



SQL Anywhere® Studio Security Guide

Part number: 38177-01-0900-01

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 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, PowerJ, 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 SQL Desktop, Sybase SQL Lifecycle, Sybase SQL 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 | v |
| SQL Anywhere Studio documentation | vi |
| Documentation conventions | ix |
| The Adaptive Server Anywhere sample database | xi |
| Finding out more and providing feedback | xii |
| | |
| I Basic Security Information | 1 |
| | |
| 1 Keeping Your Data Secure | 3 |
| Security features overview | 4 |
| Controlling database access | 6 |
| Auditing database activity | 9 |
| Running the database server in a secure fashion | 14 |
| Encrypting a database | 16 |
| Security tips | 20 |
| | |
| II Configuring Adaptive Server Anywhere in a C2-Compliant Manner | 23 |
| | |
| 2 Installation | 25 |
| Hardware installation | 26 |
| Operating system installation | 27 |
| Adaptive Server Anywhere software installation | 28 |
| Creating a database | 32 |
| Running the database engine | 34 |
| | |
| 3 Auditing | 37 |
| Enabling and disabling auditing | 38 |
| Reading auditing output | 39 |
| Audit records | 40 |
| Administration of audit records | 47 |
| Auditing of database utilities | 48 |
| Correlating audit records | 49 |
| | |
| 4 Restrictions and Other Security Concerns | 51 |
| Restrictions | 52 |
| Security warnings | 55 |

| | |
|--|-----------|
| Changing ownership on nested objects | 56 |
| Revoking DBA authority | 58 |
| The TCB subset | 59 |
| 5 Restricted Syntax | 61 |
| Restricted syntax | 62 |
| Database engine/server | 63 |
| Initialization utility | 67 |
| Service creation utility | 68 |
| Transaction log utility | 69 |
| Interactive SQL utility | 70 |
| 6 Integrated Logins | 71 |
| Using integrated login | 72 |
| 7 Connecting to the Adaptive Server Anywhere Service | 73 |
| Connecting to the Adaptive Server Anywhere service | 74 |
| 8 The Adaptive Server Anywhere C2 Patch | 75 |
| The Adaptive Server Anywhere C2 patch | 76 |
| 9 More Information | 77 |
| Where to look for more information | 78 |
| Index | 79 |

About This Manual

Subject

This book describes security features available in SQL Anywhere Studio. It includes basic security information, as well as instructions on how to operate the current version of SQL Anywhere Studio in a manner that is comparable to the C2-certified environment.

This book does not include all information on security-related features.

Current software is not C2 certified

Adaptive Server Anywhere version 7.0 achieved the C2 security certification of the US federal government. The C2 section of this manual describes how to operate the current version of Adaptive Server Anywhere in a manner comparable to the C2-certified configuration.

This book is *not* the certified document describing C2 compliance. The certified documentation is available from the Sybase Web site at <http://my.sybase.com/detail?id=1010458>. Nothing in this document should be taken to suggest that the current version of the software is C2 compliant. Use of the phrase “equivalent to the C2-certified configuration” and similar phrases does not imply actual C2 compliance. The *only* way to operate in a C2-certified manner is to use the C2-certified release of the software according to the C2-certified documentation.

Audience

This manual is for users of Adaptive Server Anywhere who wish to make use of the security features in the software, or run the software in a manner equivalent to the C2-certified configuration.

SQL Anywhere Studio documentation

The 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 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.

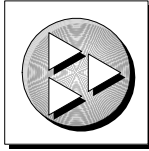
[**QUOTES** { **ON** | **OFF** }]

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

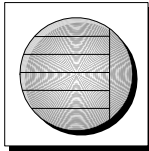
Graphic icons

The following icons are used in this documentation.

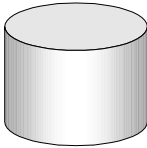
- ◆ A client application.



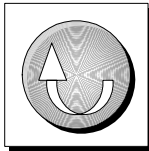
- ◆ A database server, such as Sybase Adaptive Server Anywhere.



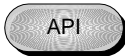
- ◆ A database. In some high-level diagrams, the icon may be used to represent both the database and the database server that manages it.



- ◆ 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.



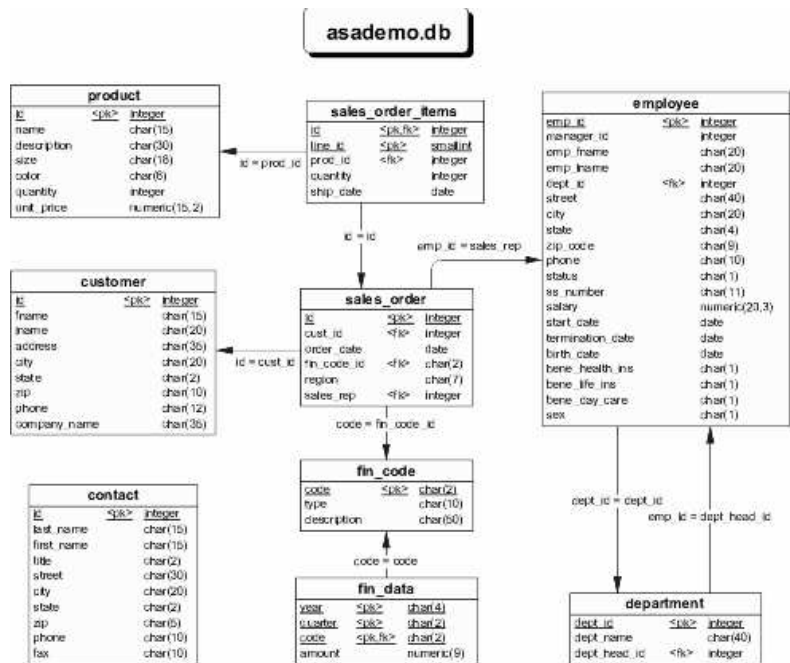
The Adaptive Server Anywhere sample database

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).

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

BASIC SECURITY INFORMATION

This part describes basic security features in SQL Anywhere Studio

CHAPTER 1

Keeping Your Data Secure

About this chapter

This chapter describes Adaptive Server Anywhere features that help make your database secure. In particular, this chapter describes auditing, database encryption, and C2 certification. It presents overviews of other security features, providing pointers to where you can find more detailed information.

Database administrators are responsible for data security. In this chapter, unless otherwise noted, you require DBA authority to carry out the tasks described.

☞ User IDs and permissions are major security-related topics. For information on these topics, see “Managing User IDs and Permissions” [*ASA Database Administration Guide*, page 389].

Contents

| Topic: | page |
|---|------|
| Security features overview | 4 |
| Controlling database access | 6 |
| Auditing database activity | 9 |
| Running the database server in a secure fashion | 14 |
| Encrypting a database | 16 |
| Security tips | 20 |

Security features overview

Since databases may contain proprietary, confidential, or private information, ensuring that the database and the data in it are designed for security is very important.

Adaptive Server Anywhere has several features to assist in building a secure environment for your data:

- ◆ **User identification and authentication** These features control who has access to a database.

☞ For information on these subjects, see “Creating new users” [ASA Database Administration Guide, page 394].

- ◆ **Discretionary access control features** These features control the actions a user can carry out while connected to a database.

☞ For more information, see “Database permissions overview” [ASA Database Administration Guide, page 390].

- ◆ **Auditing** This feature helps you maintain a record of actions on the database.

☞ For more information, see [“Auditing database activity” on page 9](#).

- ◆ **Database server options** These features let you control who can carry out operations (for example, loading databases). These options are set when you start the database server.

☞ For more information, see “Controlling permissions from the command line” [ASA Database Administration Guide, page 12].

- ◆ **Views and stored procedures** These features allow you to specify the data a user can access and the operations a user can execute.

☞ For more information, see “Using views and procedures for extra security” [ASA Database Administration Guide, page 415].


- ◆ **Database encryption** Database encryption features allow you to choose the level of database encryption. You can choose to secure your database either with simple encryption, or with strong encryption. Simple encryption is equivalent to obfuscation. Strong encryption renders the database completely inaccessible without the key.

☞ For more information, see “Initialization utility options” [ASA Database Administration Guide, page 487].


- ◆ **Communication encryption** You can encrypt client/server communications with simple or strong encryption for greater security as they pass over the network. Strong encryption is only supported over the

TCP/IP port on Solaris, Linux, NetWare, and all supported Windows operating systems except Windows CE.

Communication encryption is a separately licensable component and must be ordered before you can install it. To order this component, see the card in your SQL Anywhere Studio package or see <http://www.sybase.com/detail?id=1015780>.

 For more information, see “[Encrypting client/server communications](#)” on page 15.

- ◆ **C2 certification** C2 is a set of security guidelines established by the U.S. government to maintain consistency within their organization. If you are running Adaptive Server Anywhere 7.0, and if you have the appropriate hardware, you can set up your machine to run in a C2 certified manner. The C2-certified documentation is available at <http://my.sybase.com/detail?id=1010458>.

 For information on running the current version of Adaptive Server Anywhere in a manner equivalent to the C2-certified environment, see “[Installation](#)” on page 25.

Controlling database access

By assigning user IDs and passwords, the database administrator controls who has access to a database. By granting permissions to each user ID, the database administrator controls what tasks each user can carry out when connected. This section describes the features available for controlling database access.

Permission scheme is based on user IDs

When you log onto the database, you have access to all database objects that meet *any* of the following criteria:

- ◆ objects you created.
- ◆ objects to which you received explicit permission.
- ◆ objects to which a group you belong to received explicit permission.

The user cannot access any database object that does not meet these criteria. In short, users can access only the objects they own or objects to which they explicitly received access permissions.

☞ For more information, see the following:

- ◆ “Managing User IDs and Permissions” [*ASA Database Administration Guide*, page 389]
- ◆ “CONNECT statement [ESQL] [Interactive SQL]” [*ASA SQL Reference*, page 287]
- ◆ “GRANT statement” [*ASA SQL Reference*, page 456]
- ◆ “REVOKE statement” [*ASA SQL Reference*, page 530]

Using integrated logins

Integrated logins allow users to use a single login name and password to log onto both the Windows NT/2000/XP operating systems and onto a database. An external login name is associated with a database user ID. When you attempt an integrated login, you log onto the operating system by giving both a login name and password. The operating system then tells the server who you are, and the server logs you in as the associated database user ID. No additional login name or password are required.

There are some security implications of integrated logins to consider. For example, leaving the user profile Guest enabled with a blank password can permit unrestricted access to a database that is hosted by that server. Literally any user can log in to the server using any login ID and any password because they are logged in by default to the Guest user profile.

☞ For more information, see the following:

- ◆ “Security concerns: unrestricted database access” [ASA Database Administration Guide, page 89]
- ◆ “Using integrated logins” [ASA Database Administration Guide, page 85]
- ◆ “LOGIN_MODE option [database]” [ASA Database Administration Guide, page 602]

Increasing password security

Passwords are an important part of any database security system. To be secure, passwords must be difficult to guess, and they must not be easily accessible on users’ hard drives or other locations.

