central, using the Create Write File wizard.
- At the command prompt, using the *dbwrite* command. This is useful for incorporating into batch or command files.

The Write File utility runs against a database file. The database must not be running on a server when you run the Write File utility.

The database file cannot be modified after the write file is created, otherwise the write file becomes invalid.

Creating a write file using the Create Write File wizard

To create a write file for a database

- 1. In the left pane, select the Adaptive Server Anywhere 9 plug-in.
- 2. In the right pane, click the Utilities tab.
- In the right pane, double-click Create Write File. The Create Write File wizard appears.
- 4. Follow the instructions in the wizard.

Tip

You can also access the Create Write File wizard from within Sybase Central using any of the following methods:

choosing Tools \succ Adaptive Server Anywhere 9 \succ Create Write File.

Right-clicking the server, and choosing Create Write File from the popup menu.

Selecting a server in the left pane, and choosing Create Write File from the File menu.

Creating a write file using the dbwrite command-line utility

Syntax

dbwrite [options] database-file [write-name]

Option	Description
- c	Create a new write file
–d database-file	Point a write file to a different database
– ek key	Specify encryption key
-ep	Prompt for encryption key
– f database-file	Force the write file to point at a file
– m mirror-name	Set the transaction log mirror name
–o filename	Log output messages to a file
- q	Quiet mode-do not print messages
- s	Report the write file status only
-t log-name	Set the transaction log name
- y	Erase old files without confirmation

Description

If any changes are made to the original database (not using the write file), the write file will no longer be valid. This happens if you start the server using the original database file, so you should create your write file from an archived copy of a database.

If a write file becomes invalid, you can discard all of your changes and create a new write file with the following command.

dbwrite -c db-name write-name

The log-name and mirror-name parameters are used only when creating a

new write file. The *write-name* parameter is used only with the -c and -d parameters. Note that the *database_file* parameter must be specified before the *write-name* parameter.

Exit codes are 0 (success) or non-zero (failure).

For more information about the Write file utility options, see "Write file utility options" on page 553.

This utility accepts @filename parameters. For more information, see "@filename server option" on page 131.

Write file utility options

Create a new write file (-c) If an existing write already exists, any information in the old write file will be lost. If no write file name is specified in the command, the write filename defaults to the database name with the extension *wrt*. If no transaction log name is specified, the log file name will default to the database name with the extension *wlg*.

Change the database file to which an existing write file points (-d) If a database file is moved to another directory, or renamed, this option allows you to maintain the link between the write file and the database file. This option is available only when you run this utility from the command prompt.

Specify encryption key (-ek) This option allows you to specify the encryption key for strongly encrypted databases directly in the command. If you have a strongly encrypted database, you must provide the encryption key to use the database or transaction log in any way. For strongly encrypted databases, you must specify either -ek or -ep, but not both. The command will fail if you do not specify a key for a strongly encrypted database.

Prompt for encryption key (–ep) This option allows you to specify that you want to be prompted for the encryption key. This option causes a dialog box to appear, in which you enter the encryption key. It provides an extra measure of security by never allowing the encryption key to be seen in clear text. For strongly encrypted databases, you must specify either –ek or –ep, but not both. The command will fail if you do not specify a key for a strongly encrypted database.

Force a write file to point to a file (-f) This option is available only when you run this utility from the command prompt. This option is for use when a write file is being created and the database file is held on a Novell NetWare or other network path, for operating systems on which they cannot be entered directly. By providing the full Novell path name for the database file (for example: *SYS:\asademo.db*), you can avoid dependencies on local mappings of the NetWare path. Unlike with the -d option, no checking is

done on the specified path.

Set mirror name (-m) Set the name of the transaction log mirror file. This can only be used in conjunction with-c.

Log output messages to file (-o) Write output messages to the named file.

Operate quietly (-q) Do not display output messages. This option is available only when you run this utility from the command prompt.

Report the write file status only (-s) This displays the name of the database to which the write file points. This option is available only when you run this utility from the command prompt.

Set the transaction log filename (-t) If you are creating a new write file (-c), you can use this option to set the name of the transaction log file.

Operate without confirming actions (–y) Choosing this option replaces the existing write file without prompting you for confirmation.

CHAPTER 16

Database Options

About this chapter

This chapter describes the database, replication, compatibility and Interactive SQL options that can be set to customize and modify database behavior.

Contents	Торіс:	page
	Introduction to database options	556
	Database options	561
	Compatibility options	566
	Replication options	570
	Interactive SQL options	571
	Alphabetical list of options	574

Introduction to database options

Database options control many aspects of database behavior. For example, you can use database options for the following purposes:

- **Compatibility** You can control how much like Adaptive Server Enterprise your Adaptive Server Anywhere database operates, and whether SQL that does not conform to SQL/92 generates errors.
- Error handling You can control what happens when errors such as dividing by zero, or overflow errors, occur.
- Concurrency and transactions You can control the degree of concurrency, and details of COMMIT behavior.

Setting options

You set options with the SET OPTION statement. It has the following general syntax:

```
SET [ EXISTING ] [ TEMPORARY ] OPTION [
userid. | PUBLIC. ]option-name = [
option-value ]
```

Specify a user ID or group name to set the option for that user or group only. Every user belongs to the PUBLIC group. If no user ID or group is specified, the option change is applied to the currently logged on user ID that issued the SET OPTION statement.

For example, the following statement applies an option change to the user DBA, if DBA is the user that issues it:

SET OPTION login_mode = mixed

The following statement applies a change to the **PUBLIC** user ID, a user group to which all users belong.

```
SET OPTION Public.login_mode = standard
```

Caution

Changing option settings while fetching rows from a cursor is not supported because it can lead to ill-defined behavior. For example, changing the DATE_FORMAT setting while fetching from a cursor would lead to different date formats among the rows in the result set. Do not change option settings while fetching rows.

For more information, see "SET OPTION statement" [ASA SQL Reference, page 556].

Finding option settings

You can obtain a list of option settings, or the values of individual options, in a variety of ways.

Getting a list of option values

 Current option settings for your connection are available as a subset of connection properties. You can list all connection properties using the sa_conn_properties system procedure.

call sa_conn_properties

To order this list, you can call sa_conn_properties_by_name.

If you want to filter the result or order by anything other than name, you could also use a SELECT statement. For example:

```
SELECT *
FROM sa_conn_properties()
WHERE PropDescription like '%cache%'
ORDER BY PropNum
```

For more information, see "sa_conn_properties_by_name system procedure" [*ASA SQL Reference*, page 712], and "sa_conn_properties system procedure" [*ASA SQL Reference*, page 711].

• In Interactive SQL, the SET statement with no arguments lists the current setting of options.

SET

- In Sybase Central, right-click on a database, and choose Options from the popup menu.
- Use the following query on the SYSOPTIONS system view:

SELECT * FROM SYSOPTIONS

This displays all PUBLIC values, and those USER values that have been explicitly set.

Getting individual option You can obtain a single setting using the connection_property system function. For example, the following statement reports the value of the ANSI_INTEGER_OVERFLOW option:

SELECT CONNECTION_PROPERTY ('ANSI_INTEGER_OVERFLOW')

For more information, see "CONNECTION_PROPERTY function [System]" [ASA SQL Reference, page 106].

Scope and duration of database options

You can set options at 3 levels of scope: public, user, and temporary.

Temporary options take precedence over user and public settings. User level options take precedence over public settings. If you set a user level option for the current user, the corresponding temporary option is set as well.

Some options (such as COMMIT behavior) are database-wide in scope. Setting these options requires DBA permissions. Other options (such as ISOLATION_LEVEL) can also be applied to just the current connection, and need no special permissions.

Changes to option settings take place at different times, depending on the option. Changing a global option such as RECOVERY_TIME takes place the next time the server is started.

Options that affect the current connection only generally take place immediately. You can change option settings in the middle of a transaction, for example. One exception to this is that *changing options when a cursor is open can lead to unreliable results*. For example, changing DATE_FORMAT may not change the format for the next row when a cursor is opened. Depending on the way the cursor is being retrieved, it may take several rows before the change works its way to the user.

Setting temporary options Adding the TEMPORARY keyword to the SET OPTION statement changes the duration of the change. Ordinarily an option change is permanent. It does not change until it is explicitly changed using the SET OPTION statement.

When the SET TEMPORARY OPTION statement is executed, the new option value takes effect only for the current connection, and for the duration of the connection.

When the SET TEMPORARY OPTION is used to set a **PUBLIC** option, the change is in place for as long as the database is running. When the database is shut down, temporary options for the **PUBLIC** user ID revert back to their permanent value.

Setting an option for the **PUBLIC** user ID temporarily offers a security advantage. For example, when the LOGIN_MODE option is enabled, the database relies on the login security of the system on which it is running. Enabling it temporarily means that a database relying on the security of a Windows NT/2000/XP domain will not be compromised if the database is shut down and copied to a local machine. In this case, the LOGIN_MODE option will revert to its permanent value, which could be Standard, a mode where integrated logins are not permitted.

Setting public options

DBA authority is required to set an option for the PUBLIC user ID.

Changing the value of an option for the **PUBLIC** user ID sets the value of the option for all users who have not SET their own value. An option value cannot be set for an individual user ID unless there is already a **PUBLIC** user ID setting for that option.

Some options which can only be set for the PUBLIC user take effect immediately for existing connections, even though the changed setting will not be visible to users via the connection_property() function. An example of this is the global_database_id option. For this reason, PUBLIC-only options should not be changed while other users are connected to the database.

Deleting option settings

If *option-value* is omitted, the specified option setting will be deleted from the database. If it was a personal option setting, the value reverts back to the PUBLIC setting. If a TEMPORARY option is deleted, the option setting reverts back to the permanent setting.

For example, the following statement resets the ANSI_INTEGER_OVERFLOW option to its default value:

SET OPTION ANSI_INTEGER_OVERFLOW =

For more information, see "SET OPTION statement" [ASA SQL Reference, page 556].

Option classification

Adaptive Server Anywhere provides many options. It is convenient to divide them into a few general classes. The classes of options are:

- "Database options" on page 561
- "Compatibility options" on page 566
- "Replication options" on page 570
- "Interactive SQL options" on page 571

Initial option settings

Connections to Adaptive Server Anywhere can be made through the TDS protocol (Open Client and jConnect JDBC connections) or through the Adaptive Server Anywhere protocol (ODBC and Embedded SQL).

If you have users who use both TDS and the Adaptive Server Anywhere-specific protocol, you can configure their initial settings using stored procedures. As it is shipped, Adaptive Server Anywhere uses this method to set Open Client connections and jConnect connections to reflect default Adaptive Server Enterprise behavior.

The initial settings are controlled using the LOGIN_PROCEDURE option. This option names a stored procedure to run when users connect. The default setting is to use the sp_login_environment system procedure. If you wish to change this behavior you can do so.

In turn, sp_login_environment checks to see if the connection is being made over TDS. If it is, it calls the sp_tsql_environment procedure, which sets several options to new "default" values for the current connection.

For more information, see "LOGIN_PROCEDURE option [database]" on page 603, "sp_login_environment system procedure" [ASA SQL Reference, page 746], and "sp_tsql_environment system procedure" [ASA SQL Reference, page 751].
Database options

This section lists all database options.

OPTION	VALUES	DEFAULT
"AUDITING option [database]" on page 578	ON, OFF	OFF
"BACKGROUND_PRIORITY op- tion [database]" on page 580	ON, OFF	OFF
"BLOCKING option [database]" on page 581	ON, OFF	ON
"BLOCKING_TIMEOUT option [database]" on page 581	Integer	0
"CHECKPOINT_TIME option [database]" on page 582	Number of minutes	60
"CIS_ROWSET_SIZE option [database]" on page 582	Integer	50
"COOPERATIVE_COMMIT TIMEOUT option [database]" on page 586	Integer	250
"COOPERATIVE_COMMITS op- tion [database]" on page 586	ON, OFF	ON
"DATE_FORMAT option [compati- bility]" on page 587	String	'YYYY-MM-DD'
"DATE_ORDER option [compatibil- ity]" on page 588	'YMD', 'DMY', 'MDY'	'YMD'
"DEFAULT_TIMESTAMP INCREMENT option [database]" on page 589	Integer	1
"DELAYED_COMMIT_TIMEOUT option [database]" on page 589	Integer	500
"DELAYED_COMMITS option [database]" on page 590	ON, OFF	OFF
"EXTENDED_JOIN_SYNTAX op- tion [database]" on page 592	ON/OFF	ON

OPTION	VALUES	DEFAULT
"EXCLUDE_OPERATORS option [database]" on page 592	reserved	reserved
"FIRST_DAY_OF_WEEK option [database]" on page 593	1, 2, 3, 4, 5, 6, 7	7
"GLOBAL_DATABASE_ID option [database]" on page 595	Integer	2147483647
"ISOLATION_LEVEL option [com- patibility]" on page 597	0, 1, 2, 3	0
"JAVA_HEAP_SIZE option [database]" on page 601	Number of bytes	1 000 000
"JAVA_NAMESPACE_SIZE option [database]" on page 602	Number of bytes	4 000 000
"LOGIN_MODE option [database]" on page 602	STANDARD, MIXED, IN- TEGRATED	STANDARD
"LOGIN_PROCEDURE option [database]" on page 603	String	sp_login environment
"MAX_CURSOR_COUNT option [database]" on page 605	Integer	50
"MAX_HASH_SIZE option [database]" on page 606	Integer	10
"MAX_PLANS_CACHED option [database]" on page 606	Integer	20
"MAX_STATEMENT_COUNT op- tion [database]" on page 608	Integer >=0	50
"MAX_WORK_TABLE_HASH SIZE option [database]" on page 607	Integer	20
"MIN_PASSWORD_LENGTH op- tion [database]" on page 608	Integer	0
"MIN_TABLE_SIZE_FOR HISTOGRAM option [database]" on page 609	Integer	1 000 rows

OPTION	VALUES	DEFAULT
"ON_CHARSET_CONVERSION FAILURE option [database]" on page 611	IGNORE, WARNING, ERROR	IGNORE
"OPTIMISTIC_WAIT_FOR COMMIT" on page 612	ON, OFF	OFF
"OPTIMIZATION_GOAL option [database]" on page 613	First-row or all-rows	all-rows
"PINNED_CURSOR_PERCENT OF_CACHE option [database]" on page 617	0-100	10
"PRECISION option [database]" on page 617	Number of digits	30
"PREFETCH option [database]" on page 618	ON, OFF	ON
"PRESERVE_SOURCE_FORMAT option [database]" on page 618	ON, OFF	ON
"PREVENT_ARTICLE_PKEY UPDATE option [database]" on page 619	ON, OFF	ON
"READ_PAST_DELETED option [database]" on page 620	ON, OFF	ON
"RECOVERY_TIME option [database]" on page 621	Number of minutes	2
"RETURN_DATE_TIME_AS STRING option [database]" on page 622	ON, OFF	OFF
"RETURN_JAVA_AS_STRING op- tion [database]" on page 623	ON, OFF	OFF
"ROW_COUNTS option [database]" on page 624	ON, OFF	OFF
"SCALE option [database]" on page 624	Number of digits	6

OPTION	VALUES	DEFAULT
"SORT_COLLATION option [database]" on page 624	any valid collation name or collation_ID	Internal
"SUBSUME_ROW_LOCKS option [database]" on page 627	ON, OFF	ON
"SUPPRESS_TDS_DEBUGGING option [database]" on page 627	ON, OFF	OFF
"TDS_EMPTY_STRING_IS_NULL option [database]" on page 628	ON, OFF	OFF
"TIME_FORMAT option [compati- bility]" on page 628	String	'HH:NN:SS.SSS'
"TIME_ZONE_ADJUSTMENT op- tion [database]" on page 628	Integer, or Negative inte- ger enclosed in quotation marks, or String rep- resenting time in hours and minutes, pre- ceded by + or -	Set by either the client's or the server's time zone, depending on the client's connection type
"TIMESTAMP_FORMAT option [compatibility]" on page 629	String	'YYYY-MM-DD HH:NN:SS.SSS'
"TRUNCATE_DATE_VALUES op- tion [database]" on page 630	ON, OFF	ON (post version 7 databases) OFF (pre version 7 databases)
"TRUNCATE_TIMESTAMP VALUES option [database]" on page 631	ON, OFF	OFF
"TRUNCATE_WITH_AUTO COMMIT option [database]" on page 631	ON, OFF	OFF

OPTION	VALUES	DEFAULT
"USER_ESTIMATES option [database]" on page 633	ENABLED, OVERRIDE- MAGIC, DIS- ABLED	OVERRIDE- MAGIC
"UPDATE_STATISTICS option [database]" on page 633	ON, OF	ON
"WAIT_FOR_COMMIT option [database]" on page 634	ON, OFF	OFF

Compatibility options

The following options allow Adaptive Server Anywhere behavior to be made compatible with Adaptive Server Enterprise, or to support both old behavior and allow ISO SQL/92 behavior.

For further compatibility with Adaptive Server Enterprise, some of these options can be set for the duration of the current connection using the Transact-SQL SET statement instead of the Adaptive Server Anywhere SET **OPTION** statement.

For a listing, see "SET statement [T-SQL]" [ASA SQL Reference, page 550].

The default setting for some of these options differs from the Adaptive Default settings Server Enterprise default setting. For compatibility, you should explicitly set the options to ensure compatible behavior.

> When a connection is made using the Open Client or JDBC interfaces, some option settings are explicitly set for the current connection to be compatible with Adaptive Server Enterprise. These options are listed in the following table.

For information on how the settings are made, see "System and catalog stored procedures" [ASA SQL Reference, page 707].

d JDBC connection		
mpatibility with	Option	Setting
laptive Server	ALLOW_NULLS_BY_DEFAULT	OFF
	ANSI_BLANKS	ON
	ANSINULL	OFF
	CHAINED	OFF
	CONTINUE_AFTER_RAISERROR	ON
	DATE_FORMAT	YYYY-MM-DD
	DATE_ORDER	MDY
	ESCAPE_CHARACTER	OFF
	FLOAT_AS_DOUBLE	ON
	ISOLATION_LEVEL	1
	ON_TSQL_ERROR	CONDITIONAL
	QUOTED_IDENTIFIER	OFF

Options for Open Client an CO Ad En

Option	Setting
TSQL_HEX_CONSTANT	ON
TSQL_VARIABLES	ON
TIME_FORMAT	HH:NN:SS.SSS
TIMESTAMP_FORMAT	YYYY-MM-DD HH:NN:SS.SSS

Transact-SQL and SQL/92 compatibility options

The following table lists the compatibility options, their allowed values, and their default settings.

OPTION	VALUES	DEFAULT
"ALLOW_NULLS_BY DEFAULT option [compatibility]" on page 574	ON, OFF	ON
"ANSI_BLANKS option [compati- bility]" on page 574	ON, OFF	OFF
"ANSI_CLOSE_CURSORS_ON ROLLBACK option [compatibil- ity]" on page 574	ON, OFF	OFF
"ANSI_INTEGER_OVERFLOW option [compatibility]" on page 575	ON, OFF	OFF
"ANSI_PERMISSIONS option [compatibility]" on page 575	ON, OFF	ON
"ANSI_UPDATE CONSTRAINTS option [com- patibility]" on page 576	OFF, CUR- SORS, STRICT	CURSORS
"ANSINULL option [compatibil- ity]" on page 577	ON, OFF	ON
"AUTOMATIC_TIMESTAMP op- tion [compatibility]" on page 579	ON, OFF	OFF
"CHAINED option [compatibil- ity]" on page 581	ON, OFF	ON
"CLOSE_ON_ENDTRANS option [compatibility]" on page 583	ON, OFF	ON

OPTION	VALUES	DEFAULT
"CONTINUE_AFTER RAISERROR option [compati- bility]" on page 584	ON,OFF	ON
"CONVERSION_ERROR option [compatibility]" on page 585	ON,OFF	ON
"DIVIDE_BY_ZERO_ERROR option [compatibility]" on page 591	ON, OFF	ON
"ESCAPE_CHARACTER option [compatibility]" on page 592	System use	OFF
"FIRE_TRIGGERS option [com- patibility]" on page 592	ON, OFF	ON
"FLOAT_AS_DOUBLE option [compatibility]" on page 593	ON, OFF	OFF
"NEAREST_CENTURY option [compatibility]" on page 609	Integer	50
"NON_KEYWORDS option [com- patibility]" on page 610	comma- separated keywords list	No keywords turned off
"ON_TSQL_ERROR option [com- patibility]" on page 612	STOP, CON- DITIONAL, CONTINUE	CONDITIONAL
"PERCENT_AS_COMMENT op- tion [compatibility]" on page 616	ON, OFF	ON
"QUERY_PLAN_ON_OPEN op- tion [compatibility]" on page 619	ON, OFF	OFF
"QUOTED_IDENTIFIER option [compatibility]" on page 620	ON, OFF	ON
"RI_TRIGGER_TIME option [compatibility]" on page 623	BEFORE, AFTER	AFTER
"SQL_FLAGGER_ERROR LEVEL option [compatibility]" on page 625	E, I, F, W	W
"SQL_FLAGGER_WARNING LEVEL option [compatibility]" on page 625	E, I, F, W	W

OPTION	VALUES	DEFAULT
"STRING_RTRUNCATION op- tion [compatibility]" on page 626	ON, OFF	OFF
"TSQL_HEX_CONSTANT option [compatibility]" on page 632	ON, OFF	ON
"TSQL_VARIABLES option [compatibility]" on page 632	ON, OFF	OFF

Replication options

The following options are included to provide control over replication behavior. Some replication options are useful for SQL Remote replication, and some are relevant for use with Sybase Replication Server.

OPTION	VALUES	DEFAULT
"BLOB_THRESHOLD option [replication]" on page 580	Integer	256
"COMPRESSION option [replication]" on page 584	Integer, from -1 to 9	6
"DELETE_OLD_LOGS option [replication]" on page 590	ON, OFF	OFF
"QUALIFY_OWNERS option [replication]" on page 619	ON, OFF	ON
"QUOTE_ALL IDENTIFIERS option [repli- cation]" on page 620	ON, OFF	OFF
"REPLICATE_ALL option [replication]" on page 621	ON, OFF	OFF
"REPLICATION_ERROR option [replication]" on page 621	Procedure-name	(no procedure)
"REPLICATION_ERROR PIECE option [replication]" on page 622	Procedure-name	(no procedure)
"SUBSCRIBE_BY REMOTE option [replica- tion]" on page 627	ON, OFF	ON
"VERIFY_ALL COLUMNS option [repli- cation]" on page 634	ON, OFF	OFF
"VERIFY_THRESHOLD option [replication]" on page 634	Integer	1 000

Interactive SQL options

Syntax 1	SET [TEMPORARY] OPTION
	[userid. PUBLIC .]option-name = [option-value]
Syntax 2	SET PERMANENT
Syntax 3	SET
Parameters	userid: identifier, string or host-variable option-name: identifier, string or host-variable
	option-value: host-variable (indicator allowed), string, identifier, or number
Description	SET PERMANENT (syntax 2) stores all current Interactive SQL options in the SYSOPTION system table. These settings are automatically established every time Interactive SQL is started for the current user ID.
	Syntax 3 is used to display all of the current option settings. If any options have temporary settings for Interactive SQL or the database server, these are

OPTION	VALUES	DEFAULT
"AUTO_COMMIT option [ISQL]" on page 579	ON, OFF	OFF
"AUTO_REFETCH option [ISQL]" on page 579	ON, OFF	ON
"BELL option [ISQL]" on page 580	ON, OFF	ON
"COMMAND_DELIMITER option [ISQL]" on page 583	String	۰,۰

displayed; otherwise, the permanent option settings are displayed.

OPTION	VALUES	DEFAULT
"COMMIT_ON_EXIT option [ISQL]" on page 584	ON, OFF	ON
"DESCRIBE_JAVA_FORMAT option [ISQL]" on page 591	Varchar, binary	Varchar
"ECHO option [ISQL]" on page 591	ON, OFF	ON
"INPUT_FORMAT option [ISQL]" on page 595	ASCII, DBASE, DBA- SEII, DBASEIII, EX- CEL, FIXED, FOX- PRO, LOTUS	ASCII
"ISQL_COMMAND_TIMING option [ISQL]" on page 597	ON, OFF	ON
"ISQL_ESCAPE CHARACTER option [ISQL]" on page 597	Character	• \ •
"ISQL_FIELD_SEPARATOR option [ISQL]" on page 598	String	۰, ۰ ,
"ISQL_LOG option [ISQL]" on page 599	File-name	(empty string)
"ISQL_PLAN option [ISQL]" on page 599	NONE, SHORT, LONG, GRAPHICAL, GraphicalWithStatis- tics	SHORT
"ISQL_PRINT_RESULT_SET option [ISQL]" on page 600	LAST, ALL, NONE	LAST
"ISQL_QUOTE option [ISQL]" on page 600	String	4
"NULLS option [ISQL]" on page 610	String	'(NULL)'
"ON_ERROR option [ISQL]" on page 611	STOP, CON- TINUE, PROMPT, EXIT, NOTIFY CONTINUE, NOTIFY_STOP, NOTIFY_EXIT	PROMPT

OPTION	VALUES	DEFAULT
"OUTPUT_FORMAT option [ISQL]" on page 614	ASCII, DBASEII, DBASEIII, EXCEL, FIXED, FOXPRO, HTML, LOTUS, SQL, XML,	ASCII
"OUTPUT_LENGTH option [ISQL]" on page 615	Integer	0
"OUTPUT_NULLS option [ISQL]" on page 616	String	'NULL'
"STATISTICS option [ISQL]" on page 626	0,3,4,5,6	3
"TRUNCATION_LENGTH op- tion [ISQL]" on page 632	Integer	256

Alphabetical list of options

This section lists options alphabetically.

ALLOW_NULLS_BY_DEFAULT option [compatibility]

Function	Controls whether new columns that are created without specifying either NULL or NOT NULL are allowed to contain NULL values.
Allowed values	ON, OFF
Default	ON
	OFF for Open Client and JDBC connections
Description	The ALLOW_NULLS_BY_DEFAULT option is included for Transact-SQL compatibility.
	For more information, see "Setting options for Transact-SQL compatibility" [ASA SQL User's Guide, page 451].

ANSI_BLANKS option [compatibility]

Function	Controls behavior when character data is truncated at the client side.
Allowed values	ON, OFF
Default	OFF
	ON for Open Client and JDBC connections
Description	The ANSI_BLANKS option has no effect unless the database was created with the $-b$ command-line option. It forces a truncation error whenever a value of data type CHAR(N) is read into a C char(M) variable for values of N greater than or equal to M . With ANSI_BLANKS set to OFF , a truncation error occurs only when at least one non-blank character is truncated.
	For Embedded SQL, if ANSI_BLANKS is ON when you supply a value of data type DT_STRING, you must set the sqllen field to the length of the value, including space for the terminating null character. With ANSI_BLANKS off, the length is determined solely by the position of the null character.
	The ANSI_BLANKS setting persists for the life of a connection. Changing it after a connection has been established does not affect that connection.
ANSI CLOSE C	URSORS ON ROLL BACK option [compatibility]

ANSI_CLOSE_CURSORS_ON_ROLLBACK option [compatibility]

Function	Controls whether cursors t	that were opened	WITH HOLD are	closed when a

	ROLLBACK is performed.
Allowed values	ON, OFF
Default	OFF
Description	The draft SQL/3 standard requires all cursors be closed when a transaction is rolled back. By default, on a rollback Adaptive Server Anywhere closes only those cursors that were opened without a WITH HOLD clause. This option allows you to force closure of all cursors.
	The CLOSE_ON_ENDTRANS option overrides the ANSI_CLOSE_CURSORS_ON_ROLLBACK option.
ANSI_INTEGER_O	VERFLOW option [compatibility]
Function	Controls what happens when an arithmetic expression causes an integer overflow error.
Allowed values	ON, OFF
Default	OFF
Description	The ISO SQL/92 standard requires integer overflow should result in a SQLSTATE = 22003 - overflow error. Adaptive Server Anywhere behavior was previously different from this. The ANSI_INTEGER_OVERFLOW option can be set to OFF to maintain compatibility with previous releases of

ANSI_PERMISSIONS option [compatibility]

the software.

Function	Controls permissions checking for DELETE and UPDATE statements.
Allowed values	ON, OFF
Scope	Can be set for the PUBLIC group only. Takes effect immediately. DBA authority is required to set this option.
Default	ON
Description	With ANSI_PERMISSIONS ON, the SQL/92 permissions requirements for DELETE and UPDATE statements are checked. The default value is OFF in Adaptive Server Enterprise. The following table outlines the differences.

SQL Statement	Permissions required with ANSI_PERMISSIONS off	Permissions re- quired with ANSI PERMISSIONS on
UPDATE	UPDATE permission on the columns where values are being set	UPDATE permission on the columns where values are being set SELECT permission on all columns appearing in the WHERE clause SELECT permis- sion on all columns on the right side of the set clause
DELETE	DELETE permission on the table	DELETE permission on the table SELECT per- mission on all columns appearing in the WHERE clause

The ANSI_PERMISSIONS option can be set only for the PUBLIC group. No private settings are allowed.

ANSI_UPDATE_CONSTRAINTS option [compatibility]

Function	Controls the range of updates that are permitted.	
Allowed values	OFF, CURSORS, STRICT	
Default	CURSORS in new databases.	
	OFF in databases created before version 7.0.	
See also	"UPDATE statement" [ASA SQL Reference, page 592]	
Description	Adaptive Server Anywhere provides several extensions that allow updates which are not permitted by the ANSI SQL standard. These extensions provide powerful, efficient mechanisms for performing updates. However, in some cases, they cause behavior that is not intuitive. This behavior can produce anomalies such as lost updates if the user application is not designed to expect the behavior of these extensions.	
	The ANSI_UPDATE_CONSTRAINTS option controls whether updates are restricted to those permitted by the SQL/92 standard.	
	If the option is set to STRICT, the following updates are prevented:	
	 Updates of cursors containing JOINS 	
	• Updates of columns that appear in an ORDER BY clause	

• The FROM clauses is not allowed in UPDATE statements

If the option is set to CURSORS, these same restrictions are in place, but only for cursors. If a cursor is not opened with FOR UPDATE or FOR READ ONLY, the database server chooses updatability based on the SQL/92 standard. If the ANSI_UPDATE_CONSTRAINTS option is CURSORS or STRICT, cursors containing an ORDER BY clause default to FOR READ ONLY; otherwise, they continue to default to FOR UPDATE.

ANSINULL option [compatibility]

Function	Controls the interpretation of NULL values.
Allowed values	ON, OFF
Default	ON
Description	This option is implemented primarily for Transact-SQL (Adaptive Server Enterprise) compatibility. ANSINULL affects the results of comparison predicates with NULL constants, and also affects warnings issued for grouped queries over NULL values.
	With ANSINULL ON , ANSI three-valued logic is used for all comparison predicates in a WHERE or HAVING clause, or in an ON condition. Any comparisons with NULL using = or != evaluate to unknown.
	Setting ANSINULL to OFF means that Adaptive Server Anywhere uses two-valued logic for the following four conditions:
	<expr> = NULL</expr>
	<expr> != NULL</expr>
	$\langle expr \rangle = @var // @var is a procedure variable, or a host variable$
	$\langle expr \rangle \coloneqq @$ var
	In each case, the predicate evaluates to either true or false - never unknown. In such comparisons, the NULL value is treated as a special value in each domain, and an equality (=) comparison of two NULL values will yield true. Note that the expression <i>expr</i> must be a relatively simple expression, referencing only columns, variables, and literals; subqueries and functions are not permitted.
	With ANSINULL ON , the evaluation of any aggregate function, except COUNT(*), on an expression that contains at least one NULL value, may generate the warning 'null value eliminated in aggregate function' (SQLSTATE=01003). With ANSINULL OFF, this warning does not appear.

- Setting ANSINULL to OFF affects only WHERE, HAVING, or ON predicates in SELECT, UPDATE, DELETE, and INSERT statements. The semantics of comparisons in a CASE or IF statement, or in IF expressions, are unaffected.
- Adaptive Server Enterprise 12.5 introduced a change in behavior for LIKE predicates with a NULL pattern string when ANSINULL is **OFF**. In Adaptive Server Anywhere, LIKE predicates remain unaffected by the setting of ANSINULL.

AUDITING option [database]

Function	Enables and disables auditing in the database.
Allowed values	ON, OFF
Scope	Can be set for the PUBLIC group only. Takes effect immediately. DBA permissions are required to set this option.
Default	OFF
See Also	"sa_enable_auditing_type" [ASA SQL Reference, page 714]
	"sa_disable_auditing_type" [ASA SQL Reference, page 713]
Description	This option turns auditing on and off.
	Auditing is the recording of detailed information about many events in the database in the transaction log. Auditing provides some security features, at the cost of some performance.
	For the AUDITING option to work, you must set the AUDITING option to ON, and also specify which types of information you want to audit using the sa_enable_auditing_type system procedure. Auditing will not take place if either of the following are true:
	• AUDITING is set to OFF
	 Auditing options have been disabled
	If you set AUDITING ON, and do not specify auditing options, all types of auditing information are recorded. Alternatively, you can choose to record any combination of the following: audit permission checks, connection attempts, DDL statements, public options, and triggers.
Example	• Turn on auditing
	SET ODTION DIDITC AUDITING - (ON)

AUTO_COMMIT option [ISQL]

Controls whether a COMMIT is performed after each statement.
ON, OFF
OFF
If AUTO_COMMIT is ON , a database COMMIT is performed after each successful statement. If the COMMIT fails, you have the option to execute additional SQL statements and perform the COMMIT again, or execute a ROLLBACK statement.
By default, a COMMIT or ROLLBACK is performed only when the user issues a COMMIT or ROLLBACK statement or a SQL statement that causes an automatic commit (such as the CREATE TABLE statement).

AUTO_REFETCH option [ISQL]

Function	Controls whether query results are fetched again after deletes, updates, and inserts.
Allowed values	ON, OFF
Default	ON
Description	If AUTO_REFETCH is on, the current query results that appear on the Results tab in the Interactive SQL Results pane will be refetched from the database after any INSERT, UPDATE, or DELETE statement. Depending on how complicated the query is, this may take some time. For this reason, it can be turned off.

AUTOMATIC_TIMESTAMP option [compatibility]

Function	Controls interpretation of new columns with the TIMESTAMP data type.
Allowed values	ON, OFF
Default	OFF
Description	Controls whether any new columns with the TIMESTAMP data type that do not have an explicit default value defined are given a default value of the Transact-SQL timestamp value as a default. The AUTOMATIC_TIMESTAMP option is included for Transact-SQL compatibility. The default is OFF.
	For more information, see "Setting options for Transact-SQL compatibility" [<i>ASA SQL User's Guide</i> , page 451].

BACKGROUND_PRIORITY option [database]

Function	To limit impact on the performance of connections other than the current connection.
Allowed values	ON, OFF
Scope	Can be set for an individual connection or for the PUBLIC group. Takes effect immediately.
	If you set this option temporarily, that setting applies to the current connection only. Different connections under the same user ID can have different settings for this option.
Default	OFF
Description	When set to ON , it requests that the current connection have minimal impact on the performance of other connections. This option allows tasks for which responsiveness is critical to coexist with other tasks for which performance is not as important.

BELL option [ISQL]

Function	Controls whether the bell sounds when an error occurs.
Allowed values	ON, OFF
Default	ON
Description	Set this option according to your preference.

BLOB_THRESHOLD option [replication]

Function	Controls the size of value that the Message Agent treats as a long object (blob).
Allowed values	Integer, in kilobytes
Default	256
Description	Any value longer than the BLOB_THRESHOLD option is replicated as a blob. That is, it is broken into pieces and replicated in chunks, before being reconstituted using a SQL variable and concatenating the pieces at the recipient site.
	If you set BLOB_THRESHOLD to a high value in remote Adaptive Server Anywhere databases, blobs are not broken into pieces, and operations can be applied to Adaptive Server Enterprise by the Message Agent. Each SQL

statement must fit within a message, so this only allows replication of small blobs.

BLOCKING option [database]

Function	Controls the behavior in response to locking conflicts.
Allowed values	ON, OFF
Scope	Can be set for an individual connection or for the PUBLIC group. Takes effect immediately.
Default	ON
Description	If BLOCKING is ON, any transaction attempting to obtain a lock that conflicts with an existing lock held by another transaction waits until every conflicting lock is released. At that time, the write goes through. If BLOCKING is OFF, the transaction that attempts to obtain a conflicting lock receives an error.

BLOCKING_TIMEOUT option [database]

Function	To control how long a transaction waits to obtain a lock.
Allowed values	Integer, in milliseconds
Scope	Can be set for an individual connection or for the PUBLIC group. Takes effect immediately.
Default	0
See also	"BLOCKING option [database]" on page 581
Description	When BLOCKING is ON, any transaction attempting to obtain a lock that conflicts with an existing lock waits for BLOCKING_TIMEOUT milliseconds for the conflicting lock to be released. If the lock is not released within BLOCKING_TIMEOUT milliseconds, then an error is returned for the waiting transaction.
	Setting this option to 0 forces all transactions attempting to obtain a lock to wait until all conflicting transactions release their locks.

CHAINED option [compatibility]

Function

Controls transaction mode in the absence of a BEGIN TRANSACTION statement.

Allowed values	ON, OFF
Default	ON
	OFF for Open Client and JDBC connections
Description	Controls the Transact-SQL transaction mode. In Unchained mode (CHAINED = OFF), each statement is committed individually unless an explicit BEGIN TRANSACTION statement is executed to start a transaction. In chained mode (CHAINED = ON) a transaction is implicitly started before any data retrieval or modification statement.

CHECKPOINT_TIME option [database]

Function	Set the maximum number of minutes that the database server will run without doing a checkpoint.
Allowed values	Integer
Scope	Can be set only for the PUBLIC group. DBA authority is required to set the option. You must shut down and restart the database server for the change to take effect.
Default	60
Description	This option is used with the "RECOVERY_TIME option [database]" on page 621 to decide when checkpoints should be done.
	For information on checkpoints, see "Checkpoints and the checkpoint log" on page 360.

CIS_OPTION option [database]

Function	Controls debugging for remote data access.
Allowed values	0, 7
Scope	Can be set for an individual connection or for the PUBLIC group.
Default	0
Description	Setting this option to 7 turns on debugging for remote data access.

CIS_ROWSET_SIZE option [database]

Function	Set the number of rows that are returned from remote servers for each fetch.
Allowed values	Integer
Scope	Can be set for an individual connection or for the PUBLIC group. Takes effect when a new connection is made to a remote server.

CLOSE_ON_ENDTRANS option [compatibility]		
Description	This option sets the ODBC FetchArraySize value when using ODBC to connect to a remote database server.	
Default	50	

Function	Controls the closing of cursors at the end of a transaction.	
Allowed values	ON, OFF	
Default	ON	
Description	When CLOSE_ON_ENDTRANS is set to ON , cursors are closed whenev a transaction is committed unless the cursor was opened WITH HOLD. The behavior when a transaction is rolled back is governed by the ANSI_CLOSE_CURSORS_AT_ROLLBACK option.	
	When CLOSE_ON_ENDTRANS is set to OFF , cursors are not closed at either a commit or a rollback, regardless of the ANSI_CLOSE_CURSORS_AT_ROLLBACK option setting or whether the cursor was opened WITH HOLD or not.	
	Setting this to OFF provides Adaptive Server Enterprise compatible behavior.	

COMMAND_DELIMITER option [ISQL]

Function	Sets the string that indicates the end of a statement in Interactive SQL.		
Allowed values	String		
Default	Semi-colon (;)		
See also	"The Interactive SQL utility" on page 492		
Description	You can specify a command delimiter to indicate the end of a SQL statement. This command delimiter can be any sequence of characters (including numbers, letters, and punctuation), but it cannot contain embedded blanks. As well, it can contain a semicolon, but only as the first character.		
	If the command delimiter is set to a string beginning with a character that is valid in identifiers, the command delimiter must be preceded by a space. This command delimiter is case sensitive. You must enclose the new command delimiter in single quotation marks.		
	Note that if the command delimiter is a semicolon (the default), no space is required before the semicolon.		

Example 1	SET OPTION COMMAND_DELIMITER='~'
Example 2	<pre>set temporary option command_delimiter = ' select'; message 'hello' select</pre>
Example 3	You can also use Interactive SQL's -d command line option to set the command delimiter without including a SET TEMPORARY OPTION COMMAND_DELIMITER statement in a .SQL file. For example, if you have a script file named test.sql which uses tildes (~) as command delimiters, you could run:

dbisql -d "~" test.sql

COMMIT_ON_EXIT option [ISQL]

Function	Controls behavior when Interactive SQL disconnects or terminates.		
Allowed values	ON, OFF		
Default	ON		
Description	Controls whether a COMMIT or ROLLBACK is done when you leave Interactive SQL. When COMMIT_ON_EXIT is set to ON , a COMMIT i done: otherwise a ROLLBACK is done.		

COMPRESSION option [replication]

Function	Set the level of compression for SQL Remote messages.		
Allowed values	Integer, from -1 to 9		
Default	6		
Description	• The values have the following meanings:		
	 -1 Send messages in Version 5 format. Message Agents (both <i>dbremote</i> and <i>ssremote</i>) from previous versions of SQL Remote cannot read 		

messages sent in Version 6 format. You should ensure that COMPRESSION is set to -1 until all Message Agents in your system are upgraded to Version 6.

- 0 No compression.
- ◆ 1 to 9 Increasing degrees of compression. Creating messages with high compression can take longer than creating messages with low compression.

CONTINUE_AFTER_RAISERROR option [compatibility]

Function

Controls behavior following a RAISERROR statement.

Allowed values	ON, OFF		
Default	OFF for databases created with Version 5.x.		
	ON for databases created with Version 6 or later.		
See also	"ON_TSQL_ERROR option [compatibility]" on page 612		
Description	The RAISERROR statement is used within procedures and triggers to generate an error. When this option is set to OFF , the execution of the procedure or trigger is stopped whenever the RAISERROR statement is encountered.		
	If you set the CONTINUE_AFTER_RAISERROR option to ON , the RAISERROR statement no longer signals an execution-ending error. Instead, the RAISERROR status code and message are stored and the most recent RAISERROR is returned when the procedure completes. If the procedure that caused the RAISERROR was called from another procedure, the RAISERROR is not returned until the outermost calling procedure terminates.		
	Intermediate RAISERROR statuses and codes are lost after the procedure terminates. If, at return time, an error occurs along with the RAISERROR, then the information for the new error is returned and the RAISERROR information is lost. The application can query intermediate RAISERROR statuses by examining the @@error global variable at different execution points.		
	The setting of the CONTINUE_AFTER_RAISERROR option is used to control behavior following a RAISERROR statement <i>only</i> if the ON_TSQL_ERROR option is set to CONDITIONAL (the default). If you set the ON_TSQL_ERROR option to STOP or CONTINUE, the ON_TSQL_ERROR setting takes precedence over the CONTINUE_AFTER_RAISERROR setting.		
	ROR option [compatibility]		
Function	Controls the reporting of data type conversion failures on fetching information from the database.		
Allowed values	ON, OFF		

Default

ON

Description This option controls whether data type conversion failures, when data is fetched from the database or inserted into the database, are reported by the database as errors (CONVERSION_ERROR set to **ON**) or as a warning (CONVERSION_ERROR set to **OFF**).

When CONVERSION_ERROR is set to **ON**, the SQLE_CONVERSION_ERROR error is generated. If the option is set to **OFF**, the warning SQLE_CANNOT_CONVERT is produced.

If conversion errors are reported as warnings only, the NULL value is used in place of the value that could not be converted. In Embedded SQL, an indicator variable is set to -2 for the column or columns that cause the error.

COOPERATIVE_COMMIT_TIMEOUT option [database]

Function	Governs when a COMMIT entry in the transaction log is written to disk.		
Allowed values	Integer, in milliseconds		
Scope	Can be set for an individual connection or for the PUBLIC group. Takes effect immediately.		
Default	250		
Description	This option has meaning only when COOPERATIVE_COMMITS is set to ON . The database server waits for the specified number of milliseconds for other connections to fill a page of the log before writing to disk. The default setting is 250 milliseconds.		

COOPERATIVE_COMMITS option [database]

Function	Controls when commits are written to disk.	
Allowed values	ON or OFF	
Scope	Can be set for an individual connection or for the PUBLIC group. Takes effect immediately.	
Default	ON	
Description	If COOPERATIVE_COMMITS is set to OFF , a COMMIT is written to disk as soon as the database server receives it, and the application is then allowed to continue.	
	If COOPERATIVE_COMMITS is set to ON (the default) and if there are other active connections, the database server does not immediately write the COMMIT to the disk. Instead, the application waits for up to the maximum length set by the COOPERATIVE_COMMIT_TIMEOUT option for something else to put on the pages before they are written to disk.	
	Setting COOPERATIVE_COMMITS to ON , and increasing the COOPERATIVE_COMMIT_TIMEOUT setting, increases overall database server throughput by cutting down the number of disk I/Os, but at the expense of a longer turnaround time for each individual connection.	

If both COOPERATIVE_COMMITS and DELAYED_COMMITS are set to **ON**, and the COOPERATIVE_COMMIT_TIMEOUT interval passes without the pages getting written, the application is resumed (as if the commit had worked), and the remaining interval (DELAYED_COMMIT_TIMEOUT - COOPERATIVE_COMMIT_TIMEOUT) is used as a DELAYED_COMMIT_INTERVAL. The pages will then be written, even if they are not full.

DATE_FORMAT option [compatibility]

Function	Sets the format for dates retrieved from the database.	
Allowed values	String	
Scope	Can be set for an individual connection or for the PUBLIC group. Takes effect immediately.	
Default	'YYYY-MM-DD'. This corresponds to ISO date format specifications.	
Description	The format is a string using the following symbols:	

.

Symbol	Description		
уу	Two digit year		
уууу	Four digit year		
mm	Two digit month, or two digit minutes if following a colon (as in hh:mm)		
mmm[m]	Character short form for months—as many characters as there are "m"s		
d	Single digit day of week, $(0 = $ Sunday, $6 =$ Saturday $)$		
dd	Two digit day of month		
ddd[d]	Character short form for day of the week		
hh	Two digit hours		
nn	Two digit minutes		
ss[.ss]	Seconds and parts of a second		
aa	AM or PM (12 hour clock)		
pp	PM if needed (12 hour clock)		
jij	Day of the year, from 1 to 366		

	Each symbol is substituted with the appropriate data for the date that is being formatted.			
	For symbols that represent case of the output as follow	For symbols that represent character data (such as mmm), you can control the case of the output as follows:		
	 Type the symbol in all u case. For example, MMM 	 Type the symbol in all upper case to have the format appear in all upper case. For example, MMM produces JAN. 		
	 Type the symbol in all l case. For example, mmm 	• Type the symbol in all lower case to have the format appear in all lower case. For example, mmm produces jan.		
	• Type the symbol in mixed case to have Adaptive Server Anywhere choose the appropriate case for the language that is being used. For example, in English, typing Mmm produces May, while in French it produces mai.			
	For symbols that represent numeric data, you can control zero-padding with the case of the symbols:			
	 Type the symbol in same-case (such as MM or mm) to allow zero padding. For example, yyyy/mm/dd could produce 2002/01/01. 			
	 Type the symbol in mixed case (such as Mm) to suppress zero padding. For example, yyyy/Mm/Dd could produce 2002/1/1. 			
Examples	 The following table illu the output from the foll 2001. 	strates DATE_FORMAT settings, together with owing statement, executed on Thursday May 21,		
	SELECT CURRENT DAT	Έ		
	DATE_FORMAT	SELECT CURRENT DATE		
	yyyy/mm/dd/ddd	2001/05/21/thu		
	jjj	141		
	mmm yyyy	May 1998		
	mm-yyyy	05-1998		

DATE_ORDER option [compatibility]

Function	Controls the interpretation of date formats.	
Allowed values	'MDY', 'YMD', or 'DMY'	
Scope	Can be set for an individual connection or for the PUBLIC group. Takes effect immediately.	
Default	YMD. This corresponds to ISO date format specifications.	

588

For Open Client and JDBC connections, the default is set to MDY.

Description The database option DATE_ORDER is used to determine whether 10/11/12 is Oct 11 1912, Nov 12 1910, or Nov 10 1912. The option can have the value 'MDY', 'YMD', or 'DMY'.

DEFAULT_TIMESTAMP_INCREMENT option [database]

Function	Specifies the number of microseconds to add to a column of type TIMESTAMP in order to keep values in the column unique.	
Allowed values	Integer, between 1 and 60 000 000 inclusive.	
Scope	Can be set for an individual connection or for the PUBLIC group. Takes effect immediately.	
Default	1	
See also	"TRUNCATE_TIMESTAMP_VALUES option [database]" on page 631	
Description	Since a TIMESTAMP value is precise to six decimal places in Adaptive Server Anywhere, by default 1 microsecond (0.000001 of a second) is added to differentiate between two identical TIMESTAMP values.	
	Some software, such as Microsoft Access, truncates TIMESTAMP values to three decimal places, making valid comparisons a problem. You can set the TRUNCATE_TIMESTAMP_VALUES option to ON to specify the number of decimal place values Adaptive Server Anywhere stores to maintain compatibility.	

DELAYED_COMMIT_TIMEOUT option [database]

Function	Determines when the server returns control to an application following a COMMIT.
Allowed values	Integer, in milliseconds.
Scope	Can be set for an individual connection or for the PUBLIC group. Takes effect immediately.
Default	500
Description	This option has meaning only when DELAYED_COMMITS is set to ON . it governs when a COMMIT entry in the transaction log is written to disk. With DELAYED_COMMITS set to ON , the database engine waits for the number of milliseconds set in the DELAYED_COMMIT_TIMEOUT option for other connections to fill a page of the log before writing the current page contents to disk.
	For more information, see "DELAYED_COMMITS option [database]"

on page 590.

DELAYED_COMMITS option [database]

Function	Determines when the server returns control to an application following a COMMIT.	
Allowed values	ON, OFF	
Scope	Can be set for an individual connection or for the PUBLIC group. Takes effect immediately.	
Default	OFF. This corresponds to ISO COMMIT behavior.	
Description	When set to ON , the database server replies to a COMMIT statement immediately instead of waiting until the transaction log entry for the COMMIT has been written to disk. When set to OFF , the application must wait until the COMMIT is written to disk.	
	When this option is ON , the log is written to disk when the log page is full or according to the DELAYED_COMMIT_TIMEOUT option setting, whichever is first. There is a slight chance that a transaction may be lost even though committed if a system failure occurs after the server replies to a COMMIT, but before the page is written to disk. Setting DELAYED_COMMITS to ON , and the DELAYED_COMMIT_TIMEOUT option to a high value, promotes a quick response time at the cost of security.	
	If both COOPERATIVE_COMMITS and DELAYED_COMMITS are set to ON , and if the COOPERATIVE_COMMIT_TIMEOUT interval passes without the pages getting written, the application is resumed (as if the commit had worked), and the remaining interval (DELAYED_COMMIT_TIMEOUT - COOPERATIVE_COMMIT_TIMEOUT) is used as a DELAYED_COMMIT_interval after which the pages will be written, even if they are not full.	

DELETE_OLD_LOGS option [replication]

Function	Controls whether transaction logs are deleted when their messages have been replicated or synchronized.
Allowed values	ON, OFF, DELAY
Default	OFF
Description	This option is used at MobiLink clients, by SQL Remote, and by the Adaptive Server Anywhere Replication Agent. The default setting is OFF . When it is set to ON , each old transaction log is deleted when all the

changes it contains have been sent and confirmed as received. When it is set to **DELAY**, each old transaction log with a file name indicating that it was created on the current day is not deleted.

For more information about how to use the DELETE_OLD_LOGS option in conjunction with the Backup statement to delete old copies of transaction logs, see the "BACKUP statement" [*ASA SQL Reference*, page 263].

DESCRIBE_JAVA_FORMAT option [ISQL]

Function	Controls whether Java objects are interpreted as strings (for display) or as binary (for loading and unloading).	
Allowed values	varchar, binary	
Default	Varchar	
Description	When set to varchar , Interactive SQL converts all data fetched from t database to strings. The database server calls the toString() method Java columns to provide formatted output for the Results tab in the Re pane.	
	When set to binary , Interactive SQL does no data conversion.	

DIVIDE_BY_ZERO_ERROR option [compatibility]

Function	Controls the reporting of division by zero.	
Allowed values	ON, OFF	
Default	ON	
Description	This option indicates whether division by zero is reported as an error. If the option is set ON, then division by zero results in an error with SQLSTATE 22012.	
	If the option is set OFF, division by zero is not an error. Instead, a NULL is returned.	

ECHO option [ISQL]

Function	Controls whether statements are echoed before they are executed.	
Allowed values	ON, OFF	
Default	ON	
Description	This option is most useful when you use the READ statement to execute a Interactive SOL command file.	

ESCAPE_CHARACTER option [compatibility]

This option is reserved for system use. Do not change the setting of this option.

EXCLUDE_OPERATORS option [database]

This option is reserved for system use. Do not change the setting of this option.

EXTENDED_JOIN_SYNTAX option [database]

Function	Controls whether queries with duplicate correlation names syntax for multi-table joins are allowed, or reported as an error.	
Allowed values	ON, OFF	
Default	ON	
See also	"REWRITE function [Miscellaneous]" [ASA SQL Reference, page 177]	
Description	If this option is set to ON then Adaptive Server Anywhere allows duplicate correlation names to be used in the null-supplying side of the outer joins. The tables or views specified with the same correlation name are interpreted as the same instance of the table or view.	
	The following FROM clause illustrates the Adaptive Server Anywhere interpretation of a join using duplicate correlation names:	
(R left outer join T on (Cl), T join S on (C2))		
	where C1 and C2 are search conditions.	
	If the option is set to ON , this join is interpreted as follows:	
	(R left outer join T on (Cl)) join S on (C2)	
	If the option is set to OFF , the following error is generated:	
	ASA Error -137: Table 'T' requires a unique correlation name.	
Note:	To see the result of eliminating duplicate correlation names, you can view the rewritten statement using the REWRITE function with the second argument set to ANSI.	

FIRE_TRIGGERS option [compatibility]

Function

Controls whether triggers are fired in the database.

Allowed values	ON, OFF	
Default	ON	
Description	When set to ON , triggers are fired. When set to OFF , no triggers are fired, including referential integrity triggers (such as cascading updates and deletes). Only a user with DBA authority can set this option. The option is overridden by the -gf command-line option, which turns off all trigger firing regardless of the FIRE_TRIGGERS setting.	
	This option is relevant when replicating data from Adaptive Server Enterprise to Adaptive Server Anywhere because all actions from Adaptive Server Enterprise transaction logs are replicated to Adaptive Server Anywhere, including actions carried out by triggers.	

FIRST_DAY_OF_WEEK option [database]

Function	Sets the numbering of the days of the week.	
Allowed values	1 2 3 4 5 6 7	
Default	7 (Sunday is the first day of the week)	
Description	The values have the following meaning:	

Value	Meaning
1	Monday
2	Tuesday
3	Wednesday
4	Thursday
5	Friday
6	Saturday
7	Sunday

FLOAT_AS_DOUBLE option [compatibility]

Function	Controls the interpretation of the FLOAT keyword.
Allowed values	ON, OFF
Default	OFF
	ON for Open Client and JDBC connections
Description	The FLOAT_AS_DOUBLE option makes the FLOAT keyword behave like

Adaptive Server Enterprise's FLOAT keyword when a precision is not specified.

When enabled (set to **ON**), all occurrences of the keyword FLOAT are interpreted as equivalent to the keyword DOUBLE within SQL statements.

By default, Adaptive Server Anywhere FLOAT values are interpreted by Adaptive Server Enterprise as REAL values. Since Adaptive Server Enterprise treats its own FLOAT values as DOUBLE, enabling this option makes Adaptive Server Anywhere treat FLOAT values in the same way Enterprise treats FLOAT values.

REAL values are four bytes; DOUBLE values are eight bytes. According to the ANSI SQL/92 specification, FLOAT can be interpreted based on the platform. It is up to the database to decide what size the value is, so long as it can handle the necessary precision. Adaptive Server Enterprise and Adaptive Server Anywhere exhibit different default behavior.

The FLOAT_AS_DOUBLE option takes effect only when no precision is specified. For example, the following statement is not affected by the option setting:

```
create table t1(
    c1 float(5)
)
```

The following statement is affected by the option setting:

```
create table t2(
    c1 float)
// affected by option setting
```

FOR_XML_NULL_TREATMENT option [database]

Function	Controls the treatment of NULL values in queries that use the FOR XML clause.
Allowed values	EMPTY, OMIT
Default	OMIT
See also	"Using the FOR XML clause to retrieve query results as XML" [ASA SQL User's Guide, page 491]
	"SELECT statement" [ASA SQL Reference, page 541]
Description	If you execute a query that includes the FOR XML clause, the FOR_XML_NULL_TREATMENT option determines how NULL values are treated. By default, elements and attributes that contain NULL values are omitted from the result. Setting this option to EMPTY generates empty

elements or attributes if the value is NULL.

GLOBAL_DATABASE_ID option [database]

Function	For use in generating unique primary keys in a replication environment. Controls the beginning of the range of values for columns created with DEFAULT GLOBAL AUTOINCREMENT.
Allowed values	Integer
Default	2147483647
Scope	PUBLIC setting only.
See also	"CREATE TABLE statement" [ASA SQL Reference, page 361]
	"Database-level properties" on page 664
	"Ensuring unique primary keys" [SQL Remote User's Guide, page 129]
Description	If you create a column with DEFAULT GLOBAL AUTOINCREMENT, its value is incremented. The initial value is determined by the GLOBAL_DATABASE_ID value and the partition size. For more information on the partition size, see "CREATE TABLE statement" [<i>ASA SQL Reference</i> , page 361].
	Setting GLOBAL_DATABASE_ID to the default value indicates that GLOBAL DEFAULT AUTOINCREMENT is disabled. In this case NULL is generated as a default.
	You can find the value of the option in the current database using the following statement:
	<pre>select db_property('GlobalDBId')</pre>
	This feature is of particular use in replication environments to ensure unique primary keys.
	For more information, see "Ensuring unique primary keys" [<i>SQL Remote User's Guide</i> , page 129] and "Maintaining unique primary keys using global autoincrement" [<i>MobiLink Synchronization User's Guide</i> , page 82].

INPUT_FORMAT option [ISQL]

Function	Sets the default data format expected by the INPUT statement.
Allowed values	String. See below.
Default	ASCII
Description	Certain file formats contain information about column names and types.

Using this information, the INPUT statement will create the database table if it does not already exist. This is a very easy way to load data into the database. The formats that have enough information to create the table are: DBASEII, DBASEIII, FOXPRO, and LOTUS.

Allowable input formats are:

◆ ASCII Input lines are assumed to be ASCII characters, one row per line, with values separated by commas. Alphabetic strings may be enclosed in apostrophes (single quotes) or quotation marks (double quotes). Strings containing commas must be enclosed in either single or double quotes. If single or double quotes are used, double the quote character to use it within the string. Optionally, you can use the DELIMITED BY clause to specify a different delimiter string than the default, which is a comma (,).

Three other special sequences are also recognized. The two characters \n represent a newline character, \represents a single backslash character, and the sequence \xDD represents the character with hexadecimal code DD.

- **DBASE** The file is in dBASE II or dBASE III format. Interactive SQL will attempt to determine which format, based on information in the file. If the table doesn't exist, it will be created.
- **DBASEII** The file is in dBASE II format. If the table doesn't exist, it will be created.
- **DBASEIII** The file is in dBASE III format. If the table doesn't exist, it will be created.
- **EXCEL** Input file is in the format of Microsoft Excel 2.1. If the table doesn't exist, it will be created.
- **FIXED** Input lines are in fixed format. The width of the columns can be specified using the COLUMN WIDTHS clause. If they are not specified, column widths in the file must be the same as the maximum number of characters required by any value of the corresponding database column's type.
- **FOXPRO** The file is in FoxPro format. The FoxPro memo field is different than the dBASE memo field. If the table doesn't exist, it will be created.
- LOTUS The file is a Lotus WKS format worksheet. INPUT assumes that the first row in the Lotus WKS format worksheet consists of column names. If the table doesn't exist, it will be created. In this case, the types and sizes of the columns created may not be correct because the information in the file pertains to a cell, not to a column.
ISOLATION_LEVEL option [compatibility]

Function	Controls the locking isolation level.	
Allowed values	0, 1, 2, or 3	
Scope	Can be set for an individual connection or for the PUBLIC group. Takes effect immediately.	
Default	0	
	1 for Open Client and JDBC connections	
Description	This option controls the locking isolation level as follows.	
	• 0 Allow dirty reads, non-repeatable reads, and phantom rows.	
	• 1 Prevent dirty reads. Allow non-repeatable reads and phantom rows.	
	• 2 Prevent dirty reads and guarantee repeatable reads. Allow phantom rows.	
	• 3 Serializable. Do not allow dirty reads, guarantee repeatable reads, and do not allow phantom rows.	
	For more information, see "Isolation levels and consistency" [ASA SQL User's Guide, page 104].	

ISQL_COMMAND_TIMING option [ISQL]

Function	Controls whether SQL statements are timed or not.
Allowed values	ON, OFF
Default	ON
Description	This boolean option controls whether SQL statements are timed or not. If you set the option to ON, YES, or 1, the time of execution appears in the Messages pane after you execute a statement. If you set the option to OFF, NO, or 0, the time does not appear.
	Verse 1. Out the station of the Management of the Outlines fight

You can also set this option on the Messages tab of the Options dialog.

ISQL_ESCAPE_CHARACTER option [ISQL]

Function	Controls the escape character used in place of unprintable characters in data exported to ASCII files.
Allowed values	Any single character
Default	A backslash (\)

Description	When Interactive SQL exports strings that contain unprintable characters (such as a carriage return), it converts each unprintable character into a hexadecimal format and precedes it with an escape character. The character you specify for this setting is used in the output if your OUTPUT statement does not contain an ESCAPE CHARACTER clause. This setting is used only if you are exporting to an ASCII file.
Example	• Create a table that contains one string value with an embedded carriage return (denoted by the "\n" in the INSERT statement). Then export the data to <i>c</i> :\ <i>escape.txt</i> with a # sign as the escape character.
	CREATE TABLE escape_test(TEXT varchar(10)); INSERT INTO escape_test VALUES('one\ntwo'); SET TEMPORARY OPTION ISQL_ESCAPE_CHARACTER='#'; SELECT * FROM escape_test; OUTPUT TO c:\escape.txt FORMAT ASCII
	This code places the following data in <i>escape.txt</i> :
	'one#x0Atwo'
	where # is the escape character and xOA is the hexadecimal equivalent of the "\n" character.

The start and end characters (in this case, single quotation marks) depend on the ISQL_QUOTE setting.

ISQL_FIELD_SEPARATOR option [ISQL]

Function	Controls the default string used for separating values in data exported to ASCII files.
Allowed values	String
Default	A comma (,)
Description	Controls the default string used for separating (or delimiting) values in data exported to ASCII files. If an OUTPUT statement does not contain a DELIMITED BY clause, the value of this setting is used.
Example	• Set the field separator to a colon in the data exported to <i>c:\employee.txt</i> .
	SET TEMPORARY OPTION ISQL_FIELD_SEPARATOR=':'; SELECT emp_lname, emp_fname FROM employee WHERE emp_id < 150;
	OUTPUT TO c:\employee.txt FORMAT ASCII
	This code places the following data in <i>employee.txt</i> :
	'Whitney': 'Fran'
	'Cobb':'Matthew'
	'Chin':'Philip'

'Jordan':'Julie'

The start and end characters (in this case, single quotation marks) depend on the ISQL_QUOTE setting.

ISQL_LOG option [ISQL]

Function	Controls logging behavior.
Allowed values	String, containing a file name.
Default	Empty string.
Description	If ISQL_LOG is set to a non-empty string, all Interactive SQL statements are added to the end of the named file. Otherwise, if ISQL_LOG is set to the empty string, Interactive SQL statements are not logged.
	Individual session only
	This option logs an individual Interactive SQL session only. See "Backup and Data Recovery" on page 337 for a description of the transaction log

that logs all changes to the database by all users.

ISQL_PLAN option [ISQL]

Function	Controls the amount of information that appears on the Plan tab and the UltraLite Plan tab in the Results pane after you execute statements.		
Allowed values	SHORT, LONG, GRAPHICAL, GRAPHICALWITHSTATISTICS, NONE		
Default	SHORT		
Description	After you execute a SQL statement in Interactive SQL, the optimizer provides information about how it optimized the execution on the Plan tab. To set the level of detail for the plan the optimizer provides, you can choose one of the following values:		
	SHORT provides basic information about the optimizer plan.		
	LONG provides detailed information about the optimizer plan.		
	GRAPHICAL provides a tree diagram of the query. You can click on a node in the plan diagram to see details about that part of the query.		
	GRAPHICALWITHSTATISTICS provides a tree diagram of the query, as well as statistics which indicate the resources used by the part of the query that is selected. The UltraLite plan treats this value the same as the GRAPHICAL value.		

NONE means no optimizer information can be accessed. This parameter is deprecated.

The plan is computed only when you click the Plan tab.

You can also specify the plan type on the Plan tab of the Options dialog.

ISQL_PRINT_RESULT_SET option [ISQL]

Function	This option specifies which result set(s) are printed when a .SQL file is run.
	The ISQL_PRINT_RESULT_SET option takes effect only when you run DBISQL within a command window (for example, using the -nogui option).
Allowed values	LAST, ALL, NONE
Default	LAST
Description	This option allows you to specify which result set(s) are printed when a .SQL file is run. It is remembered on a per-machine basis (not per-user), and is case-insensitive.
	You can choose one of the following print options:
	LAST prints the result set from the last statement in the file.
	ALL prints result sets from each statement in the file which returns a result set.
	NONE does not print any result sets.
	Although this option has no affect when Interactive SQL is running in windowed mode, you can still view and set the option in windowed mode. Choose Tools ➤ Options, then select the appropriate action in the Console Mode box on the Results page. Remember to choose the Make Permanent option if you want to set this option for all Interactive SQL sessions. Otherwise the change will be discarded when you exit Interactive SQL.

ISQL_QUOTE option [ISQL]

Function	Controls the default string that begins and ends all strings in data exported to ASCII files.
Allowed values	String
Default	A single apostrophe (')
Description	Controls the default string that begins and ends all strings in data exported to ASCII files. If an OUTPUT statement does not contain a QUOTE clause, this value is used by default.

Example	٠	To change the default string that begins and ends all strings to a double
		quote character.

SET TEMPORARY OPTION ISQL_QUOTE='"'; SELECT emp_lname, emp_fname FROM employee WHERE emp_id < 150; OUTPUT TO c:\employee.txt FORMAT ASCII

This code places the following data in *employee.txt*:

"Whitney", "Fran"

"Cobb","Matthew"

"Chin","Philip"

"Jordan","Julie"

The separator characters (in this case, commas) depend on the ISQL_FIELD_SEPARATOR setting.

JAVA_HEAP_SIZE option [database]

Function	To limit the memory used by Java applications for a connection.	
Allowed values	Integer	
Scope	Can be set for an individual connection or for the PUBLIC group. Takes effect immediately. DBA permissions are required to set this option for <i>any</i> connection.	
Default	1 000 000	
Description	This option sets the maximum size (in bytes) of the memory that is allocated to Java applications on a per-connection basis. Per-connection memory allocations typically consist of the user's working set of allocated Java variables and Java application stack space.	
	While a Java application is executing on a connection, the per-connection allocations come out of the fixed cache of the database server, so it is important that a run-away Java application is prevented from using up too much memory.	

JAVA_INPUT_OUTPUT option [database]

Function	To enable file access from Java in the database
Allowed values	ON, OFF
	Supported on NT only.
Scope	DBA authority required.

Default	OFF
See also	"Security management for Java" [ASA Programming Guide, page 96]
Description	By default, the java.io package is only partially supported. In particular, classes that handle file access are disabled. If JAVA_INPUT_OUTPUT is set to ON , Java file access classes in java.io are enabled.

JAVA_NAMESPACE_SIZE option [database]

Function	To limit the memory used by Java applications for a connection.
Allowed values	Integer
Scope	Can be set only for the PUBLIC group. Takes effect immediately.
Default	4 000 000
Description	This option sets the maximum size (in bytes) of the memory that is allocated to Java applications on a per-database basis.
	Per-database memory allocations include Java class definitions. Because class definitions are effectively read-only, they are shared between connections. Consequently, their allocations come right out of the fixed cache, and this option sets a limit on the size of these allocations.

LOGIN_MODE option [database]

Function	Controls the use of integrated logins for the database.
Allowed values	Standard, Mixed, or Integrated
Scope	Can be set only for the PUBLIC group. Takes effect immediately.
Default	Standard
Description	This option specifies whether integrated logins are permitted. The following values are accepted (the values are case insensitive):
	• Standard This is the default setting, which does not permit integrated logins. An error occurs if an integrated login connection is attempted.
	• Mixed With this setting, both integrated logins and standard logins are allowed.

• **Integrated** With this setting, all logins to the database must be made using integrated logins.

Caution

Setting the LOGIN_MODE database option to Integrated restricts connections to only those users who have been granted an integrated login mapping. Attempting to connect with a user ID and password generates an error. The only exceptions to this are users with DBA authority (full administrative rights).

For more information on integrated logins see "Using integrated logins" on page 85.

LOGIN_PROCEDURE option [database]

Function	A login procedure that sets connection compatibility options at startup. By default the procedure calls the sp_login_environment procedure to determine which options to set.
Allowed values	String
Scope	DBA authority required.
Default	sp_login_environment
Description	This login procedure calls the sp_login_environment procedure at run time to determine the database connection settings.
	You can customize the default database option settings by creating a new procedure and setting LOGIN_PROCEDURE to call the new procedure. You should not edit either sp_login_environment or sp_tsql_environment.
Examples	 The following example shows how you can disallow a connection by signaling the INVALID_LOGON error.
	<pre>create procedure DBA.login_check() begin declare INVALID_LOGON exception for sqlstate '28000'; // Allow a maximum of 3 concurrent connections if(db_property('ConnCount') > 3) then signal INVALID_LOGON; else call sp_login_environment; end if; end go grant execute on DBA.login_check to PUBLIC go set option PUBLIC.Login_procedure='DBA.login_check' go</pre>

For an alternate way to disallow connections, see "RAISERROR statement [T-SQL]" [ASA SQL Reference, page 515].

The following example shows how you can block connection attempts if the number of failed connections for a user exceeds 3 within a 30 minute period. All blocked attempts during the block out period receive an invalid password error and are logged as failures. The log is kept long enough for a DBA to analyze it.

```
create table DBA.ConnectionFailure(
    pk int primary key default autoincrement,
    user_name char(128) not null,
    tm timestamp not null default current timestamp
)
go
create index ConnFailTime on DBA.ConnectionFailure(
    user_name, tm )
go
```

```
create event ConnFail type ConnectFailed
handler
begin
    declare usr char(128);
    set usr = event_parameter( 'User' );
    // Put a limit on the number of failures logged.
    if (select count(*) from DBA.ConnectionFailure
        where user_name = usr
        and tm >= dateadd( minute, -30,
            current timestamp )) < 20 then
        insert into DBA.ConnectionFailure( user_name )
            values( usr );
        commit;
        // Delete failures older than 7 days
        delete DBA.ConnectionFailure
        where user_name = usr
        and tm < dateadd( day, -7, current timestamp );
        commit;
    end if;
end
go
create procedure DBA.login_check()
begin
    declare usr char(128);
    declare INVALID_LOGON exception for sqlstate '28000';
    set usr = connection_property( 'Userid' );
    // Block connection attempts from this user
    // if 3 or more failed connection attempts have occurred
    // within the past 30 minutes.
    if (select count(*) from DBA.ConnectionFailure
        where user_name = usr
        and tm >= dateadd( minute, -30,
            current timestamp ) ) >= 3 then
        signal INVALID_LOGON;
    else
        call sp_login_environment;
    end if;
end
go
grant execute on DBA.login_check to PUBLIC
qo
set option PUBLIC.Login_procedure='DBA.login_check'
qo
```

MAX_CURSOR_COUNT option [database]

Function

A resource governor to limit the maximum number of cursors that a connection can use at once.

Allowed values

Integer

Scope	Can be set for an individual connection or for the PUBLIC group. Takes effect immediately. DBA permissions are required to set this option for <i>any</i> connection.
Default	50
Description	This resource governor allows a DBA to limit the number of cursors per connection that a user can use. If an operation would exceed the limit for a connection, an error is generated, indicating that the governor for the resource has been exceeded.
	If a connection executes a stored procedure, that procedure is executed under the permissions of the procedure owner. However, the resources used by the procedure are assigned to the current connection.
	\Leftrightarrow You can remove resource limits by setting the option to 0 (zero).

MAX_HASH_SIZE option [database]

Function	Specify the default maximum hash size for new indexes.
Allowed values	Integer, from 2 to 64 inclusive.
Scope	Can be set for the PUBLIC group only. Takes effect immediately. DBA permissions are required to set this option.
Default	10
See also	"CREATE INDEX statement" [ASA SQL Reference, page 319]
	"CREATE TABLE statement" [ASA SQL Reference, page 361]
Description	This option controls the default hash size for indexes. This option is deprecated.

MAX_PLANS_CACHED option [database]

Specify the maximum number of execution plans to be stored in a cache.
Integer
Can be set for an individual connection or for the PUBLIC group. Takes effect immediately. DBA permissions are required to set this option for the PUBLIC group.
20
"Access plan caching" [ASA SQL User's Guide, page 375]
"Connection-level properties" on page 647
This option specifies the maximum number of plans cached for each

connection. The optimizer caches the execution plan for queries, INSERT, UPDATE, and DELETE statements that are performed inside stored procedures, functions, and triggers. After a statement in a stored procedure, stored function, or trigger is executed several times by a connection, the optimizer builds a reusable plan for the statement.

Reusable plans do not use the values of host variables for selectivity estimation or rewrite optimizations. As a result of this, the reusable plan can have a higher cost than if the statement was re-optimized. When the cost of the reusable plan is close to the best observed cost for a statement, the optimizer adds the plan to the plan cache.

The cache is cleared when you execute statements, such as CREATE TABLE and DROP TABLE, that modify the table schema. Statements that reference declared temporary tables are not cached.

Setting this option to 0 disables plan caching.

MAX_RECURSIVE_ITERATIONS [database]

Function	Limits the maximum number of iterations a recursive common table expression can make.
Allowed values	Integer, from 1 to [UNDER CONSTRUCTION]
Scope	Can be set for an individual connection or for the PUBLIC group. Takes effect immediately. DBA permissions are required to set this option for the PUBLIC group.
Default	100
See also	"Common Table Expressions" [ASA SQL User's Guide, page 305]
Description	Computation of a recursive common table expressions aborts and an error is generated if they fail to terminate within the specified number of iterations. Recursive subqueries often increase geometrically in the amount of resources required for each additional iteration. Set this option so as to limit the amount of time and resources that will be consumed before infinite recursion is detected, yet permit your recursive common table expressions to work as intended.

MAX_WORK_TABLE_HASH_SIZE option [database]

Function	Specify the maximum hash size used during query optimization for internal temporary tables.
Allowed values	Integer, from 2 to 64 inclusive.
Scope	Can be set for the PUBLIC group only. Takes effect immediately. DBA

permissions are required to set this option.Default20See also"Use of work tables in query processing" [ASA SQL User's Guide, page 185].DescriptionIn general, you should not need to use this option as the query optimizer
allocates hash sizes for the internal temporary tables based on the data
distribution within the table. The option allows you to override the optimizer
behavior to either increase the maximum hash size it can use beyond 20, or
to restrict it further. This option is deprecated.

MAX_STATEMENT_COUNT option [database]

Function	A resource governor to limit the maximum number of prepared statements that a connection can use at once.
Allowed values	Integer >=0
Scope	Can be set for an individual connection or for the PUBLIC group. Takes effect immediately. DBA permissions are required to set this option for <i>any</i> connection.
Default	50
Description	This resource governor allows a DBA to limit the number of prepared statements per connection a user can use. If an operation would exceed the limit for a connection, an error is generated, indicating that the governor for the resource has been exceeded.
	If a connection executes a stored procedure, that procedure is executed under the permissions of the procedure owner. However, the resources used by the procedure are assigned to the current connection.
	rightarrow You can remove resource limits by setting the option to 0 (zero).

MIN_PASSWORD_LENGTH option [database]

Function	Sets the minimum length for new passwords in the database.
Allowed values	Integer, greater than or equal to zero.
	The value is in bytes. For single-byte character sets, this is the same as the number of characters.
Scope	Can be set for the PUBLIC group. Takes effect immediately. DBA permissions are required to set this option.
Default	0 characters
Description	This option allows the database administrator to impose a minimum length

on all new passwords for greater security. Existing passwords are not affected.

• Set the minimum length for new passwords to 6 bytes.

SET OPTION PUBLIC.MIN_PASSWORD_LENGTH = 6

MIN_TABLE_SIZE_FOR_HISTOGRAM option [database]

Example

Function	Specify the minimum table size for which histograms are created.
Allowed values	Integer, greater than or equal to zero.
Scope	Can be set for the PUBLIC group only.
Default	1 000 rows
See also	"Optimizer estimates" [ASA SQL User's Guide, page 369]
Description	The optimizer uses histograms to estimate the selectivity of search conditions. For small tables, a histogram does not significantly improve the optimizer's ability to choose an efficient plan. This option lets you specify the minimum number of rows a table must have in order for the optimizer to create a histogram for columns in the table.
	You can change the setting to build histograms for smaller tables, or to build them only for very large tables. This option is ignored by the CREATE STATISTICS statement: when a CREATE STATISTICS statement is executed, a histogram is created regardless of the number of rows in the table.

NEAREST_CENTURY option [compatibility]

Function	Controls the interpretation of two-digit years in string-to-date conversions.
Allowed values	Integer, between 0 and 100 inclusive.
Default	50 for databases created with Version 6 or later.
	0 for databases created with Version 5.5 or earlier.
Description	This option controls the handling of two-digit years when converting from strings to dates or timestamps.
	The NEAREST_CENTURY setting is a numeric value that acts as a rollover point. Two digit years less than the value are converted to 20yy, while years greater than or equal to the value are converted to 19yy.
	The historical Adaptive Server Anywhere behavior is to add 1900 to the year. Adaptive Server Enterprise behavior is to use the nearest century, so any year where value <i>yy</i> is less than 50, the year is set to 20yy.

NON_KEYWORDS option [compatibility]

Function	Turns off individual keywords, allowing their use as identifiers.
Allowed values	String
Default	The empty string
See also	"Keywords" [ASA SQL Reference, page 4]
Description	This option turns off individual keywords or all keywords introduced since a specific release of the product. This provides a way of ensuring that applications created with older versions of the product are not broken by new keywords. If you have an identifier in your database that is now a keyword, you can either add double quotes around the identifier in all applications or scripts, or turn off the keyword using the NON_KEYWORDS option.
	In addition to specifying individual keywords, you can turn off all keywords since a specified release, using one of the following special values in the list of keywords:
	<pre>keywords_4_0_d, keywords_4_0_c, keywords_4_0_b, keywords_4_0_a, keywords_4_0, keywords_5_0_01, keywords_5_0</pre>
	The following statement prevents TRUNCATE and SYNCHRONIZE from being recognized as keywords:
	SET OPTION NON_KEYWORDS = 'TRUNCATE, SYNCHRONIZE'
	The following statement prevents all keywords introduced since release 4.0d from being recognized as keywords:
	SET OPTION NON_KEYWORDS = 'keywords_4_0_d'
	Each new setting of this option replaces the previous setting. The following statement clears all previous settings.
	SET OPTION NON_KEYWORDS =

A side-effect of this options is that SQL statements that use a turned off keyword cannot be used: they produce a syntax error.

NULLS option [ISQL]

Function	Specifies how NULL values in the database are displayed.
Allowed values	String
Default	(NULL)

Description	Set this according to y	our preference.
-------------	-------------------------	-----------------

ON_CHARSET_CONVERSION_FAILURE option [database]

Function	Controls what happens if an error is encountered during character conversion.
Allowed values	String. See below for allowed values.
Default	IGNORE
Description	Controls what happens if an error is encountered during character conversion, as follows:
	• IGNORE Errors and warnings do not appear.
	• WARNING Reports substitutions and illegal characters as warnings. Illegal characters are not translated.
	• ERROR Reports substitutions and illegal characters as errors.
	Single-byte to single-byte converters are not able to report substitutions and illegal characters, and must be set to IGNORE.
ON_ERROR option	n [ISQL]
Function	Controls what happens if an error is encountered while reading statements from a command file.

Allowed values *String*. See below for allowed values.

Default PROMPT

Description Controls what happens if an error is encountered while reading statements from a command file, as follows:

- **STOP** Interactive SQL stops reading statements from the file and returns to the SQL Statements pane for input.
- **PROMPT** Interactive SQL prompts the user to see if the user wishes to continue.
- ♦ CONTINUE The error is ignored and Interactive SQL continues reading statements from the command file. The INPUT statement continues with the next row, skipping the row that caused the error.
- **EXIT** Interactive SQL terminates.
- **NOTIFY_CONTINUE** The error appears in a message box with a single button. Execution continues once the button is clicked.

- **NOTIFY_STOP** The error appears in a message box with a single button. Execution of the script stops once the button is clicked.
- **NOTIFY_EXIT** The error appears in a message box with a single button. Interactive SQL terminates once the button is clicked.

ON_TSQL_ERROR option [compatibility]

Function	Controls error-handling in stored procedures.
Allowed values	String. See below for allowed values.
Default	CONDITIONAL
See also	"CREATE PROCEDURE statement" [ASA SQL Reference, page 324]
	"CREATE PROCEDURE statement [T-SQL]" [ASA SQL Reference, page 332]
	"Transact-SQL procedure language overview" [ASA SQL User's Guide, page 463]
Description	This option controls error handling in stored procedures.
	• STOP Stop execution immediately upon finding an error.
	• CONDITIONAL If the procedure uses ON EXCEPTION RESUME, and the statement following the error handles the error, continue, otherwise exit.
	• CONTINUE Continue execution, regardless of the following statement. If there are multiple errors, the first error encountered in the stored procedure is returned.
	Both CONDITIONAL and CONTINUE settings for On_TSQL_Error are used for Adaptive Server Enterprise compatibility, with CONTINUE most closely simulating ASE behavior. The CONDITIONAL setting is recommended, particularly when developing new Transact-SQL stored procedures, as it allows errors to be reported earlier.
	When this option is set to STOP or CONTINUE, it supercedes the setting of the CONTINUE_AFTER_RAISERROR option. However, when this option is set to CONDITIONAL (the default), behavior following a RAISERROR statement is determined by the setting of the CONTINUE_AFTER_RAISERROR option.