Implement minimum password lengths

By default, passwords can be any length. For greater security, you can enforce a minimum length requirement on all new passwords. You do this by setting the MIN_PASSWORD_LENGTH database option to a value greater than zero. The following statement enforces passwords to be at least 8 bytes long.

```
SET OPTION PUBLIC.MIN_PASSWORD_LENGTH = 8
```

☞ For more information, see “MIN_PASSWORD_LENGTH option [database]” [ASA Database Administration Guide, page 608].

Do not include passwords in ODBC data sources

Passwords are the key to accessing databases. They should not be easily available to unauthorized people in a security-conscious environment.

When you create an ODBC data source, or a Sybase Central connection profile, you can optionally include a password. Avoid including passwords for greater security.

☞ For information on creating ODBC data sources, see “Creating an ODBC data source” [ASA Database Administration Guide, page 53].

Encrypt command files containing passwords

When you create a configuration file, you can optionally include password information. To protect your passwords, consider hiding the contents of configuration files with simple encryption, using the File Hiding [dbfhide] utility.

☞ For information on the File Hiding [dbfhide] utility, see “The File Hiding utility” [ASA Database Administration Guide, page 466].

Controlling the tasks users can perform

Users can access only those objects to which they have been granted access.

You grant permission on an object to another user with the GRANT statement. You can also grant a user permission to pass on the permissions

on an object to other users.

The GRANT statement also gives more general permissions to users. Granting CONNECT permissions to a user allows them to connect to the database and change their passwords. Granting RESOURCE authority allows the user to create tables, views, procedures, and so on. Granting DBA authority to a user gives that user the ability to see and do anything in the database. The DBA would also use the GRANT statement to create and administer groups.

The REVOKE statement is the opposite of the GRANT statement—any permission that GRANT has explicitly given, REVOKE can take away. Revoking CONNECT from a user removes the user from the database, including all objects owned by that user.

Negative permissions

Adaptive Server Anywhere does not support **negative permissions**. This means that you cannot revoke a permission that was not explicitly granted.

For example, suppose user bob is a member of a group called sales. If a user grants DELETE permission on a table, T, to sales, then bob can delete rows from T. If you want to prevent bob from deleting from T, you cannot simply execute a REVOKE DELETE on T from bob, since the DELETE ON T permission was never granted directly to bob. In this case, you would have to revoke bob's membership in the sales group.

☞ For more information, see:

- ◆ “GRANT statement” [ASA SQL Reference, page 456]
- ◆ “REVOKE statement” [ASA SQL Reference, page 530]

Designing database objects for security

Views and stored procedures provide alternative ways of tuning the data users can access and the tasks they can perform.

☞ For more information on these features, see:

- ◆ “Benefits of procedures and triggers” [ASA SQL User's Guide, page 612]
- ◆ “Using views and procedures for extra security” [ASA Database Administration Guide, page 415]


Auditing database activity

Auditing is a way of keeping track of the activity performed on a database. The record of activities stays in the transaction log. By turning on auditing, the DBA increases the amount of data saved in the transaction log to include the following:

- ◆ All login attempts (successful and failed), including the terminal ID.
- ◆ Accurate timestamps of all events (to a resolution of milliseconds)
- ◆ All permissions checks (successful and failed), including the object on which the permission was checked (if applicable)
- ◆ All actions that require DBA authority.

The transaction log

Each database has an associated transaction log file. The transaction log is used for database recovery. It is a record of transactions executed against a database.

 For information about the transaction log, see “The transaction log” [ASA Database Administration Guide, page 343].

The transaction log stores all executed data definition statements, and the user ID that executed them. It also stores all updates, deletes, and inserts and which user executed those statements. However, this is insufficient for some auditing purposes. By default, the transaction log does not contain the time of the event, just the order in which events occurred. It also contains neither failed events, nor select statements.

Turning on auditing

The database administrator can turn on **auditing** to add security-related information to the transaction log.

Auditing is off by default. To enable auditing on a database, the DBA must set the value of the public option AUDITING to ON. Auditing then remains enabled until explicitly disabled, by setting the value of the AUDITING option to OFF. You must have DBA permissions to set this option.

❖ To turn on auditing

1. Ensure that your database is upgraded to at least version 6.0.2.
2. If you had to upgrade your database, create a new transaction log.
3. Execute the following statement:

```
SET OPTION PUBLIC.AUDITING = 'ON'
```

☞ For more information, see “AUDITING option [database]” [ASA Database Administration Guide, page 578].

Retrieving audit information

You can use the Log Translation [dbtran] utility to retrieve audit information. You can access this utility from Sybase Central or from the command prompt. It operates on a transaction log to produce a SQL script containing all of the transactions, along with some information on what user executed each command. By using the `-g` option, *dbtran* includes more comments containing the auditing information.

To ensure a complete and readable audit record, the `-g` option automatically sets the following options:

- ◆ **-d** Display output in chronological order.
- ◆ **-t** Include trigger-generated operations in the output.
- ◆ **-a** Include rolled back transactions in the output.

You can run the Log Translation Utility against a running database server or against a database log file.

❖ To retrieve auditing information from a running database server

1. Make sure your user ID has DBA authority.
2. With the database server running, execute the following statement at a system command prompt:

```
dbtran -g -c "uid=DBA;pwd=SQL;..." -n asademo.SQL
```

☞ For information about connection strings, see “Connection parameters” [ASA Database Administration Guide, page 70].

❖ **To retrieve auditing information from a transaction log file**

1. Close the database server to ensure the log file is available.
2. At a system command prompt, execute the following statement to place the information from the file *asademo.log* and into the file *asademo.SQL*.

```
dbtran -g asademo.log
```

The `-g` option includes auditing information in the output file.

☞ For more information, see “The Log Translation utility” [ASA Database Administration Guide, page 508].

Adding audit comments

You can add comments to the audit trail using the `sa_audit_string` system stored procedure. It takes a single argument, which is a string of up to 200 bytes. You must have DBA permissions to call this procedure.

For example:

```
call sa_audit_string( 'Started audit testing here.' )
```

This comment is stored in the transaction log as an audit statement.

An auditing example

This example shows how the auditing feature records attempts to access unauthorized information.

1. As database administrator, turn on auditing.

You can do this from Sybase Central as follows:

- ◆ Connect to the ASA 9.0 Sample data source. This connects you as the **DBA** user.
- ◆ Right-click the **asademo** database icon and choose Options from the popup menu.
- ◆ Select Auditing from the list of options, and enter the value ON in the Public Setting box. Click Set Permanent Now to set the option and Close to exit.

Alternatively, you can use Interactive SQL. Connect to the sample database from Interactive SQL as user ID **DBA** with password **SQL** and execute the following statement:

```
SET OPTION PUBLIC.AUDITING = 'ON'
```

-
2. Add a user to the sample database, named **BadUser**, with password **BadUser**. You can do this from Sybase Central. Alternatively, you can use Interactive SQL and enter the following statement:

```
GRANT CONNECT TO BadUser  
IDENTIFIED BY 'BadUser'
```

3. Using Interactive SQL, connect to the sample database as **BadUser** and attempt to access confidential information in the **employee** table with the following query:

```
SELECT emp_lname, salary  
FROM DBA.employee
```

You receive an error message: do not have permission to select from employee.

4. From a command prompt, change directory to your Adaptive Server Anywhere installation directory, which holds the sample database, and execute the following command:

```
dbtran -g -c "dsn=ASA 7.0 Sample" -n asademo.SQL
```

This command produces a file named *asademo.SQL*, containing the transaction log information and a set of comments holding audit information. The lines indicating the unauthorized **BadUser** attempt to access the employee table are included in the file as follows:

```
--AUDIT-1001-0000287812 -- 1999/02/11 13:59:58.765 Checking  
Select permission on employee - Failed  
--AUDIT-1001-0000287847 -- 1999/02/11 13:59:58.765 Checking  
Select permission on employee(salary) - Failed
```

5. Restore the sample database to its original state so other examples you try in this documentation give the expected results.

Connect as the **DBA** user, and carry out the following operations:

- ◆ Revoke Connect privileges from the user ID **BadUser**.
- ◆ Set the **PUBLIC.AUDITING** option to 'OFF'.

Auditing actions outside the database server

Some database utilities act on the database file directly. In a secure environment, only trusted users should have access to the database files.

To provide auditing of actions, under Windows NT only, any use of *dbtran*, *dbwrite*, and *dblog* generates a text file in the same directory as the database file, with the extension *.alg*. For example, for *asademo.db*, the file is called *asademo.alg*. Records containing the tool name, Windows user name, and

date/time are appended to this file. Records are only added to the *.alg* file if the AUDITING option is set to ON.

Running the database server in a secure fashion

There are several security features you can set either when starting the database server or during server operation, including:

- ◆ **Starting and stopping databases** By default, any user can start an extra database on a running server. The `-gd` option allows you to limit access to this option to users with a certain level of permission in the database to which they are already connected. The permissible values include **DBA**, **all**, or **none**.

☞ For more information, see “-gd server option” [ASA Database Administration Guide, page 145].

- ◆ **Creating and deleting databases** By default, any user can use the CREATE DATABASE statement to create a database file. The `-gu` option allows you to limit access to this option to users with a certain level of permission in the database to which they are connected. The permissible values include **DBA**, **all**, **none**, or **utility_db**.

☞ For information, see “-gu server option” [ASA Database Administration Guide, page 150].

- ◆ **Stopping the server** The `dbstop` utility stops a database server. It is useful in batch files, or in other cases where interactive stopping of the server (by clicking Shutdown on the server window) is impractical. By default, any user can run `dbstop` to shut down a server. The `-gk` option allows you to limit access to this option to users with a certain level of permission in the database. The permissible values include **DBA**, **all**, or **none**.

☞ For more information, see “-gk server option” [ASA Database Administration Guide, page 146].

- ◆ **Loading and unloading data** The LOAD TABLE, UNLOAD TABLE, and UNLOAD statements all access the file system on the database server machine. If you are running the personal database server, you already have access to the file system and this is not a security issue. If you are running the network database server, unwarranted file system access may be a security issue. The `-gl` option allows you to control the database permissions required to carry out loading and unloading of data. The permissible values are **DBA**, **all**, or **none**.

☞ For more information, see “-gl server option” [ASA Database Administration Guide, page 147].

- ◆ **Encrypting client/server communications over the network** For greater security, you can force client/server network communications to be encrypted as they pass over the network.

☞ For more information, see “Encrypting client/server communications” on page 15.

Encrypting client/server communications

Client/server communication encryption is a separately licensable component and must be ordered before you can install it. To order this component, see the card in your SQL Anywhere Studio package or see <http://www.sybase.com/detail?id=1015780>.

You can set client/server encryption when you start the database server or in the client connection parameters. You can encrypt all native Adaptive Server Anywhere packets (Embedded SQL, ODBC, and OLEDB) that are transmitted to and from all clients. TDS packets (Java connections, including Sybase Central and Interactive SQL, as well as Open Client connections) are not encrypted.

When you use strong encryption by specifying `-ec ECC_TLS` or `RSA_TLS` in the server command, all connections to the server must perform a Certicom handshake. This handshake cannot be faked and Certicom encryption ensures that invalid packets, which may be intended to harm the server, are discarded.

❖ To force encryption of client/server communications from the server

1. Start the database server using the `-ec` option. For example:

```
dbsrv9 -ec simple,ECC_TLS -x tcpip "c:\Program Files\Sybase\
SQL Anywhere 9\asademo.db"
```

☞ For more information, see “-ec server option” [ASA Database Administration Guide, page 141].

❖ To force encryption of client/server communications from a particular client

1. Add the **Encryption (ENC)** connection parameter to your connection string.

```
"UID=DBA;PWD=SQL;ENG=myeng;LINKS=tcpip; Encryption=ECC_TLS
(trusted_certificates=sample.crt)"
```

You can also set this parameter on the Advanced tab of the Connect dialog and on the Network tab of the ODBC data source dialog.

☞ For more information, see “Encryption connection parameter [ENC]” [ASA Database Administration Guide, page 188].

Encrypting a database

As a database administrator, you can use database encryption to make it more difficult for someone to decipher the data in your database. You can choose to secure your database either with simple or with strong encryption.

Simple encryption

Simple encryption is equivalent to obfuscation and makes it more difficult for someone using a disk utility to look at the file to decipher the data in your database. Simple encryption does not require a key to encrypt the database. Simple encryption technology is supported in previous versions of SQL Anywhere.

Strong encryption

Strong database file encryption technology makes the database inoperable without the key (password). As well, it scrambles the information contained in your database and transaction log files so they cannot be deciphered simply by looking at the files using a disk utility. The data is completely inaccessible without the key.

Two algorithms have been chosen to implement strong encryption: AES, a block encryption algorithm chosen as the new Advanced Encryption Standard (AES) for block ciphers by the National Institute of Standards and Technology (NIST); and MDSR, a new 56-bit algorithm developed by Casio.

A database can be strongly encrypted using the `ENCRYPTION` and `KEY` options with the `CREATE DATABASE` statement. Similarly, the database administrator can initialize a database using the `dbinit` utility in combination with various options to enable strong encryption. You can also use the Sybase Central Create Database wizard to create a strongly encrypted database. Using the `dbinit` utility with the `-ea` option enables strong encryption and sets the algorithm to either AES or MDSR. Using the `dbinit` utility in combination with the `-ek` or `-ep` option enables strong encryption and indicates whether the key is to be specified in a prompt box or at the command prompt.

❖ To create a strongly encrypted database (SQL)

1. Connect to an existing database from Interactive SQL.
2. Execute a `CREATE DATABASE` statement that includes the `ENCRYPTION` and `KEY` options. For example, the following statement creates a database file named *myencrypteddb.db* in the C:\directory.

```
CREATE DATABASE 'c:\\myencrypteddb'  
TRANSACTION LOG ON  
ENCRYPTED ON  
KEY '0kz2o52AK#'  
ALGORITHM 'MDSR'
```

❖ **To create a strongly encrypted database (command prompt)**

1. At a command prompt, use the *dbinit* utility to create a database. You must include the following options:

- ◆ `-ea` to specify the encryption algorithm.
- ◆ `-ek` or `-ep` to specify the encryption key and whether you want to enter it at the command prompt or in a dialog box.

The following command (entered all on one line) creates a strongly encrypted database and specifies that the encryption key is entered as part of the command.

```
dbinit -ea MDSR -ek "0kZ2o56AK#" "myencrypteddb.db"
```

2. Start the database from the command prompt.

```
dbeng9 myencrypteddb.db -ek "0kZ2o56AK#"
```

☞ For more information about the encryption key, see “Encryption Key connection parameter [DBKEY]” [*ASA Database Administration Guide*, page 190].

As with most passwords, it is best to choose a key value that cannot be easily guessed. We recommend that you choose a value for your key that includes between 8 and 30 digits, a combination of upper and lower case characters, and numbers, letters, and special characters.

Caution

Be sure to store a copy of your key in a safe location. You require the key each time you want to start or modify the database. A lost key will result in a completely inaccessible database, from which there is no recovery.

Controlling strong encryption

In Adaptive Server Anywhere, the database administrator has control over four aspects of strong encryption, including: strong encryption status, the encryption key, protection of the encryption key, and the encryption algorithm.

Strong encryption status

Although you can't simply turn strong encryption on or off in an existing database, you can choose from two options when it comes to implementing strong encryption. You can either create a database from scratch with strong encryption, or you can rebuild an existing database and change the encryption status at that time. Rebuilding the database unloads all of the data and schema of an existing database, creates a new database (at which point you can change a variety of settings including strong encryption status), and

reloads the data into the new database. You need to know the key to unload a strongly encrypted database.

☞ For more information on these features, see

- ◆ “Reloading a Database” [*ASA SQL User’s Guide*, page 543]
- ◆ “CREATE DATABASE statement” [*ASA SQL Reference*, page 292]

The encryption key

As with most passwords, it is best to choose a key value that cannot be easily guessed. The key can be of arbitrary length, but generally the longer the key, the better because a shorter key is easier to guess than a longer one. As well, including a combination of numbers, letters, and special characters decreases the chances of someone guessing the key. You must supply this key each time you want to start the database. Lost or forgotten keys result in completely inaccessible databases.

Protection of the encryption key

You can choose whether the encryption key is entered at the command prompt (the default) or into a prompt box. Choosing to enter the key in a prompt box provides an extra measure of security because the key is never visible in plain sight. Clients are required to specify the key each time they start the database. In cases where the database administrator starts the database, clients never need to have access to the key.

☞ For more information, see “-ep server option” [*ASA Database Administration Guide*, page 143].

The encryption algorithm

When you strongly encrypt a database, you can choose to use either the AES algorithm (the default if an algorithm isn’t specified explicitly) or MDSR.

AES has recently been through a period of international evaluation and has now been chosen as the new Advanced Encryption Standard block cipher algorithm. It has many properties that lend itself well to encryption of Adaptive Server Anywhere databases in terms of performance and size.

☞ For more information about database encryption algorithms, see:

- ◆ “Initialization utility options” [*ASA Database Administration Guide*, page 487]
- ◆ “CREATE DATABASE statement” [*ASA SQL Reference*, page 292]

Performance issues

Performance of Adaptive Server Anywhere is somewhat slower when the database is encrypted. The performance impact depends on how often pages are read from or written to disk, and can be minimized by ensuring that the server is using an adequate cache size.


☞ You can increase the starting size of the cache with the `-c` option when you start the server. For operating systems that support dynamic resizing of the cache, the cache size that is used may be restricted by the amount of memory that is available; to increase the cache size, increase the available memory.

☞ For more information, see:

- ◆ “Using the cache to improve performance” [*ASA SQL User’s Guide*, page 176]
- ◆ “`-c` server option” [*ASA Database Administration Guide*, page 133]

Security tips


As database administrator, there are many actions you can take to improve the security of your data. For example, you can:



- ◆ **Change the default user ID and password** The default user ID and password for a newly created database is **DBA** and **SQL**. You should change this password before deploying the database.
- ◆ **Require long passwords** You can set the `MIN_PASSWORD_LENGTH` public option to disallow short (and therefore easily guessed) passwords.
 For information, see “`MIN_PASSWORD_LENGTH` option [database]” [*ASA Database Administration Guide*, page 608].
- ◆ **Restrict DBA authority** You should restrict DBA authority only to users who absolutely require it since it is very powerful. Users with DBA authority can see and do anything in the database.

You may consider giving users with DBA authority two user IDs: one with DBA authority and one without, so they can connect as DBA only when necessary.
- ◆ **Drop external system functions** The following external functions present possible security risks: `xp_cmdshell`, `xp_startmail`, `xp_startsmtp`, `xp_sendmail`, `xp_stopmail`, and `xp_stopsmtmp`.

The `xp_cmdshell` procedure allows users to execute operating system commands or programs.

The e-mail commands allow users to have the server send e-mail composed by the user. Malicious users could use either the e-mail or command shell procedures to perform operating-system tasks with authorities other than those they have been given by the operating system. In a security-conscious environment, you should drop these functions.

 For information on dropping procedures, see “`DROP` statement” [*ASA SQL Reference*, page 408].
- ◆ **Protect your database files** You should protect the database file, log files, dbspace files, and write files from unauthorized access. Do not store them within a shared directory or volume.
- ◆ **Protect your database software** You should similarly protect Adaptive Server Anywhere software. Only give users access to the applications, DLLs, and other resources they require.

- ◆ **Run the database server as a service or a daemon** To prevent unauthorized users from shutting down or gaining access to the database or log files, run the database server as a Windows service. On UNIX, running the server as a daemon serves a similar purpose.
 For more information, see “Running the server outside the current session” [ASA Database Administration Guide, page 21].
- ◆ **Set ASTMP to a unique directory** To make the engine secure on UNIX platforms, set ASTMP to a unique directory, and make the directory read, write, and execute protected against all other users. Doing so forces all connections to use TCP/IP, which is more secure than the shared memory connection.
- ◆ **Strongly encrypt your database** Strongly encrypting your database makes it completely inaccessible without the key. You cannot open the database, or view the database or transaction log files using any other means.
 For more information, see “-ep server option” [ASA Database Administration Guide, page 143] and “-ek database option” [ASA Database Administration Guide, page 168].

PART II

CONFIGURING ADAPTIVE SERVER ANYWHERE IN A C2-COMPLIANT MANNER

This part describes the mechanics of setting up, installing and running Adaptive Server Anywhere in a C2-compliant manner. It also contains additional information you may find useful when operating Adaptive Server Anywhere in a manner equivalent to the C2-certified configuration.

CHAPTER 2

Installation

About this chapter

This chapter describes the procedures for installing Adaptive Server Anywhere (ASA) in a manner equivalent to the C2 certified configuration. The instructions contained in this document must be followed exactly to ensure an environment equivalent to the certified environment.

Contents

| Topic: | page |
|--|-------------|
| Hardware installation | 26 |
| Operating system installation | 27 |
| Adaptive Server Anywhere software installation | 28 |
| Creating a database | 32 |
| Running the database engine | 34 |

Hardware installation

Set up the hardware as specified in the Hardware User's Manual with the restrictions listed in the *Microsoft Windows C2 NT Administrator's and User's Security Guide*, chapter 4.

Additional hardware information is available in the Final Evaluation Report (FER), which is available on Sybase's website.

Operating system installation

The first step in creating a C2 certified configuration involves installing and setting up the operating system.

❖ To install and set up the operating system

1. Install Windows NT 4.0 in the C2 certified configuration (including Service Pack 6a and the C2 security hotfix), as specified in the Microsoft Windows NT C2 Administrator's and User's Security Guide, Chapter 4.
2. Log in to Windows NT as Administrator.
3. From the Start menu, choose Programs ► Administrative Tools (Common) ► User Manager for Domains.
4. Using the User Manager, create a user called sybase.
 - ◆ Give this user a secure password.
 - ◆ Add the user to *only* the Users group.
 - ◆ Clear the User Must Change Password at Next Logon checkbox.
 - ◆ Click Add, and then Close.
5. From the Policies menu, choose User Rights.
6. Check the Show Advanced User Rights checkbox, and then select Log On As A Service from the Right dropdown list.
7. Click Add.