OPTIMISTIC_WAIT_FOR_COMMIT

Function	$Controls \ locking \ behavior \ for \ WAIT_FOR_COMMIT \ option.$
Allowed values	ON, OFF

Default	OFF
Description	By default, OPTIMISTIC_WAIT_FOR_COMMIT is OFF.
	Locking behavior is changed as follows when both OPTIMISTIC_WAIT_FOR_COMMIT and WAIT_FOR_COMMIT are ON :
	• No locks are placed on primary key rows when adding a foreign key row without a matching primary key row.
	• If a transaction adds a primary row, it will be allowed to commit only if the transaction has exclusive locks on all foreign rows that reference the primary row at the time the primary row is added.
	This feature is for use in migrating 5.x applications to 8.x by mimicking 5.x locking behavior when transactions add foreign rows before primary rows (as long as no two transactions concurrently add foreign rows with the same key value).
	Note that this option is not recommended for general use as there are a number of situations in which transactions will not be allowed to commit, including:
	• If transactions concurrently add foreign rows with the same key value when the primary row does not exist, at most 1 of the transactions will be allowed to commit (the one that adds the corresponding primary row).
	• If a transaction deletes a primary row and then adds it back, it likely will not be allowed to commit (unless it has somehow obtained exclusive locks on all of the matching foreign rows).
OPTIMIZATION_G	DAL option [database]
Function	Determines whether query processing is optimized towards returning the first row quickly, or minimizing the cost of returning the complete result set.
Allowed values	first-row, all-rows
Default	all-rows
Description	The OPTIMIZATION_GOAL option controls whether Adaptive Server Anywhere optimizes SQL "data manipulation language (DML)" [<i>ASA Getting</i> <i>Started</i> , page 157] statements for response time or total resource consumption.
	If the option is set to all-rows (the default), then Adaptive Server Anywhere optimizes a query so as to choose an access plan with the minimal estimated total retrieval time. Setting OPTIMIZATION_GOAL to all-rows may be appropriate for applications that intend to process the entire result set, such as PowerBuilder DataWindow applications. A setting of all-rows is also

appropriate for insensitive (ODBC static) cursors since the entire result is materialized when the cursor is opened. It may also be appropriate for scroll (ODBC keyset-driven) cursors, since the intent of such a cursor is to permit scrolling through the result set.

If the option is set to **first-row**, Adaptive Server Anywhere chooses an access plan that is intended to reduce the time to fetch the first row of the query's result, possibly at the expense of total retrieval time. In particular, the Adaptive Server Anywhere optimizer will typically avoid, if possible, access plans that require the materialization of results in order to reduce the time to return the first row. With this setting, the optimizer favors access plans that utilize an index to satisfy a query's ORDER BY clause, rather than plans that require an explicit sorting operation.

You can use the FASTFIRSTROW table hint in a query's FROM clause to set the optimization goal for a specific query to **first-row**, without having to change the OPTIMIZATION_GOAL setting.

For more information about using the FASTFIRSTROW table hint, see "FROM clause" [*ASA SQL Reference*, page 445].

OUTPUT_FORMAT option [ISQL]

Function	Sets the output format for the data retrieved by the SELECT statement and redirected into a file, or output using the OUTPUT statement.
Allowed values	String. See below for allowed values.
Default	ASCII
Description	The valid output formats are:
	◆ ASCII The output is an ASCII format file with one row per line in the file. All values are separated by commas, and strings are enclosed in apostrophes (single quotes). The delimiter and quote strings can be changed using the DELIMITED BY and QUOTE clauses. If ALL is specified in the QUOTE clause, then all values (not just strings) will be quoted.
	Three other special sequences are also used. The two characters \n represent a newline character; \represents a single backslash character, and the sequence \xDD represents the character with hexadecimal code DD.
	◆ DBASEII The output is a dBASE II format file with the column definitions at the top of the file. Note that a maximum of 32 columns can be output. Column names are truncated to 11 characters, and each row of data in each column is truncated to 255 characters.

- ◆ **DBASEIII** The output is a dBASE III format file with the column definitions at the top of the file. Note that a maximum of 128 columns can be output. Column names are truncated to 11 characters, and each row of data in each column is truncated to 255 characters.
- **EXCEL** The output is an Excel 2.1 worksheet. The first row of the worksheet contains column labels (or names if there are no labels defined). Subsequent worksheet rows contain the actual table data.
- FIXED The output is fixed format, with each column having a fixed width. The width for each column can be specified using the COLUMN WIDTH clause. If this clause is omitted, the width for each column is computed from the data type for the column, and is large enough to hold any value of that data type. No column headings are output in this format.
- ◆ FOXPRO The output is a FoxPro format file (the FoxPro memo field is different than the dBASE memo field) with the column definitions at the top of the file. Note that a maximum of 128 columns can be output. Column names are truncated to 11 characters, and each row of data in each column is truncated to 255 characters.
- **HTML** The output is in the Hyper Text Markup Language format.
- ◆ LOTUS The output is a Lotus WKS format worksheet. Column names will be put as the first row in the worksheet. Note that there are certain restrictions on the maximum size of Lotus WKS format worksheets that other software (such as Lotus 1-2-3) can load. There is no limit to the size of file Interactive SQL can produce.
- **SQL** The output is an Interactive SQL INPUT statement required to recreate the information in the table.
- ◆ XML The output is an XML file encoded in UTF-8 and containing an embedded DTD. Binary values are encoded in CDATA blocks with the binary data rendered as 2-hex-digit strings.

OUTPUT_LENGTH option [ISQL]

Function	Controls the length used when Interactive SQL exports information to an external file.
Allowed values	Integer
Default	0 (no truncation)
Description	This option controls the length used when Interactive SQL exports information to an external file (using output redirection with the OUTPUT statement). This option affects only ASCII, HTML, and SQL output formats.

OUTPUT_NULLS option [ISQL]

Function	Controls the way NULL values appear in result sets.
Allowed values	String
Default	'NULL'
Description	This option controls the way NULL values appear in result sets. Every time a NULL value is found in the result set, the string from this option is returned instead. This setting applies to data displayed in Interactive SQL on the Results tab in the Results pane as well as to data in output files generated by the OUTPUT statement. This option affects only ASCII, HTML, and SQL output formats.

PERCENT_AS_COMMENT option [compatibility]

Function	Controls the interpretation of the percent character.
Allowed values	ON, OFF
Default	ON
Description	It is recommended that you not use % as a comment marker.
	Versions of this product before Version 6 treated the percent character (%) in SQL statements exclusively as a comment delimiter. Since Version 5, alternative comment markers such as //, /* */, and – (double dash) have been available. The double-dash style is the SQL/92 comment delimiter.
	Adaptive Server Enterprise treats % as a modulo operator and does not support the Adaptive Server Anywhere mod function. Writing a statement that works in both environments and performs a modulo operation was previously impossible.
	The PERCENT_AS_COMMENT option controls the meaning of %. The default setting is ON for backwards compatibility. You can set the option to OFF for compatibility with Adaptive Server Enterprise.
	Procedures, triggers and views that were created with %-style comments are converted to double-dash comments when they are stored in the catalog.
	Existing procedures must be recreated before changing option Any existing procedures that contain %-style comments must be recreated before you change the option setting; otherwise, the procedures will fail to load.

PINNED_CURSOR_PERCENT_OF_CACHE option [database]

Function	Specifies how much of the cache can be used for pinning cursors.
Allowed values	Integer, between 0-100
Scope	Can be set for the PUBLIC group.
Default	10
Description	The server uses pages of virtual memory for the data structures needed to implement cursors. These pages are kept locked in memory between fetch requests so they are readily available when the next fetch request arrives.
	To prevent these pages from occupying too much of the cache in low memory environments, a limit is placed on the percentage of the cache allowed to be used for pinning cursors. You can use the PINNED_CURSOR_PERCENT_OF_CACHE option to adjust this limit.
	The option value is specified as a percentage from 0 to 100, with a default of 10. Setting the option to 0 means that cursor pages will not be pinned between fetch requests.

PRECISION option [database]

Function	Specifies the maximum number of digits in the result of any decimal arithmetic.
Allowed values	Integer, between 0 and 127 inclusive.
Scope	Can be set for an individual connection or for the PUBLIC group. Takes effect immediately.
Default	30
Description	Precision is the total number of digits to the left and right of the decimal point. The "SCALE option [database]" on page 624 specifies the minimum number of digits after the decimal point when an arithmetic result is truncated to the maximum PRECISION.
	Multiplication, division, addition, subtraction, and aggregate functions can all have results that exceed the maximum precision.
	For example, when a DECIMAL(8,2) is multiplied with a DECIMAL(9,2), the result could require a DECIMAL(17,4). If PRECISION is 15, only 15 digits will be kept in the result. If SCALE is 4, the result will be a DECIMAL(15,4). If SCALE is 2, the result will be a DECIMAL(15,2). In both cases, there is a possibility of overflow.

PREFETCH option [database]

Function	The PREFETCH option acts as a toggle allowing you to turn fetching on and off.
Allowed values	ON, OFF
Scope	Can be set for an individual connection or for the PUBLIC group. Takes effect immediately.
Default	ON
Description	This option controls whether rows are fetched to the client side in advance of being made available to the client application. Fetching a number of rows at a time, even when the client application requests rows one at a time (for example, when looping over the rows of a cursor) cuts down on response time and improves overall throughput by cutting down the number of requests to the database.
	The setting of PREFETCH is ignored by Open Client and JDBC connections.
	For more information, see "Prefetching rows" [<i>ASA Programming Guide</i> , page 42] and "DisableMultiRowFetch connection parameter [DMRF]" on page 187.
PRESERVE_SOUR	CE_FORMAT option [database]
Function	Controls whether the original source definition of procedures, triggers,

Function	Controls whether the original source definition of procedures, triggers, views, and event handlers is saved in system files. If saved, it is saved in the column source in SYSTABLE, SYSPROCEDURE, SYSTRIGGER, and SYSEVENT.
Allowed values	ON, OFF
Default	ON
Description	When PRESERVE_SOURCE_FORMAT is ON, the server saves the formatted source from CREATE and ALTER statements on procedures, views, triggers, and events, and puts it in the appropriate system table's source column.
	Unformatted source text is stored in the same system tables, in the columns proc_defn, trigger_defn, and view_defn. However, these definitions are not easy to read in Sybase Central. The formatted source column allows you to view the definitions with the spacing, comments, and case that you want.
	This option can be turned off to reduce space used to save object definitions

in the database. The option can be set only for the user PUBLIC.

PREVENT_ARTICLE_PKEY_UPDATE option [database]

Function	Controls updates to the primary key columns of tables involved in publications.
Allowed values	ON, OFF
Default	ON
Description	Setting this option to ON disallows updates to the primary key columns of tables that are part of a publication. This option helps ensure data integrity, especially in a replication and synchronization environment

QUALIFY_OWNERS option [replication]

Function	Controls whether SQL statements being replicated by SQL Remote should use qualified object names.
Allowed values	ON, OFF
Default	The default in Adaptive Server Anywhere is ON .
	The default in Adaptive Server Enterprise is OFF .
Description	Qualifying owners in Adaptive Server Enterprise setups is rarely needed because it is common for objects to be owned by dbo . When qualification is not needed in Adaptive Server Anywhere setups, messages will be slightly smaller with the option OFF.

QUERY_PLAN_ON_OPEN option [compatibility]

Function	Controls whether a plan is returned when a cursor is opened.
Allowed values	ON, OFF
Default	OFF
Description	In early versions of the software, each time an OPEN was done on a cursor, the server would return in the SQLCA sqlerrmc field a string representing the query plan (limited to 70 bytes). A more complete description can be obtained using the EXPLAIN statement or the PLAN function. For this reason, computing and returning the query plan on an OPEN is needed only for compatibility with old applications. The QUERY_PLAN_ON_OPEN option controls whether the plan is returned on an OPEN. By default, the setting is OFF .

QUOTE_ALL_IDENTIFIERS option [replication]

Function	Controls whether SQL statements being replicated by SQL Remote should use quoted identifiers
Allowed values	ON, OFF
Default	OFF
Description	When this option is off, <i>dbremote</i> quotes identifiers that require quotes by Adaptive Server Anywhere (as it has always done) and <i>ssremote</i> does not quote any identifiers.
	When the option is on, all identifiers are quoted.

QUOTED_IDENTIFIER option [compatibility]

Function	Controls the interpretation of strings that are enclosed in double quotes.
Allowed values	ON, OFF
Default	ON
	OFF for Open Client and JDBC connections
Description	This option controls whether strings that are enclosed in double quotes are interpreted as identifiers (ON) or as literal strings (OFF). The QUOTED_IDENTIFIER option is included for Transact-SQL compatibility.
	Note that Interactive SQL automatically executes the equivalent of
	SET TEMPORARY OPTION QUOTED_IDENTIFIER = ON
	when it connects to a database. TEMPORARY options apply only to the current connection, and last only until the connection is closed.
	For more information, see "Setting options for Transact-SQL compatibility" [ASA SQL User's Guide, page 451].

READ_PAST_DELETED option [database]

Function	Controls server behavior on uncommitted deletes at isolation levels 1 and 2
Allowed values	ON, OFF
Default	ON
Description	If READ_PAST_DELETED is ON (the default), sequential scans at isolation 1 and 2 will skip uncommitted deleted rows. If OFF , sequential scans will block on uncommitted deleted rows at isolation levels 1 and 2

(until the deleting transaction commits or rolls back). This option changes server behaviour at isolation levels 1 and 2.

For most purposes, this option should be left **ON**. If set to **OFF**, the blocking behaviour depends on the plan chosen by the optimizer (if there is an index that could possibly be used).

RECOVERY_TIME option [database]

Function	Sets the maximum length of time, in minutes, that the database server will take to recover from system failure.
Allowed values	Integer, in minutes
Scope	Can be set only for the PUBLIC group. Takes effect when server is restarted.
Default	2
Description	This option is used with the "CHECKPOINT_TIME option [database]" on page 582 to decide when checkpoints should be done.
	Adaptive Server Anywhere uses a heuristic to estimate the recovery time based on the operations that have been performed since the last checkpoint. Thus, the recovery time is not exact.
	For more information, see "The automatic recovery process" on page 363.

REPLICATE_ALL option [replication]

Function	Allows an entire database to act as a primary site in a Replication Server setup.
Allowed values	ON, OFF
Default	OFF
Description	This option is used by the LTM only. When it is set to ON , the entire database is set to act as a primary site in a Replication Server installation. All changes to the database are sent to Replication Server by the LTM.

REPLICATION_ERROR option [replication]

Function	For SQL Remote, allows you to specify a stored procedure to be called by the Message Agent when a SQL error occurs.
Allowed values	Stored procedure name
Default	No procedure

Description	For SQL Remote, the REPLICATION_ERROR option allows you to specify a stored procedure to be called by the Message Agent when a SQL error occurs. By default no procedure is called.
	The procedure must have a single argument of type CHAR, VARCHAR, or LONG VARCHAR. The procedure is called once with the SQL error message and once with the SQL statement that causes the error.
	Although the option allows you to track and monitor SQL errors in replication, you must still design them out of your setup; this option is not intended to resolve such errors.
REPLICATION_ER	ROR_PIECE option [replication]

Function	For SQL Remote, the REPLICATION_ERROR_PIECE option works in conjunction with the REPLICATION_ERROR option to allow you to specify a long varchar stored procedure to be called by the Message Agent when a SQL error occurs.
	This is a compatibility feature most useful for databases created with older versions of Adaptive Server Enterprise which do not support long varchar.
Allowed values	Stored procedure name
Default	No procedure
Description	If an error occurs and REPLICATION_ERROR is defined then the REPLICATION_ERROR procedure is called with the full error string.
	If REPLICATION_ERROR and REPLICATION_ERROR _PIECE are both defined then the error is broken up into varchar pieces. REPLICATION_ERROR is called with the first piece and REPLICATION_ERROR_PIECE is called repeatedly with the remaining pieces.

RETURN_DATE_TIME_AS_STRING option [database]

Function	To control how a date, time, or timestamp value is passed to the client application when queried.
Allowed values	ON, OFF
Scope	Can be set as a temporary option only, for the duration of the current connection.
Default	OFF
See also	"DATE_FORMAT option [compatibility]" on page 587
	"TIME_FORMAT option [compatibility]" on page 628

"TIMESTAMP_FORMAT option [compatibility]" on page 629

Description This option indicates whether date, time, and timestamp values are returned to applications as a date or time datatype or as a string.

When this option is set to ON, the engine converts the date, time, or timestamp value to a string before it is sent to the client in order to preserve the TIMESTAMP_FORMAT, DATE_FORMAT, or TIME_FORMAT option setting.

Sybase Central and Interactive SQL automatically turn the RETURN_DATE_TIME_AS_STRING option ON.

RETURN_JAVA_AS_STRING option [database]

Function	To control how a Java object is passed to the client application when queried.
Allowed values	ON, OFF
Scope	Can be set as a temporary option only, for the duration of the current connection.
Default	OFF
Description	The option indicates how a Java object is returned to applications connecting over Open Client or jConnect.
	By default, a Java object is returned over TDS as a Sun serialization of the object. The client, or receiver of the object serialization, is responsible for deserializing the object into an instance.
	If RETURN_JAVA_AS_STRING is set to ON , the object is first converted to an instance of java.lang.String using the toString () method, and the String object is returned to the client.
RI_TRIGGER_TI	ME option [compatibility]
Function	Controls the relative timing of referential integrity checks and trigger actions.
Allowed values	BEFORE, AFTER
Scope	Can be set for the PUBLIC option only. DBA authority is required to set this option.
Default	AFTER
Description	The option can be set to either BEFORE or AFTER. When it's set to AFTER, referential integrity actions are executed after the UPDATE or DELETE.

Only the PUBLIC setting can be used; any other setting is ignored.

ROW_COUNTS option [database]

Function	Specifies whether the database will always count the number of rows in a query when it is opened.
Allowed values	ON, OFF
Scope	Can be set for an individual connection or for the PUBLIC group. Takes effect immediately.
Default	OFF
Description	If this option is set to OFF , the row count is usually only an estimate. If this option is set to ON , the row count is always accurate.
	<i>Warning:</i> When ROW_COUNTS is set to ON , it may take significantly longer to execute queries. In fact, it will usually cause Adaptive Server Anywhere to execute the query twice, doubling the execution time.

SCALE option [database]

Function	Specifies the minimum number of digits after the decimal point when an arithmetic result is truncated to the maximum PRECISION.
Allowed values	Integer, between 0 and 127 inclusive.
Scope	Can be set for an individual connection or for the PUBLIC group. Takes effect immediately.
Default	6
Description	Multiplication, division, addition, subtraction, and aggregate functions can all have results that exceed the maximum precision. See "PRECISION option [database]" on page 617 for an example.

SORT_COLLATION option [database]

Function	Allows implicit use of the SORTKEY function on ORDER BY expressions.
Allowed values	INTERNAL, collation_name, or collation_id
Default	INTERNAL
See also	"SORTKEY function [String]" [ASA SQL Reference, page 184]
Description	When the value of this option is INTERNAL , the ORDERBY clause remains unchanged.

When the value of this option is set to a valid *collation name* or *collation ID*, any string expression in the ORDER BY clause is treated as if the SORTKEY function had been invoked.

Examples

Set the sort collation to binary:

set temporary option sort_collation='binary'

Having the sort collation set to binary transforms

SELECT name, id FROM product ORDER BY name, id

and

```
SELECT name, id
FROM product
ORDER BY 1,2
```

into

```
SELECT name, id
FROM product
ORDER BY SORTKEY(name, 'binary'), id
```

SQL_FLAGGER_ERROR_LEVEL option [compatibility]

Function	Controls the response to any SQL that is not part of a specified set of SQL/92.
Allowed values	E, I, F, or W
Default	W
Description	This option flags any SQL that is not part of a specified set of SQL/92 as an error.
	The allowed values of <i>level</i> are as follows:
	• E Flag syntax that is not entry-level SQL/92 syntax
	• I Flag syntax that is not intermediate-level SQL/92 syntax
	• F Flag syntax that is not full-SQL/92 syntax
	♦ W Allow all supported syntax
SQL FLAGGER V	VARNING LEVEL option [compatibility]

SQL_FLAGGER_WARNING_LEVEL Option [compatibility]

Function

Controls the response to any SQL that is not part of a specified set of SQL/92.

Allowed values	E, I, F, or W
Default	W
Description	This option flags any SQL that is not part of a specified set of SQL/92 as a warning.
	The allowed values of <i>level</i> are as follows:
	• E Flag syntax that is not entry-level SQL/92 syntax
	• I Flag syntax that is not intermediate-level SQL/92 syntax
	• F Flag syntax that is not full-SQL/92 syntax
	◆ ₩ Allow all supported syntax

STATISTICS option [ISQL]

Function	Controls whether the execution time is enabled.
Allowed values	Any non-negative integer
Default	7
See also	"ISQL_COMMAND_TIMING option [ISQL]" on page 597
	"ISQL_PLAN option [ISQL]" on page 599
Description	This option is an older version of the ISQL_PLAN option.
	When you set the STATISTICS option to 0, no information appears in the Messages pane after you execute a SQL statement. When you set the option to any other number, the execution time appears in the Messages pane after you execute a statement. The number you specify also becomes the default height (in lines) of the Messages pane.

STRING_RTRUNCATION option [compatibility]

Function	Determines whether an error is raised when an INSERT or UPDATE truncates a CHAR or VARCHAR string.
Allowed values	ON, OFF
Default	OFF
Description	If the truncated characters consist only of spaces, no exception is raised. The setting of ON corresponds to ANSI/ISO SQL/92 behavior. When it is set to OFF , the exception is not raised and the character string is silently truncated.

SUBSCRIBE_BY_REMOTE option [replication]

Function	Controls interpretation of NULL or empty-string SUBSCRIBE BY values.
Allowed values	ON, OFF
Default	ON
Description	When the option is set to ON , operations from remote databases on rows with a SUBSCRIBE BY value that is NULL or an empty string assume that the remote user is subscribed to the row. When it is set to OFF , the remote user is assumed not to be subscribed to the row.
SUBSUME_ROW_I	-OCKS option [database]
Function	Controls when the engine acquires individual row locks for a table.
Allowed values	ON, OFF
Default	ON
Description	If the SUBSUME_ROW_LOCKS option is ON (the default) then whenever a table t is locked exclusively with LOCK TABLE t IN EXCLUSIVE MODE, the engine will no longer acquire individual row locks for t.
	This can result in a significant performance improvement if extensive updates are made to <i>t</i> in a single transaction, especially if <i>t</i> is large relative to cache size. It also allows for atomic update operations that are larger than the lock table can currently handle ($> \sim 2-4m$ rows).
	When this option is <i>ON</i> , keyset cursors over a table locked in this fashion will return row changed warnings for every row in the cursor, if any row in the database has been modified. Note that the engine could turn an updateable cursor with an ORDER BY into a keyset cursor as a result.
SUPPRESS_TDS_I	DEBUGGING option [database]
Function	Determines whether TDS debugging information appears in the server window.
Allowed values	ON, OFF
Default	OFF
Description	When the server is started with the $-z$ option, debugging information appears in the server window, including debugging information about the TDS protocol.
	The SUPPRESS_TDS_DEBUGGING option restricts the debugging

information about TDS that appears in the server window. When this option is set to **OFF** (the default) TDS debugging information appears in the server window.

TDS_EMPTY_STRING_IS_NULL option [database]

Function	Controls whether empty strings are returned as NULL or a string containing one blank character for TDS connections.
Allowed values	ON, OFF
Default	OFF
Description	By default, this option is set to OFF and empty strings are returned as a string containing one blank character for TDS connections. When this option is set to ON , empty strings are returned as NULL strings for TDS connections. Non-TDS connections distinguish empty strings from NULL strings.

TIME_FORMAT option [compatibility]

Function	Sets the format for times retrieved from the database.
Allowed values	String, composed of the symbols listed below.
Scope	Can be set for an individual connection or for the PUBLIC group. Takes effect immediately.
Default	HH:NN:SS.SSS
Description	The format is a string using the following symbols:
	• hh Two digit hours (24 hour clock)
	• nn Two digit minutes
	• mm Two digit minutes if following a colon (as in hh:mm)
	• ss[.s] Two digit seconds plus optional fraction
	Each symbol is substituted with the appropriate data for the time that is being formatted. Any format symbol that represents character rather than digit output can be put in uppercase, which causes the substituted characters to also be in uppercase. For numbers, using mixed case in the format string suppresses leading zeros.

TIME_ZONE_ADJUSTMENT option [database]

Function Allows a connection's time zone adjustment to be modified.

Allowed values	Integer, (for example, 300), or
	Negative integer, enclosed in quotation marks (for example, '-300'), or
	<i>String</i> , representing a time in hours and minutes, preceded by + or -, enclosed in quotation marks (for example, '+5:00', or '-5:00').
Default	If the client is connecting via ESQL, ODBC, OLE DB, or ADO, the default value is set according to the client's time zone. If the client is connecting via jConnect or OpenClient, the default is based on the server's time zone.
See also	"Connection-level properties" on page 647
Description	The TIME_ZONE_ADJUSTMENT option value is the same value as that returned by connection_property('TimeZoneAdjustment'). The value represents the number of minutes that must be added to the Coordinated Universal Time (UTC) to display time local to the connection.

TIMESTAMP_FORMAT option [compatibility]

Function	Sets the format for timestamps that are retrieved from the database.
Allowed values	String, composed of the symbols listed below.
Scope	Can be set for an individual connection or for the PUBLIC group. Takes effect immediately.
Default	YYYY-MM-DD HH:NN:ss.SSS
	For Open Client and JDBC connections the default is set to YYYY-MM-DD HH:NN:SS.SSS.
Description	The format is a string using the following symbols:

Symbol	Description
уу	Two digit year
уууу	Four digit year
mm	Two digit month, or two digit minutes if following a colon (as in 'hh:mm')
mmm[m]	Character short form for months—as many characters as there are "m"s
dd	Two digit day of month
ddd[d]	Character short form for day of the week
hh	Two digit hours

Symbol	Description
nn	Two digit minutes
SS.SSSSSS	Seconds and fractions of a second, up to six decimal places. Not all platforms support timestamps to a precision of six places.
aa	am or pm (12 hour clock)
рр	pm if needed (12 hour clock)
f	Use French days and months (deprecated)

Each symbol is substituted with the appropriate data for the date that is being formatted.

For symbols that represent character data (such as mmm), you can control the case of the output as follows:

- Type the symbol in all upper case to have the format appear in all upper case. For example, MMM produces JAN.
- Type the symbol in all lower case to have the format appear in all lower case. For example, mmm produces jan.
- Type the symbol in mixed case to have Adaptive Server Anywhere choose the appropriate case for the language that is being used. For example, in English, typing Mmm produces May, while in French it produces mai.

For symbols that represent numeric data, you can control zero-padding with the case of the symbols:

- Type the symbol in same-case (such as MM or mm) to allow zero padding. For example, yyyy/mm/dd could produce 2002/01/01.
- Type the symbol in mixed case (such as Mm) to suppress zero padding. For example, yyyy/Mm/Dd could produce 2002/1/1.

TRUNCATE_DATE_VALUES option [database]

Function	To change the storage of time values within DATE data type values.
Allowed values	ON, OFF
Default	ON for databases created with Version 7 or later software, or upgraded to Version 7 or later.
	OFF for older databases

See also	"DATE_FORMAT option [compatibility]" on page 587
	"DATE data type [Date and Time]" [ASA SQL Reference, page 69]
Description	When the option is set to ON , columns or variables declared as DATE data types have no hours and minutes stored.
	When the option is set to OFF , DATE columns or variables have hours and minutes stored also. These time components are typically not displayed because of the DATE_FORMAT setting. However, exact comparisons between DATE values may unexpectedly fail because of non-matching time components.
TRUNCATE_TIMESTAMP_VALUES option [database]	

Function	Limits the resolution of timestamp values.
Allowed values	ON, OFF
Scope	Can be set for the PUBLIC group. This option should not be enabled for databases already containing timestamp data.
Default	OFF
Description	A TIMESTAMP value is precise to six decimal places in Adaptive Server Anywhere. However, to maintain compatibility with other software, which may truncate the TIMESTAMP value to three decimal places, you can set the TRUNCATE_TIMESTAMP_VALUES option to ON to limit the number of decimal places Adaptive Server Anywhere stores. The DEFAULT_TIMESTAMP_INCREMENT option determines the number of decimal places to which the TIMESTAMP value is truncated.
Example	Setting the DEFAULT_TIMESTAMP_INCREMENT option to 100 000 causes truncation after the first decimal place in the seconds component, allowing a value such as '2000/12/05 10:50:53:700' to be stored.

TRUNCATE_WITH_AUTO_COMMIT option [database]

Function	To speed up TRUNCATE TABLE statements
Allowed values	ON, OFF
Scope	Can be set only for the PUBLIC group.
Default	ON
See also	"TRUNCATE TABLE statement" [ASA SQL Reference, page 584]
Description	If TRUNCATE_WITH_AUTO_COMMIT is set to ON , then a COMMIT is executed both before and after the TRUNCATE TABLE statement is

executed. The primary purpose of the option is to enable faster table truncation (delete of all rows).

There are some cases where a fast TRUNCATE cannot be done:

- If there are foreign keys either to or from the table
- ♦ If the TRUNCATE TABLE statement is executed within a trigger
- If the TRUNCATE TABLE statement is executed within an atomic statement

TRUNCATION_LENGTH option [ISQL]

Function	Controls the truncation of wide columns for displays to fit on a screen.
Allowed values	Integer
Default	256
Description	The Truncation_length option limits the length of displayed column values. The unit is in characters. A value of 0 means that column values are not truncated. The default truncation length is 256.

TSQL_HEX_CONSTANT option [compatibility]

Function	Controls whether hexadecimal constants are treated as binary typed constants.
Allowed values	ON, OFF
Default	ON
Description	When this option is set to ON , hexadecimal constants are treated as binary typed constants. To get the historical behavior, set the option to OFF.

TSQL_VARIABLES option [compatibility]

Function	Controls whether the @ sign can be used as a prefix for Embedded SQL host variable names.
Allowed values	ON, OFF
Default	OFF
	ON for Open Client and JDBC connections
Description	When this option is set to ON , you can use the @ sign instead of the colon as a prefix for host variable names in Embedded SQL. This is implemented primarily for Transact-SQL compatibility.
UPDATE_STATISTICS option [database]

Function	Controls the gathering of statistics during query execution		
Allowed values	ON, OFF		
Default	ON		
Description	The server collects statistics during normal query execution and uses the gathered statistics to self-tune the column histograms. You can set the UPDATE_STATISTICS option OFF to disable the gathering of statistics during query execution.		
	The UPDATE_STATISTICS option does not affect changes to statistics as a result of updates to data (LOAD/INSERT/UPDATE/DELETE).		
	Under normal circumstances it should not be necessary to turn this option off.		

USER_ESTIMATES option [database]

Function	Controls whether or not user selectivity estimates in query predicates are respected or ignored by the query optimizer.		
Allowed values	ENABLED, DISABLED, OVERRIDE-MAGIC		
Scope	Can be set for an individual connection or for the PUBLIC group. Takes effect immediately.		
Default	OVERRIDE-MAGIC		
Description	Adaptive Server Anywhere allows you to specify user selectivity estimates can improve the optimizer's performance when the server is unable to accurately predict the selectivity of a predicate. However, user selectivity estimates should be used only in appropriate circumstances. For example, it may be useful to supply a selectivity estimate for a predicate that involves one or more functions if the OVERRIDE-MAGIC selectivity estimate used by the optimizer is significantly different from the actual selectivity.		
	If you have used selectivity estimates that are inaccurate as a workaround to performance problems where the software-selected access plan was poor, it is recommended that you set this option to DISABLED. The server may not select an optimal plan if you use inaccurate estimates.		
	For more information about user selectivity estimates, see "Explicit selectivity estimates" [ASA SQL Reference, page 30].		
	When a user selectivity estimate is supplied with a predicate, the estimate is respected or ignored based on the setting of this option. The following		

values are accepted:

- **ENABLED** All user-supplied selectivity estimates are respected. You can also use ON to turn on this option.
- OVERRIDE-MAGIC A user selectivity estimate is respected and used only if the optimizer would otherwise choose to use its last-resort, heuristic value (also called the magic value).
- **DISABLED** All user estimates are ignored and magic values are used when no other estimate data is available. You can also use OFF to turn off this option.

VERIFY_ALL_COLUMNS option [replication]

Function	Controls whether messages that contain updates published by the local database are sent with all column values included.		
Allowed values	ON, OFF		
Default	OFF		
Description	This option is used by SQL Remote only. When it is set to ON , messages that contain updates published by the local database are sent with all column values included, and a conflict in any column triggers a RESOLVE UPDATE trigger at the subscriber database.		

VERIFY_THRESHOLD option [replication]

Function	Controls which columns are verified when updates are replicated.
Allowed values	Integer, in bytes
Default	1 000
Description	This option is used by SQL Remote only. If the data type of a column is longer than the threshold, old values for the column are not verified when an UPDATE is replicated. This keeps the size of SQL Remote messages down, but has the disadvantage that conflicting updates of long values are not detected.

WAIT_FOR_COMMIT option [database]

Function	Determines when foreign key integrity is checked, as data is manipulated.
Allowed values	ON or OFF
Scope	Can be set for an individual connection or for the PUBLIC group. Takes effect immediately.

Default	OFF
Description	If this option is set to ON , the database does not check foreign key integrity until the next COMMIT statement. Otherwise, all foreign keys that are not created with the CHECK_ON_COMMIT option are checked as they are inserted, updated or deleted.

CHAPTER 17

Database Performance and Connection Properties

About this Chapter	This chapter contains information and tables relating to database performance monitoring and database connection parameters.	
Contents	Торіс:	page
	Database performance statistics	638
	Database properties	647

Database performance statistics

Adaptive Server Anywhere provides a set of statistics that can be used to monitor database performance. These are accessible from Sybase Central, and client applications can access the statistics as functions. In addition, these statistics are made available by the server to the Windows Performance Monitor.

Note: The Windows Performance Monitor is available in Windows NT/2000/XP.

This section describes how to access performance and related statistics from client applications, how to monitor database performance using Sybase Central, and how to monitor database performance using the Windows Performance Monitor.

Performance Monitor statistics

The Windows Performance Monitor is an application for viewing the behavior of objects such as processors, memory, and applications. Adaptive Server Anywhere provides many statistics for the Performance Monitor to display.

For information on how to use the Performance Monitor, see "Monitoring database statistics from Windows Performance Monitor" [ASA SQL User's Guide, page 190].

Adaptive Server Anywhere makes statistics available for the Performance Monitor. Rates are reported per second. The statistics are grouped into the following areas:

- "Cache statistics" on page 639
- "Checkpoint and recovery statistics" on page 639
- "Communications statistics" on page 640
- "Disk I/O statistics" on page 641
- "Disk read statistics" on page 642
- "Disk write statistics" on page 642
- "Index statistics" on page 643
- "Java VM statistics" on page 643
- "Memory pages statistics" on page 644

- "Request statistics" on page 645
- "Miscellaneous statistics" on page 645

Cache statistics These statistics describe the use of the cache.

Statistic	Scope	Description
Cache Hits/sec	c Connection and Database	The rate at which database page lookups are satisfied by finding the page in the cache.
Cache Reads: dex Interior/se	In- Connection and Database	The rate at which index internal-node pages are read from the cache.
Cache Reads: dex Leaf/sec	In- Connection and Database	The rate at which index leaf pages are read from the cache.
Cache Reads: ble/sec	Ta- Connection and Database	The rate at which table pages are read from the cache.
Cache Reads: tal Pages/sec	To- Connection and Database	The rate at which database pages are looked up in the cache.
Cache Size: C rent	Cur- Engine	The current size of the database server cache, in kilobytes.
Cache Size: N imum	Iax- Engine	The maximum allowed size of the database server cache, in kilobytes.
Cache Size: M mum	Iini- Engine	The minimum allowed size of the database server cache, in kilobytes.
Cache Size: Po	eak Engine	The peak size of the database server cache, in kilobytes.

Checkpoint and recovery
statisticsThese statistics isolate the checkpoint and recovery actions performed when
the database is in an idle state.

Statistic	Scope	Description
Checkpoint Flushes/sec	Database	The rate at which ranges of adja- cent pages are written out during a checkpoint.
Checkpoint Log	Database	The rate at which the transaction log is checkpointed.

Statistic	Scope	Description
Checkpoint Ur- gency	Database	Checkpoint Urgency, expressed as a percentage.
Checkpoints/sec	Database	The rate at which checkpoints are performed.
Idle Actives/sec	Database	Database The rate at which the en- gine's idle thread becomes active to do idle writes, idle checkpoints, etc.
Idle Checkpoint Time	Database	The total time spent doing idle check- points, in seconds.
Idle Check- points/sec	Database	The rate at which checkpoints are completed by the engine's idle thread. An idle checkpoint occurs whenever the idle thread writes out the last dirty page in the cache.
Idle Writes/sec	Database	The rate at which disk writes are issued by the engine's idle thread.
Recovery I/O Es- timate	Database	The estimated number of I/O opera- tions required to recover the database.
Recovery Urgency	Database	Recovery Urgency expressed as a percentage.

Communications statistics

These statistics describe client/server communications activity.

Statistic	Scope	Description
Comm: Buffer Misses	Server	The total number of network buffer allocations exceeding the connection buffer pool.
Comm: Bytes Received/sec	Connection and Server	The rate at which network data (in bytes) are received.
Comm: Bytes Received Uncom- pressed/sec	Connection and Server	The rate at which bytes would have been received if compression was disabled.
Comm: Bytes Sent/sec	Connection and Server	The rate at which bytes are transmitted over the network.

Statistic	Scope	Description
Comm: Bytes Sent Uncom- pressed/sec	Connection and Server	The rate at which bytes would have been sent if compression was dis- abled.
Comm: Free Buffers	Server	Number of free network buffers.
Comm: Licenses in Use	Server	The number of unique client network addresses connected.
Comm: Multi- packets Re- ceived/sec	Server	The rate at which multi-packet deliv- eries are received.
Comm: Multi- packets Sent/sec	Server	The rate at which multi-packet deliv- eries are transmitted.
Comm: Packets Received/sec	Connection and Server	The rate at which network packets are received.
Comm: Packets Received Uncom- pressed/sec	Connection and Server	The rate at which network packets would have been received if compres- sion was disabled.
Comm: Packets Sent/sec	Connection and Server	The rate at which network packets are transmitted.
Comm: Pack- ets Sent Uncom- pressed/sec	Connection and Server	The rate at which network packets would have been transmitted if com- pression was disabled.
Comm: Send Fails/sec	Server	The rate at which the underlying protocol(s) failed to send a packet.
Comm: Total- Buffers	Server	The total number of network buffers.

Disk I/O statistics These statistics combine disk reads and disk writes to give overall information about the amount of activity devoted to disk I/O.

Statistic	Scope	Description
Disk: Active I/O	Database	The current number of file I/Os issued by the engine which have not yet completed.
Disk: Maximum I/O	Database	The maximum value "Disk Reads: Active I/O" has reached.

Disk read statistics

These statistics describe the amount and type of activity devoted to reading information from disk.

Statistic	Scope	Description
Disk Reads: Total Pages/sec	Connection and Database	The rate at which pages are read from a file.
Disk Reads: Ac- tive	Database	The current number of file reads issued by the engine which have not yet completed.
Disk Reads: Index interior/sec	Connection and Database	The rate at which index internal-node pages are being read from disk.
Disk Reads: Index leaf/sec	Connection and Database	The rate at which index leaf pages are being read from disk.
Disk Reads: Ta- ble/sec	Connection and Database	The rate at which table pages are being read from disk.
Disk Reads: Max- imum Active	Database	The maximum value "Disk Reads: Active" has reached.

Disk write statistics These statistics describe the amount and type of activity devoted to writing information to disk.

Statistic	Scope	Description
Disk Writes: Ac- tive	Database	The current number of file writes issued by the engine which are not yet completed.
Disk Writes: Maximum Active	Database	The maximum value "Disk Writes: Active" has reached.

Statistic	Scope	Description
Disk Writes: Commit Files/sec	Database	The rate at which the engine forces a flush of the disk cache. Win- dows NT/2000/XP and NetWare plat- forms use unbuffered (direct) I/O, so the disk cache does not need to be flushed.
Disk Writes: Database Ex- tends/sec	Database	The rate at which the database file is extended, in pages/sec.
Disk Writes: Temp Extends/sec	Database	The rate at which temporary files are extended, in pages/sec.
Disk Writes: Pages/sec	Connection and Database	The rate at which modified pages are being written to disk.
Disk Writes: Transaction Log/sec	Connection and Database	The rate at which pages are written to the transaction log.
Disk Writes: Transaction Log Group Commits	Connection and Database	Occurs when a commit of the trans- action log is requested but the log has already been written (so the commit was done for "free").