A dialog appears.
8. In the List Names From dropdown list, select \\machine_name.
9. In the Add Names field, type **sybase**, and click OK.
10. Click OK to close the dialog.
11. If you wish to audit logons and logoffs of users (which can help in correlating Adaptive Server Anywhere audit records with Windows NT users) choose Policies ► Auditing, and:
 - ◆ Select the Audit These Events option.
 - ◆ Check the Logon and Logoff checkbox under Success.
 - ◆ Select any other events you want audited, and click OK.
12. Close the User Manager (optional).

Adaptive Server Anywhere software installation

Next, you have to install Adaptive Server Anywhere in a C2-compliant manner. For C2 compliance you must use Adaptive Server Anywhere version 7.0.0, English only, without any EBFs (express bug fixes), in a standalone environment. Most of this book describes how to operate the current version of the software, but this section refers specifically to the C2-certified release.

❖ To install Adaptive Server Anywhere 7.0.0

1. Log in to Windows NT as administrator.
2. Download the Adaptive Server Anywhere C2 patch from www.sybase.com/developer.
3. Run *ASAC2Patch.exe* and save the files into the default directory (*C:\ASAC2Patch*).

ASAC2Patch.exe is a self-extracting archive.

☞ For information on this patch, see [“The Adaptive Server Anywhere C2 patch” on page 76](#).

4. Open a command prompt window.

The Adaptive Server Anywhere installation includes MDAC (Microsoft Data Access Components). The MDAC installation replaces some Windows NT system DLLs which are part of the Windows NT TCB (trusted computing base). To avoid this, you must first make copies of these DLLs, and then replace them after the Adaptive Server Anywhere installation. The Adaptive Server Anywhere C2 Patch includes three batch files to facilitate this procedure.

The first batch file creates a temporary directory and copies fourteen *.dll* files and one *.exe* file from the *C:\winnt\system32* directory. To run the first batch file, enter the following commands at the command prompt:

```
C:
cd \ASAC2Patch
mdac1
exit
```

5. Install the Adaptive Server Anywhere 7.0.0 software, using the following guidelines:
 - ◆ Clear the Adaptive Server Anywhere for NetWare checkbox.
 - ◆ Clear the Adaptive Server Anywhere for Windows CE checkbox.
 - ◆ Clear the UltraLite development components checkbox.

- ◆ Clear all options under Synchronization.
 - ◆ Clear the PowerDynamo 3.5, PowerDesigner, and Infomaker 7 options.
 - ◆ If available, clear the Encryption for MobiLink Synchronization checkbox.
 - ◆ Use the default values for installation directories.
6. Reboot your machine after the installation is complete.
 7. Log in to Windows NT as an administrator.
 8. Install the Adaptive Server Anywhere C2 patch according to the directions in *readme.txt* (located in *C:\ASAC2Patch*).
You do not need to reboot the machine after this step.
 9. Set permissions on the software directory as follows:
 - ◆ Double-click My Computer. Right-click the directory containing the Adaptive Server Anywhere software (it should be *C:\Program Files\Sybase*), and choose Properties.
 - ◆ Open the Security tab and then click the Permissions button.
 - ◆ Select Everyone, and change the Type of Access to Read.
 - ◆ Click Add. On the dialog that appears, select *\\machine_name* from the List Names From dropdown list. Select Administrators from the Names list and click Add.
 - ◆ Click Show Users. Select sybase from the Names list and click Add. Change Type of Access to Full Control, and click OK.
 - ◆ Make sure the list contains only the three entries mentioned above.
 - ◆ Check the Replace Permissions on Subdirectories checkbox.
 - ◆ Click OK, and answer Yes to the prompt.
 10. Create a folder for the database and transaction log files. For example, you may create a folder *C:\Databases*. In the remainder of this document, this folder is referred to as the **C2 database folder**. Set the permissions on this folder as follows:
 - ◆ Double-click My Computer. Right-click the Databases folder and select Properties.
 - ◆ Click the Security tab and click the Permissions button.
 - ◆ Remove the Everyone entry.
 - ◆ Click Add. On the dialog that appears, select *\\machine_name* in the List Names From dropdown list, and then type **sybase** in the Add Names field. Change Type of Access to Full Control, and click OK.

◆ Click OK.

11. Create a folder under *C:* called *ASTMP* for the engine to use as temporary storage space. Set the same permissions as for the Databases folder in the previous step.
12. Set the System environment variable *ASTMP* to the temporary folder just created by right-clicking the My Computer icon, and choosing Properties. Click the Environment tab. In the Upper listbox, click any entry. Change the Variable entry to *ASTMP*, and change the Value entry to *C:\ASTMP*. Click Set, and then click OK.
13. The second batch file contained in the Adaptive Server Anywhere C2 Patch copies the *.dll* and *.exe* files from the temporary directory created by *mdac1.bat* into the *C:\winnt\system32* directory. To run the second batch file, from the Start menu, choose Programs ► Command Prompt. At the command prompt, enter the following commands:

```
C:
cd \ASAC2Patch
mdac2
exit
```

14. When putting Windows NT into the certified configuration, several registry keys are deleted. During Adaptive Server Anywhere installation, two of these keys are re-created. For Windows NT to remain in its certified configuration, these keys must be deleted again. Use *regedt32.exe* to delete the following registry keys:

| Key | HKEY_LOCAL_MACHINE\SOFTWARE |
|--------|---|
| Subkey | Microsoft\OS/2 Subsystem for Windows NT |
| Entry | delete all subkeys |
| Key | HKEY_LOCAL_MACHINE\SYSTEM |
| Subkey | CurrentControlSet\Control\Session Manager\Environment |
| Entry | Os2LibPath |
| Value | delete entry |

15. You must also ensure that these files have the correct permissions as shown below:

| Files | C2-Level Permissions |
|-------------------------------|--|
| BOOT.INI, NTDETECT.COM, NTLDR | Administrators: Full Control SYSTEM: Full Control |

16. Close all open windows and reboot your machine.

You must reboot your machine for the Service Control Manager to read changes to system environment variables.

17. Log in to Windows NT as administrator.

18. The third batch file contained in the Adaptive Server Anywhere C2 Patch cleans up the temporary directory created by *mdac1.bat*. To run the third batch file, open a command prompt window. At the command prompt, enter the following commands:

```
C:  
cd \ASAC2Patch  
mdac3  
exit
```

Creating a database

To operate in a C2 compliant configuration, your database must be C2 compliant as well. All connections to the database must use the integrated login mechanism. Standard connections to the database (for example, specifying a user ID and password) are not allowed in the certified configuration.

❖ To create a C2 compliant database

1. Log in as sybase.
2. From the Start menu, choose Programs ► Command Prompt.
3. Use the dbinit utility to create a database with the following restrictions:
 - ◆ You must use the `-i` switch to disable jConnect support.
 - ◆ You must not use the `-k`, or `-n` switches.
 - ◆ You must put the database file in your C2 database folder.
 - ◆ If you specify a transaction log file using the `-t` switch, or a transaction log mirror file using the `-m` switch, the files specified must be in your C2 database folder.
 - ☞ For information on using the dbinit utility in the certified configuration, see [“Initialization utility” on page 67](#). For information about the database folder, see [“Adaptive Server Anywhere software installation” on page 28](#).
4. Once the database is created, you need to connect to the database.

This connection must only be used to set the `min_password_length` option and the DBA's password.
5. At a command prompt, type **dbisqlc -c**
UID=DBA;PWD=SQL;DBF=file where *file* is the full path of the database file created above.

Interactive SQL appears after a few seconds.

 - ☞ For information on using the dbisqlc utility in the certified configuration, see [“Interactive SQL utility” on page 70](#) and [“Restrictions” on page 52](#).
6. Type set option `public.min_password_length=6` (or higher) and click Execute.
7. Type grant connect to DBA identified **by newpw** where *newpw* is the new password for the DBA account, and click Execute.

The new password must be at least as long as the number entered in step 5, and should not be easy to guess.

8. Type `grant integrated login to sybase as user DBA`, and click Execute.
9. Type `set option public.login_mode='Integrated'`, and click Execute.
10. Exit Interactive SQL by clicking the X in the top right corner of the window.

Running the database engine

1. Log in to Windows NT as administrator.

You require administrator privileges to create, start, and stop services.

2. Open a command prompt.
3. Use the dbsvc utility to create a service with the following restrictions:
 - ◆ You must use the `-a` switch to specify the sybase account, and the `-p` switch to specify its password.
 - ◆ You must not use the `-as` or `-i` switches.
 - ◆ The executable name should be

`C:\Program Files\Sybase\SQL Anywhere 9\win32\dbeng9.exe`
for the personal database server, or

`C:\Program Files\Sybase\SQL Anywhere 9\win32\dsrv9.exe`
for the database server.

- ◆ You must use the following engine parameters:
 - `-n engine name`
 - `-sc`
 - `-gd DBA`
 - `-gk DBA`
 - `-gl DBA`
 - `-gu DBA`
 - `-x namedpipes(TDS=NO)`

4. Enter the full path to any database files you wish to run.

The path should be in the format *database-folder\filename.db*, where *database-folder* is your C2 database folder, and include any other relevant parameters.

For example, the following command line creates a service called `asa_svc` that starts manually, and refers to a network server. It runs under the sybase account, whose password is `sybase_password`. It executes the following command:

```
C:\Program Files\Sybase\SQL Anywhere 9\win32\
dsrv9.exe -n asa_c2 -sc -gd DBA -gk DBA
-gl DBA -gu DBA -x namedpipes(TDS=NO)
database-folder\c2test.db
dbsvc -a sybase -p sybase_password -s manual
-t network -w asa_svc C:\Program Files\Sybase\
SQL Anywhere 9\win32\dsrv9.exe -n asa_c2 -sc
-gd DBA -gk DBA -gl DBA -gu DBA
-x namedpipes(TDS=NO) database-folder\c2test.db
```

☞ For information on using the engine and the server in the certified configuration, see [“Database engine/server” on page 63](#).

5. To start and stop the service, run the Windows NT service manager from the control panel. From the Start menu, choose Settings ► Control Panel, and then double-click Services.

The service you just created appears under Adaptive Server Anywhere - svc where svc is the service name you specified on the dbsvc command line.

6. Use the Start and Stop buttons to start and stop the service.

CHAPTER 3

Auditing

About this chapter

This chapter contains information on reading auditing output, and correlating Adaptive Server Anywhere auditing output with Windows NT auditing.

Contents

| Topic: | page |
|---------------------------------|-------------|
| Enabling and disabling auditing | 38 |
| Reading auditing output | 39 |
| Audit records | 40 |
| Administration of audit records | 47 |
| Auditing of database utilities | 48 |
| Correlating audit records | 49 |

Enabling and disabling auditing

Auditing is OFF when you create a database. However, you can enable and disable auditing using the auditing public option at any time.