Index statistics These statistics describe the use of the index.

Statistic	Scope	Description
Index: Adds/sec	Connection and Database	The rate at which entries are added to indexes.
Index: Lookups/sec	Connection and Database	The rate at which entries are looked up in indexes.
Index: Full Com- pares/sec	Connection and Database	The rate at which comparisons beyond the hash value in an index must be performed.

Java VM statistics These statistics describe the memory used by the Java VM.

Statistic	Scope	Description
JVM: Global Fixed Size	Server	The total bytes allocated to the Java VM fixed heap.
JVM: Heap Size	Connection and Database	The total bytes allocated to connection Java VMs
JVM: Namespace Size	Database	The total number of bytes allocated to the Java VM namespace.

Memory pages statistics These statistics describe the amount and purpose of memory used by the database server.

Statistic	Scope	Description
Mem Pages: Locked Heap	Server	The number of heap pages locked in the cache.
Mem Pages: Lock Table	Database	The number of pages used to store lock information.
Mem Pages: Roll- back Log	Connection and Database	The number of pages in the rollback log.
Mem Pages: Main Heap	Server	The number of pages used for global engine data structures.
Mem Pages: Map Pages	Database	The number of map pages used for accessing the lock table, frequency table, and table layout.
Mem Pages: Pro- cedure Definitions	Database	The number of relocatable heap pages used for procedures.
Mem Pages: Re- locatable	Database	The number of pages used for relo- catable heaps (cursors, statements, procedures, triggers, views, etc.).
Mem Pages: Re- locations/sec	Database	The rate at which relocatable heap pages are read from the temporary file.
Mem Pages: Trig- ger Definitions	Database	The number of relocatable heap pages used for triggers.
Mem Pages: View Definitions	Database	The number of relocatable heap pages used for views.

Request statistics These statistics describe the database server activity devoted to responding to requests from client applications.

Statistic	Scope	Description
Requests	Server	The rate at which the engine is entered to allow it to handle a new request or continue processing an existing request.
Requests: Active	Server	The number of engine threads cur- rently handling a request.
Requests: Un- scheduled	Server	The number of requests that are cur- rently queued up waiting for an avail- able engine thread.
Cursors	Connection	The number of declared cursors cur- rently maintained by the engine.
Cursors Open	Connection	The number of open cursors currently maintained by the engine.
Statements	Connection	The number of prepared statements currently maintained by the engine.
Statement Pre- pares	Connection	The rate at which statement prepares are being handled by the engine.
Transaction Com- mits	Connection	The rate at which Commit requests are handled.
Transaction Roll- backs	Connection	The rate at which Rollback requests are handled.

Miscellaneous statistics

Statistic	Scope	Description
Avail IO	Server	Expressed as a count.
Connection Count	Database	The number of connections to this database.
Main Heap Bytes	Server	The number of bytes used for global engine data structures.
Query Low Mem- ory Strategy	Connection and Database	The number of times the server changed its execution plan during execution because of low memory conditions.
Temporary Table Pages	Connection and Database	The number of pages in the temporary file used for temporary tables.

Database properties

Adaptive Server Anywhere provides a set of properties that are made available to client applications. These properties describe aspects of connection, database, and database server behavior.

Accessing properties Each type of property can be accessed by supplying its name as an argument to a system function.

To access connection properties

1. Use the connection_property system function: the following statement returns the number of pages that have been read from file by the current connection.

select connection_property ('DiskRead')

To access database properties

1. Use the db_property system function. For example, the following statement returns the page size of the current database:

select db_property ('PageSize')

To access database server properties

1. Use the property system function: the following statement returns the number of pages that have been read from file by the current connection.

```
select property ( 'MainHeapPages' )
```

Connection-level properties

The following table lists properties available for each connection.

Examples

* To retrieve the value of a connection property

1. Use the connection_property system function. The following statement returns the number of pages that have been read from file by the current connection.

```
select connection_property ( 'DiskRead' )
```

✤ To retrieve the values of all connection properties

1. Use the sa_conn_properties system procedure:

call sa_conn_properties

A separate row appears for each connection.

Descriptions

Property	Description
Allow_nulls_by_default	"ALLOW_NULLS_BY_DEFAULT option [compatibility]" on page 574
Ansi_blanks	"ANSI_BLANKS option [compatibility]" on page 574
Ansi_close_cursors_on rollback	"ANSI_CLOSE_CURSORS_ON ROLLBACK option [compatibility]" on page 574
Ansi_integer_overflow	"ANSI_INTEGER_OVERFLOW option [compatibility]" on page 575
Ansi_permissions	"ANSI_PERMISSIONS option [compatibil- ity]" on page 575
Ansi_update_constraints	"ANSI_UPDATE_CONSTRAINTS option [compatibility]" on page 576
Ansinull	"ANSINULL option [compatibility]" on page 577
AppInfo	"AppInfo connection parameter [APP]" on page 175
AuditingTypes	The types of auditing currently enabled. "AU- DITING option [database]" on page 578
Automatic_timestamp	"AUTOMATIC_TIMESTAMP option [com- patibility]" on page 579
Background_priority	"BACKGROUND_PRIORITY option [database]" on page 580
BlockedOn	If the current connection is not blocked, this is zero. If it is blocked, the connection number on which the connection is blocked due to a locking conflict.
Blocking	"BLOCKING option [database]" on page 581

Property	Description
BytesReceived	The number of bytes received during client/server communications.
BytesReceivedUncomp	The number of bytes that would have been received during client/server communications if compression was disabled. (This value is the same as the value for BytesReceived if compression is disabled.)
BytesSent	The number of bytes sent during client/server communications.
BytesSentUncomp	The number of bytes that would have been sent during client/server communications if compression was disabled. (This value is the same as the value for BytesSent if compression is disabled.)
CacheHits	The number of successful reads of the cache.
CacheRead	The number of database pages that have been looked up in the cache.
CacheReadIndInt	The number of index internal-node pages that have been read from the cache.
CacheReadIndLeaf	The number of index leaf pages that have been read from the cache.
CacheReadTable	The number of table pages that have been read from the cache.
Chained	"CHAINED option [compatibility]" on page 581
CharSet	The character set used by the connection.
Checkpoint_time	"CHECKPOINT_TIME option [database]" on page 582
CIS_option	Turns on debugging for remote data access.
Cis_rowset_size	Reserved
ClientLibrary	jConnect for jConnect connections; CT Library for Open Client connections; Cmd- Seq for ODBC and Embedded SQL connec- tions;

Property	Description
Close_on_EndTrans	"CLOSE_ON_ENDTRANS option [compati- bility]" on page 583
Commit	The number of Commit requests that have been handled.
CommLink	The communication link for the connection. This is one of the network protocols supported by Adaptive Server Anywhere, or is local for a same-machine connection.
CommNetworkLink	The communication link for the connection. This is one of the network protocols sup- ported by Adaptive Server Anywhere. Values can include SharedMemory , TCPIP , SPX or NamedPipes . The CommLinkNetwork property always returns the name of the link, regardless of whether it is same-machine or not.
CommProtocol	Returns CmdSeq for Adaptive Server Any- where protocol (ODBC and Embedded SQL) or TDS for Open Client and jConnect connec- tions.
Compression	Returns ON or OFF, to indicate whether com- munication compression is enabled on the connection.
Connection authentication	A string used to authenticate the client. Au- thentication is required before the database can be modified.
Conversion_error	"CONVERSION_ERROR option [compatibil- ity]" on page 585
Cooperative_commit timeout	"COOPERATIVE_COMMIT_TIMEOUT op- tion [database]" on page 586
Cooperative_commits	"COOPERATIVE_COMMITS option [database]" on page 586
Cursor	The number of declared cursors that are cur- rently being maintained by the server.
CursorOpen	The number of open cursors that are currently being maintained by the server.

Property	Description
Database_authentication	A string used to authenticate the database. Authentication is required before the database can be modified.
Date_format	"DATE_FORMAT option [compatibility]" on page 587
Date_order	"DATE_ORDER option [compatibility]" on page 588
DBNumber	The ID number of the database.
Default_timestamp increment	"DEFAULT_TIMESTAMP_INCREMENT option [database]" on page 589
Delayed_commit_timeout	"DELAYED_COMMIT_TIMEOUT option [database]" on page 589
Delayed_commits	"DELAYED_COMMITS option [database]" on page 590
DiskRead	The number of pages that have been read from disk.
DiskReadIndInt	The number of index internal-node pages that have been read from disk.
DiskReadIndLeaf	The number of index leaf pages that have been read from disk.
DiskReadTable	The number of table pages that have been read from disk.
DiskWrite	The number of modified pages that have been written to disk.
Divide_by_zero_error	"DIVIDE_BY_ZERO_ERROR option [com- patibility]" on page 591
Encryption	"Encryption connection parameter [ENC]" on page 188
Escape_character	"ESCAPE_CHARACTER option [compatibil ity]" on page 592
EventName	The name of the associated event if the connection is running an event handler. Otherwise, the result is NULL.

Property	Description
Extended_join_syntax	"EXTENDED_JOIN_SYNTAX option [database]" on page 592
Fire_triggers	"FIRE_TRIGGERS option [compatibility]" on page 592
Float_as_double	"FLOAT_AS_DOUBLE option [compatibil- ity]" on page 593
FullCompare	The number of comparisons that have been performed beyond the hash value in an index.
IdleTimeout	The idle timeout value of the connection. For more information, see "Idle connection parameter [IDLE]" on page 192.
IndAdd	The number of entries that have been added to indexes.
IndLookup	The number of entries that have been looked up in indexes.
Isolation_level	"ISOLATION_LEVEL option [compatibil- ity]" on page 597
Java_heap_size	"JAVA_HEAP_SIZE option [database]" on page 601
Java_input_output	"JAVA_INPUT_OUTPUT option [database]" on page 601
Java_namespace_size	"JAVA_NAMESPACE_SIZE option [database]" on page 602
Java_page_buffer_size	The page buffer size used by the Java VM.
JavaHeapSize	The heap size per Java VM.
Language	The locale language.
LastIdle	The number of ticks between requests.
LastReqTime	The time at which the last request for the specified connection started.
LastStatement	The most recently prepared SQL statement for the current connection.
	For more information, see "-zl server option" on page 166.

Property	Description
LivenessTimeout	The liveness timeout period for the current connection.
	For more information, see "LivenessTimeout connection parameter [LTO]" on page 194.
Lock_rejected_rows	Reserved
LockName	A 64-bit unsigned integer value representing the lock for which a connection is waiting.
LogFreeCommit	The number of Redo Free Commits. A "Redo Free Commit" occurs when a commit of the transaction log is requested but the log has already been written (so the commit was done for "free").
Login_mode	"LOGIN_MODE option [database]" on page 602
Login_procedure	"LOGIN_PROCEDURE option [database]" on page 603
LogWrite	The number of pages that have been written to the transaction log.
Max_cursor_count	"MAX_CURSOR_COUNT option [database]" on page 605
Max_plans_cached	"MAX_PLANS_CACHED option [database]" on page 606
Max_statement_count	"MAX_STATEMENT_COUNT option [database]" on page 608
Min_password_length	"MIN_PASSWORD_LENGTH option [database]" on page 608
Min_table_size_for histogram	"MIN_TABLE_SIZE_FOR_HISTOGRAM option [database]" on page 609
Name	The name of the current connection.
Nearest_century	"NEAREST_CENTURY option [compatibil- ity]" on page 609
NodeAddress	The node for the client in a client/server connection.

Property	Description
Non_keywords	"NON_KEYWORDS option [compatibility]" on page 610
Number	The ID number of the connection.
On_tsql_error	"ON_TSQL_ERROR option [compatibility]" on page 612
Optimization_goal	"OPTIMIZATION_GOAL option [database]" on page 613
Optimization_level	Reserved
PacketsReceived	The number of client/server communication packets received.
PacketsReceivedUncomp	The number of packets that would have been received during client/server communications if compression was disabled. (This value is the same as the value for PacketsReceived if compression is disabled.)
PacketsSent	The number of client/server communication packets sent.
PacketsSentUncomp	The number of packets that would have been sent during client/server communications if compression was disabled. (This value is the same as the value for PacketsSent if compres- sion is disabled.)
PacketSize	The packet size used by the connection, in bytes.
Percent_as_comment	"PERCENT_AS_COMMENT option [com- patibility]" on page 616
Precision	"PRECISION option [database]" on page 617
Prefetch	"PREFETCH option [database]" on page 618
Prepares	The number of statement preparations carried out.
PrepStmt	The number of prepared statements currently being maintained by the server.
Query_plan_on_open	"QUERY_PLAN_ON_OPEN option [compat- ibility]" on page 619

Property	Description
QueryBypassed	The number of requests optimized by the optimizer bypass.
QueryCachePages	The number of pages used to cache execution plans.
QueryCachedPlans	The number of query execution plans currently cached for the connection.
QueryLowMemoryStrat- egy	The number of times the server changed its execution plan during execution as a result of low memory conditions. The strategy can change because less memory is available than the optimizer estimated, or because the execution plan required more memory than the optimizer estimated.
QueryOptimized	The number of requests that have been fully optimized.
QueryReused	The number of requests that have been reused from the plan cache.
Quoted_identifier	"QUOTED_IDENTIFIER option [compatibil- ity]" on page 620
Recovery_time	"RECOVERY_TIME option [database]" on page 621
Replicate_all	"REPLICATE_ALL option [replication]" on page 621
ReqType	A string for the type of the last request.
RI_trigger_time	"RI_TRIGGER_TIME option [compatibility]" on page 623
Rlbk	The number of Rollback requests that have been handled.
RollbackLogPages	The number of pages in the rollback log.
Row_counts	"ROW_COUNTS option [database]" on page 624
Scale	"SCALE option [database]" on page 624
SQL_flagger_error_level	"SQL_FLAGGER_ERROR_LEVEL option [compatibility]" on page 625

Property	Description
SQL_flagger_warning level	"SQL_FLAGGER_WARNING_LEVEL op- tion [compatibility]" on page 625
String_rtruncation	"STRING_RTRUNCATION option [compati- bility]" on page 626
Suppress_TDS debugging	"SUPPRESS_TDS_DEBUGGING option [database]" on page 627
TDS_empty_string_is null	"TDS_EMPTY_STRING_IS_NULL option [database]" on page 628
TempTablePages	The number of pages in the temporary file used for temporary tables.
Time_format	"TIME_FORMAT option [compatibility]" on page 628
Timestamp_format	"TIMESTAMP_FORMAT option [compatibil- ity]" on page 629
TimeZoneAdjustment	The number of minutes that must be added to the Coordinated Universal Time (UTC) to display time local to the connection. By default, the value is set according to the client's time zone.
Truncate_timestamp values	"TRUNCATE_TIMESTAMP_VALUES op- tion [database]" on page 631
Truncate_with autocommit	"TRUNCATE_WITH_AUTO_COMMIT op- tion [database]" on page 631
Tsql_hex_constant	"TSQL_HEX_CONSTANT option [compati- bility]" on page 632
Tsql_variables	"TSQL_VARIABLES option [compatibility]" on page 632
UncommitOp	The number of uncommitted operations.
User_estimates	"USER_ESTIMATES option [database]" on page 633
Userid	The user ID for the connection.

Property	Description
UtilCmdsPermitted	Returns ON or OFF to indicate whether utility commands such as CREATE DATABASE, DROP DATABASE, and RESTORE DATABASE are permitted for the connec- tion.
	For more information, see "-gu server option" on page 150.
Wait_for_commit	"WAIT_FOR_COMMIT option [database]" on page 634

Server-level properties

The following table lists properties that apply across the server as a whole.

Examples

To retrieve the value of a server property

1. Use the property system function. For example, the following statement returns the number of cache pages being used to hold the main heap:

select property ('MainHeapPages')

* To retrieve the values of all server properties

1. Use the sa_eng_properties system procedure:

call sa_eng_properties

Descriptions

Property	Description
ActiveReq	The number of server threads that are currently handling a request.
AvailIIO	Reserved
BuildChange	Reserved
BuildClient	Reserved
BuildReproducible	Reserved
BytesReceived	The number of bytes received during client/server communications.

Property	Description
BytesReceivedUn- comp	The number of bytes that would have been received during client/server communications if compres- sion was disabled. (This value is the same as the value for BytesReceived if compression is dis- abled.)
BytesSent	The number of bytes sent during client/server communications.
BytesSentUncomp	The number of bytes that would have been sent dur- ing client/server communications if compression was disabled. (This value is the same as the value for BytesSent if compression is disabled.)
C2	Returns YES if the -sc option was used when the server was started. Otherwise, returns NO.
	For more information, see "-sc server option" on page 158.
CacheHitsEng	The number of database page lookups.
CacheReplacements	The number of pages in the cache that have been replaced.
CharSet	The character set in use by the database server.
CommandLine	The command line that was used to start the server.
CompactPlatformVer	A condensed version of the PlatformVer property.
CompanyName	The name of the company owning this software.
ConnsDisabled	Returns ON or OFF, to indicate the current setting of the server option to disable new connections.
	For information, see "sa_server_option system procedure" [ASA SQL Reference, page 739].
CurrentCacheSize	The current cache size, in kilobytes.
DefaultCollation	The collation that would be used for new databases, if none is explicitly specified.
FreeBuffers	The number of available network buffers.
IdleTimeout	The default idle timeout.
	For more information, see "-ti server option" on page 159.

Property	Description
IsIQ	Returns YES if the server is an IQ server, and NO otherwise.
IsJavaAvailable	Returns YES if the JavaVM is installed, and NO if the JavaVM is not installed. This property only indicates if the Java VM is available, not whether it is currently being used.
IsNetworkServer	Returns YES if connected to a network database server, and NO if connected to a personal database server.
IsRuntimeServer	Returns YES if connected to the limited desktop runtime database server, and NO otherwise.
JavaGlobFix	Java VM global fixed size.
Language	The locale language for the server.
LegalCopyright	The copyright string for the software.
LegalTrademarks	Trademark information for the software.
LicenseCount	The number of licensed seats.
LicensedCompany	The name of the licensed company.
LicensedUser	The name of the licensed user.
LicenseType	The license type. Can be either concurrent or networked seat (per-seat).
LicensesinUse	The number of concurrent users currently con- nected to the network server, as determined by the number of unique client network addresses connected to the server.
LivenessTimeout	The client liveness timeout default.
LockedHeapPages	The number of heap pages locked in the cache.
MachineName	The name or IP address of the computer running a database server.
MainHeapBytes	The number of bytes used for global server data structures.
MainHeapPages	The number of pages used for global server data structures.

Property	Description
MaxCacheSize	The maximum allowed cache size, in kilobytes.
MaxMessage	The current maximum line number that can be retrieved from the server's message window. This represents the most recent message displayed in the server's message window.
Message, linenumber	A line from the server's message window, prefixed by the date and time the message appeared. The second parameter specifies the line number.
MessageText, linenumber	The text associated with the specified line number in the server's message window, without a date and time prefix. The second parameter specifies the line number.
MessageTime, linenumber	The date and time associated with the specified line number in the server's message window. The second parameter specifies the line number.
MessageWindowSize	The maximum number of lines that can be retrieved from the server's message window.
MinCacheSize	The minimum allowed cache size, in kilobytes.
MultiPacketsRe- ceived	The number of multi-packet deliveries received during client/server communications.
MultiPacketsSent	The number of multi-packet deliveries sent during client/server communications.
Name	The name of the server.

Property	Description
NativeProcessorAr- chitecture	On platforms where a processor can be emulated (such as X86 on Win64), returns a string that iden- tifies the native processor type. In all other cases, it returns the same value as property('ProcessorAr- chitecture'). Values can include: 32-bit Windows (not CE) - X86 NetWare - X86 Intel Solaris - X86 CE - SH3, SH4, MIPS or ARM 64-bit Windows - IA64 or AMD64 64-bit Unix - IA64 or AMD64 Solaris - SPARC AIX - PPC MAC OS - PPC HD. DA DISC
	HP - PA_RISC DEC Univ. ALPHA
	Linux - X86, SPARC, IA64
NumProcessorsAvail	The number of processors on the server.
NumProcessorsMax	The maximum number of processors used. Nor- mally this should be 2 for <i>dbeng.exe</i> and 0 for <i>dbsrv.exe</i> .
PacketsReceived	The number of client/server communication packets received.
PacketsReceivedUn- comp	The number of packets that would have been received during client/server communications if compression was disabled. (This value is the same as the value for PacketsReceived if compression is disabled.)
PacketsSent	The number of client/server communication packets sent.
PacketsSentUncomp	The number of packets that would have been sent during client/server communications if compres- sion was disabled. (This value is the same as the value for PacketsSent if compression is disabled.)

Property	Description
PageSize	The size of the database server cache pages. This can be set using the -gp option, otherwise, it is the maximum database page size of the databases specified on the command line.
PeakCacheSize	The largest value the cache has reached in the current session, in kilobytes.
Platform	The operating system on which the software is running.
PlatformVer	The operating system on which the software is running, including build numbers, service packs, etc.
ProcessCPU	CPU usage statistics for the server process. Values are in seconds. This property is supported on Windows NT/2000/XP, Windows 95/98/Me, and UNIX. This property is not supported on Windows CE or NetWare.
ProcessCPUSystem	Process CPU system usage. Values are in sec- onds. This property is supported on Windows NT/2000/XP, Windows 95/98/Me, and UNIX. This property is not supported on Windows CE or Net- Ware.
ProcessCPUUser	Process CPU user usage. Values are in sec- onds. This property is supported on Windows NT/2000/XP, Windows 95/98/Me, and UNIX. This property is not supported on Windows CE or Net- Ware.

Property	Description
ProcessorArchitec- ture	A string that identifies the processor type. Values can include:
	32-bit Windows (not CE) - X86
	NetWare - X86
	Intel Solaris - X86
	CE - SH3, SH4, or ARM
	64-bit Windows - IA64 or AMD64 64-bit
	Unix - IA64 or AMD64
	Solaris - SPARC
	AIX - PPC
	MAC OS - PPC
	HP - PA_RISC
	DEC Unix - ALPHA
ProductName	The name of the software.
ProductVersion	The version of the software being run.
QuittingTime	Shutdown time for the server. If none is specified, the value is none .
Req	The number of times the server has been entered to allow it to handle a new request or continue processing an existing request.
RequestLogFile	The name of the request-level logging filename. An empty string is returned if there is no request level logging.
	For information, see "sa_server_option system procedure" [ASA SQL Reference, page 739].
RequestLogging	ALL, SQL, or NONE.
	For information, see "sa_server_option system procedure" [ASA SQL Reference, page 739].
RequestQueueWait	The number of times the server has had to wait for room in the request queue.
SendFail	The number of times that the underlying communi- cations protocols have failed to send a packet.
StartTime	The date/time that the server started

Property	Description
Tempdir	The directory in which temporary files are stored by the server.
TimeZoneAdjust- ment	The number of minutes that must be added to the Coordinated Universal Time (UTC) to display time local to the server.
TotalBuffers	The total number of network buffers.
UnschReq	The number of requests that are currently queued up waiting for an available server thread.

Database-level properties

The following table lists properties available for each database on the server.

Examples

To retrieve the value of a database property

1. Use the db_property system function. For example, the following statement returns the page size of the current database:

select db_property ('PageSize')

* To retrieve the values of all database properties

1. Use the sa_db_properties system procedure:

call sa_db_properties

Descriptions

Property	Description
Alias	The database name.
BlankPadding	The status of the blank padding feature. Returns ON if the database has blank padding enabled. Otherwise, it returns OFF.
BlobArenas	The status of the BlobArenas feature. Returns ON if the database stores extension (blob) pages separately from table pages for the database. Otherwise, returns OFF.

Property	Description
CacheHits	The number of database page lookups satisfied by finding the page in the cache.
CacheRead	The number of database pages that have been looked up in the cache.
CacheReadIndInt	The number of index internal-node pages that have been read from the cache.
CacheReadIndLeaf	The number of index leaf pages that have been read from the cache.
CacheReadTable	The number of table pages that have been read from the cache.
Capabilities	The capability bits enabled for the database. This property is primarily for use by technical support.
CaseSensitive	The status of the case sensitivity feature. Returns ON if the database is case sensitive. Otherwise, it returns OFF
CharSet	The character set of the database.
CheckpointUrgency	The time that has elapsed since the last checkpoint as a percentage of the checkpoint time setting of the database.
Chkpt	The number of checkpoints that have been per- formed.
ChkptFlush	The number of ranges of adjacent pages written out during a checkpoint.
ChkptPage	The number of transaction log checkpoints.
CommitFile	The number of times the server has forced a flush of the disk cache. On Windows NT/2000/XP and NetWare platforms, the disk cache does not need to be flushed if unbuffered (direct) I/O is used.
CompressedBTrees	Returns ON if Compressed B-tree indexes are supported. Otherwise, returns OFF.

Property	Description
Compression	The compression status of the database. Returns either ON (meaning the database is compressed) or OFF. If a write file is created on a compressed database, the write file is NOT compressed. Start- ing a write file created on a compressed database and selecting db_property('icompression'), re- turns OFF.
ConnCount	The number of connections to the database.
CurrentRedoPos	The current offset in the transaction log file where the next database operation is to be logged.
CurrIO	The current number of file I/Os that were issued by the server but have not yet completed.
CurrRead	The current number of file reads that were issued by the server but have not yet completed.
CurrWrite	The current number of file writes that were issued by the server but have not yet completed.
DBFileFragments	The number of database file fragments. This property is supported on Windows NT/2000/XP.
DiskRead	The number of pages that have been read from disk.
DiskReadIndInt	The number of index internal-node pages that have been read from disk.
DiskReadIndLeaf	The number of index leaf pages that have been read from disk.
DiskReadTable	The number of table pages that have been read from disk.
DiskWrite	The number of modified pages that have been written to disk.
DriveType	The drive on which the database file is located. Returns CD, FIXED, RAMDISK, REMOTE, REMOVABLE, and UNKNOWN. Returns UN- KNOWN on non-Windows platforms.
Encryption	The type of encryption applied to the database. Returns None, Simple, AES, or MDSR.

Property	Description
ExtendDB	The number of pages by which the database file has been extended.
ExtendTempWrite	The number of pages by which temporary files have been extended.
File	The file name of the database root file, including path.
FileSize dbspace	When used with db_property , this property returns the file size of the system dbspace, in pages. When used with db_extended_property , you
	<i>Dbspace</i> can be either the <i>name</i> of the dbspace, the <i>file_id</i> of the dbspace, or <i>temporary</i> to refer to the temporary dbspace.
	You can also specify <i>translog</i> to return the size of the log file.
	Finally, you can specify <i>writefile</i> to refer to the write file. When using a write file, FileSize on a dbspace returns the amount of space in the virtual dbspace, represented by the underlying dbspace plus the modifications to that dbspace that have been stored in the write file.
	Leaving the <i>dbspace</i> unspecified, or using <i>system</i> , both refer to the system dbspace.
	If the specified dbspace does not exist, the prop- erty function returns null. If the name of a dbspace is specified and an id or name of a database which is not the database of the current connection is also specified, the function also returns null.
FileVersion	The version of the database file. This does not correspond to a software release version.

Property	Description	
FreePages dbspace	FreePages is only supported on databases created with 8.0.0 or later.	
	When used with db_property , this property returns the number of free pages in the system dbspace.	
	When used with db_extended_property , you can specify which dbspace you want the number of free pages for. <i>Dbspace</i> can be either the <i>name</i> of the dbspace, the <i>file_id</i> of the dbspace, or <i>temporary</i> to refer to the temporary dbspace.	
	You can also specify <i>translog</i> to return the size of the log file.	
	Finally, you can specify <i>writefile</i> to refer to the write file. When using a write file, FreePages on a dbspace returns the number of free pages in the virtual dbspace, represented by the underlying dbspace plus the modifications to that dbspace that have been stored in the write file.	
	Leaving the <i>dbspace</i> unspecified, or using <i>system</i> both refer to the system dbspace.	
	If the specified dbspace does not exist, the prop- erty function returns null. If the name of a dbspace is specified and an id or name of a database which is not the database of the current connection is also specified, the function also returns null.	
FreePageBitMaps	Returns ON if free database pages are managed via bitmaps. Otherwise, returns OFF.	
FullCompare	The number of comparisons that have been per- formed beyond the hash value in an index.	
GlobalDBId	The value of the GLOBAL_DATABASE_ID option used to generate unique primary key values in a replication environment.	
Histograms	Returns ON if optimizer statistics are maintained as histograms. Otherwise, returns OFF.	
IdleCheck	The number of times that the server's idle thread has become active to do idle writes, idle check- points, and so on.	
Property	Description	
-------------------	---	--
IdleChkpt	The number of checkpoints completed by the server's idle thread. An idle checkpoint occurs whenever the idle thread writes out the last dirty page in the cache.	
IdleChkTime	The number of 100ths of a second spent check- pointing during idle I/O.	
IdleWrite	The number of disk writes that have been issued by the server's idle thread.	
IndAdd	The number of entries that have been added to indexes.	
IndLookup	The number of entries that have been looked up in indexes.	
IOToRecover	The estimated number of I/O operations required to recover the database.	
IQStore	Reserved.	
JavaHeapSize	Heap size per Java VM.	
JavaNSSize	Java VM Namespace size.	
JDKVersion	The Java runtime library version used by this database.	
Language	The locale language of the database.	
LargeProcedureIDs	Returns ON if 32-bit stored procedure IDs are supported for the database. Otherwise, returns OFF.	
LockTablePages	The number of pages used to store lock informa- tion.	
LogFileFragments	The number of log file fragments. This property is supported on Windows NT/2000/XP.	
LogFreeCommit	The number of Redo Free Commits. A "Redo Free Commit" occurs when a commit of the transaction log is requested but the log has already been written (so the commit was done for "free").	
LogName	The file name of the transaction log, including path.	

Property	Description
LogWrite	The number of pages that have been written to the transaction log.
LTMGeneration	The generation number of the LTM, or Replica- tion Agent. This property is primarily for use by technical support.
LTMTrunc	The minimal confirmed log offset for the Replica- tion Agent.
MapPages	The number of map pages used for accessing the lock table, frequency table, and table layout.
MaxIO	The maximum value that CurrIO has reached.
MaxRead	The maximum value that CurrRead has reached.
MaxWrite	The maximum value that CurrWrite has reached.
MultiByteCharSet	Returns ON if the database uses a multi-byte character set. Otherwise, returns OFF.
Name	The database name (identical to alias).
PageRelocations	The number of relocatable heap pages that have been read from the temporary file.
PageSize	The page size of the database, in bytes.
PreserveSource	Returns ON if the database preserves the source for procedures and views. Otherwise, returns OFF.
ProcedurePages	The number of relocatable heap pages that have been used for procedures.
QueryBypassed	The number of requests optimized by the opti- mizer bypass.
QueryCachePages	The number of pages used to cache execution plans.
QueryCachedPlans	The number of cached execution plans across all connections.

Property	Description
QueryLowMemoryS- trategy	The number of times the server changed its ex- ecution plan during execution as a result of low memory conditions. The strategy can change be- cause less memory is available than the optimizer estimated, or because the execution plan required more memory than the optimizer estimated.
QueryOptimized	The number of requests fully optimized.
QueryBypassed	The number of requests reused from the plan cache.
ReadOnly	Returns ON if the database is being run in read- only mode. Otherwise, returns OFF.
RecoveryUrgency	An estimate of the amount of time required to recover the database.
RelocatableHeapPages	The number of pages used for relocatable heaps (cursors, statements, procedures, triggers, views, etc.).
RemoteTrunc	The minimal confirmed log offset for the SQL Re- mote Message Agent.
RollbackLogPages	The number of pages in the rollback log.
SeparateCheckpoint- Log	Returns ON if the checkpoint log for the database is maintained at the end of the SYSTEM dbspace. Otherwise, returns OFF.
SeparateForeignKeys	Returns ON if primary and foreign keys are stored separately. Otherwise, returns OFF.
SyncTrunc	The minimal confirmed log offset for the Mo- biLink client <i>dbmlsync</i> executable.
TableBitMaps	Returns ON if the database supports table bitmaps. Otherwise, returns OFF.
TempFileName	The file name of the database temporary file, including path.
TempTablePages	The number of pages in the temporary file used for temporary tables.
TransactionsSpanLogs	Returns ON if transactions can span multiple log files. Otherwise, returns OFF.

Property	Description
TriggerPages	The number of relocatable heap pages used for triggers.
VariableHashSize	Returns ON if the hash size can be specified for B-tree indexes. Otherwise, returns OFF.
ViewPages	The number of relocatable heap pages used for views.

CHAPTER 18

Physical Limitations

About this chapter	This chapter describes the limitations on size and number of objects i Adaptive Server Anywhere databases.	
Contents	Торіс:	page
	Size and number limitations	674

Size and number limitations

The following table lists the physical limitations on size and number of objects in Adaptive Server Anywhere database. The memory, CPU, and disk drive of the computer are more limiting factors in most cases.

Item	Limitation
Database size	13 files per database. For each file, the largest file allowed by operating system and file system
Field size	2 Gb
File size (FAT 12)	16 Mb
File size (FAT 16)	2 Gb
File size (FAT 32)	4 Gb
File size for NTFS, NetWare (NSS volumes), HP-UX 11.0 and later, Solaris 2.6 and later, Linux 2.4 and later)	 256 Gb for 1 kb pages 512 Gb for 2 kb pages 1 Tb for 4 kb pages
NetWare (traditional volumes)	4 Gb
File size (all other platforms and file systems)	2 Gb
Maximum cache size (non-AWE cache) (Windows 95/98, Windows NT, Win- dows 2000 Professional, Windows 2000 Server, Windows XP Home Edition, Windows XP Professional, Windows Server 2003 Web Edition, Windows Server 2003 Standard Edition)	1.8 Gb
Maximum cache size (non-AWE cache) (Windows NT Advanced Server, Win- dows 2000 Advanced Server, Windows 2000 Enterprise Server, Windows 2000 Datacenter Server, Windows Server 2003 Enterprise Edition, Windows Server 2003 Datacenter Edition)	2.7 Gb
Maximum cache size (Windows CE)	32 Mb

Item	Limitation
Maximum cache size (UNIX—Solaris, x86 Linux, AIX, HP)	2 Gb for 32-bit engines
Maximum cache size (Win 64)	limited by physical memory on 64-bit engines
Maximum cache size (UNIX—Tru64 UNIX, Itanium Linux, Itanium HP-UX)	limited by physical memory on 64-bit engines
Maximum cache size (NetWare)	2 Gb
Maximum index entry size	No limit
Number of databases per server	255
Number of columns per table	((page size)/4)^2. Note: An excessive number of columns, although allowed, will affect performance.
Number of indexes	The maximum number of indexes per table is based on the database page size:
	 4 kb or less 2048 indexes 8 kb 1024 indexes 16 kb 512 indexes 32 kb 256 indexes
Number of rows per database	Limited by file size
Number of rows per table	Limited by file size
Number of tables per database	2^32 - 2^20 - 1 = 4 293 918 719
Number of temporary tables per connec- tion	2^20 = 1 048 576
Number of tables referenced per trans- action	No limit
Number of stored procedures per database	2^32 - 1 = 4 294 967 295
Number of events per database	2^31 - 1 = 2 147 483 647
Number of triggers per database	2^32 - 1 = 4 294 967 295

Item	Limitation
Row size	Limited by file size
Table size	Maximum file size. User-created indexes for the table can be stored separately from the table

Index

Symbols

–? server option		
database server	133	
–A option		
Log Transfer Manger [dbltm] utility 501, 502		
–C option		
Log Transfer Manger [dbltm] utilit 501, 502	y	
-l option	10.4	
Log Transfer Manger [dblmit] utility 501, 502	486 .y	
service creation [dbsvc] utility	519	
upgrade [dbupgrad] utility	543	
validation [dbvalid] utility	548	
–M option		
Log Transfer Manger [dbltm] utilit 501, 502	y	
-ODBC option		
Interactive SQL [dbisql] utility 495	492,	
–S option		
Log Transfer Manger [dbltm] utilit 501, 502	y	
–a option		
database server	167	
Log Translation [dbtran] utility 510	508,	
ping [dbping] utility	514	
service creation [dbsvc] utility519,	521	
-ac option		
unload [dbunload] utility 534,	537	
-an option		
unload [dbunload] utility 534,	537	
-ar option	520	
unioad [dbunioad] utility 517, 534,	538	
-as option	521	
b option	521	
data source [dbdsn] utility 472	475	
database server	133	

initialization [dbinit] utility	486, 487
-c option	150 160
backup [dbbackup] utility	459,460
collation [dbcollat] utility	463
console [dbconsole] utility	4/1
data source [dbdsn] utility	472,476
database server	133
histogram [dbhist] utility	481, 482
information [dbinfo] utility	483
initialization [dbinit] utility	486, 487
Interactive SQL [dbisql] utili 494	ty 492,
Log Translation [dbtran] utili	ity 508,
	514
ping [doping] utility	514
stop [dbstop] utility	525, 526
unload [dbunload] utility	534, 538
upgrade [dbupgrad] utility	543, 544
validation [dbvalid] utility	548, 549
write file [dbwrite] utility	552, 553
-ca option	
database server	135
-ch option	
database server	136
-cl option	
data source [dbdsn] utility	472
database server	136
-codepage option	
Interactive SQL [dbisql] utili 494	ty 492,
-cs option	
database server	137
-ct option	
database server	137
-cw option	
data source [dbdsn] utility	472
database server	138
-d option	
backup [dbbackup] utility	459, 460
collation [dbcollat] utility	463
data source [dbdsn] utility	472, 475
database server	141

Interactive SQL [dbisql] utility 492, 494
language [db]ang] utility 497 498
Log Translation [dbtran] utility 508
511
ping [dbping] utility 514, 515
service creation [dbsvc] utility519, 521
stop [dbstop] utility 525, 526
unload [dbunload] utility 534, 538
write file [dbwrite] utility 552, 553
-d1 option
Interactive SQL [dbisql] utility 494
-datasource option
Interactive SQL [dbisql] utility 492,
494
-dl option
Interactive SOL [dbisa]] utility 492
Log Transfer Manger [dbltm] utility
501 503
e option
collation [dbcollat] utility 463 464
initialization [dbinit] utility 465, 464
unload [dburload] utility 524,529
unioad [dounioad] utility 534, 538
initialization [dbinit] utility 486, 488
unioad [dbunioad] utility 534, 538
-ec option
data source [dbdsn] utility 4/2, 4/6
database server 141
-ek option
compression [dbshrink] utility468, 469
database server 168
erase [dberase] utility 479
initialization [dbinit] utility 486, 488
Log Transfer Manger [dbltm] utility 501, 502
Log Translation [dbtran] utility 508,
transaction log [dblog] utility 527 529
uncompression [dbeypand] utility 531
532
unload [dbunload] utility 534, 539
write file [dbwrite] utility 552, 553
-ep option
compression [dbshrink] utility468, 469
database server 143
erase [dberase] utility 479

initialization [dbinit] utility Log Transfer Manger [dbltm] 501 502	486, 488 utility
Log Translation [dbtran] utilit 511	ty 508,
transaction log [dblog] utility	527, 529
uncompression [dbexpand] ut 532	ility 531,
unload [dbunload] utility	534, 539
write file [dbwrite] utility	552, 553
-f option	
database server	168
Log Translation [dbtran] utilit 512	ty 508,
spawn [dbspawn] utility	523, 524
validation [dbvalid] utility	548, 549
write file [dbwrite] utility	552, 553
-fd option	
validation [dbvalid] utility	548, 549
-fi option	
validation [dbvalid] utility	548, 550
-fx option	
validation [dbvalid] utility	548, 550
-g option	
data source [dbdsn] utility	472, 475
Log Translation [dbtran] utilit	ty 508,
512	
service creation [dbsvc] utility	519, 521
transaction log [dblog] utility	527, 529
-ga option	
database server	144
-gb option	
database server	144
-gc option	
database server	144
-gd option	
database server	145
-ge option	146
database server	146
-gr option	146
database server	146
-gk option	146
uatabase server	146
-gi option	1 47
uatabase server	14/
-gm option	1 47
database server	147