❖ To start auditing on a particular database

1. Turn the option ON using the following SQL statement:

```
SET OPTION public.auditing='on'
```

Only users with DBA authority can set public options. Once this option has been turned on, all permission checks and connection attempts are audited.

❖ To stop (disable) auditing on a particular database

1. Turn the option OFF using the following SQL statement:

```
SET OPTION public.auditing = 'off'
```

Only a user with DBA authority can issue this statement.

☞ For more information and a complete list of the types of audit records that the engine or server can generate, see [“Audit records” on page 40](#).

Note

Auditing is optional when running in a C2 certified configuration.

Reading auditing output

You can use the `dbtran` utility to retrieve audit records from the transaction log. The transaction log file is usually found in the `dbname.log` file, located in the same directory as the database file.

The `-g` switch tells `dbtran` to include audit records in the output. The output from `dbtran` is a SQL script with comments interspersed. This SQL script can be used to recover the database if a failure occurs. When using the `-g` option, the output file is entirely comments, since the `-g` option implies the `-d` option (which records transaction log information in the order in which it was contained in the log, not in the default commit order). Do not use output in this format for recovery of a database. Each line is commented to avoid accidental use of this file for recovery.

When a user connects to the database, an audit record is generated:

```
-CONNECT-1001-0000198970-dba-1998/dec/03 14:54
```

The data following the `CONNECT` are interpreted as follows:

- ◆ 1001 is the connection ID assigned to this connection. Any transactions listed below with connection ID 1001 belong to this connection, until another `CONNECT-1001` is found.
- ◆ 0000198970 is the byte offset of the record in the transaction log.
- ◆ dba is the user name logged in on this connection.
- ◆ 1998/dec/03 14:54 is the date and time of the connection.

Other records have the connection ID and byte offset, but only the `CONNECT` record has the user name and date/time. Note that disconnects are not logged. If another `CONNECT` record is generated with the same connection ID as a previous `CONNECT` record, you can assume that the first user has disconnected. Although the connection ID is reused, the second connection is entirely unrelated to the first.

Audit records

This section identifies the different audit records that may be generated by the engine or server, the information contained in the record, and when the record is generated. Descriptions of the audit records generated by the three database utilities dblog, dbtran, and dbwrite in the .alg file appear in [“Auditing of database utilities” on page 48](#).

| Type | Information | Use |
|------------------------------------|--|--|
| Attempting Operation | date/time, SQL of attempted operation | <p>This record displays the operation being attempted. This is necessary because of the way the transaction log works.</p> <p>The transaction log contains SQL to replicate changes made to the database data or schema if recovery becomes necessary. Audit records become part of this log so that each permission check is recorded as it happens, and so that the activity on the database can be recreated later.</p> <p>However, if a permissions check fails, then the operation being attempted doesn’t actually happen, and therefore doesn’t get logged. In this case, there is no way of knowing what was being attempted. This is especially important when a non-DBA user attempts something that requires DBA authority.</p> <p>For this reason, all DDL statements (and a few other statements as well) are recorded before they are attempted.</p> |
| Operation Succeeded / Failed | date/time, success or failure | <p>This record indicates the success or failure of the most recent Operation Attempt, Attempting to set public option, or Attempting SETUSER record for the same connection ID.</p> |

| Type | Information | Use |
|---------------------|---|--|
| Checking permission | date/time, type of permission / authority, table name (if applicable), column name (if applicable), procedure / function name (if applicable) | <p>This record indicates that a permission or authority check of some kind took place. The permission in question is indicated, and can be one of:</p> <p>DBA / Resource authority</p> <p>Insert / Update / Select / Delete / Alter / Resource permission on a table</p> <p>Update / Select / Resource permission on a table and column</p> <p>Grant Insert / Update / Select / Delete / Alter / Resource permission on a table</p> <p>Grant Update / Select / Resource permission on a table and column</p> <p>Execute permission on a procedure or function</p> <p>Grant Execute permission on a procedure or function</p> |
| Checking user | date/time, user name | <p>This record indicates that a user check took place. This can help determine ownership of objects, for example, user bob owns table T. If an insert is attempted on table T, we must check to see if the current user is user bob. The text of the record is Checking to see if user is user name.</p> |
| Set Public Option | date/time, name of option | <p>This record indicates that a user attempted to set an option owned by the PUBLIC user. Only users with DBA authority are allowed to do this, so this check will always be followed by a DBA authority check. An Operation Succeeded/Failed record indicates success or failure.</p> |

| Type | Information | Use |
|-----------------------------|--|---|
| Auditing Enabled / Disabled | date/time | This record indicates that the auditing public option has been changed. This record will always follow a Set Public Option record. This record is generated whether auditing is turned on or off. However, this record will not be generated if the user sets the auditing variable to ON when auditing is already on, or if the user sets the variable to OFF when auditing is already off. |
| Attempting SETUSER | date/time, name of user | This record indicates that a user has attempted a SETUSER command with a parameter. Only users with DBA authority are allowed to do this, so this record will always be followed by a DBA authority check. An Operation Succeeded/Failed record indicates success or failure. Note that the SETUSER command with no arguments is neither audited nor logged, since any user can execute that statement. |
| Attempting Connection | date/time, user name (if successful), machine address (local if the same machine), port type, success or failure | This record indicates that a connection attempt took place. |

| Type | Information | Use |
|-------------------------------|-------------------------------|--|
| Trigger firing / finishing | date/time, name of trigger | This record indicates that a trigger has fired or finished executing. All audit records for the same connection in between these two records are auditing the trigger execution. Note that triggers execute with the permission of the table owner, not the caller, so any permission checks audited in between Trigger firing and Trigger finishing records are done with respect to the table owner. Examining the SQL statement that caused the trigger to fire will reveal the table owner. Look at the SQL statement for the same connection immediately preceding the Trigger firing record. It will be an insert, update, or delete on a table. The table name will be in the format owner.table. |
| String | date/time, string | Records of this type can be inserted into the audit trail using a system stored procedure called sa_audit_string. This procedure is executable only by users with DBA authority. Any string (up to 128 characters) can be specified. |

Table 6.2 – Format of audit records – fixed

| Type | Format |
|-------------------------|--|
| Transaction redo header | 1 byte |
| Connection identifier | 3 bytes |
| date / time | 11 bytes: <ul style="list-style-type: none"> ♦ 2 bytes year (for example, 1998) ♦ 1 byte month (1–12) ♦ 1 byte day (1–31) ♦ 1 byte hour (0–23) ♦ 1 byte minute (0–59) ♦ 1 byte second (0–59) ♦ 4 bytes microsecond (0–999999) |
| Audit type | 1 byte |

Table 6.3 – Format of audit records – variable by type

| Type | Format |
|-----------------------|--|
| AUDIT_ENABLE_AUDITING | ♦ 1 byte 1 (auditing enabled) or 0 (auditing disabled) |
| AUDIT_SET_PUB_OPTION | ♦ 2 bytes length of following string (n) ♦ n bytes option name |
| AUDIT_OP_ATTEMPT | ♦ 2 bytes length of following string (n) ♦ n bytes SQL of attempted operation |
| AUDIT_OP_SUCCESS | ♦ 1 byte 1 (operation succeeded) or 0 (operation failed) |

| Type | Format |
|------------------|--|
| AUDIT_PERM_CHECK | <ul style="list-style-type: none"> ◆ 1 byte 1 (success) or 0 (failure) ◆ 2 bytes length of following string (n) ◆ n bytes permission type (for example, select, update, or execute) ◆ 2 bytes length of following string (n) ◆ n bytes object (table, view, procedure, etc.) name ◆ 2 bytes length of following string (n) ◆ n bytes column name, if applicable |
| AUDIT_USER_CHECK | <ul style="list-style-type: none"> ◆ 1 byte 1 (success) or 0 (failure) ◆ 2 bytes length of following string (n) ◆ n bytes user name |
| AUDIT_CONNECTION | <ul style="list-style-type: none"> ◆ 1 byte 1 (success) or 0 (failure) ◆ 2 bytes length of following string (n) ◆ n bytes user name (if connection succeeded) ◆ 2 bytes length of following string (n) ◆ n bytes machine ID |
| AUDIT_SETUSER | <ul style="list-style-type: none"> ◆ 2 bytes length of following string (n) ◆ n bytes user name |
| AUDIT_TRIGGER | <ul style="list-style-type: none"> ◆ 2 bytes length of following string (n) ◆ n bytes name of trigger ◆ 3 bytes fired or finished |

| Type | Format |
|--------------|---|
| AUDIT_STRING | <ul style="list-style-type: none">◆ 2 bytes length of following string (n)◆ n bytes variable text string |

Administration of audit records

The Log translation [dbtran] utility can retrieve audit records from the transaction log. Using the -u or -x switches when invoking dbtran, records can be filtered depending on the user name. Audit records cannot be deleted. However, the transaction log can be purged or truncated using the dblog or dbbackup utilities.

☞ For more information about purging the transaction log, see [“Transaction log utility” on page 69](#).

If the audit log (in the case of Adaptive Server Anywhere, the transaction log) becomes full, the engine or server will rollback all pending transactions and fail all subsequent requests. At this point, the transaction log must be truncated in order to continue using the database. It is strongly recommended that you back up the transaction log before truncating it. The easiest way to back up the transaction log is to stop the engine, and then copy the file to another disk. You can then delete the old transaction log file and restart the engine or server. A new transaction log file will then be created.

Auditing of database utilities

Some database utilities perform actions that must be audited, but do not necessarily communicate with a running engine or server. These utilities must be audited separately. The utilities in question are dblog, dbwrite, and dbtran. These utilities check the database or transaction log to see if auditing is enabled. If so, they audit their invocation by writing to a file called *dbname.alg*, located in the same directory as the database file.

The *.alg* file is a text file, and can be viewed with any standard editor, such as Notepad. You can also use text-file sort and filter utilities (such as grep) to retrieve audit records for a particular user or utility.

Each audit record consists of a single line, in the following format:

```
2000/07/07 15:31:17.316 - User NT user name invoking  
utility name
```

You can delete records from this file at any time, simply by deleting them in the editor and saving the file. You can also delete the file at any time.

Utilities that generate records into this file will fail if they cannot write to this file (for example, if the file system is full). Accesses to the *.alg* file can be audited using the Windows NT audit mechanism.

Correlating audit records

In some cases, it may be useful to know the name of the user who was logged into Windows NT at the time that some audit records were generated. For example, if a DBA notices a lot of failed logon attempts grouped together, he may want to know who was logged into Windows NT at the time that these attempts were made. There are two ways to do this, depending on the type of information that is required.

In the above example, you would simply record the time at which the audit event in question took place – all audit events include the date and time of the event. Then, log into Windows NT as the administrator, and run the Event Viewer application. From the Log menu, choose Security to see the logon and logout audit records. Locate and double click the Logon/Logoff event immediately before the date and time of the audit event in question. It should be a Successful Logon event. The user name and domain of the user that logged on will appear, and tell you who was logged into the Windows NT workstation at the time that the audited event occurred. Note that this is only possible if auditing of Windows NT logons and logouts was enabled during [“Operating system installation” on page 27](#).

If the audit log contains information about a specific connection, and you need to correlate that with a particular Windows NT user, this second method is easier. Since integrated login is used for all connections, the database user is mapped to a particular Windows NT user. Since this mapping must be one-to-one, we know that no other Windows NT user can be mapped to this database user. To find the name of the Windows NT user given the database login ID, execute the following SQL statement:

```
SELECT lg.integrated_login_id
FROM syslogin lg
KEY JOIN sysuserperm p
WHERE p.user_name='login ID'
```


CHAPTER 4

Restrictions and Other Security Concerns

About this chapter

This chapter describes C2 certification restrictions and other security concerns.

Contents

| Topic: | page |
|--|--------------------|
| Restrictions | 52 |
| Security warnings | 55 |
| Changing ownership on nested objects | 56 |
| Revoking DBA authority | 58 |
| The TCB subset | 59 |

Restrictions

The following restrictions are required for Adaptive Server Anywhere to run in the certified C2 configuration.

1. Do not delete, modify, or replace any files under the Adaptive Server Anywhere installation directory, with the following exceptions:
 - ◆ *win32\util_db.ini* – this file may be modified as required.
 - ◆ *win32\asasrv.ini* – this file may be modified or deleted as required.
 - ◆ *win32\rebuild.bat* – this file may be modified as required.
 - ◆ *win32\backup.syb* – this file may be modified or deleted as required.
 - ◆ *win32\procdebug.bat* – this file may be modified as required.
 - ◆ *win32\custom.SQL* – this file may be modified as required.
 - ◆ *win32\tjava.pdf* – this file may be deleted as required.
2. Do not add any new files under the Adaptive Server Anywhere installation directory.
3. The sybase account password should only be given to one person.
4. The path for the sybase account should not contain any directories other than *%SystemRoot%\system32*, *%SystemRoot%*, and the Adaptive Server Anywhere *win32* directory.
5. Grant *only* the Login as a Service privilege to the sybase account.
6. DBA authority is very powerful. Only grant DBA authority to those users who require it. The number of DBA users should be kept to a minimum. However, each person who requires DBA authority should be given a separate account with DBA authority granted to it (for example, do not use shared DBA accounts).
7. DBAs who will be using the database outside of their DBA capacity should be given two different Adaptive Server Anywhere user accounts—one with DBA authority and one without. DBAs should only use the account with DBA authority when necessary.
8. The password for the DBA account must be changed upon creation of a new database.
9. The value for the *min_password_length* public option must be set to at least 6 upon creation of a new database.
10. The database engine or server must be run as a Windows NT service. Adaptive Server Anywhere is only certified when running as a service.

11. The following switches must be specified on the engine or server start line

```
-sc -gd DBA -gk DBA -gl DBA -gu DBA  
-x namedpipes(TDS=NO)
```

The engine or server start line is specified when executing the dbsvc utility, so these switches must be included in the Details part of the dbsvc command.

 For more information, see [“Service creation utility” on page 68](#) for details on dbsvc.

12. Do not use the `-x` parameter to start up any ports other than Named Pipes. Adaptive Server Anywhere is only certified in a standalone environment.
13. Do not grant REMOTE_DBA authority to any user.
14. Do not grant execute permission on the following system procedures to any user or group:
- ◆ xp_cmdshell
 - ◆ xp_startmail
 - ◆ xp_sendmail
 - ◆ xp_stopmail
 - ◆ xp_read_file
 - ◆ xp_write_file
 - ◆ sp_audit_string
 - ◆ java_debug_version
 - ◆ java_debug_connect
 - ◆ java_debug_disconnect
 - ◆ java_debug_get_existing_vms
 - ◆ java_debug_free_existing_vms
 - ◆ java_debug_wait_for_debuggable_vm
 - ◆ java_debug_get_vm_name
 - ◆ java_debug_release_vm
 - ◆ java_debug_attach_to_vm
 - ◆ java_debug_detach_from_vm
 - ◆ java_debug_detach_request
 - ◆ Any system procedures introduced after version 7.
15. Do not create stored procedures or functions owned by any user with DBA authority.

-
16. Do not create triggers on any tables owned by any user with DBA authority.
 17. Upgrade older databases by running the dbupgrad utility before using them.
 - ☞ For more information about upgrading a database, see “Upgrading a database using the dbupgrad command-line utility” [*ASA Database Administration Guide*, page 543].
 18. Databases must use a transaction log file. Do not use the `-n` switch (no transaction log) when creating a database and do not execute `dblog -n` (do not use a transaction log or mirror) on a database.
 19. All database, transaction log, dbspace, write file, and mirror files should be stored in non-shared, protected directories.
 - ☞ For guidelines on how to protect a directory, see “[Adaptive Server Anywhere software installation](#)” on page 28.
 20. The java.net package is disabled in the engine or server. Java running in the database will not be able to use this package.
 21. The java_input_output public option must always be set to OFF (the default).
 22. Do not create a database user called guest. Such a user would allow any Windows NT user to connect to the database using integrated login.
 23. Always set the login_mode public option to Integrated during database installation.
 - ☞ For more information, see “[Creating a database](#)” on page 32.
 24. All connections to the database must use the integrated login mechanism. Standard connections to the database (that is, those specifying user ID and password) are not allowed in the certified configuration.
 25. All integrated login mappings must be one-to-one. No two Windows NT user names may be mapped to the same database user.
 26. Embedded SQL programs must not use the db_delete_file function because the name of the file being deleted is not audited.
 27. Do not grant SELECT access on sys.sysuserperm or sys.syslogin to any non-DBA user.

Security warnings

Below are some other security issues to be aware of:

1. Since triggers execute with the permission of the table owner, it is possible for any user with ALTER permission on a table to write a trigger that accesses other tables owned by the same user. Please be aware that by granting ALTER permission on a table to another user, you are effectively granting all permissions on all of your tables to that user.
2. Audit records are created when a trigger is fired, and when the stored procedure executed by the trigger finishes. The user ID listed in these audit records is that of the owner of the table on which the trigger is defined.
3. Stored procedures may contain the GRANT command. When such a procedure is executed, the GRANT is done with the permissions of the owner of the stored procedure, not those of the caller. Be aware of this when creating stored procedures containing GRANT statements.
4. Windows NT has the ability to audit actions taken by users. It is recommended that users configure Windows NT to audit the sybase user. Note that such auditing could produce a large amount of data.

☞ For more information, see [“Operating system installation” on page 27](#).

5. Permissions on tables and columns are cumulative, but independent. This means that if executing two different GRANT statements gives overlapping permissions, revoking one of the two does not revoke the other.

For example, if user fred executes `GRANT UPDATE (Street) on the Employee table to sue`, Sue can update the Street column of table Employee.

If user fred subsequently executes `GRANT UPDATE on the Employee table to sue`, Sue is then able to update any column of the Employee table.

If user fred then executes `REVOKE UPDATE on Employee from sue`, the second grant is revoked, but the first grant is still in effect. Sue still has the ability to update the Street column of table Employee.

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.

Revoking DBA authority

Since the engine does not generally allow you to revoke DBA authority from a user while that user is connected to the database, the easiest way to revoke DBA authority is simply to wait until the user has disconnected, and then issue a REVOKE DBA statement.

However, it may be necessary to immediately revoke DBA authority from a user who is currently connected to the database, before the user has a chance to do anything else. Assume for this example you are trying to revoke DBA authority from user fred.

❖ To revoke DBA authority from a connected user

1. Connect to the same database as a *different* user with DBA authority.

For example, use a user ID other than fred.

2. Disable connections to the server by executing the following statement:

```
CALL sa_server_option('disable_connections', 'ON')
```

This prevents fred from connecting again once his existing connections have been dropped.

3. List all the connections to the database by executing the following statement:

```
CALL sa_conn_info( )
```

4. Write down the value of the Number column for each row containing fred in the Userid column.

5. For each connection number you wrote down in step 4, execute the following statement:

```
DROP CONNECTION number
```

This immediately drops each connection, rolling back any uncommitted transactions. Note that any transactions committed by fred, as well as any DDLs executed by fred before the DROP statement was executed, are not rolled back and must be manually undone.

6. Execute the following SQL statement:

```
REVOKE DBA FROM fred
```

7. Re-enable connections to the server by executing the following statement:

```
CALL sa_server_option('disable_connections', 'OFF')
```

The TCB subset

Following are the software modules and files that comprise the TCB (trusted computing base) included in the certified configuration. (Note that all .exe and .dll files are located in the *win32* subdirectory of your Adaptive Server Anywhere directory.)

1. Database engine / server

- ◆ *dbeng9.exe*
- ◆ *dbsrv9.exe*
- ◆ *dbserv9.dll*
- ◆ *dbctr9.dll*
- ◆ *libsybbr.dll*
- ◆ *dblgen9.dll*
- ◆ *dbcis9.dll*
- ◆ *dbjava9.dll*
- ◆ *.sql in the *scripts* directory
- ◆ *.zip in the *java* directory

2. Interactive SQL

- ◆ *dbisqlc.exe*
- ◆ *dbcon9.dll*
- ◆ *dblgen9.dll*
- ◆ *dbtool9.dll*
- ◆ *dblib9.dll*

3. Database utilities

- ◆ *dbackup.exe*
- ◆ *dbcollat.exe*
- ◆ *dbdsn.exe*
- ◆ *dberase.exe*
- ◆ *dbexpand.exe*
- ◆ *dbinfo.exe*
- ◆ *dbinit.exe*
- ◆ *dblog.exe*
- ◆ *dbping.exe*
- ◆ *dbshrink.exe*

-
- ◆ *dbstop.exe*
 - ◆ *dbsvc.exe*
 - ◆ *dbtran.exe*
 - ◆ *dbunload.exe*
 - ◆ *dbupgrad.exe*
 - ◆ *dbvalid.exe*
 - ◆ *dbwrite.exe*
 - ◆ *sqlpp.exe*
 - ◆ *dblgen9.dll*
 - ◆ *dbtool9.dll*
 - ◆ *dblib9.dll*

CHAPTER 5

Restricted Syntax

About this chapter

This chapter lists the syntax for the engine and server, as well as several database utilities used in the certified configuration.

Contents

| Topic: | page |
|--|--------------------|
| Restricted syntax | 62 |
| Database engine/server | 63 |
| Initialization utility | 67 |
| Service creation utility | 68 |
| Transaction log utility | 69 |
| Interactive SQL utility | 70 |

Restricted syntax

This section lists the syntax for the engine and server, as well as several database utilities used in the certified configuration. These tools are documented in “Database Administration Utilities” [*ASA Database Administration Guide*, page 455], but appear here for convenience, and also to emphasize the required or restricted switches in the C2 certified configuration. Note that where optional switches are listed, *only* those switches listed may be used. Any switches that may be documented or listed in the usage screen of the utility but are not listed here are not allowed in the certified configuration.