-gn option		initialization [dbinit] utility	486, 488
database server	148	upgrade [dbupgrad] utility	543, 545
-gp option		–jdk option	
database server	149	initialization [dbinit] utility	486, 489
-gr option		upgrade [dbupgrad] utility	543, 545
database server	149	-jr option	,
-gss option		unload [dbunload] utility	534
database server	149	upgrade [dbupgrad] utility	543.545
-gt option		-k option	,
database server	150	initialization [dbinit] utility	486.489
-gu option		-l option	,,
database server	150	backup [dbbackup] utility	459, 460
-gx option	100	data source [dbdsn] utility	472, 475
database server	150	initialization [dbinit] utility	486, 490
-host option	100	license [dblic] utility	499 500
Interactive SOL [dbisal] utility	492	Log Translation [dbtran] utili	tv 510
495	472,	ning [dbning] utility	514 515
_i option		service creation [dbsvc] utilit	v519 521
initialization [dbinit] utility	188	m option	y517, 521
service creation [dbsvc] utility	521	-moption database server	151 160
upgrade [dbupgrad] utility	544	initialization [dbinit] utility	191,107
validation [dbyalid] utility	550	Log Translation [dbtran] utili	400, 490
ii option	550		ty 508,
unload [dhunload] utility 5	24 520	D12 ping [dhping] utility	514 515
il option	54, 559	transaction log [dblog] utility	514, 515
-II Option	27 520	unload [dhunload] utility	521, 529
in antian	21, 329	unioad [dounioad] unity	550 554
	500	write me [downie] unity	552, 554
Log Translation [dbtran] utility	508,	-n option	150 160
	27 520	backup [dbbackup] utility	459,460
transaction log [dblog] utility 5	27, 529	database option	1/0
-is option	500	nistogram [dbnist] utility	481, 482
Log Translation [dbtran] utility	508,	initialization [dbinit] utility	486, 490
512		Log Translation [dbtran] utili	ty 508,
-it option	500	512	1.50
Log Translation [dbtran] utility	508,	server option	152
512		setting database name	170
-ix option		transaction log [dblog] utility	527, 530
unload [dbunload] utility 5	34, 539	unload [dbunload] utility	534, 540
–j option		–nogui option	
Log Translation [dbtran] utility	508,	Interactive SQL [dbisql] utili	ty 492,
512		495	
unload [dbunload] utility 5	34, 539	–o option	
upgrade [dbupgrad] utility 5	43, 545	backup [dbbackup] utility	459, 460
–jConnect option		collation [dbcollat] utility	463, 464
Interactive SQL [dbisql] utility	492,	compression [dbshrink] utilit	y468, 469
495		data source [dbdsn] utility	472
–ja option		database server	153

erase [dberase] utility	479, 480
information [dbinfo] utility	483, 484
initialization [dbinit] utility	486, 490
license [dblic] utility	499, 500
Log Transfer Manger [dbltm]	utility
501, 503	-
Log Translation [dbtran] utili	ty 508,
operating quietly	130
ning [dhning] utility	514 516
server location [dblocate] util	514, 510
ston [dbston] utility	525 526
transaction log [dblog] utility	527,530
uncompression [dbexpand] u	tility 531
532	unity 551,
unload [dbunload] utility	534 540
ungrade [dbungrad] utility	543 546
validation [dbyalid] utility	548 550
write file [dbwrite] utility	552 554
o server option	552, 554
data source [dbdsn] utility	476
-oe option	470
logging startup errors	153
operating quietly	130
operating quictly	150
Interactive SOL [dbisal] utili	ty /02
	ty 4 <i>72</i> ,
-os option	
database server	154
Log Transfer Manger [db]tm	1.04 I utility
501 503	Jutility
-ot option	
Log Transfer Manger [dbltm]	l utility
501 503	juliili
-p option	
data source [dbdsn] utility	472, 476
database server	172, 170
initialization [dbinit] utility	486 490
license [dblic] utility	499, 500
service creation [dbsvc] utilit	v519 521
spown [dbspown] utility	573 574
spawn [ubspawn] unity	525, 524 534 540
nc option	554, 540
database server	154
ning [dhning] utility	514 515
_nd option	514, 515
ping [dbping] utility	514 515
ping [uoping] utility	517, 515

-port option	
Interactive SQL [dbisql] utilit	y 492,
496	
-ps option	
ping [dbping] utility	514, 516
-pt option	
database server	155
-q option	
backup [dbbackup] utility	459.461
collation [dbcollat] utility	463, 464
compression [dbshrink] utility	468.469
data source [dbdsn] utility	472.476
erase [dberase] utility	479 480
information [dbinfo] utility	483 484
initialization [dbinit] utility	486 490
Interactive SOL [dbisal] utilit	v /07
	y 472,
language [db]ang] utility	107 108
license [dblie] utility	400 500
Log Transfer Mangar [db]tm]	499, 500
501 502	utility
501, 505	5 00
Log Translation [dotran] utili	ty 508,
512	514 516
ping [dbping] utility	514, 516
server location [dblocate] util	ity 518
service creation [dbsvc] utility	y519, 521
spawn [dbspawn] utility	523, 524
stop [dbstop] utility	525, 526
transaction log [dblog] utility	527, 530
uncompression [dbexpand] ut	ility 531,
532	
unload [dbunload] utility	534, 540
upgrade [dbupgrad] utility	543, 546
validation [dbvalid] utility	548, 550
write file [dbwrite] utility	552, 554
-qi option	
database server	156
operating quietly	130
-qq option	
data source [dbdsn] utility	476
-qs option	
operating quietly	130
-r option	
backup [dbbackup] utility	459, 461
data source [dbdsn] utility	472, 476
database	170
database server	157

Log Translation [dbtran] utility 508, 512, 513
transaction log [dblog] utility 527, 530 unload [dbunload] utility 534, 540
-rg option
service creation [dbsvc] utility519, 521
service creation [dbsvc] utility519, 521
-rsu option Log Translation [dbtran] utility 508, 513
-s option
database server 158
Log Transfer Manger [dbltm] utility
Log Translation [dbtran] utility 508, 513
service creation [dbsvc] utility519, 521
write file [dbwrite] utility 552, 554
-sh ontion
database server 158
-sc option
database server 158
sroption
-sr option
-sr option Log Translation [dbtran] utility 508, 513
-sr option Log Translation [dbtran] utility 508, 513 -t option
-sr option Log Translation [dbtran] utility 508, 513 -t option backup [dbbackup] utility 459, 461
-sr option Log Translation [dbtran] utility 508, 513 -t option backup [dbbackup] utility 459, 461 histogram [dbhist] utility 481, 482
-sr option Log Translation [dbtran] utility 508, 513 -t option backup [dbbackup] utility 459, 461 histogram [dbhist] utility 481, 482 initialization [dbinit] utility 486 490
-sr option Log Translation [dbtran] utility 508, 513 -t option backup [dbbackup] utility 459, 461 histogram [dbhist] utility 481, 482 initialization [dbinit] utility 486, 490 Log Translation [dbtran] utility 508
-sr option Log Translation [dbtran] utility 508, 513 -t option backup [dbbackup] utility 459, 461 histogram [dbhist] utility 481, 482 initialization [dbinit] utility 486, 490 Log Translation [dbtran] utility 508 service creation [dbtran] utility 519
-sr option Log Translation [dbtran] utility 508, 513 -t option backup [dbbackup] utility 459, 461 histogram [dbhist] utility 481, 482 initialization [dbinit] utility 486, 490 Log Translation [dbtran] utility 508 service creation [dbsvc] utility519, 521 transaction log [dblog] utility527, 530
-sr option Log Translation [dbtran] utility 508, 513 -t option backup [dbbackup] utility 459, 461 histogram [dbhist] utility 481, 482 initialization [dbint] utility 486, 490 Log Translation [dbtran] utility 508 service creation [dbsvc] utility519, 521 transaction log [dblog] utility 527, 530 unload [dbunload] utility 524, 540
-sr option Log Translation [dbtran] utility 508, 513 -t option backup [dbbackup] utility 459, 461 histogram [dbhist] utility 481, 482 initialization [dbint] utility 486, 490 Log Translation [dbtran] utility 508 service creation [dbsvc] utility519, 521 transaction log [dblog] utility 527, 530 unload [dbunload] utility 534, 540 validation [dbvalid] utility 534, 550
-sr option Log Translation [dbtran] utility 508, 513 -t option backup [dbbackup] utility 459, 461 histogram [dbhist] utility 481, 482 initialization [dbinit] utility 486, 490 Log Translation [dbtran] utility 508 service creation [dbtran] utility 508 service creation [dbtran] utility 527, 530 unload [dbunload] utility 534, 540 validation [dbvaid] utility 548, 550 unitial [dbvaid] utility 522, 554
-sr option Log Translation [dbtran] utility 508, 513 -t option backup [dbbackup] utility 459, 461 histogram [dbhist] utility 481, 482 initialization [dbtran] utility 508 service creation [dbtran] utility 508 service creation [dbsvc] utility519, 521 transaction log [dblog] utility 527, 530 unload [dbunload] utility 534, 540 validation [dbvaid] utility 548, 550 write file [dbwrite] utility 552, 554
-sr option Log Translation [dbtran] utility 508, 513 -t option backup [dbbackup] utility 459, 461 histogram [dbhist] utility 481, 482 initialization [dbtran] utility 508 service creation [dbtran] utility 508 service creation [dbtran] utility 508 service creation [dbsvc] utility519, 521 transaction log [dblog] utility 527, 530 unload [dbunload] utility 534, 540 validation [dbvalid] utility 548, 550 write file [dbwrite] utility 552, 554 -ti option
-sr option Log Translation [dbtran] utility 508, 513 -t option backup [dbbackup] utility 459, 461 histogram [dbhist] utility 481, 482 initialization [dbtnit] utility 486, 490 Log Translation [dbtran] utility 508 service creation [dbsvc] utility519, 521 transaction log [dblog] utility 527, 530 unload [dbunload] utility 534, 540 validation [dbvalid] utility 548, 550 write file [dbwrite] utility 552, 554 -ti option database server 159
-sr option Log Translation [dbtran] utility 508, 513 -t option backup [dbbackup] utility 459, 461 histogram [dbhist] utility 481, 482 initialization [dbinit] utility 486, 490 Log Translation [dbtran] utility 508 service creation [dbsvc] utility519, 521 transaction log [dblog] utility 527, 530 unload [dbunload] utility 534, 540 validation [dbvalid] utility 548, 550 write file [dbwrite] utility 552, 554 -ti option database server 159 -tl option
-sr option Log Translation [dbtran] utility 508, 513 -t option backup [dbbackup] utility 459, 461 histogram [dbhist] utility 481, 482 initialization [dbinit] utility 486, 490 Log Translation [dbtran] utility 508 service creation [dbsvc] utility519, 521 transaction log [dblog] utility 527, 530 unload [dbunload] utility 534, 540 validation [dbvalid] utility 548, 550 write file [dbwrite] utility 552, 554 -ti option database server 159 -tl option data source [dbdsn] utility 472, 477
-sr option Log Translation [dbtran] utility 508, 513 -t option backup [dbbackup] utility 459, 461 histogram [dbhist] utility 481, 482 initialization [dbinit] utility 486, 490 Log Translation [dbtran] utility 508 service creation [dbsvc] utility519, 521 transaction log [dblog] utility 527, 530 unload [dbunload] utility 534, 540 validation [dbvalid] utility 548, 550 write file [dbwrite] utility 552, 554 -ti option database server 159 -tl option data source [dbdsn] utility 472, 477 database server 159
-sr option Log Translation [dbtran] utility 508, 513 -t option backup [dbbackup] utility 459, 461 histogram [dbhist] utility 481, 482 initialization [dbtnit] utility 486, 490 Log Translation [dbtran] utility 508 service creation [dbsvc] utility519, 521 transaction log [dblog] utility 527, 530 unload [dbunload] utility 534, 540 validation [dbvalid] utility 548, 550 write file [dbwrite] utility 552, 554 -ti option database server 159 -tl option data source [dbdsn] utility 472, 477 database server 159 -tmf option
-sr option Log Translation [dbtran] utility 508, 513 -t option backup [dbbackup] utility 459, 461 histogram [dbhist] utility 481, 482 initialization [dbinit] utility 486, 490 Log Translation [dbtran] utility 508 service creation [dbsvc] utility519, 521 transaction log [dblog] utility 527, 530 unload [dbunload] utility 534, 540 validation [dbvalid] utility 548, 550 write file [dbwrite] utility 552, 554 -ti option database server 159 -tl option database server 159 -tm option database server 159
-sr option Log Translation [dbtran] utility 508, 513 -t option backup [dbbackup] utility 459, 461 histogram [dbhist] utility 481, 482 initialization [dbinit] utility 486, 490 Log Translation [dbtran] utility 508 service creation [dbsvc] utility519, 521 transaction log [dblog] utility 527, 530 unload [dbunload] utility 534, 540 validation [dbvalid] utility 548, 550 write file [dbwrite] utility 552, 554 -ti option database server 159 -tl option database server 159 -tm option database server 160 -tmt option
-sr option Log Translation [dbtran] utility 508, 513 -t option backup [dbbackup] utility 459, 461 histogram [dbhist] utility 481, 482 initialization [dbinit] utility 486, 490 Log Translation [dbtran] utility 508 service creation [dbsvc] utility519, 521 transaction log [dblog] utility 527, 530 unload [dbunload] utility 534, 540 validation [dbvalid] utility 548, 550 write file [dbwrite] utility 552, 554 -ti option database server 159 -tl option database server 159 -tmf option database server 160 -tmt option database server 160
-sr option Log Translation [dbtran] utility 508, 513 -t option backup [dbbackup] utility 459, 461 histogram [dbhist] utility 481, 482 initialization [dbint] utility 486, 490 Log Translation [dbtran] utility 508 service creation [dbsvc] utility519, 521 transaction log [dblog] utility 527, 530 unload [dbunload] utility 534, 540 validation [dbvalid] utility 548, 550 write file [dbwrite] utility 552, 554 -ti option database server 159 -tl option database server 159 -tmf option database server 160 -tmt option database server 160

–u option	
database server	161
histogram [dbhist] utility	481, 482
information [dbinfo] utility	483, 484
license [dblic] utility	499, 500
Log Translation [dbtran] utilit	ty 508,
513	
service creation [dbsvc] utility	y 519
unload [dbunload] utility	534, 540
-ud option	
database server	162
Log Transfer Manger [dbltm] 501, 503	utility
-ut option	
database server	162
–v option	
data source [dbdsn] utility	472, 476
database server	162
Log Transfer Manger [dbltm] 501, 503	utility
unload [dbunload] utility	534, 540
–w option	
backup [dbbackup] utility	459, 461
data source [dbdsn] utility	472, 475
service creation [dbsvc] utility	y519, 522
-x option	
backup [dbbackup] utility	459, 461
collation [dbcollat] utility	463, 464
data source [dbdsn] utility	472, 477
database server	162
Interactive SQL [dbisql] utilit	y 492,
496	
Log Translation [dbtran] utili 513	ty 508,
service creation [dbsvc] utility	y 519
stop [dbstop] utility	525, 526
transaction log [dblog] utility	527, 530
-xi option	
unload [dbunload] utility	534, 540
-xo option	
backup [dbbackup] utility	459, 461
-xs option	
database server	164
-xx option	
unload [dbunload] utility	534, 540
-y option	
backup [dbbackup] utility	459, 461

collation [dbcollat] utility	463, 465
compression [dbshrink] utility	468, 469
data source [dbdsn] utility	472, 476
database server	165
erase [dberase] utility	479, 480
Log Translation [dbtran] utilit	y 508,
513	
service creation [dbsvc] utility	519, 522
stop [dbstop] utility	525, 526
uncompression [dbexpand] ut	ility 531,
unload [dbunload] utility	534, 540
write file [dbwrite] utility	552, 554
-z option	
collation [dbcollat] utility	463 465
data source [dbdsn] utility	472. 477
database server	34, 165
initialization [dbinit] utility	486, 491
Log Translation [dbtran] utilit	v 513
ping [dbping] utility	514.516
transaction log [dblog] utility	527, 530
-zl option	021,000
database server	166
-zo option	
database server	166
-zr option	
database server	167
-zs option	
database server	167
-cl option	
data source [dbdsn] utility	475
-qp option	
database server	156
-qs option	
database server	156
-qw option	
database server	157
% comment indicator	616
% operator	
modulo function	616
& UNIX command line	21
· · · · · · · · · · · · · · · · · · ·	
Α	

ActiveReq property description

Adaptive Server Anywhere

480	Server	10)9
508,	Address Windowing Extensions	3	
	setting cache size	13	38
522	ADO		
526	connecting	6	59
531,	AES encryption algorithm		
	initialization [dbinit] utility	48	38
540	unload [dbunload] utility	53	38
554	Alias property		
	description	66	54
465	ALL		
477	permissions	39	96
165	ALLOW_NULLS_BY_DEFAU	JLT	
491	option		
513	compatibility option	566, 56	57
516	description	57	74
530	obtaining value	64	48
	Open Client	12	20
166	alphabetic characters		
	defined	32	24
166	ALTER permissions	39	96
	ALTER PROCEDURE stateme	nt	
167	about	44	44
	effects	44	44
167	ALTER TABLE statement		
	REPLICATE ON	43	33
475	ANSI		
	about code pages	29	91
156	choosing code pages	31	12
	COMMIT behavior	586, 59	90
156	conformance	62	25
	cursors	57	74
157	delete permissions	57	75
616	update permissions	57	75
	variable behavior	57	74
616	ANSI_BLANKS option		
	compatibility option	566, 56	67
21	description	57	74
	obtaining value	64	48
	Open Client	12	20
	ANSI_CLOSE_CURSORS_ON	1_ -	
657	ROLLBACK		
0.57	option		

connecting using 9.0 sample data

Adaptive Server Anywhere as an Open

source setting up as an Open Server 48

112

compatibility option	567
description	574
obtaining value	648
ANSI INTEGER OVERFLOW	option
compatibility option	567
description	575
obtaining value	648
Open Client	120
ANSI PERMISSIONS option	
compatibility option	567
description	575
obtaining value	648
ANSI UPDATE CONSTRAIN	ГS
option	
compatibility option	567
description	576
obtaining value	648
ANSINULL option	
compatibility option	566, 567
description	577
obtaining value	648
Open Client	120
APC pw parameter	436
LTM configuration file	503
APC user parameter	436.445
LTM configuration file	503
APCs	
function APCs	444
Replication Server	427
AppInfo [APP] connection parar	neter
about	70
about connection parameters	174
description	175
introduction to connection par	ameters
39	
tips	73
using default connection para	meters 51
AppInfo property	
description	648
ASANY9 environment variable	
description	245
ASANYSH9 environment variab	ole
description	245
ASAProv	
OLE DB provider	68
asasrv.ini file	
conver information	82

troubleshooting server startup	35
ASCHARSET environment variable	
specifying character sets	245
ASCII	
character set	290
collating sequences	491
file format 595,	614
ASLANG environment variable	
specifying languages	245
ASTMP environment variable	
description	246
UNIX	248
asynchronous procedures	
about	445
Replication Server	427
user ID for	503
auditing	
controlling	578
recovering uncommitted operation 382	s
AUDITING option	
description	578
Log Translation [dbtran] utility	512
setting	561
AuditingTypes property	
obtaining value	648
AUTO_COMMIT option	
description	579
Interactive SQL settings	571
AUTO_REFETCH option	
description	579
Interactive SQL settings	571
autocommit	
ODBC configuration	57
autoexec.ncf	
automatic loading	6
AUTOMATIC TIMESTAMP option	
compatibility option	567
description	579
obtaining value	648
Open Client	120
automation	
administration tasks	267
AutoStart [ASTART] connection	
parameter	
about	70
about connection parameters	174
•	

description	176
introduction to connection parameters	
39	
tips	73
using default connection parameter	eters 51
AutoStop [ASTOP] connection par	rameter
about	70
about connection parameters	174
description	177
introduction to connection parameter	meters
39	
ODBC configuration	59
tips	73
using default connection parame	eters 51
Avail IO statistic	645
availability	
database servers	21
high	378
AvailIIO property	
description	657
AWE cache	
setting cache size	138

В

background	
running database servers	21
BACKGROUND_PRIORITY option	
description	580
obtaining value	648
setting	561
backup [dbbackup] utility	
about	458
exit codes	458
options	460
syntax	459
backup and data recovery	337
backup database wizard	
using	377
backup directory	
backup [dbbackup] utility	460
backup plans	
about	353
backups	
about	338
backup [dbbackup] utility	458
concepts	343
database only	460

349dbltm and3dbmlsync and3dbremote and3designing procedures3external3for remote databases3full3internal3	50 50 50 46 41 52 67 41 59 60 50
dbltm and3dbmlsync and3dbremote and3designing procedures3external3for remote databases3full3internal3	50 50 50 46 41 52 67 41 59 60 50
dbmisync and3dbremote and3designing procedures3external3for remote databases3full3internal3	50 50 46 41 52 67 41 59 .60 50
dbremote and3designing procedures3external3for remote databases3full3internal3	50 46 41 52 67 41 59 60 50
designing procedures3external3for remote databases3full3internal3	46 41 52 67 41 59 60 50
external 3 for remote databases 3 full 3 internal 3	41 52 67 41 59 60 50
for remote databases 3 full 3 internal 3	652 67 67 59 60 50
internal 3	67 41 59 60 50
internal 3	41 59 60 50
	60 50
internals of 3	60 50
live 3/8, 4	50 40
MobiLink and 3	
MobiLink consolidated databases 3	49
offline 3	41
online 3	41
options 4	.60
planning 3	47
rename and start new transaction log 461	5
replication 449.4	50
Replication Agent and 3	50
restrictions during 3	60
scheduling 3	47
SOL Remote and 3	50
SQL statement 3	641
strategies 3	47
Sybase Central 3	641
types of 3	641
unfinished 3	75
validating 3	54
batch files	
starting servers 5	23
batch mode	
for LTM 4	46
batch_ltl_cmds parameter	
LTM configuration file 5	03
batch_ltl_mem parameter	
LTM configuration file 5	03
batch_ltl_sz parameter	
LTM configuration file 5	03
BELL option	
description 5	80
Interactive SQL settings 5	71
bindery	
NetWare 1	00
blank padding	

initialization [dbinit] utility	487	BytesSentUncomp property	
BlankPadding property		description 6	48, 657
description	664		
blanks		C	
ANSI behavior	574	C2 property	
BLOB_THRESHOLD option		C2 property	(57
description	580	description	057
replication option	570	database server entions	120
BlobArenas property			129
description	664	maximum size	0/4
BlockedOn property		size	129
description	648	size option	12
BLOCKING option		cache bullers	25 127
description	581	performance 133, 1.	33-137
obtaining value	648	Cache Hits/sec statistic	
setting	561	Cache Reads Index Interior/sec sta	tistic
BLOCKING_TIMEOUT option			. (20
description	581	Cache Reads Index Leaf/sec statist	10 639
setting	561	Cache Reads Table/sec statistic	639
Broadcast [BCAST] communication		Cache Reads Total Pages/sec statis	tic 639
parameter		cache size	674
description	202	limit	6/4
BroadcastListener [BLISTENER]		limiting 1.	36, 137
communication parameter		setting	133
description	203	setting for windows Server 200	3 138
buffer size		setting for windows 2000	138
ODBC configuration	61	setting for windows AP	138
buffering		static	135
replication commands	446	Cache Size Current statistic	639
BufferMisses property		Cache Size Maximum statistic	639
description	657	Cache Size Minimum statistic	639
BuildChange property		Cache Size Peak statistic	639
description	657	cache statistics	639
BuildClient property		CacheHits property	10 664
description	657	description 6	48, 664
BuildReproducible property		CacheHitsEng property	
description	657	description	657
bulk loading		CacheRead property	10 664
option	13	description 6	48, 664
bulk operations		CacheReadIndInt property	
-b server option	133	description 6	48, 664
BytesReceived property		CacheReadIndLeaf property	
description 648	. 657	description 6	48, 664
BytesReceivedUncomp property	,	CacheRead lable property	10 654
description 648	, 657	description 6	48, 664
BytesSent property	,	CacheReplacements property	
description 648	. 657	description	657
010	,	caching	

server names	82	about	286
cannot find database server	78	application	303
Capabilities property		avoiding translation	319
description	664	choosing	325
capability ID	664	connection parameter	177
case sensitivity	486	definition	289
collations	323	determining	303
command line	8	encoding	286
connection parameters	72, 174	fixed width	297
database name	11	HTTP requests	234
initialization [dbinit] utility	487	LTM	447, 448
international aspects	299	multibyte	297, 315
server name	11	Replication Server	441
CaseSensitive property		server	303
description	664	single-byte	290
CD-ROM		specifying	245
databases	260	translation	329
deployment	157	Unicode	315
Certicom		UNIX default	303
encrypting client/server		variable width	297
communications	141	Windows	291
security	141	Windows default	303
CHAINED option		characters	
compatibility option	566, 567	alphabetic	324
description	581	digits	324
obtaining value	648	white space	324
Open Client	120	CharSet [CS] connection paramet	er
chained transaction mode	581	about	70
change log file settings wizard		about connection parameters	174
changing the location of a tra	ansaction	description	177
log	383	introduction to connection para	ameters
starting a transaction log min	ror for an	39	
existing database	385	overriding client character set	with 333
using	527	tips	73
changing		using default connection paran	neters 51
collations	332	CharSet property	
CHAR data type		description 648, 0	557, 664
and host variables	574	Checkpoint Flushes/sec statistic	639
character LTM	447	Checkpoint Log statistic	639
character set conversion		checkpoint logs	
enabling	137	about	360
character set translation		checkpoint statistics	639
about	329	Checkpoint Urgency statistic	639
disabling	329	CHECKPOINT_TIME option	
enabling	137, 329	description	582
error messages	317	obtaining value	648
character sets		setting	561

using	363	Windows	291
checkpoints		collation [dbcollat] utility	
backups	343	about	462
CHECKPOINT_TIME option	582	exit codes	462
interval between	144	options	463
log	360	syntax	463
scheduling	364	collation file	
urgency	363	editing	320
Checkpoints/sec statistic	639	collation sequence label	
CheckpointUrgency property		collation [dbcollat] utility	465
description	664	collation sequences	
Chkpt property		collation [dbcollat] utility	462
description	664	dbcollat command line syntax	463
ChkptFlush property		initialization [dbinit] utility	490
description	664	list	491
ChkptPage property		collation utility	
description	664	custom collations	30. 331
choosing drivers	43	collations	,
CIS option property		about	286
description	648	alternate	310
CIS ROWSET SIZE option		changing	332
description	582	choosing	807.325
setting	561	creating	330
Cis rowset size property		custom 320.3	30, 331
description	648	default	262
client liveness timeout		definition	289
data source [dbdsn] utility	477	extracting using Sybase Centra	1 462
client side		file format	320
setting encryption parameters 188	. 190	internals	320
ClientLibrary property	,	ISO 1	312
description	648	list of supplied	308
ClientPort [CPORT] communication		LTM 4	47.448
parameter		multibyte	315
description	204	OEM	313
clients		options	463
identifying	175	recommended for Windows and	1 UNIX
CLOSE_ON_ENDTRANS option		309	
compatibility option	567	Replication Server	441
description	583	sorting characters with	298
obtaining value	648	column names	
code pages		international aspects	299
ANSI	291	column permissions	396
definition	289	columns	
Interactive SQL [dbisql] utility	494	limitations	674
list	491	permissions	396
OEM	291	Comm Buffer Misses statistic	640
overview	290		

Comm Bytes Received	
Uncompressed/sec statistic	640
Comm Bytes Sent Uncompressed/sec	
statistic	640
Comm Bytes Sent/sec statistic	640
Comm Free Buffers statistic	640
Comm Licenses in Use statistic	640
Comm Multi-packets Received/sec	
statistic	640
Comm Multi-packets Sent/sec statistic	с
640	
Comm Packets Received	
Uncompressed/sec statistic	640
Comm Packets Received/sec statistic	640
Comm Packets Sent Uncompressed/set	ec
statistic	640
Comm Packets Sent/sec statistic	640
Comm Send Fails/sec statistic	640
Comm TotalBuffers statistic	640
comma delimited files 595.	614
command delimiter	
Interactive SOL [dbisg]] utility	494
command echo	591
command file	
option	131
command files	
adding simple encryption with dbf	hide
466	
command line	
case sensitivity	8
database server	124
in configuration file	131
options 10.	131
server	131
starting the server	8
command line utilities	
backup [dbbackup] syntax	459
collation [dbcollat] syntax	463
compression [dbshrink] syntax	468
console [dbconsole] syntax	471
data source [dbdsn] syntax	472
erase [dberase] syntax	479
file hiding [dbfhide] syntax	466
histogram [dbhist] syntax	481
information [dbinfo] syntax	483
initialization [dbinit] syntax	486
Interactive SOL [dbisal] syntax	492
Carlondil Plum	

language [dblang] syntax	497
license [dblic] syntax	499
log tranlsation [dbtran] syntax	508
log transfer manager [dbltm] synta	Х
501	
ping [dbping] syntax	514
rebuild syntax	517
server location [dblocate] syntax	518
service creation [dbsvc] syntax	519
spawn [dbspawn] syntax	523
stop [dbstop] syntax	525
transaction log [dblog] syntax	527
uncompression [dbexpand] syntax	531
unload [dbunload] syntax	534
upgrade [dbupgrad] syntax	543
validation [dbvalid] syntax	548
write file [dbwrite] syntax	552
COMMAND_DELIMITER option	
description	583
Interactive SQL settings	571
CommandLine property	
description	657
CommBufferSize [CBSIZE] connecti	on
parameter	
1	
about	70
about about connection parameters	70 174
about about about connection parameters description	70 174 178
about about connection parameters description introduction to connection parameter	70 174 178 ters
about about connection parameters description introduction to connection paramet 39	70 174 178 ters
about about connection parameters description introduction to connection paramet 39 TCP/IP	70 174 178 ters 95
about about connection parameters description introduction to connection paramet 39 TCP/IP tips	70 174 178 ters 95 73
about about connection parameters description introduction to connection parameter 39 TCP/IP tips using default connection parameter	70 174 178 ters 95 73 rs 51
about about connection parameters description introduction to connection parameter 39 TCP/IP tips using default connection parameter comments	70 174 178 ters 95 73 rs 51
about about connection parameters description introduction to connection parameter 39 TCP/IP tips using default connection parameter comments comment delimiters	70 174 178 ters 95 73 rs 51 616
about about connection parameters description introduction to connection parameter 39 TCP/IP tips using default connection parameter comments comment delimiters Commit property	70 174 178 ters 95 73 rs 51 616
about about connection parameters description introduction to connection paramet 39 TCP/IP tips using default connection parameter comments comment delimiters Commit property description	70 174 178 ters 95 73 rs 51 616 648
about about connection parameters description introduction to connection parameter 39 TCP/IP tips using default connection parameter comments comment delimiters Commit property description COMMIT statement	70 174 178 ters 95 73 rs 51 616 648
about about connection parameters description introduction to connection parameter 39 TCP/IP tips using default connection parameter comments comment delimiters Commit property description COMMIT statement and AUTO_COMMIT option	70 174 178 ters 95 73 rs 51 616 648 579
about about connection parameters description introduction to connection parameter 39 TCP/IP tips using default connection parameter comments comment delimiters Commit property description COMMIT statement and AUTO_COMMIT option LTM	70 174 178 ters 95 73 rs 51 616 648 579 438
about about connection parameters description introduction to connection parameter 39 TCP/IP tips using default connection parameter comments comment delimiters Commit property description COMMIT statement and AUTO_COMMIT option LTM COMMIT_ON_EXIT option	70 174 178 ters 95 73 rs 51 616 648 579 438
about about connection parameters description introduction to connection parameter 39 TCP/IP tips using default connection parameter comments comment delimiters Commit property description COMMIT statement and AUTO_COMMIT option LTM COMMIT_ON_EXIT option description	70 174 178 ters 95 73 rs 51 616 648 579 438 584
about about connection parameters description introduction to connection paramet 39 TCP/IP tips using default connection parameter comments comment delimiters Commit property description COMMIT statement and AUTO_COMMIT option LTM COMMIT_ON_EXIT option description Interactive SQL settings	70 174 178 ters 95 73 551 616 648 579 438 584 571
about about connection parameters description introduction to connection paramet 39 TCP/IP tips using default connection parameter comments comment delimiters Commit property description COMMIT statement and AUTO_COMMIT option LTM COMMIT_ON_EXIT option description Interactive SQL settings CommitFile property description	70 174 178 ters 95 73 rs 51 616 648 579 438 584 571
about about connection parameters description introduction to connection paramet 39 TCP/IP tips using default connection parameter comments comment delimiters Commit property description COMMIT statement and AUTO_COMMIT option LTM COMMIT_ON_EXIT option description Interactive SQL settings CommitFile property description commits	70 174 178 ters 95 73 rs 51 616 648 579 438 584 571 664
about about connection parameters description introduction to connection parameters 39 TCP/IP tips using default connection parameter comments comment delimiters Commit property description COMMIT statement and AUTO_COMMIT option LTM COMMIT_ON_EXIT option description Interactive SQL settings CommitFile property description commits	70 174 178 ters 95 73 rs 51 616 648 579 438 584 571 664
about about connection parameters description introduction to connection parameters 39 TCP/IP tips using default connection parameter comments comment delimiters Commit property description COMMIT statement and AUTO_COMMIT option LTM COMMIT_ON_EXIT option description Interactive SQL settings CommitFile property description commits COOPERATIVE_COMMIT	70 174 178 ters 95 73 rs 51 616 648 579 438 584 571 664

586		TDS
COOPERATIVE_COMMITS o	ption	TIME
586	•	Verify
DELAYED_COMMIT_TIMEO	UT	commun
option	589	about
DELAYED COMMITS option	590	databa
CommLink property		debug
description	648	encrvi
CommLinks [LINKS] connection		param
parameter		suppo
about	70	troubl
about connection parameters	174	commun
description	179	Compact
introduction to connection parat	neters	descri
39		company
options	16	license
narentheses	318	Company
ting	73	descri
ups	tors 51	comparin
CommNetworkLink property	21018 3 1	TIME
description	648	comparie
common table expressions	040	DATE
MAY DECUDSIVE ITEDATI	ONS	compatik
MAA_RECORSIVE_ITERATI	607	ANST
Option CommProtocol proporty	007	SOL
description	610	SQL
	048	
communication compression	101	compant
compression threshold	182	ALLU
communication parameters	172	ANSI
about	1/3	ANSI
Broadcast [BCAS1]	202	
BroadcastListener [BLISTENE]	RJ 203	
ClientPort [CPORT]	204	ANSI
DLL	206	ANSI
DoBroadcast [DOBROAD]	206	ANSI
ExtendedName [ENAME]	207	ANSI
HOST [IP]	207	AUTC
LDAP [LDAP]	97, 209	CHAI
list of	201	classif
LocalOnly [LOCAL]	209	CLOS
MyIP [ME]	213	CONT
PreFetchOnOpen	214	
ReceiveBufferSize [RCVBUFS2	Z] 214	CONV
RegisterBindery [REGBIN]	214	DATE
SearchBindery [BINSEARCH]	215	DATE
SendBufferSize [SNDBUFSZ]	215	DIVII
ServerPort [PORT]	215	ESCA

TDS	217
TIMEOUT [TO]	217
VerifyServerName [VERIFY]	218
communications	
about	93
database server 162.	164
debugging	165
encrypting 141, 143, 168, 188,	190
parameters	201
supported	94
troubleshooting	105
communications statistics	640
CompactPlatformVer	
description	657
company names	
license [dblic] utility	500
CompanyName property	
description	657
comparing	
TIMESTAMP	589
comparisons	
DATE data type	630
compatibility	
ANSI	566
SOL	566
Transact-SOL	566
compatibility options	
ALLOW NULLS BY DEFAULT	574
ANSI BLANKS	574
ANSI CLOSE CURSORS ON -	
ROLLBACK	
574	
ANSI INTEGER OVERFLOW	575
ANSI PERMISSIONS	575
ANSI UPDATE CONSTRAINTS	576
ANSINULL	577
AUTOMATIC TIMESTAMP	579
CHAINED	581
classification	559
CLOSE ON ENDTRANS	583
CONTINUE AFTER RAISERRO)R
584	
CONVERSION ERROR	585
DATE FORMAT	587
DATE ORDER	588
DIVIDE BY ZERO ERROR	591
ESCAPE CHARACTER	592

FIRE_TRIGGERS	592
FLOAT_AS_DOUBLE	593
initial settings	559
ISOLATION_LEVEL	597
list of	566, 567
NON KEYWORDS	610
ON TSOL ERROR	612
OPTIMISTIC WAIT FOR -	
COMMIT	
612	
PERCENT AS COMMENT	616
OUERY PLAN ON OPEN	619
OUOTED IDENTIFIER	620
RI TRIGGER TIME	623
SOL FLAGGER ERROR L	EVEL
625	LYEE
SQL_FLAGGER_WARNING	
	_
625	
STRING_RTRUNCATION	626
TIME FORMAT	628
TIMESTAMP FORMAT	629
TSOL HEX CONSTANT	632
TSOL VARIABLES	632
Compress [COMP] connection p	arameter
about	70
about connection parameters	174
description	181
introduction to connection par	rameters
39	
tips	73
using default connection para	meters 51
compress database wizard	
using	468
compressed databases	531
compressing with dbshrink	468
creating from Sybase Central	468
uncompressing	531
CompressedBTrees property	
description	664
compressing	
database files with dbshrink	468
packets	154
compression	101
compression [dbshrink] utility	468
options	469
performance	102
renormance	102

compression [dbshrink] utility	
about	468
exit codes	468
options	469
syntax	468
COMPRESSION option	
description	584
replication option	570
Compression property	
description	648, 664
CompressionThreshold [COMP	TH]
connection parameter	
about	70
about connection parameters	174
description	182
introduction to connection pa	arameters
39	
tips	73
using default connection para	ameters 51
configuration file	
creating LTM	445
database server	10
for the LTM	503
format for LTM	445
Log Transfer Manger [dbltm] 502] utility
LTM	436, 445
LTM command line	502
option	131
configuration files	
adding simple encryption with	h dbfhide
466	
hiding	466
configuring	
Adaptive Server Anywhere for	or
Replication Server	439, 440
interfaces file	114
LTM	445
ODBC data sources	56
SQL.ini	114
using dbconsole	471
ConnCount property	
description	664
connect dialog	
accessing	42
overview	44
connected users	405

managing	405	EncryptedPassy
connecting		Encryption [EN
ADO	69	Encryption Key
character sets	318	EngineName [H
Connect dialog overview	44	establishing con
database connection scenari	los 45	FileDataSource
firewalls 95, 203, 204, 20	7, 209, 215	ForceStart [FO]
from Interactive SQL	42	Idle [IDLE]
from Sybase Central	42	information [db
OLE DB	68	Integrated [INT
permission	394	Interactive SQL
RAS	96	introduction
starting a local server	46	Language [LA]
to a local database	46	LazyClose [LC
using a data source	48	list of
using LDAP	97, 518	LivenessTimeo
utility database	263	location of
Windows CE	63, 65, 66	Log Translation
connecting to a database	37	Logfile [LOG]
connection name		ODBC
ODBC configuration	62	overview
connection parameters		Password [PWI
about	173	ping [dbping] u
and connection strings	40	PrefetchBuffer
AppInfo [APP]	175	PrefetchRows [
AutoStart [ASTART]	176	priority
AutoStop [ASTOP]	177	ServerName [E
backup [dbbackup] utility	460	setting
case sensitivity	72, 174	StartLine [STA]
CharSet [CS]	177	stop [dbstop] ut
collation [dbcollat] utility	463	Unconditional
CommBufferSize [CBSIZE	1 178	unload [dbunlo
CommLinks [LINKS]	179	upgrade [dbupg
Compress [COMP]	181	Userid [UID]
Compression Threshold [C	OMPTH1	using default pa
182	,	validation [dbv
conflicts	73	connection proper
ConnectionName [CON]	183	Allow nulls by
console [dbconsole] utility	471	Ansi blanks
data source [dbdsn] utility	475	Ansi close cur
data sources	53	Ansi integer o
DatabaseFile [DBF]	183	Ansi permissic
DatabaseName [DBN]	184	Ansi undate co
DatabaseSwitches [DBS]	185	AnsiNull
DataSourceName [DSN]	186	AppInfo
DisableMultiRowFetch [DN	/RF1 187	AuditingTypes
embedded databases	73	Automatic tim
	10	· · · · · · · · · · · · · · · · · · ·

EncryptedPassword [ENP]	188
Encryption [ENC]	188
Encryption Key [DBKEY]	190
EngineName [ENG]	187
establishing connections	38
FileDataSourceName [FILEDSN]	191
ForceStart [FORCESTART]	191
Idle [IDLE]	192
information [dbinfo] utility	483
Integrated [INT]	192
Interactive SOL [dbisal] utility	494
introduction	39
Language [LANG]	193
LazyClose [LCLOSE]	193
list of 7	0 73
LivenessTimeout [LTO]	194
location of	77
Log Translation [dbtran] utility	511
Log Hansiation [dottan] utility	10/
ODBC	70
ouerview	174
Descuerd [DWD]	1/4
rasswold [rwD]	514
pring [uoping] utility	107
PrefetchBuller [PBUF]	19/
PreletchRows [PROwS]	198
priority	100
ServerName [ENG]	198
setting	40
StartLine [START]	199
stop [dbstop] utility	526
Unconditional [UNC]	199
unload [dbunload] utility	537
upgrade [dbupgrad] utility	544
Userid [UID]	199
using default parameters	51
validation [dbvalid] utility	549
connection properties	
Allow_nulls_by_default	648
Ansi_blanks	648
Ansi_close_cursors_on_rollback	648
Ansi_integer_overflow	648
Ansi_permissions	648
Ansi_update_constraints	648
AnsiNull	648
AppInfo	648
AuditingTypes	648
Automatic_timestamp	648