Please consult the Adaptive Server Anywhere Reference manual for more complete descriptions of each switch.

Database engine/server

Syntax 1 **dbeng9 -sc -gd dba -gk dba -gl dba -gu dba -x namedpipes(TDS=NO)**
 [*optional-engine-or-server-switches*]
 [*db-file* [*optional-database-switches*]] ...

Syntax 2 **dbsrv9 -sc -gd dba -gk dba -gl dba -gu dba -x namedpipes(TDS=NO)**
 [*optional-engine-or-server-switches*]
 [*db-file* [*optional-database-switches*]] ...

Required switches:

| Switch | Description | Reason |
|------------------------|--|--|
| -sc | Set up C2 Certified communication links. | Disallows shared memory connections. |
| -gd dba | Set starting database permission to DBA. | Non-DBA users could start their own database, connect as DBA, and then execute the UNLOAD or DROP DATABASE statements, or stop the engine or server. |
| -gk dba | Set stopping database engine or server permission to DBA. | Non-DBA users could stop the database engine or server, causing denial-of-service. |
| -gl dba | Set LOAD/UNLOAD permission to DBA. | A non-DBA user could use the UNLOAD command to write to the file system with the permissions of the sybase user. |
| -gu dba | Set utility commands permission to DBA. | Non-DBA users could use the DROP DATABASE statement to delete database files owned by the sybase user. |
| -x named-pipes(TDS=NO) | Starts the named pipes port and disallows TDS connections. | The named pipes port is the only communications mechanism supported in the certified configuration; the TDS protocol is not included in the certified configuration. |

Optional engine or server switches:

| Switch | Description | Restrictions |
|-------------------------|---|--|
| <code>-a logfile</code> | Apply named transaction log file. | Used only in recovery. |
| <code>-b</code> | Run in bulk operations mode. | |
| <code>-c size</code> | Make initial cache a maximum of <i>size</i> bytes. | |
| <code>-ca 0</code> | Disable automatic cache growth to compensate for memory allocation. | |
| <code>-ch size</code> | Set maximum cache size of <i>size</i> bytes. | |
| <code>-cl size</code> | Set minimum cache size of <i>size</i> bytes. | |
| <code>-cs</code> | Display cache sizing statistics. | |
| <code>-ct</code> | Perform client-engine or server character translation. | |
| <code>-d</code> | Disable asynchronous I/O. | |
| <code>-e</code> | Encrypt communications messages. | |
| <code>-f</code> | Force database to start without transaction log. | Used only in recovery. Note that auditing is unavailable if the engine or server is started with this switch. |
| <code>-ga</code> | Automatically shutdown after last database closed. | |
| <code>-gc num</code> | Set checkpoint timeout period to <i>num</i> minutes. | |
| <code>-ge size</code> | Set external DLL thread stack size. | |
| <code>-gf</code> | Disable firing of triggers. | |
| <code>-gm num</code> | Allow maximum <i>num</i> connections, if possible. | |
| <code>-gn num</code> | Use <i>num</i> engine or server threads. | |

| Switch | Description | Restrictions |
|-----------------------|---|---|
| <code>-gp size</code> | Set maximum page size of <i>size</i> bytes. | Note that this also truncates the audit log after checkpoint. |
| <code>-gr num</code> | Set maximum recovery time to <i>num</i> minutes. | |
| <code>-gt num</code> | Allow <i>num</i> OS threads to run concurrently. | |
| <code>-gw num</code> | Background process every <i>num</i> milliseconds. Default 500 milliseconds. | |
| <code>-gx num</code> | Use <i>num</i> OS threads. | |
| <code>-m</code> | Truncate transaction log after checkpoint. | |
| <code>-n name</code> | Name the database engine or server. | |
| <code>-o file</code> | Filename for copy of message window. | |
| <code>-os size</code> | Maximum size for the file specified by <code>-o</code> . | |
| <code>-p size</code> | Set maximum communication packet size. | |
| <code>-q</code> | Quiet mode—suppress output. | |
| <code>-r</code> | Read-only mode—database modifications not allowed. | |
| <code>-ti min</code> | Client idle time before disconnect. Default 240 minutes. | |
| <code>-tl sec</code> | Client liveness timeout in seconds. | Has no effect in certified configuration. |
| <code>-tq time</code> | Set quitting time. | |
| <code>-u</code> | Use buffered disk I/O. | |
| <code>-v</code> | Display product version information. | |
| <code>-z</code> | Display debugging information. | |

| Switch | Description | Restrictions |
|------------------|--|--------------|
| <i>-zo file</i> | Redirect request logging information to file. | |
| <i>-zr level</i> | Set request logging level. Level may be ALL, SQL, or NONE. | |
| <i>-zs size</i> | Maximum size for file specified by <i>-zo</i> . | |

db-file is a fully-qualified database file or write file name. All files must reside in your C2 database folder.

Initialization utility

Syntax

dbinit -i [*optional-switches*] *c2-database-folder\ filename*

Required switches:

| Switch | Description | Reason |
|--------|---------------------------------|---|
| -i | Do not install jConnect support | jConnect uses TCP/IP to communicate, which is not supported in the certified configuration. |

Optional switches:

| Switch | Description | Restrictions |
|----------------|--|---|
| -b | Blank padding of strings for comparisons | Full path must be specified; file must reside in your C2 database folder. |
| -c | Case sensitivity on all string comparisons | |
| -e | Encrypt database | |
| -m <i>name</i> | Set transaction log mirror name | |
| -o <i>file</i> | Log output messages to file | |
| -p <i>size</i> | Set page size | |
| -q | Quiet: do not print messages | |
| -t <i>name</i> | Transaction log file name | |
| -z <i>cs</i> | Specify collation sequence | Full path must be specified; file must reside in your C2 database folder. |

Service creation utility

| | |
|----------|---|
| Syntax 1 | dbsvc [<i>optional-switches</i>] -d <i>svc name</i> |
| Syntax 2 | dbsvc [<i>optional-switches</i>] -a sybase [<i>creation-switches</i>] -w <i>svc-name</i> <i>Details</i> |
| Syntax 3 | dbsvc [-q] -d <i>svc name</i> |
| Syntax 4 | dbsvc [-q] -l |

Required switches:

| Switch | Description | Reason |
|------------------|---------------------|---|
| -a sybase | Account name to use | The Adaptive Server Anywhere service must run as the sybase user. |

Optional switches:

| Switch | Description | Restrictions |
|-----------|--|--------------|
| -q | Do not print banner | |
| -y | Delete or overwrite service without confirmation | |

Creation switches:

| Switch | Description | Restrictions |
|--------------------------|--|--------------|
| -p <i>passwd</i> | Specify the password for the sybase account. | |
| -s <i>startup</i> | Startup option. Startup must be Automatic, Manual, or Disabled. Default is Manual. | |
| -t <i>type</i> | Type of service. Type must be Network or Standalone. Default is Standalone. | |

Notes For syntax 2, *Details* must contain the full path to the Adaptive Server Anywhere engine or server executable, as well as the parameters for that engine or server.

☞ For more information about the engine and server parameters, see [“Database engine/server” on page 63](#).

Transaction log utility

Syntax

dblog [*optional-switches*] *c2-database-folder\ database-file*

Optional switches:

| Switch | Description | Restrictions |
|----------------|---|---|
| -g <i>n</i> | Set LTM generation number. | Full path must be specified; file must reside in your C2 database folder. |
| -il | Ignore LTM truncation point. | |
| -ir | Ignore SQL Remote truncation point. | |
| -m <i>name</i> | Set transaction log mirror name. | |
| -o <i>file</i> | Log output messages to file. | |
| -q | Quiet: do not print messages. | |
| -r | Do not use a transaction log mirror. | |
| -t <i>name</i> | Set transaction log name. | Full path must be specified; file must reside in your C2 database folder. |
| -x <i>n</i> | Zap transaction log current relative offset to <i>n</i> . | |
| -z <i>n</i> | Zap transaction log starting offset to <i>n</i> . | |

Interactive SQL utility

Syntax 1 **dbisqlc** [*optional-switches*] *SQL-command*

Syntax 2 **dbisqlc** [**optional switches**] *filename*

Optional switches:

| Switch | Description | Restrictions |
|---------------------------|--|--|
| <code>-c conn_str</code> | Use connection string <i>conn_str</i> . | <i>conn_str</i> must contain “INT=YES;LINKS=namedpipes” and must not contain “UID=” or “PWD=” |
| <code>-d delimiter</code> | Specify command delimiter. | |
| <code>-q</code> | Silent mode, no window. | |
| <code>-x</code> | Syntax check only, no commands executed. | |

CHAPTER 6

Integrated Logins

About this chapter

This chapter describes how to use the integrated login in a manner equivalent to the C2 certified configuration.

Contents

Topic:

page

[Using integrated login](#)

72

Using integrated login

Adaptive Server Anywhere uses the integrated login mechanism to map a Windows NT user to an Adaptive Server Anywhere user. When a Windows NT user attempts to connect to the database, the operating system provides assurance that the user has been authenticated (usually using a password). If the database server contains a mapping between that Windows NT user and a valid Adaptive Server Anywhere user, that user can connect.

For use in the C2 certified configuration, Adaptive Server Anywhere requires the use of integrated login exclusively. An integrated login must be created for the DBA account in the database, and it is recommended that the sybase Windows NT user be used for this purpose. As well, integrated login mappings must be one-to-one. That is, two Windows NT user accounts may not be mapped to the same Adaptive Server Anywhere account.

☞ For instructions on how to create an integrated login for the sybase user, see [“Creating a database” on page 32](#).

☞ For more information on integrated login, see “Connecting to a Database” [*ASA Database Administration Guide*, page 37].

CHAPTER 7

Connecting to the Adaptive Server Anywhere Service

About this chapter

This chapter describes how to connect to the Adaptive Server Anywhere Service in a manner equivalent to the C2-certified configuration.

Contents

| Topic: | page |
|--|--------------------|
| Connecting to the Adaptive Server Anywhere service | 74 |

Connecting to the Adaptive Server Anywhere service

Once the Adaptive Server Anywhere service has been started, users can use dbisqlc to connect to the engine and execute SQL statements. There are two ways to tell dbisqlc how to connect:

1. You can use the `-c` switch and specify a connection string containing a list of parameters that tell dbisqlc which engine and database to connect to, and how to find it. For example, if your engine is named `asademo`, you can connect to it using:

```
dbisqlc -c "ENG=asademo;LINKS=namedpipes;INT=YES
```

`LINKS=namedpipes` tells dbisqlc to use named pipes to connect to the engine, and `INT=YES` tells dbisqlc to use the integrated login facility.

2. You can simply start dbisqlc, and fill in the fields on the connection dialog. Note that you must select the Use integrated login option on the Login tab, you must enter an engine name on the Database tab, and you must check the Named pipes checkbox on the Network tab.

CHAPTER 8

The Adaptive Server Anywhere C2 Patch

About this chapter

This chapter describes the C2 patch to the Adaptive Server Anywhere 7.0.0 release. This chapter does not apply when running the current software in a manner equivalent to the C2-certified environment.

Contents

| Topic: | page |
|---|--------------------|
| The Adaptive Server Anywhere C2 patch | 76 |

The Adaptive Server Anywhere C2 patch

The Adaptive Server Anywhere C2 Patch contains two DLLs, three batch files, and one text file. This section describes each file in the patch.

This section describes the C2 patch to the Adaptive Server Anywhere 7.0.0 release. It does not apply when running the current software in a manner equivalent to the C2-certified environment.

| File | Description |
|--------------------|--|
| <i>dbngen7.dll</i> | Contains English language strings used by the Adaptive Server Anywhere engine and tools. This file contains the auditing string used by dbbackup to audit the use of the -xo switch. |
| <i>dbtool7.dll</i> | Used by all of the database utilities, as well as <i>dbisqlc.exe</i> . This file contains a fix to dbbackup to audit the truncation of the transaction log. |
| <i>mdac1.bat</i> | This batch file creates a temporary directory and copies fourteen .dll files and one .exe from the C:\winnt\system32 directory into the temporary directory. The Adaptive Server Anywhere installation will replace these files, and they must be copied before Adaptive Server Anywhere installation so they can be restored after. |
| <i>mdac2.bat</i> | This batch file copies the files from the temporary directory created by <i>mdac1.bat</i> into the C:\winnt\system32 directory, overwriting those installed by Adaptive Server Anywhere. One of the files is in use by the operating system, so it is renamed before the file copy. |
| <i>mdac3.bat</i> | This batch file deletes the file that was renamed by <i>mdac2.bat</i> , as well as the temporary directory created by <i>mdac1.bat</i> . |
| <i>readme.txt</i> | This file contains instructions for installing the .dll files included in the patch. |

CHAPTER 9

More Information

About this chapter

This chapter contains a list of additional sources of information that may be helpful while operating Adaptive Server Anywhere in the C2 certified configuration.

Contents

| Topic: | page |
|--|--------------------|
| Where to look for more information | 78 |

Where to look for more information

| Subject | Source |
|---|---|
| Auditing | “Keeping Your Data Secure” on page 3 |
| Connection parameters | For a list, run <code>dbdsn -cl</code> or see “Client/Server Communications” [ASA <i>Database Administration Guide</i> , page 93]. |
| Database options | “Database Options” [ASA <i>Database Administration Guide</i> , page 555] |
| dbinit, dblog, dbtran, dbisqlc, dbbackup and other administrative utilities | “Database Administration Utilities” [ASA <i>Database Administration Guide</i> , page 455] |
| dbsvc utility | “The Service Creation utility” [ASA <i>Database Administration Guide</i> , page 519] |
| Engine / server switches | “The Database Server” [ASA <i>Database Administration Guide</i> , page 123] |
| Integrated login | “Connecting to a Database” [ASA <i>Database Administration Guide</i> , page 37] |
| Java in the Database | “Introduction to Java in the Database” [ASA <i>Programming Guide</i> , page 51] and “Using Java in the Database” [ASA <i>Programming Guide</i> , page 81] |
| Procedures, Functions, Triggers | “Using Procedures, Triggers, and Batches” [ASA <i>SQL User’s Guide</i> , page 609] |
| Security tips | “Keeping Your Data Secure” on page 3 |
| Tables, Views | “Working with Database Objects” [ASA <i>SQL User’s Guide</i> , page 25] |
| The GRANT and REVOKE SQL statements | “SQL Statements” [ASA <i>SQL Reference</i> , page 213] |
| The transaction log file | “Backup and Data Recovery” [ASA <i>Database Administration Guide</i> , page 337] |
| User IDs and permissions | “Managing User IDs and Permissions” [ASA <i>Database Administration Guide</i> , page 389] |

Index

A

| | |
|----------------------------------|------|
| access | |
| security features | 4 |
| Adaptive Server Anywhere | |
| C2 Patch | 76 |
| C2 software installation | 28 |
| administration | |
| audit records | 47 |
| AES encryption algorithm | |
| about | 16 |
| audit records | 40 |
| auditing | |
| about | 9 |
| C2 requirements | 37 |
| comments | 11 |
| correlating records | 49 |
| enabling/disabling | 38 |
| example | 11 |
| log translation [dbtran] utility | 12 |
| option | 38 |
| reading output | 39 |
| retrieving audit information | 10 |
| security features | 4, 9 |
| transaction log [dblog] utility | 12 |
| turning on | 9 |
| utilities | 48 |
| write file [dbwrite] utility | 12 |

B

| | |
|----------------|----|
| backup utility | |
| C2 security | 47 |

C

| | |
|-----------------------------------|----|
| C2 database folder | |
| C2 security | 29 |
| C2 installation | |
| Adaptive Server Anywhere software | 28 |
| hardware | 26 |
| operating system | 27 |
| C2 security | |
| about | v |

| | |
|------------------------------|------|
| creating-compliant databases | 32 |
| disclaimer | v |
| documentation | v |
| guidelines | 4 |
| more information | 78 |
| running compliant engines | 34 |
| cache size | |
| encrypted database issues | 19 |
| comments | 11 |
| connecting | |
| C2 security | 74 |
| integrated logins | 6 |
| conventions | |
| documentation | viii |
| creating | |
| C2 compliant databases | 32 |
| creating databases | |
| security | 14 |

D

| | |
|----------------------|--------|
| database access | |
| controlling | 6 |
| database files | |
| encrypting | 16 |
| security | 16, 20 |
| database servers | |
| C2 restricted syntax | 63 |
| security | 14 |
| DBA authority | |
| security tips | 20 |
| dbbackup utility | |
| C2 security | 47 |
| dbeng9 | |
| C2 restricted syntax | 63 |
| dbinit utility | |
| C2 restricted syntax | 67 |
| dbisqlc utility | |
| C2 restricted syntax | 70 |
| dblog utility | |
| auditing | 12 |
| C2 restricted syntax | 69 |
| C2 security | 47 |

| | | | |
|------------------------------------|--------|-----------------------------|--------|
| dbsrv9 | | C2 restricted syntax | 70 |
| C2 restricted syntax | 63 | | |
| dbsvc utility | | K | |
| C2 restricted syntax | 68 | keeping your data secure | 3 |
| dbtran utility | | L | |
| auditing | 10, 12 | LOAD TABLE statement | |
| C2 security | 47 | security | 14 |
| dbwrite utility | | loading data | |
| auditing | 12 | security | 14 |
| C2 security | 48 | log translation utility | |
| deleting databases | | auditing | 10, 12 |
| security | 14 | C2 security | 47 |
| disabling auditing | 38 | | |
| documentation | | M | |
| conventions | viii | mappings | |
| SQL Anywhere Studio | vi | integrated logins | 72 |
| | | MDSR encryption algorithm | |
| E | | about | 16 |
| enabling auditing | 38 | N | |
| encryption | | negative permissions | 7 |
| AES algorithm | 16 | network server | |
| communications | 15 | encryption | 15 |
| database files | 16 | newsgroups | |
| MDSR algorithm | 16 | technical support | xii |
| passwords | 7 | O | |
| performance of encrypted databases | 19 | options | |
| Rijndael algorithm | 16 | auditing | 38 |
| simple | 16 | output | |
| strong | 16 | auditing | 39 |
| encryption algorithms | | P | |
| AES | 16 | passwords | |
| MDSR | 16 | length | 20 |
| Rijndael | 16 | security features | 7 |
| | | security tips | 20 |
| F | | patch | |
| feedback | | Adaptive Server Anywhere C2 | 76 |
| documentation | xii | performance | |
| providing | xii | encrypted databases | 19 |
| | | permissions | |
| I | | about | 7 |
| icons | | negative | 7 |
| used in manuals | x | | |
| integrated logins | | | |
| C2 security | 72 | | |
| security features | 6 | | |
| Interactive SQL utility | | | |

| | | | |
|-------------------------------|--------|---|--------|
| scheme | 6 | stored procedures | |
| security features | 6 | security features | 4 |
| R | | strong encryption | |
| reading | | AES algorithm | 16 |
| auditing output | 39 | database files | 16 |
| records | | Rijndael | 16 |
| administration | 47 | using MDSR algorithm | 16 |
| audit | 40 | subset | |
| correlating | 49 | TCB | 59 |
| restrictions | | support | |
| security | 52 | newsgroups | xii |
| Rijndael encryption algorithm | | T | |
| about | 16 | TCB subset | 59 |
| running | | technical support | |
| C2 compliant database engines | 34 | newsgroups | xii |
| S | | transaction log utility | |
| security | | auditing | 12 |
| about | v | C2 security | 47 |
| AES encryption | 16 | troubleshooting | |
| auditing | 9, 10 | encrypted database performance | 19 |
| C2 guidelines | 4 | Trusted Computing Base | 59 |
| creating databases | 14 | U | |
| database server | 14, 20 | UNLOAD statement | |
| deleting databases | 14 | security | 14 |
| encrypting database files | 16 | UNLOAD TABLE statement | |
| encryption | 15 | security | 14 |
| integrated logins | 6 | unloading | |
| loading data | 14 | data | 14 |
| MDSR encryption | 16 | unloading data | |
| overview | 4 | security | 14 |
| passwords | 7 | user IDs | |
| restrictions | 52 | security features | 4 |
| Rijndael encryption | 16 | security tip | 20 |
| server command line | 4 | users | |
| system functions | 20 | C2 security | 72 |
| tips | 20 | utilities | |
| unloading data | 14 | auditing | 48 |
| warnings | 55 | backup [dbbackup] in C2 security | 47 |
| service creation utility | | initialization [dbinit] | 67 |
| C2 restricted syntax | 68 | Interactive SQL [dbisql] | 70 |
| services | | log translation [dbtran] auditing | 10, 12 |
| connecting to | 74 | log translation [dbtran] in C2 security | 47 |
| simple encryption | 16 | service creation [dbsvc] | 68 |
| SQL Anywhere Studio | | transaction log [dblog] | 69 |
| documentation | vi | | |

| | |
|--|----|
| transaction log [dblog] auditing | 12 |
| transaction log [dblog] in C2 security | 47 |
| write file [dbwrite] auditing | 12 |
| write file [dbwrite] in C2 security | 48 |

V

| | |
|-------------------|---|
| views | |
| security features | 4 |

W

| | |
|--------------------|----|
| warnings | |
| security | 55 |
| write file utility | |
| auditing | 12 |
| C2 security | 48 |

X

| | |
|-------------------------------|----|
| xp_cmdshell system procedure | |
| security features | 20 |
| xp_sendmail system procedure | |
| security features | 20 |
| xp_startmail system procedure | |
| security features | 20 |
| xp_startsmtp system procedure | |
| security features | 20 |
| xp_stopmail system procedure | |
| security features | 20 |
| xp_stopsmtmp system procedure | |
| security features | 20 |