Background_priority	648	Fire_triggers	648
BlockedOn	648	Float_as_double	648
Blocking	648	FullCompare	648
BytesReceived	648	IdleTimeout	648
BytesReceivedUncomp	648	IndAdd	648
BytesSent	648	IndLookup	648
BytesSentUncomp	648	Isolation_level	648
CacheHits	648	Java_heap_size	648
CacheRead	648	Java_input_output	648
CacheReadIndInt	648	Java_namespace_size	648
CacheReadIndLeaf	648	Java_page_buffer_size	648
CacheReadTable	648	JavaHeapSize	648
Chained	648	Language	648
CharSet	648	LastIdle	648
Checkpoint_time	648	LastReqTime	648
CIS_option	648	LastStatement	648
Cis_rowset_size	648	list of	648
ClientLibrary	648	LivenessTimeout	648
Close_on_endtrans	648	Lock_rejected_rows	648
Commit	648	LockName	648
CommLink	648	LogFreeCommit	648
CommNetworkLink	648	Login_mode	648
CommProtocol	648	Login_procedure	648
Compression	648	LogWrite	648
Continue_after_raiserror	648	Max_cursor_count	648
Conversion_error	648	Max_plans_cached	648
Cooperative_commit_timeout	648	Max_statement_count	648
Cooperative_commits	648	Min_password_length	648
Cursor	648	Min_table_size_for_histogram	648
CursorOpen	648	Name	648
Database_authentication	648	Nearest_century	648
Date_format	648	NodeAddress	648
Date_order	648	Non-keywords	648
DBNumber	648	Number	648
Default_Timestamp_Increment	648	On_TSQL_error	648
Delayed_commit_timeout	648	Optimization_goal	648
Delayed_commits	648	Optimization_level	648
DiskRead	648	PacketSize	648
DiskReadIndInt	648	PacketsReceived	648
DiskReadIndLeaf	648	PacketsReceivedUncomp	648
DiskReadTable	648	PacketsSent	648
DiskWrite	648	PacketsSentUncomp	648
Divide_by_zero_error	648	Percent_as_comment	648
Encryption	648	Precision	648
Escape_character	648	Prefetch	648
EventName	648	Prepares	648
Extended_join_syntax	648	PrepStmt	648

Query_plan_on_open	648	description	183
QueryBypassed	648	introduction to connection para	neters
QueryCachedPlans	648	39	
QueryCachePages	648	tips	73
QueryLowMemoryStrategy	648	using default connection parame	eters 51
QueryOptimized	648	connections	
QueryReused	648	about	38
Quoted_identifier	648	connection property syntax	647
Recovery_time	648	creating	435
Replicate_all	648	default parameters	51
reporting	515	definition	38
ReqType	648	details	75
RI_trigger_time	648	disallowing with RAISERROR	603
Rlbk	648	dropping	159
RollbackLogPages	648	embedded database	47
Row_counts	648	from utilities	51
Scale	648	Interactive SQL	83
SQL_Flagger_error_level	648	limit	147
SQL_Flagger_warning_level	648	list of properties	648
String_rtruncation	648	liveness	159
Suppress_TDS_debugging	648	local database	45
TDS_empty_string_is_null	648	network	49
TempTablePages	648	overview	38
Time_format	648	performance	82
Timestamp_format	648	problems	75
TimeZoneAdjustment	648	programming interfaces	38
Truncate_timestamp_values	648	setting a maximum number	603
Truncate_with_auto_commit	648	simple	45
TSQL_Hex_constant	648	syntax of server-level property	657
TSQL_variables	648	troubleshooting 75, 83, 5	14, 518
UncommitOp	648	connectivity	
User_estimates	648	iAnywhere JDBC driver	43
Userid	648	jConnect	43
UtilCmdsPermitted	648	ConnsDisabled property	
Wait_for_commit	648	description	657
connection scenarios	45	console [dbconsole] utility	
connection strings		about	471
about	40	exit codes	471
character sets	318	options	471
introduction	39	syntax	471
representing	40	CONTINUE_AFTER_RAISERRO)R
CONNECTION_PROPERTY func	tion	option	
obtaining option settings	557	compatibility option 50	56, 567
ConnectionName [CON] connection	on	description	584
parameter		obtaining value	648
about	70	Open Client	120
about connection parameters	174	conventions	

documentation	xii	Replication Server connections	435
CONVERSION_ERROR option		subscriptions for Replication Serv	er
compatibility option	567	437	
description	585	users	394
obtaining value	648	creating databases	
COOPERATIVE_COMMIT_TIME	OUT	options	487
option		Sybase Central	485
description	586	Windows CE	312
obtaining value	648	CT-Library	110
setting	561	CURRENT USER	
COOPERATIVE_COMMITS optio	n	environment settings	249
description	586	CurrentCacheSize property	
obtaining value	648	description	657
setting	561	CurrentRedoPos property	
CPU		description	664
number used 1	2, 150	CurrIO property	
create backup images wizard		description	664
using 37	1.458	CurrRead property	
create connection statement	435	description	664
Replication Server	435	CurrWrite property	
create custom collation wizard		description	664
using	462	Cursor property	
CREATE DATABASE statement		description	648
permissions 1	2. 265	CursorOpen property	
utility database	262	description	648
create database wizard	202	cursors	0.0
list of collating sequences	490	and transactions	583
using	485	ANSI CLOSE CURSORS ON	-
CREATE DBSPACE statement	102	ROLLBACK option	
using	256	574	
create replication definition stateme	nt	closing	583
qualifying table owner for Replic	ration	connection limit 421	605
Server	436	database options	558
Replication Server	436	describing behavior	57
CREATE SUBSCRIPTION stateme	nt	ODBC configuration	57
Replication Server	437	opening	619
create write file wizard	457	redescribing	57
using	551	Cursors Open statistic	645
creating	551	Cursors statistic	645
databases	485	custom collations	0+5
databases with dbinit	405	about	320
dhamaaaa	256	about	220
dospaces	406	creating databases	221
ODBC data sources	52 55	using with collation [dbcollat] util	331
ODBC data sources with dbdar	55,55 177		шу
replication definitions for Deplication	412	405	
Server		exilip directory	041
Server	430	adout	241

D

daemon		
database server as	21	
Log Transfer Manger [dbltm] ut	tility	
503		
LTM	503	
Replication Agent	503	
running the server as	162	
data source [dbdsn] utility		
about	472	(
exit codes	474	
options	475	(
syntax	472	
data source description		
ODBC	56	
data source name		
ODBC	56	
data sources		,
about	41, 53	
configuring	56	
connection parameters	39	
creating	53, 55	
creating ODBC with dbdsn	472	
data source [dbdsn] utility	475	
embedded SQL	53	
example	48	
file	62	
Interactive SOL [dbisgl] utility	494	
ODBC	53	
UNIX	63	
using with jConnect	43	
Windows CE	63	
data type conversion		
errors	585	
data types		
supported	443	
database administration utilities		
about	455	
database administrator		
defined	390	
database connections		
ping [dbping] utility	515	
database file encryption	010	
initialization [dbinit] utility	488	
database files	.00	
backups	343	
compressing with dbshrink	468	
compressing with dosinink	-00	

erasing	478
limit	256
location	8
maximum size	674
media failure	379
NDS	130
ODBC configuration	59
paths	130
UNC filenames	130
uncompressing	531
database information	
getting with dbinfo	483
database name	
case sensitivity	11
ODBC configuration	59
option	10
setting	170
database options	555
about	556
AUDITING	578
BACKGROUND_PRIORITY	580
BLOCKING	581
BLOCKING_TIMEOUT	581
CHECKPOINT_TIME	582
CIS_ROWSET_SIZE	582
classification	559
COOPERATIVE_COMMIT	
TIMEOUT	
586	
COOPERATIVE_COMMITS	586
DEFAULT_TIMESTAMP	
INCREMENT	
589	
DELAYED_COMMIT_TIMEO	UT
589	
DELAYED_COMMITS	590
deleting settings	559
ESCAPE_CHARACTER	592
EXCLUDE_OPERATORS	592
finding values	557
FIRST_DAY_OF_WEEK	593
FOR_XML_NULL_TREATME	NT
594 CLODAL DATADAGE ID	
GLOBAL_DATABASE_ID	595
initial settings	559
JAVA_HEAP_SIZE	601
JAVA_INPUT_OUTPUT	601

JAVA_NAMESPACE_SIZE	602
list of compatibility options 566,	567
list of general database options	561
list of Interactive SQL options	571
list of replication options	570
LOGIN MODE	602
LOGIN PROCEDURE	603
MAX CURSOR COUNT	605
MAX HASH SIZE	606
MAX PLANS CACHED	606
MAX RECURSIVE ITERATION	IS
607	
MAX STATEMENT COUNT	608
MAX WORK TABLE HASH S	ZE
607	
MIN PASSWORD LENGTH	608
MIN TABLE SIZE FOR -	000
HISTOGRAM	
609	
NEAREST CENTURY option	600
ON CHAPSET CONVERSION	007
EALL LIDE	-
FAILURE	
Onen Client	120
OPTIMIZATION COAL	612
OPTIMIZATION_GOAL	10
DINNED CURSOR DEDCENT	10
FINNED_CURSOR_FERCENT	
OF_CACHE	
	(17
PRECISION	61/
PREFEICH	618
PRESERVE_SOURCE_FORMAT	618
PREVENT_ARTICLE_PKEY	
UPDATE	
619 DE 1 D. D. C. DEL ETED	60 0
READ_PAST_DELETED	620
RECOVERY_TIME	621
RETURN_DATE_TIME_AS	
STRING	
622	
RETURN_JAVA_AS_STRING	
ROW_COUNTS	623
SCALE	623 624
DETILL	623 624 624
scope and duration	623 624 624 558
scope and duration setting	623 624 624 558 556
scope and duration setting setting in Interactive SQL	623 624 624 558 556 571

startup settings	120
SUBSUME_ROW_LOCKS	627
SUPPRESS_TDS_DEBUGGING	627
TDS_EMPTY_STRING_IS_NUL	L
628	
TIME_ZONE_ADJUSTMENT	628
TRUNCATE DATE VALUES	630
TRUNCATE_TIMESTAMP	
VALUES	
631	
TRUNCATE_WITH_AUTO	
COMMIT	
631	
UPDATE STATISTICS	633
USER ESTIMATES	633
WAIT FOR COMMIT	634
database pages	
displaying size of	483
database performance and connection	
properties	637
database properties	
Alias	664
BlankPadding	664
BlobArenas	664
CacheHits	664
CacheRead	664
CacheReadIndInt	664
CacheReadIndLeaf	664
CacheReadTable	664
Capabilities	664
CaseSensitive	664
CharSet	664
CheckpointUrgency	664
Chkpt	664
ChkptFlush	664
ChkptPage	664
CommitFile	664
CompressedBTrees	664
Compression	664
ConnCount	664
CurrentRedoPos	664
CurrIO	664
CurrRead	664
CurrWrite	664
DBFileFragments	664
DiskRead	664
DiskReadIndInt	664

DiskReadIndLeaf	664	QueryCachedPlans	664
DiskReadTable	664	QueryCachePages	664
DiskWrite	664	QueryLowMemoryStrategy	664
DriveType	664	QueryOptimized	664
Encryption	664	QueryReused	664
ExtendDB	664	ReadOnly	664
ExtendTempWrite	664	RecoveryUrgency	664
File	664	RelocatableHeapPages	664
FileSize	664	RemoteTrunc	664
FileVersion	664	reporting	515
FreePageBitMaps	664	RollbackLogPages	664
FreePages	664	SeparateCheckpointLog	664
FullCompare	664	SeparateForeignKeys	664
GlobalDBId	664	SyncTrunc	664
Histograms	664	syntax	647, 664
IdleCheck	664	TableBitMaps	664
IdleChkpt	664	TempFileName	664
IdleChkTime	664	TempTablePages	664
IdleWrite	664	TransactionsSpanLogs	664
IndAdd	664	TriggerPages	664
IndLookup	664	VariableHashSize	664
IOToRecover	664	ViewPages	664
IQStore	664	database servers	
JavaHeapSize	664	automatic starting	6
JavaNSSize	664	command line	124
JDKVersion	664	locating	78, 83
Language	664	name	152
LargeProcedureIDs	664	name caching	82
list of	664	name option	10
LockTablePages	664	NetWare	6,130
LogFileFragments	664	options	10
LogFreeCommit	664	preventing from starting	176
LogName	664	quiet mode	130
LogWrite	664	running as a daemon	21
LTMGeneration	664	running in the background	21
LTMTrunc	664	shutting down	161
MapPages	664	silent	130
MaxIO	664	starting	5, 8
MaxRead	664	starting on Windows CE	9
MaxWrite	664	stopping	18, 199
MultiByteCharSet	664	UNIX	6
Name	664	Windows operating systems	5
PageRelocations	664	Windows services	21
PageSize	664	database sizes	
PreserveSource	664	limit	674
ProcedurePages	664	database utilities	
QueryBypassed	664	and database connections	51

backup [dbbackup]	458	tips	73
collation [dbcollat]	462	using default connection parameter	ers 51
compression [dbshrink]	468	databases	472
console [dbconsole]	471	accessing the Connect dialog	42
data source [dbdsn]	472	allocating space	258
erase [dberase]	478	altering dbspaces	258
file hiding [dbfhide]	466	automatic stopping	144
histogram [dbhist]	481	backing up from Sybase Central	458
information [dbinfo]	483	backup	338
initialization [dbinit]	485	character set	317
Interactive SQL [dbisql]	492	compressed	531
language [dblang]	497	compressing from Sybase Central	468
license [dblic]	499	connecting from Interactive SQL	42
Log Transfer Manager [dbltm]	501	connecting from Sybase Central	42
log translation [dbtran]	508	connecting to 3	8,75
ping [dbping]	514	connecting to a local database	46
rebuild [rebuild]	517	connection scenarios	45
server location [dblocate]	518	creating	485
service creation [dbsvc]	519	creating dbspaces	256
spawn [dbspawn]	523	creating with dbinit	486
stop [dbstop]	525	custom collations	331
transaction log [dblog]	527	deleting dbspaces	258
uncompression [dbexpand]	531	erasing	478
unload [dbunload]	533	information	483
upgrade [dbupgrad]	542	initializing	485
validation [dbvalid]	547	Java-enabling	545
write file [dbwrite]	551	large databases	256
DATABASE_AUTHENTICATION	N	maximum size	674
property		multiple-file	256
description	648	new	29
DatabaseFile [DBF] connection		page usage	483
parameter		permissions	389
about	70	permissions to load	147
about connection parameters	174	permissions to start	145
description	183	permissions to stop	146
embedded databases	47	permissions to unload	147
introduction to connection para	meters	properties of	664
39		read-only 13, 157, 170	, 260
tips	73	rebuilding	517
using default connection parame	eters 51	recovery	338
DatabaseName [DBN] connection		replicating entire	450
parameter		starting	20
about	70	stopping 20	, 525
about connection parameters	174	unloading	533
description	184	upgrading	542
introduction to connection parameters	meters	utilities 458, 462, 466, 468, 471,	, 478,
39		481, 483, 485, 492, 497, 49	99,

501, 508, 514, 517–519, 5	23,	dBA
525, 527, 531, 533, 542, 5	47,	dBA
551		dBA
utility	262	dbba
validating 36	7, 547	ab
validating from Sybase Central	547	ex
write files	551	fu
DatabaseSwitches [DBS] connectio	n	liv
parameter		op
about	70	sy
about connection parameters	174	dbcc
description	185	dbco
introduction to connection param	eters	ab
39		cu
tips	73	ex
using default connection parame	ters 51	or
DataSourceName [DSN] connection	n	sv
parameter		dbco
about	70	ab
about connection parameters	174	ex
description	186	or
introduction to connection param	eters	sv
39		dbds
tips	73	ab
using default connection parame	ters 51	ev
Windows CF	63	or
DATE data type	05	or sv
time values in	630	115
DATE FORMAT option	050	dhen
compatibility option	566	co
description	587	ne
obtaining value	648	pe sv
setting	561	dher
DATE ORDER option	501	sv
compatibility option	566	dher
description	588	ab
obtaining value	500 648	au
setting	561	01
dates	501	dhev
NEAREST CENTURY option	609	ab
time values in	630	au
DR Library	110	CA OT
DB-Library DBA authority	110	or
about	300	Sy Jhfhi
defined	390	uum
aronting	205	au
granning not inhoritable	393 406	SY
not inneritable	400	DBL

dBASE file format	595
dBASE II file format	595, 614
dBASE III file format	595, 614
dbbackup utility	
about	458
exit codes	458
full backup	367
live backup	378
options	460
syntax	459
dbcc function	529
dbcollat utility	
about	462
custom collations	330, 331
exit codes	462
options	463
syntax	463
dbconsole utility	
about	471
exit codes	471
options	471
syntax	471
dbdsn utility	
about	472
exit codes	474
options	475
syntax	472
using	55
dbeng9	
command line	124
personal database server	4
syntax	124
dberase [dberase] utility	
syntax	479
dberase utility	
about	478
exit codes	478
options	479
dbexpand utility	
about	531
exit codes	532
options	532
syntax	531
dbfhide utility	
about	466
syntax	466
DBFileFragments property	

description	664	dbltm utility	
dbhist utility		about	501
about	481	exit codes	502
exit codes	481	options	502
options	482	syntax	501
syntax	481	dbmlsrv	
dbinfo utility		licensing	499
about	483	DBNumber property	
exit codes	483	description	648
options	483	dbo user	
syntax	483	system objects	533
dbinit utility		dbping utility	
about	485	about	514
exit codes	485	exit codes	514
options	487	options	514
syntax	486	syntax	514
dbisql utility		using	83
about	492	dbshrink utility	
exit codes	493, 495	about	468
options	494	exit codes	468
syntax	492	options	469
DBKEY connection parameter		syntax	468
about	190	dbspace creation wizard	
dblang utility		using	257
about	497	dbspaces	
exit codes	498	altering	258
options	498	creating	256
syntax	497	deleting	258
dblgen9.res		limit	674
locating	242	dbspawn utility	
dblic utility		about	523
about	499	exit codes	523
exit codes	499	options	524
options	500	syntax	523
syntax	499	dbsrv	
dblocate utility		licensing	499
about	518	dbsrv9	
exit codes	518	command line	124
options	518	network database server	4
syntax	518	syntax	124
dblog utility		dbstop utility	
about	527	about	525
command line	528	exit codes	525
exit codes	528	options	525
options	529	permissions	12, 146
syntax	527	syntax	525
transaction log mirrors	385	using	18

dbsvc utility		obtaining value	648
about	519	setting	561
exit codes	520	DefaultCollation property	
options	521	about	262
syntax	519	description	657
dbtran utility		defaults	
about	508	connection parameters	51
command line	510	DELAYED_COMMIT_TIMEOU	Т
exit codes	510	option	
options	510	description	589
syntax	508	obtaining value	648
transaction logs	382	setting	561
uncommitted changes	382	DELAYED_COMMITS option	
dbunload utility		description	590
about	533	obtaining value	648
exit codes	536	setting	561
options	537	DELETE permissions	396
syntax	534	DELETE statement	442
dbupgrad utility		DELETE_OLD_LOGS option	
about	542	description	590
exit codes	544	replication and synchronization	n 590
options	544	replication and synchronization	option
syntax	543	570	
dbvalid utility		resetting truncation offset	529
about	547	transaction log options	529
exit codes	549	using	450
options	549	deletes	
syntax	548	ANSI behavior	575
using	367	Transact-SQL permissions	575
dbwrite utility		deleting	
about	551	dbspaces	258
exit codes	553	erase [dberase] utility	478
options	553	groups	412
syntax	552	integrated logins	87
debugging		services	521
event handlers	276, 282	users	403
SQL scripts	492	Delphi	
decimal precision		ODBC configuration for	57
database option	617	dependencies	
default character set		of services	31, 32
about	303	setting	521
UNIX	303	DESCRIBE_JAVA_FORMAT opt	ion
Windows	303	description	591
DEFAULT_TIMESTAMP		Interactive SQL settings	571
INCREMENT		describing	
option		cursor behavior	57
description	589	dial-up networking	

connections	96	disk v
digit characters		Disk V
defined	324	Disk V
digits		
maximum number	617	Disk V
directory structure	240	
DisableMultiRowFetch [DMRF]		Disk V
connection parameter		
about	70	Disk V
about connection parameters	174	Disk V
description	187	
introduction to connection parame	ters	Disk V
39		
tips	73	Disk V
using default connection paramete	rs 51	
disconnecting users		DiskR
using the Console [dbconsole] util	ity	des
471		DiskR
DISH server		des
about	219	DiskR
creating services	222	des
disk access		DiskR
options	141	des
Disk Active I/O statistics	641	disks
disk cache		frag
operating system	161	rec
disk controllers		DiskV
transaction log management	356	des
disk crashes		distrib
about	342	enl
disk full		DIVII
error writing to transaction log	355	cor
disk I/O statistics	641	DIVII
Disk Maximum I/O statistics	641	des
disk mirroring		obt
transaction logs	356	DLL
disk read statistics	642	loc
Disk Reads Active statistics	642	DLL (
Disk Reads Index interior/sec statistic	cs	des
642		DoBro
Disk Reads Index leaf/sec statistics	642	
Disk Reads Maximum Active statistic	cs	des
642		docun
Disk Reads Table/sec statistics	642	con
Disk Reads Total Pages/sec statistics	642	SQ
disk space		Drive
event example	273	OD

disk write statistics		642
Disk Writes Active statistics		642
Disk Writes Commit Files/sec st	atisti	cs
642		
Disk Writes Database Extends/se	ec	
statistics		642
Disk Writes Maximum Active st	atisti	cs
642		
Disk Writes Pages/sec statistics		642
Disk Writes Temp Extends/sec s	tatisti	ics
642		
Disk Writes Transaction Log Gr	oup	
Commits statistics	•	642
Disk Writes Transaction Log/sec	;	
statistics		642
DiskRead property		
description	648,	664
DiskReadIndInt property	,	
description	648.	664
DiskReadIndLeaf property	,	
description	648.	664
DiskReadTable property	,	
description	648.	664
disks	,	
fragmentation and performance	re.	258
recovery from failure of		379
DiskWrite property		517
description	648	664
distributed transactions	040,	004
enlistment		160
DIVIDE BY ZERO option		100
compatibility option		567
DIVIDE BY ZERO ERROR	ntion	507
description	puon	591
obtaining value		648
DLL		040
location		242
DLL communication parameter		272
description		206
DoBroadcast [DOBROAD]		200
communication parame	oter	
description		206
documentation		200
conventions		vii
SOL Anywhere Studio		лп v
Driver Not Capable arror		л
ODBC configuration		57
ODBC configuration		51

drivers	
choosing a driver for a connection	43
DriveType property	
description	664
dropping	
groups	412
users	403
DSEdit utility	
about	112
entries	116
not included with SQL Anywhere	
Studio	112
starting	114
using 114,	430
DSN connection parameter	
about	53
Windows CE	63

Ε

EBCDIC	
collating sequence	491
ECHO option	
description	591
Interactive SQL settings	571
embedded databases	
connecting	47
connection parameters	73
Java	47
starting	47
embedded SQL	
connections	38
interface library	76
encoding	
character sets	289
definition	289
multibyte character sets	323
encrypted passwords	
ODBC configuration	59
EncryptedPassword [ENP] connec	ction
parameter	
about	70
about connection parameters	174
description	188
introduction to connection para	meters
J7 tins	73
ups using default connection param	/J
using default connection param	

encryption	
Certicom	141
client/server communications 14	1, 143,
communications	217
creating databases with dbinit	486
file hiding [dbfhide] utility	460
network packets	+00
passwords	188
strong 141 143 168 1	100 88 190
Encryption [FNC] connection para	meter
about	70
about connection parameters	174
description	188
introduction to connection param 39	neters
tips	73
using default connection parame	eters 51
encryption algorithms	
initialization [dbinit] utility	488
unload [dbunload] utility	538
Encryption Key [DBKEY] connect	tion
parameter	
description	190
encryption keys	
compression [dbshrink] utility	469
erase [dberase] utility	479
initialization [dbinit] utility	488
Log Transfer Manger [dbltm] ut 502	ility
Log Translation [dbtran] utility	511
ODBC configuration	59
transaction log [dblog] utility	529
uncompression [dbexpand] utili	ty 532
unload [dbunload] utility	539
write file [dbwrite] utility	553
Encryption property	
connection property description	648
database property description	664
EngineName [ENG] connection	
parameter	
about	70
about connection parameters	174
character sets	73
description	187
introduction to connection para	neters
57	

tips	73	escape characters	
using default connection paramet	ers 51	unload [dbunload] utility	540
enlistment		ESCAPE_CHARACTER option	L
distributed transactions	160	compatibility option	566, 567
environment variables	51	description	592
about	244	obtaining value	648
ASANY9	245	estimates	
ASANYSH9	245	recovery time	621
ASCHARSET	245	row count	624
ASLANG	245	USER_ESTIMATES option	633
ASTMP	246	ethernet	106
ERRORLEVEL	495	euro symbol	
LD_LIBRARY_PATH	246	1252LATIN1 collation	313
PATH	246	event creation wizard	
setting	244	using	280
setting TEMP directory on CE	250	event handlers	
SQLCONNECT 5	1, 247	debugging	276, 282
SQLLOCALE deprecated	247	defined	268
SQLPATH	247	internals	279
SQLREMOTE	247	event handling	
SYBASE	248	about	267
TEMP	248	event log	
TEMPDIR	248	suppressing messages	130
TMP	248	event types	
environment-variable server option		about	278
database server	131	EventName property	
erase [dberase] utility		description	648
about	478	events	
exit codes	478	about	272
options	479	defined	268
syntax	479	handling	267, 268
erase database wizard		internals	278
using	478	limitations	674
erasing databases	478	scheduling	267
Sybase Central	478	examples	
error handling		directories	241
Interactive SQL	611	Excel file format	595, 614
Transact-SQL procedures	612	EXCLUDE_OPERATORS optic	on
error messages		description	592
character set translation	317	setting	561
ERRORLEVEL environment variab	ole	executable name	
Interactive SQL return code	495	license [dblic] utility	500
errors		execution threads	
divide by zero	591	number	148
HTTP codes	235	exit codes	
in Interactive SQL	611	backup [dbbackup] utility	458
Transact-SQL procedures	612	collation [dbcollat] utility	462
compression [dbshrink] utility	468		
--	-----		
console [dbconsole] utility	471		
data source [dbdsn] utility	474		
erase [dberase] utility	478		
histogram [dbhist] utility	481		
information [dbinfo] utility	483		
initialization [dbinit] utility	485		
Interactive SQL	495		
Interactive SQL [dbisql] utility	493		
language [dblang] utility	498		
license [dblic] utility	499		
Log Transfer Manager [dbltm] util 502	ity		
log translation [dbtran] utility	510		
ping [dbping] utility	514		
rebuild [rebuild] utility	517		
server location [dblocate] utility	518		
service creation [dbsvc] utility	520		
spawn [dbspawn] utility	523		
stop [dbstop] utility	525		
transaction log [dblog] utility	528		
uncompression [dbexpand] utility	532		
unload [dbunload] utility	536		
upgrade [dbupgrad] utility	544		
validation utility (dbvalid)	549		
write file [dbwrite] utility	553		
expanding			
database files	531		
explicit selectivity estimates			
USER_ESTIMATES option	633		
exporting data			
output format	614		
ExtendDB property			
description	664		
extended characters			
about	290		
collation [dbcollat] utility	464		
EXTENDED_JOIN_SYNTAX option	n		
description	592		
obtaining value	648		
setting	561		
ExtendedName [ENAME]			
communication parameter			
description	207		
ExtendTempWrite property			
description	664		
external functions			

stack size	146
F	
failure	
recovery from	338
feedback	550
documentation	vvi
providing	xvi
fetch operation	AVI
suppressing warnings	57
file data sources	57
creating	62
fla hiding [dhfhida] utility	02
	100
about	400
syntax	400
nie locations	0.41
NetWare	241
Windows CE	241
File property	
description	664
FileDataSourceName [FILEDSN]	
connection parameter	
about	70
about connection parameters	174
description	191
introduction to connection parame 39	ters
referencing file data sources	53
tips	73
using default connection paramete	rs 51
Windows CE	63
filename server option	
database server	131
files	
location	242
FileSize property	
description	664
FileVersion property	
description	664
FIPS	
SOL FLAGGER ERROR option	625
FIRE TRIGGERS ontion	020
compatibility option	567
description	592
obtaining value	6/8
firewalls	0-0
111 C 17 U110	

connecting across 95, 97,	203, 204,	GlobalDBId property	
207, 209, 215, 518		description	664
first row optimization option		GRANT statement	
OPTIMIZATION_GOAL	613	creating groups	406
FIRST_DAY_OF_WEEK optio	n	DBA authority	395
description	593	group membership	407
setting	561	new users	394
FIXED file format	595, 614	passwords	395
fixed width character sets		permissions	396
about	297	procedures	400
FLOAT_AS_DOUBLE option		RESOURCE authority	395
compatibility option	566, 567	table permissions	396
description	593	WITH GRANT OPTION	399
obtaining value	648	without password	410
Open Client	120	granting	
follow bytes		remote permissions	402
about	298, 318	group creation wizard	
FOR_XML_NULL_TREATME	ENT	using	406
option		group dependencies	
description	594	setting	521
ForceStart [FORCESTART] con	nnection	GROUP permissions	
parameter		not inheritable	406
description	191	groups	
format		creating	406
input file	595	deleting	412
forward log	343	deleting integrated logins	87
FoxPro file format	595, 614	granting integrated logins	85
fragmentation and performance	258	leaving	408
frame type	106	managing	406
FreeBuffers property		membership	407
description	657	of services	32
FreePageBitMaps property		permissions	392, 409
description	664	PUBLIC	411
FreePages property		remote permissions	402
description	664	revoking membership	408
full backups		setting dependencies	521
performing	367	setting options	393
FullCompare property		SYS	411
description	648, 664	without passwords	410
functions		-	
replicating	444	Н	
support for APC format	444	handling events	767 760
-		handming events	207, 208
G		transaction loss	254
		transaction logs	530

hash size

MAX_HASH_SIZE option

606

G

GLOBAL_DATABASE_ID option	
description	595
setting	561

MAX_WORK_TABLE_HASH_SIZE		
option	607	
hexadecimal constants		
data type	632	
high availability		
database servers	21	
live backups	378	
histogram [dbhist] utility		
about	481	
exit codes	481	
options	482	
syntax	481	
histograms		
minimum table size	609	
viewing with dbhist	481	
Histograms property		
description	664	
Host [IP] communication parameter		
description	207	
HTML file format	614	
HTTP		
default services	230	
server configuration	164	
HTTP server		
about	219	
character sets	234	
creating services	222	
errors	235	
interpreting URIs	227	
listening for HTTP requests	225	
quick start	221	
request handlers	229	
variables	232	
HTTPS		
server configuration	164	
I		
I/O		
idle	364	
operations and the database server	141	

215

43

495

29

xiv

IANA

icons

port number

iAnywhere JDBC driver

for running services

used in manuals

Interactive SQL [dbisql] utility

client applications		175
identifiers		
case insensitivity		299
international aspects		299
Idle [IDLE] connection parameter	er	
about		70
about connection parameters		174
description		192
introduction to connection par 39	amet	ers
tips		73
using default connection para	meter	s 51
Idle Actives/sec statistic		639
Idle Checkpoint Time statistic		639
Idle Checkpoints/sec statistic		639
idle I/O		
task		364
idle server		
event example		274
idle timeout		
ODBC configuration		60
Idle Writes/sec statistic		639
IdleCheck property		
description		664
IdleChkpt property		
description		664
IdleChkTime property		
description		664
IdleTime event		
polling		278
IdleTimeout property		
description	648,	657
IdleWrite property		
description		664
IN keyword		
CREATE TABLE statement		256
incremental backups		
backup [dbbackup] utility		458
IndAdd property		
description	648,	664
Index Adds/sec statistics		643
Index Full Compares/sec statistic	cs	643
Index Lookups/sec statistics		643
indexes		
hash size	606,	607
limitations		674

identification

statistics	643	security features	89,90
IndLookup property		using	85, 88
description	648, 664	Interactive SQL	
information [dbinfo] utility		AUTO_COMMIT option	579
about	483	command line	493
exit codes	483	exit codes	495
options	483	Java output	591
syntax	483	list of options 571, 579, 580, 58	3, 584,
INI files		591, 595, 597–600, 610,	511,
about	249	614–616, 626, 632	
initialization [dbinit] utility		setting options	571
about	485	starting from Sybase Central	492
exit codes	485	Interactive SQL [dbisql] utility	
options	487	about	492
syntax	486	exit codes	493
initializing databases with dbinit	t 486	options	494
INPUT_FORMAT option		syntax	492
description	595	Interactive SQL options	
Interactive SQL settings	571	AUTO_COMMIT	579
INSERT permissions	396	AUTO_REFETCH	579
INSERT statement	442	BELL	580
truncation of strings	626	classification	559
installation		COMMAND_DELIMITER	583
directory	240	COMMIT_ON_EXIT	584
NetWare	241	DESCRIBE_JAVA_FORMAT	591
registry settings	250	ECHO	591
Windows CE	241	initial settings	559
integer overflow		INPUT_FORMAT	595
ANSI behavior	575	ISQL_COMMAND_TIMING	597
Integrated [INT] connection para	ameter	ISQL_ESCAPE_CHARACTER	. 597
about	70	ISQL_FIELD_SEPARATOR	598
about connection parameters	174	ISQL_LOG	599
description	192	ISQL_PLAN	599
introduction to connection pa	rameters	ISQL_PRINT_RESULT_SET	600
	70	ISQL_QUOTE	600
tips	/3	list of	5/1
using default connection para	meters 51	NULLS	610
integrated login creation wizard	07	ON_ERROR	611
using	87	OUTPUT_FORMAT	614
integrated logins	05	OUTPUT_LENGTH	615
	85	OUTPUT_NULLS	010 571
	80	SELLING	5/1
delating	91	STATISTICS TRUNCATION LENCTH	620
network aspects	0/ 01	interface libraries	032
ODPC configuration	91 50		20
operating systems	20 05	locating	38 76
operating systems	65	locating	/0

interfaces file	
configuring	114
Log Transfer Manger [dbltm] utilit 502	У
Open Servers	430
internals	
backups	359
event handlers	279
event handling	278
events	278
schedules	278
international languages and character 285	sets
IOToRecover property	
description	664
IP address	
about	116
ping	105
IQStore property	
description	664
IsIQ property	
description	657
IsJavaAvailable property	
description	657
IsNetworkServer property	
description	657
ISO_1 collation	
about	312
isolation levels	
ODBC configuration	56
ISOLATION_LEVEL option	
compatibility option	566
description	597
obtaining value	648
Open Client	120
setting	561
ISQL_COMMAND_TIMING option	
description	597
Interactive SQL settings	571
ISQL_ESCAPE_CHARACTER optic	on
description	597
Interactive SQL settings	571
ISQL_FIELD_SEPARATOR option	
description	598
Interactive SQL settings	571
ISQL_LOG option	
description	599

Interactive SQL settings	571
ISQL_PLAN option	
description	599
Interactive SQL settings	571
ISQL_PRINT_RESULT_SET opt	tion
description	600
ISQL_QUOTE option	
description	600
Interactive SQL settings	571
IsRuntimeServer property	
description	657

J

Java		
and memory use	601,	602
connection parameters		73
enabling a database	2	545
file access		601
Java classes		
upgrade [dbupgrad] utility	1	545
Java in the database		
upgrading	1	544
Java VM statistics		643
java.io package		
support		601
JAVA_HEAP_SIZE option		
description		601
obtaining value		648
setting		561
JAVA_INPUT_OUTPUT option		
description		601
obtaining value		648
JAVA_NAMESPACE_SIZE optic	on	
description		602
obtaining value		648
setting	-	561
Java_page_buffer_size property		
description		648
JavaGlobFix property		
description		657
JavaHeapSize property		
description	648,	664
JavaNSSize property		
description		664
jConnect		
initialization [dbinit] utility	4	488
Interactive SQL [dbisql] utility	y .	495

TDS
upgrade [dbupgrad] utility
jConnect driver
JDBC
compatibility options
JDBC connectivity
JDKVersion property
description
JVM Global Fixed Size statistics
JVM Heap Size statistics
JVM Namespace Size statistics
jxmp directory
about

Κ

keyboard mapping	
about	289
keywords	
NON_KEYWORDS option	610
turning off	610

L

LANalyzer	106
language	
locale	301
registry settings	250, 497
language [dblang] utility	
about	497
exit codes	498
options	498
syntax	497
Language [LANG] connect	ion parameter
connection parameter	193
language DLL	
locating	242
registry settings	497
Language property	
description	648, 657, 664
language resource library	
messages file	289
registry settings	497
language support	
about	286
collations	325
multibyte character sets	315
overview	286
languages	

110	specifying	245
544	Langugae [LANG] connection para	ameter
43	about connection parameters	174
	LargeProcedureIDs property	
566	description	664
43	LastIdle property	
	description	648
664	LastReaTime property	0.0
643	description	648
643	LastStatement property	0.0
643	description	648
015	LazyClose [LCLOSE] connection	010
241	parameter	
211	about	70
	about connection parameters	174
	description	193
	introduction to connection parar	neters
289	39	neters
610	using default connection parame	eters 51
610	LD_LIBRARY_PATH environmen	t
	variable	
	description	246
100	LDAP [LDAP] communication par	ameter
106	description	209
201	LDAP servers	
301	connecting with 97, 20)9, 518
497	leaving	
	groups	408
497	LegalCopyright property	
498	description	657
498	LegalTrademarks property	
497	description	657
eter	libctl.cfg file	
193	DSEdit	115
	libraries	
242	ping [dbping] utility	515
497	license [dblic] utility	
	about	499
664	exit codes	499
	options	500
289	syntax	499
497	license type	
	license [dblic] utility	500
286	specifying with Log Transfer M	anger
325	[dbltm] utility	502
315	specifying with Log Translation	202
286	[dbtran] utility	510
	[uoualij ullity	510

description657language301LicensedCompany propertysetting327description657localhost machine name116LicensedUser propertyLocallolly [LOCAL] communication106description657parameter209adding with license [dblic] utility499LocalSystem account23description657options28LicensesinUse propertyabout23description657cocalSystem account243description657Lock_rejected_rows property243database servers499LockedHeapPages property48database servers499LockedHeapPages property57limitlocking657lockingconnections147OPTIMISTIC_WAIT_FORlimitations674612cache size674locking conflictscolumns674BLOCKING potion581databases674LockTablePages propertyindexes674description664live backups378log files528, 529backup [dbackup] utility460changing name from Sybase Centralregular backups and357527102liveness159rollback360connections159rollback362ubackup and70about361databases159rollback362connection parameters174cockraper Manager [dbltm] utili	LicenseCount property		character sets	317
LicensedCompany property setting 327 description 657 LicensedUser property description 657 licenses 657 Licenses 750 description 657 description 657 licenses 657 description 657 licenses database servers 499 synchronization servers 499 description 657 limit 00PTIMISTIC_WAIT_FOR COMMIT Adaptive Server Anywhere 674 databases 674 databases 674 databases 674 databases 674 description 658 databases 674 description 648 databases 674 description 648 databases 674 description 648 databases 674 description 648 databases 674 databases 674 description 648 databases 674 description 644 description 644 descri	description	657	language	301
description657localhost machine name116LicensedUser propertyparameterdescription657parameterlicensesdescription209adding with license [dblic] utility499LocalSystem accountLicenseStrype propertyabout23description657options28LicenseType propertyLOCATION registry entry243description657Lock_rejected_rows property468database servers499LockedHeapPages property57limitlocking657limitlocking657limitations147OPTIMISTIC_WAIT_FOR COMMIT657limitations674612cache size674locking conflictscolumns674BLOCKING option581databases674LockTablePages property581databases674description664live backups378log files528, 529backup [dbbackup] utility460changing name from Sybase Centralregular backups and357527527livenesscheckpoint log360connection parameters174components427description194exit codes502indexes174components427description194exit codes502indexes194components427description194exit codes502	LicensedCompany property		setting	327
LicensedUser property descriptionLocalOnly [LOCAL] communication parameterlicenses657parameterlicensesdescription209adding with license [dblic] utility499LocalSystem accountLicensesinUse propertyabout23description657options28LicenseType propertyLOCATION registry entry243description657Lock_rejected_rows propertydescription657lockingdatabase servers499description657limitlockingoptimistic657connections147OPTIMISTIC_WAIT_FORlimitlocking conflicts581databases674612cache size674BLOCKING option581events674BLOCKING option581events674LockName property581indexes674LockTablePages property648tables674LockTablePages property648tables674LockTablePages property648tables674LockTablePages property64temporary tables674LockTablePages property64twees674LockTablePages property64tables674LockTablePages property64tables674LockTablePages property64tables674LockTablePages property64tables674LockTablePages property64temporar	description	657	localhost machine name	116
description657parameter209licensesdescription203adding with license [dblic] utility499LocalSystem account23description657options28LicenseType propertyLOCATION registry entry243description657Lock-rejected_rows property483description657Lock-rejected_rows property483database servers499LockedHeapPages property591synchronization servers499description657limitlockingCOMMITconnections147OPTIMISTIC_WAIT_FORlimitationsCOMMITCOMMITAdaptive Server Anywhere674612columns674BLOCKING option581databases674BLOCKING option581events674LockTablePages property581indexes674LockTablePages property581indexes674description664live backups378log files528, 529backup [dbbackup] utility460changing name from Sybase Central662regular backups and357527111parametercomponents174components427description60transaction343LivenessTimeout [LTO] connection10g transfer Manager [dbltm] utility30502introduction to connection parameters174components427description194	LicensedUser property		LocalOnly [LOCAL] communication	n
licenses description 209 adding with license [dblic] utility 499 LicensesinUse property about 23 description 657 LicenseType property LocalSystem account 243 description 657 LicenseType property LocATION registry entry 243 description 657 Lock_rejected_rows property description 648 database servers 499 LockedHeapPages property 657 limit Locking Connections 147 Adaptive Server Anywhere 674 cache size 674 databases 674 databases 674 Adaptive Server Anywhere 674 cache size 674 limits 674 databases 674 bevents 674 databases 674 databases 674 bevents 674 tables 674 table	description	657	parameter	
adding with license [dblic] utility499LocalSystem accountLicensesinUse propertyabout23description657options28LicenseType propertyLOCATION registry entry243description657Lock_rejected_rows property48database servers499LockedHeapPages property57limitoptimisticdescription657limitoptimisticlocking657connections147OPTIMISTIC_WAIT_FOR COMMITCOMMITAdaptive Server Anywhere674612612cache size674locking conflicts581databases674BLOCKING option581databases674LockName property581indexes674LockTablePages property581indexes674LockTablePages property581indexes674LockTablePages property528, 529backup [dbbackup] utility460changin name from Sybase Centralregular backups and357527527liveness159rollback362ODBC configuration60transaction343LivenessTimeout [LTO] connection100syntax501about70about501components427description194exit codes502introduction to connection parameters174components427dscription194exit codes502introductio	licenses		description	209
LicensesinUse propertyabout23description657options28LicenseType propertyLock_rejected_rows property243description657Lock_rejected_rows propertylicensingdescription648database servers499LockedHeapPages propertysynchronization servers499descriptionlimitlocking0connections1470Adaptive Server Anywhere674612cache size674BLOCKING optioncolumns674BLOCKING optiondatabases674BLOCKING optiondatabases674LockTablePages propertyindexes674LockTablePages propertyindexes674BLOCKING optiontables674LockTablePages propertytemporary tables674descriptionfive backups378log filessout159rollbackconnections159rollbackconnection159rollbackconnection159rollbackconnection174about70about70about70about70about70about70about73using default connection parameters39log transfer Manager [dbltm] utilityuips73uips73uips73uips73uips73 <td>adding with license [dblic] utility</td> <td>499</td> <td>LocalSystem account</td> <td></td>	adding with license [dblic] utility	499	LocalSystem account	
description657options28LicenseType property657LOCATION registry entry243description657Lock_rejected_rows property443licensingdescription648database servers499LockedHeapPages propertysynchronization servers499description657limitlocking0connections147OPTIMISTIC_WAIT_FORlimitations674612cache size674612columns674BLOCKING option581databases674BLOCKING option581databases674LockTablePages propertyindexes674LockTablePages propertytemporary tables674descriptionfive backups378log files528, 529backup [dbbackup] utility460changing name from Sybase Centralregular backups and357527livenesscheckpoint log360connections159rollback362ODBC configuration60transaction343LivenessTimeout [LTO] connectionLog Transfer Manager [dbltm] utilityabout501about70about501components427description194exit codes502introduction to connection parameterssyntax501inters73options502introduction to connection parameterssyntax501ips73 <t< td=""><td>LicensesinUse property</td><td></td><td>about</td><td>23</td></t<>	LicensesinUse property		about	23
License Type property description 657 Lock_rejected_rows property description 648 database servers 499 limit connections 147 Adaptive Server Anywhere 674 cache size 674 databases 674 tables 674 tables 674 tables 674 tables 674 tables 674 LockING option 581 BLOCKING option 581 BLOCKING option 581 BLOCKING option 581 LockTablePages property COMMIT Adaptive Server Anywhere 674 tables 674 tables 674 tables 674 LockTablePages property temporary tables 674 liveness 159 ODBC configuration 60 connection parameters 174 about 70 about 70 about 70 about 501 connection parameters 51 lock amager [dbltm] utility about 501 connection parameters 51 log transfer Manager [dbltm] utility about 501 components 427 options 510 log transfer manager [dbltm] utility about 502 log transfer manager [dbltm] utility about 508 description 648, 657 exit codes 510 options 510	description	657	options	28
description657Lock_rejected_rows property description648licensingdatabase servers499LockedHeapPages property description657limitlocking657limitlocking657limitlockingCOMMITAdaptive Server Anywhere674612cache size674locking conflictscolumns674BLOCKING optiondatabases674BLOCKING optiondatabases674LockTablePages propertyindexes674LockTablePages propertyindexes674LockTablePages propertyindexes674LockTablePages propertytemporary tables674LockTablePages propertytemporary tables674LockTablePages propertytemporary tables674LockTablePages propertytemporary tables674descriptionregular backups and357527livenesscheckpoint log360connections159rollbackobout70about501about70about501about73options502introduction to connection parameters57439log transfer manager [dbltm] utilitytips73options502log transfer manager [dbltm] utilityabout508description648, 657exit codes510local machine648, 657exit codes510	LicenseType property		LOCATION registry entry	243
licensing description 648 database servers 499 synchronization servers 499 limit 657 limit 70 Adaptive Server Anywhere 674 cache size 674 columns 674 databases 674 databases 674 databases 674 databases 674 databases 674 databases 674 description 581 events 674 tables 674 live backups 378 liveness 159 connections 159 connection parameters 174 description 194 about 70 about connection parameters 174 components 427 description 194 introduction to connection parameters 174 components 427 description 194 introduction to connection parameters 174 components 427 description 194 introduction to connection parameters 174 components 427 description 194 exit codes 502 syntax 501 log transfer Manager [dbltm] utility about 501 conso 502 syntax 501 log transfer manager [dbltm] utility about 508 description 648, 657 exit codes 510 options 510	description	657	Lock_rejected_rows property	
database servers499LockedHeapPages propertysynchronization servers499description657limitlockinglockingconnections147OPTIMISTIC_WAIT_FORImitationsCOMMITAdaptive Server Anywhere674612cache size674locking conflictscolumns674BLOCKING option581databases674BLOCKING_TIMEOUT option581databases674LockName propertyindexes674description648tables674description664live backups378log files528, 529backup [dbbackup] utility460changing name from Sybase Centralregular backups and357527livenesscheckpoint log360connections159rollback362ODBC configuration60transaction343LivenessTimeout [LTO] connectionLog Transfer Manager [dbltm] utilityabout70about501about connection parameters174components427description194exit codes502introduction to connection parameterssyntax5013910g transfer manager [dbltm] utilityoptions510local machine648, 657exit codes510local machine648, 657exit codes510	licensing		description	648
synchronization servers499description657limitlockingconnections147ImitationsOPTIMISTIC_WAIT_FOR COMMITAdaptive Server Anywhere674612612cache size674columns674databases674backups674backup Server Se	database servers	499	LockedHeapPages property	
limitlockingconnections147OPTIMISTIC_WAIT_FORlimitations674612cache size674locking conflictscolumns674BLOCKING option581databases674BLOCKING_TIMEOUT option581events674LockName propertyindexes674LockName propertyindexes674LockTablePages propertytemporary tables674descriptionlive backups378log files528, 529backup [dbbackup] utility460changing name from Sybase Centralregular backups and357527livenesscheckpoint log360connections159rollback362ODBC configuration60transaction343LivenessTimeout [LTO] connectionLog Transfer Manager [dbltm] utility4bout501about70about501about501about connection parameters174components427description194exit codes502introduction to connection parameters10g transfer manager [dbltm] utilityips73options502using default connection parameters10g transfer intralign [dbltran] utilityips73options502using default connection parameters10g transfer intralign [dbltran] utilityips73options502using default connection parameters10g transfer intralign [dbltran] utility <td>synchronization servers</td> <td>499</td> <td>description</td> <td>657</td>	synchronization servers	499	description	657
connections147OPTIMISTIC_WAIT_FOR COMMITImitationsCOMMITAdaptive Server Anywhere674612cache size674locking conflictscolumns674BLOCKING option581databases674BLOCKING_TIMEOUT option581events674LockName property581indexes674LockName property64tables674LockTablePages property64tables674description664live backups378log files528, 529backup [dbbackup] utility460changing name from Sybase Central regular backups and557livenesscheckpoint log360connections159rollback362ODBC configuration60transaction343LivenessTimeout [LTO] connection parameterLog Transfer Manager111parameter174components427description194exit codes502introduction to connection parameters73options5023910g transfer manager [dbltm] utility10g translation [dbtran] utility508description648, 657exit codes510local machine648, 657exit codes510	limit		locking	
limitations COMMIT Adaptive Server Anywhere 674 612 cache size 674 locking conflicts columns 674 BLOCKING option 581 databases 674 BLOCKING_TIMEOUT option 581 databases 674 LockName property 581 events 674 description 648 tables 674 LockTablePages property 674 tables 674 description 664 live backups 674 description 664 live backups 674 description 664 live backups and 357 527 197 liveness checkpoint log 360 connections 159 rollback 362 ODBC configuration 60 transaction 343 LivenessTimeout [LTO] connection Log Transfer Manager [dbltm] utility 427 description 194 exit codes 501 about 70 about 501 about connection parameters 174 components	connections	147	OPTIMISTIC_WAIT_FOR	
Adaptive Server Anywhere 674 612 cache size 674 locking conflictscolumns 674 BLOCKING option 581 databases 674 BLOCKING_TIMEOUT option 581 events 674 LockName property 581 indexes 674 description 648 tables 674 LockTablePages property 664 live backups 378 log files 528 , 529 backup [dbbackup] utility 460 changing name from Sybase Centralregular backups and 357 527 livenesscheckpoint log 360 connections 159 rollback 362 ODBC configuration 60 transaction 343 LivenessTimeout [LTO] connectionLog Transfer Manager [dbltm] utility 400 about 70 about 501 about connection parameters 174 components 427 description 194 exit codes 502 introduction to connection parameters 573 options 501 39 Iog transfer manager [dbltm] utility 501 39 Iog transfer manager [dbltm] utility 501 194 exit codes 502 using default connection parameters 51log translation [dbtran] utility 39 Iog translation [dbtran] utility 402 501 39 Iog translation [dbtran] utility 402 502 using default connection parameters 502 10	limitations		COMMIT	
cache size 674 locking conflictscolumns 674 BLOCKING option 581 databases 674 BLOCKING_TIMEOUT option 581 events 674 LockName propertyindexes 674 description 648 tables 674 LockTablePages propertytemporary tables 674 description 664 live backups 378 log files $528, 529$ backup [dbbackup] utility 460 changing name from Sybase Centralregular backups and 357 527 livenesscheckpoint log 360 connections 159 rollback 362 ODBC configuration 60 transaction 343 Liveness Timeout [LTO] connectionLog Transfer Manager 111 parameterLog Transfer Manager 501 about 70 about 501 about connection parameters 174 components 39 Log transfer manager [dbltm] utilitytips 73 options 502 using default connection parameters 51log translation [dbtran] utility 508 description $648, 657$ exit codes 510 local machine $648, 657$ exit codes 510	Adaptive Server Anywhere	674	612	
columns 674 BLOCKING option 581 databases 674 BLOCKING_TIMEOUT option 581 events 674 LockName propertyindexes 674 description 648 tables 674 LockTablePages propertytemporary tables 674 description 664 live backups 378 log files $528, 529$ backup [dbbackup] utility 460 changing name from Sybase Centralregular backups and 357 527 livenesscheckpoint log 360 connections 159 rollback 362 ODBC configuration 60 transaction 343 LivenessTimeout [LTO] connectionLog Transfer Manager [dbltm] utility $about$ 501 about 70 about 501 $about$ 501 about connection parameters 174 components 427 description 194 exit codes 502 introduction to connection parameters $syntax$ 501 39 log transfer manager [dbltm] utility 501 using default connection parameters 51log translation [dbtran] utility 508 using default connection parameters 51log translation [dbtran] utility 508 description $648, 657$ exit codes 510 local machine 506 exit codes 510	cache size	674	locking conflicts	
databases674BLOCKING_TIMEOUT option581events674LockName propertyindexes674description648tables674LockTablePages propertytemporary tables674description664live backups378log files528, 529backup [dbbackup] utility460changing name from Sybase Centralregular backups and357527livenesscheckpoint log360connections159rollback362ODBC configuration60transaction343LivenessTimeout [LTO] connectionLog Transfer Manager111parameterLog Transfer Manager111about70about501about connection parameters174components427description194exit codes502introduction to connection parameterssyntax50139log transfer manager [dbltm] utility502using default connection parameters 51log translation [dbtran] utilitytips73options502using default connection parameters 51log translation [dbtran] utilityLivenessTimeout propertyabout508description648, 657exit codes510local machineoptions510	columns	674	BLOCKING option	581
events 674 LockName propertyindexes 674 description 648 tables 674 LockTablePages propertytemporary tables 674 description 664 live backups 378 log files $528, 529$ backup [dbbackup] utility 460 changing name from Sybase Centralregular backups and 357 527 livenesscheckpoint log 360 connections 159 rollback 362 ODBC configuration 60 transaction 343 LivenessTimeout [LTO] connectionLog Transfer Manager 111 parameter $Log Transfer Manager [dbltm] utility427about70about501about connection parameters174components4273910g transfer manager [dbltm] utility502using default connection parameters 51log translation [dbtran] utility508description648, 657exit codes510local machine648, 657exit codes510$	databases	674	BLOCKING_TIMEOUT option	581
indexes674description648tables674LockTablePages propertytemporary tables674description664live backups378log files528, 529backup [dbbackup] utility460changing name from Sybase Centralregular backups and357527livenesscheckpoint log360connections159rollback362ODBC configuration60transaction343LivenessTimeout [LTO] connectionLog Transfer Manager111parameterLog Transfer Manager [dbltm] utility427about70about501about connection parameters174components427description194exit codes502introduction to connection parameterssyntax50139log transfer manager [dbltm] utilitytips73options502using default connection parameters 51log translation [dbtran] utilityLivenessTimeout propertyabout508description648, 657exit codes510local machineoptions510	events	674	LockName property	
tables674LockTablePages propertytemporary tables674description664live backups378log files528, 529backup [dbbackup] utility460changing name from Sybase Centralregular backups and357527livenesscheckpoint log360connections159rollbackODBC configuration60transactionparameterLog Transfer Manager111parameterLog Transfer Manager [dbltm] utilityabout70about501about connection parameters174components427description194exit codes502introduction to connection parameterssyntax50139log transfer manager [dbltm] utility502using default connection parameters 51log translation [dbtran] utility508description648, 657exit codes510local machineoptions510508	indexes	674	description	648
temporary tables674description664live backups378log files528, 529backup [dbbackup] utility460changing name from Sybase Centralregular backups and357527livenesscheckpoint log360connections159rollback362ODBC configuration60transaction343LivenessTimeout [LTO] connectionLog Transfer Manager111parameterLog Transfer Manager [dbltm] utilityabout501about70about501about connection parameters174components427description194exit codes502introduction to connection parameterssyntax50139log transfer manager [dbltm] utility502using default connection parameters 51log translation [dbtran] utility508description648, 657exit codes510local machineoptions510	tables	674	LockTablePages property	
live backups378log files528, 529backup [dbbackup] utility460changing name from Sybase Centralregular backups and357527livenesscheckpoint log360connections159rollback362ODBC configuration60transaction343LivenessTimeout [LTO] connectionLog Transfer Manager111parameterLog Transfer Manager [dbltm] utilityabout501about70about501about connection parameters174components427description194exit codes502introduction to connection parameterssyntax50139log transfer manager [dbltm] utility502using default connection parameters 51log translation [dbtran] utility508description648, 657exit codes510local machineoptions510	temporary tables	674	description	664
backup [dbbackup] utility460 regular backups andchanging name from Sybase Central 527liveness357527liveness159checkpoint log360connections159rollback362ODBC configuration60transaction343LivenessTimeout [LTO] connectionLog Transfer Manager111parameterLog Transfer Manager [dbltm] utilityabout501about70about501about connection parameters174components427description194exit codes502introduction to connection parameterssyntax50139log transfer manager [dbltm] utility502using default connection parameters 51log translation [dbtran] utility508description648, 657exit codes510local machine0ptions510502	live backups	378	log files 528	, 529
regular backups and357527liveness527liveness159connections159rollback362ODBC configuration60transaction343LivenessTimeout [LTO] connectionLog Transfer ManagerparameterLog Transfer Manager [dbltm] utilityabout70about connection parameters174components427description194exit codes502introduction to connection parameterssyntax39log transfer manager [dbltm] utilitytips73options502using default connection parameters 51log translation [dbtran] utilityLivenessTimeout propertyabout508description648, 657exit codes510local machineoptions510	backup [dbbackup] utility	460	changing name from Sybase Cent	ral
livenesscheckpoint log360connections159rollback362ODBC configuration60transaction343LivenessTimeout [LTO] connectionLog Transfer Manager111parameterLog Transfer Manager [dbltm] utility4bout501about70about501about connection parameters174components427description194exit codes502introduction to connection parameterssyntax50139log transfer manager [dbltm] utilitytips73options502using default connection parameters 51log translation [dbtran] utility508description648, 657exit codes510local machineoptions510	regular backups and	357	527	
connections159rollback362ODBC configuration60transaction343LivenessTimeout [LTO] connectionLog Transfer Manager111parameterLog Transfer Manager [dbltm] utility101about70about501about connection parameters174components427description194exit codes502introduction to connection parameterssyntax50139log transfer manager [dbltm] utilitytips73options502using default connection parameters 51log translation [dbtran] utilityLivenessTimeout propertyabout508description648, 657exit codes510local machineoptions510	liveness		checkpoint log	360
$\begin{array}{ccccccc} ODBC \ configuration & 60 & transaction & 343 \\ LivenessTimeout [LTO] \ connection & Log \ Transfer \ Manager \ [dbltm] \ utility \\ parameter & Log \ Transfer \ Manager \ [dbltm] \ utility \\ \ about & 70 & about & 501 \\ about \ connection \ parameters \ 174 & components & 427 \\ description & 194 & exit \ codes & 502 \\ introduction \ to \ connection \ parameters & syntax & 501 \\ \ 39 & log \ transfer \ manager \ [dbltm] \ utility \\ tips & 73 & options & 502 \\ using \ default \ connection \ parameters \ 51 & log \ translation \ [dbtran] \ utility \\ LivenessTimeout \ property & about & 508 \\ description & 648, \ 657 & exit \ codes & 510 \\ local \ machine & options & 510 \end{array}$	connections	159	rollback	362
LivenessTimeout[LTO] connectionLog Transfer Manager111parameterLog Transfer Manager [dbltm] utilityabout70about connection parameters174description194exit codes502introduction to connection parameterssyntax39log transfer manager [dbltm] utilitytips73options502using default connection parameters 51log translation [dbtran] utilityLivenessTimeout propertyaboutdescription648, 657local machineoptions510	ODBC configuration	60	transaction	343
parameterLog Transfer Manager [dbltm] utilityabout70about501about connection parameters174components427description194exit codes502introduction to connection parameterssyntax50139log transfer manager [dbltm] utilitytips73options502using default connection parameters 51log translation [dbtran] utilityLivenessTimeout propertyabout508description648, 657exit codes510local machineoptions510	LivenessTimeout [LTO] connection		Log Transfer Manager	111
about70about501about connection parameters174components427description194exit codes502introduction to connection parameterssyntax50139log transfer manager [dbltm] utilitytips73options502using default connection parameters 51log translation [dbtran] utilityLivenessTimeout propertyabout508description648, 657exit codes510local machineoptions510	parameter		Log Transfer Manager [dbltm] utility	/
about connection parameters174components427description194exit codes502introduction to connection parameterssyntax50139log transfer manager [dbltm] utilitytips73options502using default connection parameters 51log translation [dbtran] utilityLivenessTimeout propertyabout508description648, 657exit codes510local machineoptions510	about	70	about	501
description194exit codes502introduction to connection parameterssyntax50139log transfer manager [dbltm] utilitytips73options502using default connection parameters 51log translation [dbtran] utilityLivenessTimeout propertyabout508description648, 657exit codes510local machineoptions510	about connection parameters	174	components	427
introduction to connection parameterssyntax50139log transfer manager [dbltm] utilitytips73options502using default connection parameters 51log translation [dbtran] utilityLivenessTimeout propertyabout508description648, 657exit codes510local machineoptions510	description	194	exit codes	502
39log transfer manager [dbltm] utilitytips73options502using default connection parameters 51log translation [dbtran] utilityLivenessTimeout propertyabout508description648, 657exit codes510local machineoptions510	introduction to connection parame	eters	syntax	501
tips73options502using default connection parameters 51log translation [dbtran] utilityLivenessTimeout propertyabout508description648, 657exit codes510local machineoptions510	39		log transfer manager [dbltm] utility	
using default connection parameters 51log translation [dbtran] utilityLivenessTimeout propertyabout508description648, 657exit codes510local machineoptions510	tips	73	options	502
LivenessTimeout property descriptionabout508 exit codeslocal machine648, 657exit codes510options510	using default connection parameter	ers 51	log translation [dbtran] utility	
description648, 657exit codes510local machineoptions510	LivenessTimeout property		about	508
local machine options 510	description 648	8,657	exit codes	510
*	local machine		options	510
environment settings 249 syntax 508	environment settings	249	syntax	508
locales log translation utility 382	locales		log translation utility	382
about 301 Logfile [LOG] connection parameter	about	301	Logfile [LOG] connection parameter	

about	70
about connection parameters	174
description	194
introduction to connection part	rameters
39	
tips	73
using default connection para	meters 51
LogFileFragments property	
description	664
LogFreeCommit property	
description	648, 664
logging off	
keeping server running	162
LOGIN_MODE option	
description	602
integrated logins	86
obtaining value	648
setting	561
LOGIN_PROCEDURE option	
description	603
disallowing connections with	
RAISERROR	603
obtaining value	648
setting	561
logins	
integrated	85
LogName property	
description	664
LogWrite property	
description	648, 664
LOTUS file format	
INPUT_FORMAT option	595
OUTPUT_FORMAT option	614
lower code page	
about	290
LTM	
and interfaces file	502
character set configuration	448
character sets	447
collations	447, 448
configuration	430, 445
configuration file	436, 503
Open Client/Open Server cha	racter
sets	447
Open Client/Open Server coll	ations
447	
starting	436

supported operations	442
transaction log management	450
transaction log options	528, 529
LTM configuration file	
about	445
character sets	448
creating	445
format	145
I TM utility	44.5
about	501
about	407
supported operating systems	427
syntax	501
LTM_admin_pw parameter	436
LTM configuration file	503
LTM_admin_user parameter	436
LTM configuration file	503
LTM_charset parameter	436
LTM configuration file	448, 503
LTM_language parameter	
LTM configuration file	448
LTM sortorder parameter	
LTM configuration file	448
LTMGeneration property	
description	664
I TMTrunc property	004
description	664
description	004

Μ

MachineName property	
description	657
MAGIC	
USER_ESTIMATES option	633
Main Heap Bytes statistic	645
MainHeapBytes property	
description	657
MainHeapPages property	
description	657
maintenance user	
primary sites	431
replicate sites	434
user ID	439
managing connected users	405
MapPages property	
description	664
MAX_CURSOR_COUNT option	
description	605
obtaining value	648

setting	561
MAX_HASH_SIZE option	
description	606
setting	561
MAX_PLANS_CACHED option	
description	606
obtaining values	648
setting	561
MAX_RECURSIVE_ITERATIONS	
option	
description	607
MAX_STATEMENT_COUNT option	n
description	608
obtaining value	648
setting	561
MAX_WORK_TABLE_HASH_SIZI	E
option	
description	607
setting	561
MaxCacheSize property	
description	657
maximum size	
database files	674
databases	674
MaxIO property	
description	664
MaxMessage property	
description	657
MaxRead property	
description	664
MaxWrite property	
description	664
MDSR encryption algorithm	
initialization [dbinit] utility	488
unload [dbunload] utility	538
media failures	
about	342
protection	355
recovery	344
recovery of	379
Mem Pages Lock Table statistics	644
Mem Pages Locked Heap statistics	644
Mem Pages Main Heap statistics	644
Mem Pages Map Pages statistics	644
Mem Pages Procedure Definitions	
statistics	644
Mem Pages Relocatable statistics	644

Mem Pages Relocations/sec statistics	644
Mem Pages Rollback Log statistics	644
Mem Pages Trigger Definitions statist	ics
644	
Mem Pages View Definitions statistic	s
644	
membership	
revoking group membership	408
memory	
cache size 133, 135-	-138
connection limit	421
Java and memory use	601
memory pages statistics	644
memory use	
and Java	602
Message Agent	
transaction log management	352
Message property	
description	657
messages	
language resource library	289
MessageText property	
description	657
MessageTime property	
description	657
MessageWindowSize property	
description	657
Microsoft Access	
ODBC configuration for	57
TIMESTAMP comparison	589
Microsoft Visual Basic	
ODBC configuration for	57
MIN PASSWORD LENGTH option	
description	608
obtaining value	648
setting	561
MIN TABLE SIZE FOR -	
HISTOGRAM	
option	
description	609
obtaining value	648
setting	561
MinCacheSize property	
description	657
minimal column definitions	443
mirror	
transaction log 343. 344. 384.	385
<i></i>	

miscellaneous statistics	645	starting protocol	16
MobiLink synchronization		supported protocols	94
backups	350	national language support	
purposes	426	about	286
modulo operator		collations	325
PERCENT_AS_COMMENT	option	multibyte character sets	315
616		NativeProcessorArchitecture pro	operty
monitoring		description	657
performance with histograms	481	NDIS	
who is logged on	471	drivers	105
monitoring performance	638	NDS	
MSDASQL OLE DB provider		filenames	130
about	68	NEAREST_CENTURY option	
multi-processing		compatibility option	567
support	14	description	609
multi-processor machines		obtaining value	648
option	12	net.cfg file	106
multi-processor support	14	NetWare	
multi-tasking	14	bindery	100
multibyte character sets		cache buffers	133
about	297	database servers	6,130
using	315	enabling character set conver	sion 137
multibyte characters		file locations	241
encodings	323	installation	241
properties	324	network adapter settings	106
MultiByteCharSet property		server performance	133
description	664	subdirectories	241
MultiPacketsReceived property		threading behavior	14
description	657	network adapters	
MultiPacketsSent property		drivers	105
description	657	network communications	34, 35
multiple databases		command line options	201
DSEdit entries	116	network connections	
multiple record fetching		option	16
data source [dbdsn] utility	476	network drives	
ODBC configuration	62	database files	8
MyIP [ME] communication param	neter	network protocols	
description	213	about	93
		network connections	49
Ν		ODBC configuration	60
name		supported	94
database	170	troubleshooting	105
Name property	170	network server	
description 649	657 664	about	4
Named Pines	0.57,004	connecting to	49
about	101	new databases	29
about	101		

newsgroups

162

server configuration

technical support	xvi	using with jConnect	43
NodeAddress property		ODI	
description	648	drivers	105
NON_KEYWORDS option		OEM code pages	
compatibility option	567	about	291
description	610	choosing	312
obtaining value	648	offline backups	341
Novell client software	105	defined	346
NULL		OLE DB	
ANSI behavior	577	ASAProv provider	68
defining for output	616	connecting	68
NULLS option	610	providers	68
Transact-SQL behavior	577	OmniConnect support	111
NULLS option		ON EXCEPTION RESUME cla	use
description	610	stored procedures	612
Interactive SQL settings	571	ON_CHARSET_CONVERSION	N
Number property		FAILURE option	
description	648	611	
numeric precision		setting	561
database option	617	ON_ERROR option	
NumProcessorsAvail property		description	611
description	657	Interactive SQL settings	571
NumProcessorsMax property		ON_TSQL_ERROR option	
description	657	compatibility option	566, 567
		description	612
0		obtaining value	648
1.1.		online backups	341
objects	412	defined	346
qualified names	413	Open Client	
ODBC	52	compatibility options	566
Administrator	55	configuring	114
connection parameters	/0, 1/4	interface	110
connections	38	options	120
data sources	53	Open Server	
driver location	/6	adding	114, 430
initialization file for UNIX	63	address	116
troubleshooting	514	architecture	110
UNIX support	63	configuring servers for JDBC	119
Windows CE	63	connecting to	431
ODBC Administrator	52	deleting server entries	119
using	53	renaming server entries	119
ODBC connectivity	43	starting	112
ODBC data sources		system requirements	112
about	41	OPEN statement	
configuring	56	QUERY PLAN ON OPEN	option
creating	53, 55	619	
creating with dbdsn	472	opening cursors	619
UNIX	63	-r8 •010010	017

OPTIMISTIC_WAIT_FOR_COMM	ΛIT	TIMEOUT	
option		586	
description	612	COOPERATIVE_COMMITS	586
OPTIMIZATION_GOAL option		data source [dbdsn] utility	475
description	613	database server	124
setting	561	DATE_FORMAT	587
Optimization_goal property		DATE_ORDER	588
description	648	DEFAULT_TIMESTAMP	
Optimization_level property		INCREMENT	
description	648	589	
options		DELAYED_COMMIT_TIMEOU	JT
about	556	589	
ALLOW_NULLS_BY_DEFAUI	T 574	DELAYED_COMMITS	590
ANSI_BLANKS	574	DELETE_OLD_LOGS	590
ANSI_CLOSE_CURSORS_ON		deleting settings	559
ROLLBACK		DESCRIBE_JAVA_FORMAT	591
574		DIVIDE_BY_ZERO_ERROR	591
ANSI_INTEGER_OVERFLOW	575	ECHO	591
ANSI_PERMISSIONS	575	erase [dberase] utility	479
ANSI_UPDATE_CONSTRAIN	FS 576	ESCAPE_CHARACTER	592
ANSINULL	577	EXCLUDE_OPERATORS	592
AUDITING	578	finding values	557
AUTO_COMMIT	579	FIRE_TRIGGERS	592
AUTO_REFETCH	579	FIRST_DAY_OF_WEEK	593
AUTOMATIC_TIMESTAMP	579	FLOAT_AS_DOUBLE	593
BACKGROUND_PRIORITY	580	FOR_XML_NULL_TREATMEN	νT
backup [dbbackup] utility	460	594	
BELL	580	GLOBAL_DATABASE_ID	595
BLOB_THRESHOLD	580	histogram [dbhist] utility	482
BLOCKING	581	information [dbinfo] utility	483
BLOCKING_TIMEOUT	581	initial settings	559
CHAINED	581	initialization [dbinit] utility	487
CHECKPOINT_TIME	582	INPUT_FORMAT	595
CIS_ROWSET_SIZE	582	Interactive SQL	571
classification	559	Interactive SQL [dbisql] utility	494
CLOSE_ON_ENDTRANS	583	ISOLATION_LEVEL	597
collation [dbcollat] utility	463	ISQL_COMMAND_TIMING	597
COMMAND_DELIMITER	583	ISQL_ESCAPE_CHARACTER	597
COMMIT_ON_EXIT	584	ISQL_FIELD_SEPARATOR	598
compatibility	567	ISQL_LOG	599
COMPRESSION	584	ISQL_PLAN	599
compression [dbshrink] utility	469	ISQL_PRINT_RESULT_SET	600
console [dbconsole] utility	471	ISQL_QUOTE	600
CONTINUE_AFTER_RAISER	ROR	JAVA_HEAP_SIZE	601
584		JAVA_INPUT_OUTPUT	601
CONVERSION_ERROR	585	JAVA_NAMESPACE_SIZE	602
COOPERATIVE_COMMIT		language [dblang] utility	498

license [dblic] utility	500
list of all	574
list of compatibility options 566,	567
list of database options	561
list of Interactive SQL options	571
list of replication options	570
log transfer manager [dbltm] utility	502
log translation [dbtran] utility	510
LOGIN MODE	602
LOGIN PROCEDURE	603
MAX CURSOR COUNT	605
MAX HASH SIZE	606
MAX PLANS CACHED	606
MAX RECURSIVE ITERATION	IS
607	0
MAX STATEMENT COUNT	608
MAX WORK TABLE HASH SI	ZF
607	
MIN PASSWORD I ENGTH	608
MIN_TABLE SIZE FOR	008
HISTOCRAM	
600	
NEADEST CENTURY option	600
NON KEYWORDS	610
NULLS	610
INULLS	610
ON_CHARSE1_CONVERSION_	-
FAILURE	
	C 11
ON TSOL EDDOD	611
ON_ISQL_ERROR	612
Open Client	120
OPTIMISTIC_WAIT_FOR	
COMMIT	
612	
OPTIMIZATION_GOAL	613
OUTPUT_FORMAT	614
OUTPUT_LENGTH	615
OUTPUT_NULLS	616
PERCENT_AS_COMMENT	616
ping [dbping] utility	514
PINNED_CURSOR_PERCENT	
OF_CACHE	
617	
PRECISION	617
PREFETCH	618
PRESERVE_SOURCE_FORMAT	618
PREVENT_ARTICLE_PKEY	

UPDATE	
619	
PUBLIC options	559
OUALIFY OWNERS	619
OUERY PLAN ON OPEN	619
OUOTE ALL IDENTIFIERS	620
OUOTED IDENTIFIER	620
READ PAST DELETED	620
RECOVERY TIME	621
REPLICATE ALL	621
REPLICATION FRROR	621
REPLICATION ERROR PIECE	622
RETURN DATE TIME AS -	022
STRING	
622	
DETUDN LAVA AS STRING	672
DI TDICCED TIME	622
RI_IRIGGER_IIME	623
ROW_COUNTS	624
SCALE	024 :
scope and duration of database opt	ions
	5 10
server location [dblocate] utility	518
service creation [dbsvc] utility	521
setting database options	556
setting in Interactive SQL	571
setting temporary	558
setting user and group options	393
SORT_COLLATION	624
spawn [dbspawn] utility	524
SQL_FLAGGER_ERROR_LEVE	L
625	
SQL_FLAGGER_WARNING	
LEVEL	
625	
startup settings	120
STATISTICS	626
stop [dbstop] utility	525
STRING_RTRUNCATION	626
SUBSCRIBE_BY_REMOTE	627
SUBSUME_ROW_LOCKS	627
SUPPRESS_TDS_DEBUGGING	627
TDS_EMPTY_STRING_IS_NUL	L
628	
TIME_FORMAT	628
TIME_ZONE_ADJUSTMENT	628
TIMESTAMP_FORMAT	629
transaction log [dblog] utility	529

TRUNCATE_DATE_VALUES	630	description	648, 657
TRUNCATE_TIMESTAMP		PacketsSentUncomp property	
VALUES		description	648, 657
631		page size	
TRUNCATE_WITH_AUTO		choosing	490
COMMIT		databases	486
631		maximum allowed	149
TRUNCATION_LENGTH	632	option	12
TSQL_HEX_CONSTANT	632	page usage	
TSOL VARIABLES	632	information [dbinfo] utility	484
uncompression [dbexpand] utility	532	PageRelocations property	
unload [dbunload] utility	537	description	664
UPDATE STATISTICS	633	pages	
upgrade [dbupgrad] utility	544	displaying usage in database	files 483
USER ESTIMATES	633	PageSize property	
validation [dbvalid] utility	549	description	657,664
VERIFY ALL COLUMNS	634	parallel execution	,
VERIFY THRESHOLD	634	processors	150
WAIT FOR COMMIT	634	threads	150
write file [dbwrite] utility	553	Password [PWD] connection pa	rameter
OUTPUT FORMAT option		about	70
description	614	about connection parameters	174
Interactive SOL settings	571	description	196
OUTPUT LENGTH option		introduction to connection pa	rameters
description	615	39	
Interactive SOL settings	571	tips	73
OUTPUT NULLS option	071	using default connection para	meters 51
description	616	passwords	
Interactive SOL settings	571	changing	395
OVERRIDE-MAGIC	571	default	390
USER ESTIMATES option	633	encrypting	188
owners	000	ITM configuration file	503
about	391	minimum length	608
ussut	571	ODBC configuration	58
Р		utility database	264
-		PATH environment variable	201
packet encryption		description	246
data source [dbdsn] utility	476	PeakCacheSize property	240
packet size		description	657
data source [dbdsn] utility	476	percent sign	0.57
limiting 154	, 155	modulo or comment	616
PacketSize property		PERCENT AS COMMENT of	otion
description	648	compatibility option	567
PacketsReceived property		description	507
description 648	, 657	obtaining value	649
PacketsReceivedUncomp property		performance	040
description 648	, 657	cacha siza 122	125 120
PacketsSent property			, 155–150

compression		102
disk fragmentation		258
improving		365
LTM		446
monitoring		638
options		12
PowerBuilder DataWindow		613
prefetch		618
primary keys		357
result sets		613
server options		6
setting priorities		580
TCP/IP		95
transaction log	343.	357
transaction log mirrors	,	344
TRUNCATE TABLE stateme	nt	631
Performance Monitor		
Windows		638
permissions		000
conflicts		420
connect		394
DBA authority		390
file administration statements		265
granting passwords		394
granting remote		402
groun		406
group membership		407
group	392	409
individual	572,	394
inheriting	399	406
integrated login permissions	<i>377</i> ,	85
listing		422
loading data		147
managing		389
options		12
overview		390
passwords		395
procedures		400
RESOURCE authority	391	395
revoking	571,	402
revoking remote		402
setting table permissions		396
tables	391	396
the right to grant	571,	399
triggers	391	401
unloading data	571,	147
views 201	308	147 /15
views 391,	570,	-1J

personal server about	
about	
	4
physical layer	
troubleshooting	106
physical limitations	
Adaptive Server Anywhere	674
ping [dbping] utility	
about	514
exit codes	514
options	514
syntax	514
ping utility	
TCP/IP	105
testing networks	34
testing Open Client	118
pinging	
servers	514
PINNED CURSOR PERCEN	Г ОГ -
CACHE database opti	on
617	
planning	
backups	347, 353
plans	011,000
caching of	606
Platform property	000
description	657
description	(1)/
PlatformVer property	037
PlatformVer property description	657
PlatformVer property description	657
PlatformVer property description plug-ins registry settings	657 250
PlatformVer property description plug-ins registry settings polling frequency	657 250 30
PlatformVer property description plug-ins registry settings polling frequency port number	657 250 30
PlatformVer property description plug-ins registry settings polling frequency port number detabase server	657 250 30
PlatformVer property description plug-ins registry settings polling frequency port number database server	657 250 30 215
PlatformVer property description plug-ins registry settings polling frequency port number database server TCP/IP 113 BewerBuilder Date Wir dow	657 250 30 215 , 163, 215
PlatformVer property description plug-ins registry settings polling frequency port number database server TCP/IP 113 PowerBuilder DataWindow	657 250 30 215 , 163, 215
PlatformVer property description plug-ins registry settings polling frequency port number database server TCP/IP 113 PowerBuilder DataWindow query performance	657 250 30 , 163, 215 613
PlatformVer property description plug-ins registry settings polling frequency port number database server TCP/IP 113 PowerBuilder DataWindow query performance PRECISION option	657 250 30 , 163, 215 613
PlatformVer property description plug-ins registry settings polling frequency port number database server TCP/IP 113 PowerBuilder DataWindow query performance PRECISION option description	657 250 30 215 , 163, 215 613 617
PlatformVer property description plug-ins registry settings polling frequency port number database server TCP/IP 113 PowerBuilder DataWindow query performance PRECISION option description setting	657 250 30 215 , 163, 215 613 617 561
PlatformVer property description plug-ins registry settings polling frequency port number database server TCP/IP 113 PowerBuilder DataWindow query performance PRECISION option description setting PRECISION property	657 250 30 215 , 163, 215 613 617 561
PlatformVer property description plug-ins registry settings polling frequency port number database server TCP/IP 113 PowerBuilder DataWindow query performance PRECISION option description setting PRECISION property description	657 250 30 215 , 163, 215 613 617 561 648
PlatformVer property description plug-ins registry settings polling frequency port number database server TCP/IP 113 PowerBuilder DataWindow query performance PRECISION option description setting PRECISION property description PREFETCH option	657 250 30 215 , 163, 215 613 617 561 648
PlatformVer property description plug-ins registry settings polling frequency port number database server TCP/IP 113 PowerBuilder DataWindow query performance PRECISION option description setting PRECISION property description PREFETCH option description	657 250 30 215 , 163, 215 613 617 561 648 618
PlatformVer property description plug-ins registry settings polling frequency port number database server TCP/IP 113 PowerBuilder DataWindow query performance PRECISION option description setting PRECISION property description PREFETCH option description DisableMultiRowFetch	657 250 30 215 , 163, 215 613 617 561 648 618 187
PlatformVer property description plug-ins registry settings polling frequency port number database server TCP/IP 113 PowerBuilder DataWindow query performance PRECISION option description setting PRECISION property description PREFETCH option description DisableMultiRowFetch setting	657 250 30 215 , 163, 215 613 617 561 648 618 187 561

description	648
PrefetchBuffer [PBUF] connection	
parameter	
about	70
about connection parameters	174
description	197
introduction to connection parameter	ters
39	
tips	73
using default connection parameter	rs 51
PreFetchOnOpen communication parameter	
description	214
PrefetchRows [PROWS] connection	
parameter	
about	70
about connection parameters	174
description	198
introduction to connection parameter	ters
39	1015
tips	73
using default connection parameter	rs 51
prepared statements	
connection limits 421,	608
Prepares property	
description	648
PrepStmt property	
description	648
PRESERVE_SOURCE_FORMAT or	otion
description	618
setting	561
PreserveSource property	
description	664
PREVENT ARTICLE PKEY UPDA	ATE
option	
description	619
setting	561
primary keys	
transaction log	357
primary sites	
adding Replication Server informa	tion
431	tion
creating	429
Replication Server	427
uses of LTM at	428
priority	
process	144

ProcedurePages property	
description	664
procedures	
and Adaptive Server Anywhere 442	LTM
permissions	400
permissions for creating	391
replicating	444
security	415
ProcessCPU property	
description	657
ProcessCPUSystem property	
description	657
ProcessCPUUser property	
description	657
ProcessorArchitecture	
description	657
processors	
multiple	12, 150
number used	12, 150
ProductName property	12, 100
description	657
ProductVersion property	007
description	657
programming interfaces	0.57
connections	38
properties	50
list of connection properties	648
list of database properties	664
list of server properties	657
suntay	647
syntax	ortion
syntax of connection-level prop	erties
047	ion 661
syntax of database-level properties	657
syntax of server-level properties	057
	(17
syntax	047
TCD/ID	200
	206
protocols	0.2
about	93
disabling shared memory	158
option	16
supported	16, 94
troubleshooting	105
providers	
ASAProv	68

MSDASQL	68
OLE DB	68
PUBLIC group	411
PUBLIC options	
about	559
DBA authority required	559

Q

qualified names	
database objects	413
tables	410
QUALIFY_OWNERS option	
description	619
replication option	570
qualify_table_owner parameter	
LTM configuration file	503
Query Low Memory Strategy statistic	645
QUERY_PLAN_ON_OPEN option	
compatibility option	567
description	619
obtaining value	648
QueryBypassed	
description	664
QueryBypassed property	
description	648
QueryCachedPlans	
description	664
QueryCachedPlans property	
description	648
QueryCachePages property	
description 648,	664
QueryLowMemoryStrategy property	
description 648,	664
QueryOptimized	
description	664
QueryOptimized property	
description	648
QueryReused	
description	664
QueryReused property	
description	648
questions	
character sets	288
quiet mode	
backup [dbbackup] utility	461
collation [dbcollat] utility	464
compression [dbshrink] utility	469

data source [dbdsn] utility	476
database server	130
erase [dberase] utility	480
information [dbinfo] utility	484
initialization [dbinit] utility	490
Interactive SQL [dbisql] utility	496
language [dblang] utility	498
license [dblic] utility	500
Log Transfer Manger [dbltm] utili	ty
503	
Log Translation [dbtran] utility	512
ping [dbping] utility	516
server location [dblocate] utility	518
spawn [dbspawn] utility	524
stop [dbstop] utility	526
transaction log [dblog] utility	530
uncompression [dbexpand] utility	532
unload [dbunload] utility	540
upgrade [dbupgrad] utility	546
validation [dbvalid] utility	550
write file [dbwrite] utility	554
QuittingTime property	
description	657
QUOTE_ALL_IDENTIFIERS option	n
description	620
replication option 570	, 620
QUOTED_IDENTIFIER option	
compatibility option 566	, 567
description	620
obtaining value	648
Open Client	120

R

RAISERROR	
behavior of	584
RAISERROR statement	
CONTINUE_AFTER_R	AISERROR
option	584
ON_TSQL_ERROR opti	ion 612
RAS	
dial-up networking	96
read-only	
databases	13, 157, 170
read-only media	
modifying databases	260
READ_PAST_DELETED	option
description	620

setting	561	communication param	eter
ReadOnly property		description	214
description	664	registry	
rebuild [rebuild] utility		about	249
about	517	environment variables	244
exit codes	517	language settings	250, 497
syntax	517	location setting	250
rebuilding databases	517	modifying	244
ReceiveBufferSize [RCVBUFSZ	[]	setting SQLREMOTE environ	nment
communication parame	eter	variable	247
description	214	Sybase Central	250
recommended collations		TEMP environment variable	248
Windows and UNIX	309	tools location setting	250
recovery		Windows CE	250
about	338	Windows services	249
distributed transactions	160	RelocatableHeapPages property	
maximum time	149	description	664
media failure	344, 379	remote data access	
option	13	CIS_ ROWSET_SIZE option	582
rapid	378	remote permissions	402
server options	167	RemoteTrunc property	
system failure	340	description	664
transaction log	343	removing	
transaction log mirrors	344	users from groups	408
uncommitted changes	382	rep_func parameter	
urgency	363	LTM configuration file	503
Recovery I/O Estimate statistic	639	replicate minimal columns	
recovery mode		support for	443
Log Transfer Manger [dbltm]	utility	REPLICATE ON clause	433
502		replicate sites	
recovery statistics	639	adding Replication Server inf	ormation
Recovery Urgency statistic	639	434	
RECOVERY_TIME option		creating	429
description	621	Replication Server	427
obtaining value	648	uses of LTM at	428
setting	561	REPLICATE_ALL option	
using	363	description	621
RecoveryUrgency property		obtaining value	648
description	664	replication option	570
recursive queries		replication	
MAX_RECURSIVE_ITERAT	TIONS	backup procedures 352,	, 449, 450
option	607	benefits	426
redescribing cursors	57	buffering	446
REFERENCES permissions	396	creating a replication definition	on for
referential integrity		Replication Server	436
triggers	623	dbcc	528, 529
RegisterBindery [REGBIN]		defining	442

entire databases	450
introduction	426
list of options	570
Log Transfer Manager	501
of procedures	444
of trigger actions	592
procedures	444
Replication Server	427
replication server 528.	529
setting flag	433
stored procedures	444
transaction log management 352, 450	449,
Replication Agent	
backups	350
Log Transfer Manager [dbltm] syr 501	ıtax
replication agent	
about	501
replication options	
BLOB_THRESHOLD	580
classification	559
COMPRESSION	584
DELETE_OLD_LOGS 570.	, 590
initial settings	559
list of	570
QUALIFY_OWNERS	619
QUOTE_ALL_IDENTIFIERS	620
REPLICATE_ALL	621
REPLICATION_ERROR	621
REPLICATION_ERROR_PIECE	622
SUBSCRIBE_BY_REMOTE	627
VERIFY_ALL_COLUMNS	634
VERIFY_THRESHOLD	634
Replication Server	
Adaptive Server Anywhere charac	ter
sets	441
Adaptive Server Anywhere collation 441	ons
backup procedures	449
creating a connection	435
creating a replication definition	436
creating a subscription	437
Log Transfer Manager	501
preparing Adaptive Server Anywh	ere
databases	434
primary sites 427,	, 435

purposes	426
replicate sites 427	, 435
replicating an entire database	450
replicating procedures	444
rssetup.sql script	431
starting an Adaptive Server Anyw	here
server	430
support	111
supported versions	427
transaction log management	449
Replication Server Adaptive Server	
Anywhere configuration	439,
440	
REPLICATION_ERROR option	
description	621
replication option	570
REPLICATION_ERROR_PIECE of	otion
description	622
Req property	
description	657
ReqType property	
description	648
RequestLogFile property	
description	657
RequestLogging property	
description	657
RequestQueueWait property	
description	657
requests	
about	14
Requests Active statistic	645
Requests statistic	645
Requests Unscheduled statistic	645
RESOURCE authority	
about	391
granting	395
not inheritable	406
resource governor	
cursors	605
statements	608
restore database wizard	
using	381
return codes	
backup [dbbackup] utility	458
collation [dbcollat] utility	462
compression [dbshrink] utility	468
console [dbconsole] utility	471

data source [dodsh] utility	474
erase [dberase] utility	478
histogram [dbhist] utility	481
information [dbinfo] utility	483
initialization [dbinit] utility	485
Interactive SOL	495
Interactive SOL [dbisal] utility	493
language [dblang] utility	498
license [dblic] utility	499
L og Transfer Manager [dbltm] util	ity
502	ity
log translation [dbtran] utility	510
ning [dbning] utility	514
rebuild [rebuild] utility	517
server leastion [dbloostel utility	519
server location [dblocate] utility	520
service creation [dbsvc] utility	520
spawn [dbspawn] unity	525
stop [dbstop] utility	525
transaction log [dblog] utility	528
uncompression [dbexpand] utility	532
unload [dbunload] utility	536
upgrade [dbupgrad] utility	544
validation utility (dbvalid)	549
write file [dbwrite] utility	553
RETURN_DATE_TIME_AS_STRIN	łG
option	
description	622
•	
setting	561
setting RETURN_JAVA_AS_STRING optio	561 n
setting RETURN_JAVA_AS_STRING optio description	561 n 623
setting RETURN_JAVA_AS_STRING optio description setting	561 n 623 561
setting RETURN_JAVA_AS_STRING optio description setting revoking	561 n 623 561
setting RETURN_JAVA_AS_STRING optio description setting revoking group membership	561 n 623 561 408
setting RETURN_JAVA_AS_STRING optio description setting revoking group membership permissions	561 on 623 561 408 402
setting RETURN_JAVA_AS_STRING optio description setting revoking group membership permissions remote permissions	561 n 623 561 408 402 402
setting RETURN_JAVA_AS_STRING optio description setting revoking group membership permissions remote permissions RI_TRIGGER_TIME option	561 n 623 561 408 402 402
setting RETURN_JAVA_AS_STRING optio description setting revoking group membership permissions remote permissions RI_TRIGGER_TIME option compatibility option	561 n 623 561 408 402 402 567
setting RETURN_JAVA_AS_STRING optio description setting revoking group membership permissions remote permissions RI_TRIGGER_TIME option compatibility option description	561 n 623 561 408 402 402 402 567 623
setting RETURN_JAVA_AS_STRING optio description setting revoking group membership permissions remote permissions RI_TRIGGER_TIME option compatibility option description obtaining value	561 n 623 561 408 402 402 402 567 623 648
setting RETURN_JAVA_AS_STRING option description setting revoking group membership permissions remote permissions RI_TRIGGER_TIME option compatibility option description obtaining value Riindael encryption algorithm	561 n 623 561 408 402 402 402 567 623 648
setting RETURN_JAVA_AS_STRING option description setting revoking group membership permissions remote permissions RI_TRIGGER_TIME option compatibility option description obtaining value Rijndael encryption algorithm initialization [dbinit] utility	561 n 623 561 408 402 402 402 567 623 648 488
setting RETURN_JAVA_AS_STRING option description setting revoking group membership permissions remote permissions RI_TRIGGER_TIME option compatibility option description obtaining value Rijndael encryption algorithm initialization [dbinit] utility unload [dbunload] utility	561 n 623 561 408 402 402 402 567 623 648 488 538
setting RETURN_JAVA_AS_STRING option description setting revoking group membership permissions remote permissions RI_TRIGGER_TIME option compatibility option description obtaining value Rijndael encryption algorithm initialization [dbinit] utility unload [dbunload] utility Rlbk property	561 n 623 561 408 402 402 402 567 623 648 488 538
setting RETURN_JAVA_AS_STRING optio description setting revoking group membership permissions remote permissions RI_TRIGGER_TIME option compatibility option description obtaining value Rijndael encryption algorithm initialization [dbinit] utility unload [dbunload] utility Rlbk property description	561 n 623 561 408 402 402 402 567 623 648 488 538 648
setting RETURN_JAVA_AS_STRING optio description setting revoking group membership permissions remote permissions RI_TRIGGER_TIME option compatibility option description obtaining value Rijndael encryption algorithm initialization [dbinit] utility unload [dbunload] utility Rlbk property description rollback logs	561 n 623 561 408 402 402 402 567 623 648 488 538 648
setting RETURN_JAVA_AS_STRING optio description setting revoking group membership permissions remote permissions RI_TRIGGER_TIME option compatibility option description obtaining value Rijndael encryption algorithm initialization [dbinit] utility unload [dbunload] utility Rlbk property description rollback logs about	561 n 623 561 408 402 402 402 567 623 648 488 538 648 362
setting RETURN_JAVA_AS_STRING option description setting revoking group membership permissions remote permissions RI_TRIGGER_TIME option compatibility option description obtaining value Rijndael encryption algorithm initialization [dbinit] utility unload [dbunload] utility Rlbk property description rollback logs about POLL PACK statement	561 n 623 561 408 402 402 402 567 623 648 488 538 648 362

cursors	574
log	362
RollbackLogPages property	
description	648, 664
rounding	
SCALE option	624
routers	
broadcasting over	206
row counts	624
ROW_COUNTS option	
description	624
obtaining value	648
setting	561
RS parameter	436
LTM configuration file	503
RS_pw parameter	436
LTM configuration file	503
RS_source_db parameter	436
LTM configuration file	503
RS_source_ds parameter	436
LTM configuration file	503
RS_user parameter	436
LTM configuration file	503
rssetup.sql script	431, 434
about	439
preparing to run	439
running	440
running the database server	3

S

sa_conn	_properties	
using		557
saving		
stater	nents	599
SCALE	option	
descr	iption	624
obtair	ning value	648
settin	g	561
scan_ret	ry parameter	436
LTM	configuration file	503
schedule	es	
about		270
define	ed	268
interr	als	278
scheduli	ng	
about		267, 268, 270
backı	ıps	347, 353

events	267	server lo
server shutdown	161	exit co
schemas		syntax
unloading definition	540	server lo
scripts directory		option
about	241	server m
search conditions		displa
USER ESTIMATES option	633	limiti
SearchBindery [BINSEARCH]		loggir
communication parameter		outpu
description	215	server na
security		-n opt
about	389	case s
adding simple encryption to comm	and	data s
files	466	ODB
auditing	578	stop [
C2 158	179	server or
encrypting communications 141.	143.	databa
168 188 190	1.0,	option
event example	274	recov
file access	147	server pr
file hiding [dbfhide] utility	466	Active
integrated logins 80	900	Availl
minimum password length	608	Buffe
procedures 400	415	Build
services 400,	20	Build
temporary file	2/6	Build
utility database	240	Butos
views	204 415	Bytes
Views SocurityManagor	415	Bytes
IAVA INDUT OUTDUT option	601	Dytes
SELECT normalisations	206	C
SELECT permissions	390	C2 Cashs
LISED ESTIMATES antion	622	Cache
USEK_ESTIMATES OPHON	035	Chore
SendbullerSize [SNDBUFSZ]		Chars
description	215	Comm
	215	Comp
	(57	Comp
description	057	Conns
SeparateCneckpointLog property		Curre
description	664	Defau
SeparateForeignKeys property		FreeB
description	004	Idle I
server address	110	IsiQ
DSEdit	116	IsJava
server information	0.2	IsNet
asasrv.1n1	82	IsRun

server location [dblocate] utility	518
exit codes	518
svntax	518
server location utility	
options	518
server messages	
displaying	156, 157
limiting size of log file	154
logging startup errors	153
output to file	153
server name	
-n option	152
case sensitivity	11
data source [dbdsn] utility	477
ODBC configuration	59
stop [dbstop] syntax	525
server options	
database	168
options	10
recovery	167
server properties	
ActiveReq	657
AvailIIO	657
BufferMisses	657
BuildChange	657
BuildClient	657
BuildReproducible	657
BytesReceived	657
BytesReceivedUncomp	657
BytesSent	657
BytesSentUncomp	657
C2	657
CacheHitsEng	657
CacheReplacements	657
CharSet	657
CommandLine	657
CompactPlatformVer	657
CompanyName	657
ConnsDisabled	657
CurrentCacheSize	657
DefaultCollation	657
FreeBuffers	657
IdleTimeout	657
IsIQ	657
IsJavaAvailable	657
IsNetworkServer	657
IsRuntimeServer	657

JavaGlobFix	657	SendFail	657
Language	657	StartTime	657
LegalCopyright	657	Tempdir	657
LegalTrademarks	657	TimeZoneAdjustment	657
LicenseCount	657	TotalBuffers	657
LicensedCompany	657	UnschReq	657
LicensedUser	657	server side	
LicensesinUse	657	setting encryption parameter	s141, 143,
LicenseType	657	168	
list of	657	ServerName [ENG] connection	
LivenessTimeout	657	parameter	
LockedHeapPages	657	character sets	73
MachineName	657	description	198
MainHeapBytes	657	ServerPort (PORT) communica	tion
MainHeapPages	657	parameter	113
MaxCacheSize	657	ServerPort [PORT] communica	tion
MaxMessage	657	parameter	
MessageText	657	description	215
MessageTime	657	servers	
MessageWindowSize	657	locating	518
MinCacheSize	657	managing	519
MultiPacketsReceived	657	properties of	657
MultiPacketsSent	657	starting from batch files	523
Name	657	web	219, 221
NativeProcessorArchitecture	657	service creation [dbsvc] utility	,
NumProcessorsAvail	657	about	519
NumProcessorsMax	657	exit codes	520
PacketsReceived	657	syntax	519
PacketsReceivedUncomp	657	service creation utility	
PacketsSent	657	options	521
PacketsSentUncomp	657	service creation wizard	
PageSize	657	using	24
PeakCacheSize	657	services	
Platform	657	about	24, 29, 30
PlatformVer	657	about web	219
ProcessCPU	657	account	28
ProcessCPUSystem	657	adding	24, 522
ProcessCPUUser	657	adding new databases	29
ProcessorArchitecture	657	character sets	234
ProductName	657	configuring	26
ProductVersion	657	creating web	222
QuittingTime	657	database server for Windows	21
reporting	516	default	230
Req	657	dependencies 3	31, 32, 521
RequestLogFile	657	eligible programs	23
RequestLogging	657	errors	235
RequestQueueWait	657	event log	130

executable file	29	and Co
failure to start	27	
groups	32	server
icon on the desktop	29	shutdown
interpreting URIs	227	specify
listening for HTTP requests	225	silent
listing	521	databa
managing	24	single-by
multiple	31	about
options	27	SMP
parameters	26	numbe
pausing	30	SOAP se
registry settings	249	about
removing	24, 26	creatir
request handlers	229	software
security	29	dblog
service creation [dbsvc] utility	519	dbshri
service manager	31	dbtran
setting dependencies	521	Interac
setting group dependencies	521	licensi
setting type	521	rebuild
starting	30	versio
starting order	32	software
startup options	27	licensi
stopping	30	software
variables	232	databa
web quick start	221	sort order
Windows 23,	165, 519	collati
Windows NT Control Panel	31	definit
SET OPTION statement		SORT C
Interactive SQL syntax	571	descrij
using	556	setting
SET TEMPORARY OPTION sta	tement	sorting
Interactive SQL syntax	571	collati
using	556	sp_setrep
setting		about
database options	561	sp setrep
temporary options	558	about
setup script		spawn [d
about	439	about
preparing to run	439	exit co
running	440	option
seven-bit characters		svntax
about	290	specifvin
shared memory		drivers
about	16	SPX
		about

and CommLinks connection	n parameter
179	
server configuration	162
shutdown	1.61
specifying time	161
silent	120
database server	130
single-byte character sets	200
about	290
SMP	12 150
SOAP server	12, 150
about	210
creating services	21)
software	222
dblog	528
dbsbrink	469
dbtran	510
Interactive SOI	/03
licensing database servers	400
rebuilding	477 517
rebuilding	162
version software license	102
	400
incensing servers	499
software version	1.00
database server	162
sort order	200
collations	286
definition	289
SORI_COLLATION option	(2.1
description	624
setting	561
sorting	
collation file	321
sp_setreplicate procedure	
about	433, 444
sp_setrepproc procedure	
about	444
spawn [dbspawn] utility	
about	523
exit codes	523
options	524
syntax	523
specifying	
drivers	43
SPX	
about	100

Host communication parameter	207
starting	16
supported protocols	94
SQL Anywhere Studio	
documentation	х
SOL compatibility	
options	566
SOL file format	614
SOL Remote	01.
backup procedures	352
backup	350
options	570
purposes	126
transaction log management	420
SOL standards	332
SQL standards	567
LIDD ATE state want	507
UPDATE statement	515
SQL statements	1
capturing recently-prepared	166
SQL.ini file	430
configuring	114
SQL/92 compliance	
SQL_FLAGGER_ERROR option	625
updates	576
SQL_database parameter	436
LTM configuration file	503
SQL_FLAGGER_ERROR_LEVEL	
option	
compatibility option	567
description	625
obtaining value	648
SOL FLAGGER WARNING LEVI	EL
option	
compatibility option	567
description	625
obtaining value	648
SOL pw parameter	436
ITM configuration file	503
SOI server parameter	436
ITM configuration file	503
SOL user peremotor	126
JTM configuration flo	430 502
LIVI CONNECT and a second second	505
SQLCONNECT environment variable	e 50
connections	52
description	247
SQLDA	
and ANSI_BLANKS	574

SQLLOCALE environment variable	le
deprecated	247
SQLPATH environment variable	
description	247
SQLREMOTE environment variab	le
description	247
stack overflow	
errors	149
stack size	
external functions	146
maximum	149
start line	1.0
ODBC configuration	59
Start Line [START] connection para	ameter
about	70
about connection parameters	174
description	199
embedded databases	47
introduction to connection parar	neters
39	liciter s
options	10
tips	73
using default connection parame	eters 51
StartTime property	
description	657
Statement Prepares statistic	645
statements	
ALTER TABLE	433
COMMIT	438
CREATE DATABASE 26	52, 265
DELETE	442
INSERT	442
UPDATE	442
Statements statistic	645
statistics	
Avail IO	645
cache	639
Cache Hits/sec	639
Cache Reads Index Interior/sec	639
Cache Reads Index Leaf/sec	639
Cache Reads Table/sec	639
Cache Reads Total Pages/sec	639
Cache Size Current	639
Cache Size Maximum	639
Cache Size Minimum	639
Cache Size Peak	639
checkpoint and recovery	639
· · · · · · · · · · · · · · · · · · ·	

Checkpoint Flushes/sec	639	Id
Checkpoint Log	639	in
Checkpoint Urgency	639	In
Checkpoints/sec	639	In
Comm Buffer Misses	640	In
Comm Bytes Received		Ja
Uncompressed/sec	640	Л
Comm Bytes Sent Uncompressed	/sec	J١
640		J١
Comm Bytes Sent/sec	640	Μ
Comm Free Buffers	640	Μ
Comm Multi-packets Received/se	c 640	Μ
Comm Multi-packets Sent/sec	640	Μ
Comm Packets Received		Μ
Uncompressed/sec	640	Μ
Comm Packets Received/sec	640	Μ
Comm Packets Sent		Μ
Uncompressed/sec	640	Μ
Comm Packets Sent/sec	640	Μ
Comm Send Fails/sec	640	Μ
Comm TotalBuffers	640	m
communications	640	m
Cursors	645	m
Cursors Open	645	Q
Disk Active I/O	641	R
disk I/O	641	R
Disk Maximum I/O	641	R
disk read	642	R
Disk Reads Active	642	R
Disk Reads Index interior/sec	642	St
Disk Reads Index leaf/sec	642	St
Disk Reads Maximum Active	642	Te
Disk Reads Table/sec	642	Tı
Disk Reads Total Pages/sec	642	Tı
disk write	642	STA
Disk Write Active	642	de
Disk Write Commit Files/sec	642	In
Disk Write Database Extends/sec	642	stop
Disk Write Maximum Active	642	ab
Disk Write Pages/sec	642	ex
Disk Write Temp Extends/sec	642	op
Disk Write Transaction Log Group	р	sy
Commits	642	stop
Disk Write Transaction Log/sec	642	pe
Idle Actives/sec	639	stopp
Idle Checkpoint Time	639	da
Idle Checkpoints/sec	639	STR

Idle Writes/sec	639
index	643
Index Adds/sec	643
Index Full Compares/sec	643
Index Lookups/sec	643
Java VM	643
JVM Global Fixed Size	643
JVM Heap Size	643
JVM Namespace Size	643
Main Heap Bytes	645
Mem Pages Lock Table	644
Mem Pages Locked Heap	644
Mem Pages Main Heap	644
Mem Pages Map Pages	644
Mem Pages Procedure Definitions	644
Mem Pages Relocatable	644
Mem Pages Relocations/sec	644
Mem Pages Rollback Log	644
Mem Pages Trigger Definitions	644
Mem Pages View Definitions	644
memory pages	644
miscellaneous	645
monitoring performance	638
Query Low Memory Strategy	645
Recovery I/O Estimate	639
Recovery Urgency	639
Requests	645
Requests Active	645
Requests Unscheduled	645
Statement Prepares	645
Statements	645
Temporary Table Pages	645
Transaction Commits	645
Transaction Rollbacks	645
STATISTICS option	
description	626
Interactive SQL settings	571
stop [dbstop] utility	
about	525
exit codes	525
options	525
syntax	525
stop utility	
permissions	146
stopping	
databases	525
STRING_RTRUNCATION option	

compatibility option	567
description	626
obtaining value	648
strings	
and host variables	574
strong encryption	
client/server communications 141	. 143.
168, 188, 190	, .,
initialization [dbinit] utility	488
unload [dbunload] utility	538
subdirectories	
NetWare	241
Windows CE	241
SUBSCRIBE_BY_REMOTE option	L
description	627
replication option	570
subscriptions	
creating for Replication Server	437
SUBSUME_ROW_LOCKS option	
description	627
setting	561
support	
newsgroups	xvi
SUPPRESS_TDS_DEBUGGING op	otion
description	627
obtaining value	648
setting	561
switches	
[dbspawn] spawn utility	524
backup [dbbackup] utility	460
collation [dbcollat] utility	463
compression [dbshrink] utility	469
console [dbconsole] utility	471
data source [dbdsn] utility	475
erase [dberase] utility	479
histogram [dbhist] utility	482
information [dbinfo] utility	483
initialization [dbinit] utility	487
Interactive SOL [dbisgl] utility	494
language [dblang] utility	498
license [dblic] utility	500
log transfer manager [dbltm] utilit	tv502
log translation [dbtran] utility	510
ping [dbping] utility	514
server location [dblocate] utility	518
service creation [dbsvc] utility	521
stop [dbstop] utility	525
1 - 1 - · · ·	

uncompression [dbexpand] utility 55 unload [dbunload] utility 55 upgrade [dbupgrad] utility 55 validation [dbvalid] utility 55 write file [dbwrite] utility 55 Sybase Central backing up databases 44 changing log file names 55	32 37 44 49 53 58
unload [dbunload] utility5.upgrade [dbupgrad] utility5.validation [dbvalid] utility5.write file [dbwrite] utility5.Sybase Central5.backing up databases4.changing log file names5.	37 44 49 53 58
upgrade [dbupgrad] utility 5- validation [dbvalid] utility 5- write file [dbwrite] utility 5- Sybase Central backing up databases 4- changing log file names 5-	44 49 53 58
validation [dbvalid] utility 5- write file [dbwrite] utility 5- Sybase Central backing up databases 4- changing log file names 5-	49 53 58
write file [dbwrite] utility 55 Sybase Central backing up databases 44 changing log file names 55	53 58
Sybase Central backing up databases 4. changing log file names 55	58
backing up databases 4.	58
changing log file names 5'	
Junging rog me names J	27
compressing databases 4	68
creating custom collations 4	62
creating databases 4	85
creating write files 5	51
erasing databases 4	78
managing services	24
registry settings 2	50
starting Interactive SOL from 4	92
syntax highlighting 6	16
uncompressing database files 5	31
unloading databases 5	33
upgrading databases 54	42
validating databases 54	47
SYBASE environment variable	• •
description 24	48
DSEdit 1	15
sybinit utility	
about 1	12
sybping 4	31
synchronization	
DELETE OLD LOGS 570, 5	90
SyncTrunc property	
description 6	64
syntax	
connection properties 6	47
database properties 6	47
database-level properties 6	64
server-level properties 6	57
syntax errors	
joins 5	92
syntax highlighting	
comments 6	16
SYS group 4	11
SYSCOLAUTH view	
permissions 4	23
SYSCOLLATION table	
collation files 32	20
SYSCOLPERM table	

permissions	422	SYSUSERPERM table	
SYSDUMMY table		permissions	422
permissions	422	SYSUSERPERMS view	
SYSGROUP table		permissions	423
permissions	422		
SYSGROUPS view		Т	
permissions	423	4.11	
SYSINFO table		table names	200
collation files	320	international aspects	299
syslog		table permissions	20.6
user ID for	158	setting	396
SYSOPTION system table		table size	67 A
storing Interactive SOL options	571	limit	674
SYSPROCAUTH view		number of rows	674
permissions	423	TableBitMaps property	
SYSPROCPERM table		description	664
permissions	422	tables	
SYSTABAUTH view	122	group owners	410
permissions	423	limitations	674
SYSTABLEPERM table	723	owner	391
permissions	122	permissions	391
system events	422	qualified names	410, 413
about	272	replicating	433, 442
defined	212	tabular data streams	110
aveter feilures	208	tasks	
system failures	240	about	14
about	54Z	backup	367
recovery	540	events	280
system objects	522	recovery	367
and the dbo user	533	schedules	280
system tables	220	TCP/IP	
character sets	320	about	95
national languages	320	addresses	116
permissions	422	connecting across firewall	ls95, 97, 203,
PRESERVE_SOURCE_FORMA	41 618	204, 207, 209, 215, 518	
PREVENT_ARTICLE_PKEY		encrypting client/server	
UPDATE		communications	97
619		Host communication para	meter 207
source column 61	8, 619	Open Server	112
SYSCOLLATION	320	performance	95
SYSINFO	320	port number	113, 163, 215
users and groups	422	protocol stacks	206
system views		server configuration	162 201
permissions	422	starting	162,201
SYSUSERAUTH view		supported protocols	94
permissions	423	troubleshooting	105
SYSUSERLIST view		Windows 95/98/Me	95
permissions	423	Windows NT/2000/YP	05
		WINGOWS IV1/2000/AI	,,

TDS communication parameter		setting	561
description	217	TIME_ZONE_ADJUSTMENT opt	ion
TDS communication protocol		description	628
about	110	setting	561
TDS_EMPTY_STRING_IS_NU	JLL	Timeout [TO] communication param	meter
option		description	217
description	628	timeouts	
obtaining value	648	troubleshooting	107
setting	561	TIMESTAMP	
technical support		DEFAULT_TIMESTAMP	
newsgroups	xvi	INCREMENT option	
telnet		589	
testing networks	34	TIMESTAMP columns	579
TEMP environment variable		TIMESTAMP data type	
CE	250	comparing	589
description	248	TIMESTAMP_FORMAT option	
disk space	34	compatibility option	566
Tempdir property		description	629
description	657	obtaining value	648
TempFileName property		setting	561
description	664	TimeZoneAdjustment property	
temporary files		description 64	8,657
location	246, 248	TMP environment variable	
location on CE	250	CE	250
security	246	description	248
temporary options		TMPDIR environment variable	
setting	558	CE	250
Temporary Table Pages statistic	645	description	248
temporary tables		TotalBuffers property	
limitations	674	description	657
TempTablePages property		Transact-SQL	
description	648, 664	ALLOW_NULLS_BY_DEFAU	LT
threading behavior		option	574
NetWare	14	AUTOMATIC_TIMESTAMP or	otion
UNIX	14	579	
Windows 95/98/Me	14	column NULLs compatibility	620
threads	14	compatibility options	567
about	14	delete permissions	575
controlling behavior	15	NULL behavior	577
execution	148	QUOTED_IDENTIFIER option	620
multiple processors	150	update permissions	575
number of	150	Transaction Commits statistic	645
Windows NT/2000/XP	14	transaction log	
TIME_FORMAT option		about	343
compatibility option	566	allocating space	258
description	628	backup [dbbackup] utility 46	0, 461
obtaining value	648	changing location	383

delete after checkpoint		151
DELETE_OLD_LOGS		590
erasing		478
initialization [dbinit] utility		490
limiting size		273
location		8
Log Transfer Manager		428
Log Translation [dbtran] utilit 512	y á	508,
managing	442,	590
media failure		379
mirror 344, 384,	385,	449
option		13
placing		355
primary keys		357
size		357
transaction log [dblog] utility	528,	529
translation utilities		510
uncommitted changes		382
validating		365
write file [dbwrite] utility		554
transaction log [dblog] utility		
about		527
exit codes		528
options		529
syntax		527
transaction log mirror		
initialization [dbinit] utility		490
purpose		355
starting	384,	385
write file [dbwrite] utility		554
transaction modes		
chained/unchained		581
Transaction Rollbacks statistic		645
transactions		
closing cursors		583
distributed		160
TransactionsSpanLogs property		
description		664
translate log file wizard		
using	382,	508
trigger condition		
defined		272
TriggerPages property		
description		664
triggers		
and referential integrity action	ıs	623

and replication	592
disabling	146
permissions 391	, 401
permissions for creating	391
troubleshooting	
backups	375
common problems	106
connections 75, 514	518
database connections	75
database server	166
database server request-level loggi 167	ng
identifying client applications	175
network communications	105
ODBC	514
protocols	105
server address	118
server startup 3-	4, 35
timeouts	107
Windows CE connections	66
wiring problems	106
TRUNCATE TABLE statement	
autocommit behavior	631
TRUNCATE_DATE_VALUES optio	n
description	630
setting	561
TRUNCATE_TIMESTAMP_VALUE	ES
option	
description	631
obtaining value	648
setting	561
TRUNCATE_WITH_AUTO_COMM	1IT
option	
description	631
obtaining value	648
setting	561
truncating	
character strings	626
TRUNCATION_LENGTH option	
description	632
Interactive SQL settings	571
TSQL_HEX_CONSTANT option	
compatibility option 566	, 567
description	632
obtaining value	648
Open Client	120
TSQL_VARIABLE option	

compatibility option	566	unloading	
TSQL_VARIABLES option		databases	533
compatibility option	567	unloading databases	
description	632	from Sybase Central	533
obtaining value	648	UnschReq property	
Open Client	120	description	657
		UPDATE permissions	396
U		UPDATE statement	442
unchained transaction mode	581	truncation of strings	626
UncommitOn property	301	UPDATE_STATISTICS option	
description	648	description	633
uncompress database wizard	040	setting	561
uncompress database wizard	531	updates	
using	551	ANSI behavior	575, 576
database files	531	SQL/92behavior	576
database files from Subase Cent	531 rol 531	Transact-SQL permissions	575
uncompression [dbeypand] utility	141 551	upgrade [dbupgrad] utility	
about	531	about	542
about	532	exit codes	544
options	532	options	544
syntax	531	syntax	543
Unconditional [UNC] connection	551	upgrade database wizard	
parameter		using	542
about	70	upgrading	
about connection parameters	174	databases	542
description	100	Java in the database	544
introduction to connection parar	neters	upgrading databases	
39	neters	from Sybase Central	542
tips	73	upper code page	
using default connection parame	eters 51	about	290
Unicode character sets		URLs	
about	315	default services	230
UNIX	515	handling	229
choosing a collation	308	interpreting	227
database servers	6	usage information	
default character set	303	displaying	133
environment variables	244	user creation wizard	
ODBC support	63	using	394
threading behavior	14	user estimates	
unload [dbunload] utility		overriding	633
about	533	user IDs	
exit codes	536	default	390
syntax	534	listing	422
unload database wizard		managing	389
using	533	ODBC configuration	58
unload utility		PUBLIC options	559
options	537	user names	

license [dblic] utility	500	in
user-supplied selectivity estimate	es	ini
USER ESTIMATES option	633	ini
USER_ESTIMATES option		In
description	633	In
obtaining value	648	int
setting	561	laı
Userid [UID] connection parame	ter	laı
about	70	lic
about connection parameters	174	lic
description	199	L
introduction to connection par	ameters	Lo
39	unictors	E
tips	73	lo
using default connection para	neters 51	
Userid property	neters 51	ni
description	648	pi ni
users	040	rel
connected users	405	rel
creating	394	10
deleting	403	50
deleting integrated logins	+05 87	50
granting integrated logins	85	50
remote permissions	402	su.
removing from groups	402	sp
setting options	303	sp
UtilCmdsPermitted property	575	ste
description	619	su
utilities and See detabase		Su
healan [dhhealan]	459	u c tro
backup [dbbackup]	450	ua 110
colletion [dbackup] syntax	439	un
collation [dbcollat]	330, 331	un
communication [dbcbrink]	405	un
compression [dbshrink]	408	un
compression [doshrink] syntax	408	up
console [dbconsole]	4/1	up
data source [dbdom]	4/1	va
data source [dbdsh]	472	va
DEEdit	472	WI
DSEdit	114	WI
erase [dberase]	4/8	utility
erase [dberase] syntax	4/9	pe
	466	utility
nie hiding [dbfhide] syntax	466	ab
nistogram [dbhist]	481	all
histogram [dbhist] syntax	481	co
information [dbinfo]	483	pa

	information [dbinfo] syntax	483
	initialization [dbinit]	485
	initialization [dbinit] syntax	486
	Interactive SQL [dbisql]	492
	Interactive SQL [dbisql] syntax	492
	introduction	457
	language [dblang]	497
	language [dblang] syntax	497
	license [dblic]	499
	license [dblic] syntax	499
	Log Transfer Manager [dbltm]	501
	Log Transfer Manager [dbltm] syn	tax
	501	
	log translation [dbtran]	508
	log translation [dbtran] syntax	508
	ping [dbping]	514
	ping [dbping] syntax	514
	rebuild [rebuild]	517
	rebuild [rebuild] syntax	517
	server location [dblocate]	518
	server location [dblocate] syntax	518
	service creation [dbsvc]	519
	service creation [dbsvc] syntax	519
	spawn [dbspawn]	523
	spawn [dbspawn] syntax	523
	stop [dbstop]	525
	stop [dbstop] permissions	146
	stop [dbstop] syntax	525
	transaction log [dblog]	527
	transaction log [dblog] syntax	527
	uncompression [dbexpand]	531
	uncompression [dbexpand] syntax	531
	unload [dbunload]	533
	unload [dbunload] syntax	534
	upgrade [dbupgrad]	542
	upgrade [dbupgrad] syntax	543
	validation [dbvalid]	547
	validation [dbvalid] syntax	548
	write file [dbwrite]	551
	write file [dbwrite] syntax	552
ut	tility commands	
	permissions	150
ut	tility database	
	about	262
	allowed SQL statements	262
	connecting to	263
	passwords	264
	-	

security and passwords

V

validate database wizard		
using	369,	547
validating		
backups		354
databases	354,	547
validating databases		365
from Sybase Central		547
validation [dbvalid] utility		
about		547
exit codes		549
options		549
syntax		548
variable width character sets		
about		297
VariableHashSize property		
description		664
variables		
in web services handlers		232
verbose mode		
Log Transfer Manger [dbltm] 503	utilit	у
unload [dbunload] utility		540
VERIFY_ALL_COLUMNS opt	ion	
description		634
replication option		570
VERIFY_THRESHOLD option		
description		634
replication option		570
VerifyServerName [VERIFY]		
communication parame	eter	
description		218
version		
database server		162
version mismatch		
file locations		242
ViewPages property		
description		664
views		
owner		391
permissions	391,	398
security		415
Visual Basic		
ODBC configuration for		57

264 **W**

WAIT_FOR_COMMIT option	
description	634
obtaining value	648
setting	561
web services	
about	219
character sets	234
creating	222
default services	230
errors	235
interpreting URIs	227
listening for HTTP requests	225
quick start	221
request handlers	229
variables	232
white space characters	
defined	324
Windows	
choosing a collation	308
default character set	303
services	23
Windows 2000	
setting cache size	138
Windows 95/98/Me	
threading behavior	14
Windows CE	
character set	312
connecting from the desktop	65, 66
creating databases	312
file locations	241
installation	241
ODBC	63
setting TEMP file	250
starting the database server	9
subdirectories	241
Windows NT/2000/XP	
and TCP/IP	95
event log	130
Performance Monitor	638
threading behavior	14
Windows operating systems	
database servers	5
Windows Performance Monitor	638
Windows Server 2003	
setting cache size	138
Windows services	

- - - -- ----

registry settings	2	249	listening for HTTP requests
Windows XP			V
setting cache size	1	38	Ŷ
wiring			vear 2000
troubleshooting	1	.06	NEAREST CENTURY ont
WITH GRANT OPTION clause	3	899	
wizards			
backup database	3	377	
change log file	383, 3	885	
change log file settings	5	527	
compress database	4	68	
create backup images	371, 4	58	
create custom collation	4	62	
create database	4	85	
create write file	5	551	
dbspace creation	2	257	
erase database	4	78	
event creation	2	280	
group creation	4	06	
integrated login creation		87	
restore database	3	881	
service creation		24	
translate log file	382, 5	508	
uncompress database	5	31	
unload database	5	533	
upgrade database	5	542	
user creation	3	394	
validate database	369, 5	547	
write file [dbwrite] utility			
about	5	51	
exit codes	5	53	
options	5	53	
syntax	5	552	
write files			
about	2	260	
backup [dbbackup] utility	4	61	
creating from Sybase Central	5	51	
erasing	4	78	
managing	5	551	
managing using dbwrite	5	552	
write file [dbwrite] utility	5	53	

REST_CENTURY option

Χ

XML file format	614
XML server	
about	219
creating services	222