Adaptive Server® Anywhere C2 Security Supplement

Last modified: October 2002
Part Number: 38177-01-0802-01

Last modified October 2002. Part number 38177-01-0802-01.

Contents

About This Manual

Subject

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

This book does not include all information on security-related features. In particular, no security-related features introduced since version 7.0 are described in this book.

Current software is not C2 certified
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 only for users of Adaptive Server Anywhere who wish to run the software in a manner equivalent to the C2-certified configuration.

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 set

The SQL Anywhere Studio documentation set 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.
- **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.

- **Adaptive Server Anywhere Error Messages** This book provides a complete listing of Adaptive Server Anywhere error messages together with diagnostic information.
- **Adaptive Server Anywhere C2 Security Supplement** 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. The book does *not* include the security features added to the product since certification.
- **MobiLink Synchronization User's Guide** This book describes all aspects of 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.
- **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.
- **UltraLite User's Guide** This book describes how to build database applications for small devices such as handheld organizers using the UltraLite deployment technology for Adaptive Server Anywhere databases.
- **UltraLite User's Guide for PenRight! MobileBuilder** This book is for users of the PenRight! MobileBuilder development tool. It describes how to use UltraLite technology in the MobileBuilder programming environment.
- **SQL Anywhere Studio Help** This book is provided online only. It includes the context-sensitive help for Sybase Central, Interactive SQL, and other graphical tools.

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

Documentation formats

SQL Anywhere Studio provides documentation in the following formats:

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

  To access the online books on Windows operating systems, choose Start➤Programs➤Sybase SQL Anywhere 8➤Online Books. You can navigate the online books using the HTML Help table of contents, index, and search facility in the left pane, and using the links and menus in the right pane.

  To access the online books on UNIX operating systems, run the following command at a command prompt:

  ```
  dbbooks
  ```

- **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 following books are included in the SQL Anywhere Studio box:

  - *Introducing SQL Anywhere Studio*.
  - *Adaptive Server Anywhere Getting Started*.
  - *SQL Anywhere Studio Quick Reference*. This book is available only in printed form.

  The complete set of books is available as the SQL Anywhere Documentation set from Sybase sales or from e-Shop, the Sybase online store, at http://e-shop.sybase.com/cgi-bin/eshop.storefront/.

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 are shown 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. 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.

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

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

- **One or more options** If you choose more than one, separate your choices with commas.

  { **CONNECT**, **DBA**, **RESOURCE** }

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 ONE

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.

CHAPTER 1

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

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 (emergency 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 54.
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 44. For information about the database folder, see "Adaptive Server Anywhere software installation" on page 6.
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 47 and "Restrictions" on page 28.
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 8\win32\dbeng8.exe
     ```

     for the personal database server, or

     ```
     C:\Program Files\Sybase\SQL Anywhere 8\win32\dbsrv8.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 8\win32\
dbsrv8.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 8\win32\dbsrv8.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 41.

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 2

Auditing

About this chapter

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

Contents

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 18.

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 25.

Type	Information	Use
Attempting Operation	date/time, SQL of attempted operation	This record displays the operation being attempted. This is necessary because of the way the transaction log works. 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. 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. For this reason, all DDL statements (and a few other statements as well) are recorded before they are attempted.
Operation Succeeded / Failed	date/time, success or failure	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.

Type	Information	Use
Checking permission	date/time, type of permission / authority, table name (if applicable), column name (if applicable), procedure / function name (if applicable)	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: DBA / Resource authority Insert / Update / Select / Delete / Alter / Resource permission on a table Update / Select / Resource permission on a table and column Grant Insert / Update / Select / Delete / Alter / Resource permission on a table Grant Update / Select / Resource permission on a table and column Execute permission on a procedure or function Grant Execute permission on a procedure or function
Checking user	date/time, user name	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.
Set Public Option	date/time, name of option	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.

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.
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.

Type	Information	Use
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: ♦ 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	♦ 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	♦ 1 byte 1 (success) or 0 (failure) ♦ 2 bytes length of following string (n) ♦ n bytes user name
AUDIT_CONNECTION	♦ 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	♦ 2 bytes length of following string (n) ♦ n bytes user name
AUDIT_TRIGGER	♦ 2 bytes length of following string (n) ♦ n bytes name of trigger ♦ 3 bytes fired or finished
AUDIT_STRING	♦ 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 46.

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 5.

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 3

Restrictions and Other Security Concerns

About this chapter

This chapter describes C2 certification restrictions and other security concerns.

Contents

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 45 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" on page 522 of the book *ASA Database Administration Guide*.

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 6.

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 10.

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 5.

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.)

The C2-certified software was version 7, so file names include the number 7. To run the current software in a manner equivalent to the C2-certified configuration, use the version 8 equivalents, such as *dbeng8.exe*.

1 Database engine / server

- *dbeng7.exe*
- *dbsrv7.exe*
- *dbserv7.dll*
- *dbctrs7.dll*
- *libsybbr.dll*
- *dblgen7.dll*
- *dbcis7.dll*
- *dbjava7.dll*
- *.*sql* in the *scripts* directory
- *.*zip* in the *java* directory

2 Interactive SQL

- *dbisqlc.exe*
- *dbcon7.dll*
- *dblgen7.dll*
- *dbtool7.dll*
- *dblib7.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*
- *dblgen7.dll*
- *dbtool7.dll*
- *dblib7.dll*

PART TWO

Appendixes

The appendixes to follow contain additional information you may find useful when operating Adaptive Server Anywhere in a manner equivalent to the C2-certified configuration.

APPENDIX A

Restricted Syntax

About this appendix

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

Contents

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" on page 435 of the book *ASA Database Administration Guide*, 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

dbeng8 -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

dbsrv8 -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 namedpipes(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
–a *logfile*	Apply named transaction log file.	Used only in recovery.
–b	Run in bulk operations mode.	
–c *size*	Make initial cache a maximum of *size* bytes.	
–ca 0	Disable automatic cache growth to compensate for memory allocation.	
–ch *size*	Set maximum cache size of *size* bytes.	
–cl *size*	Set minimum cache size of *size* bytes.	
–cs	Display cache sizing statistics.	
–ct	Perform client-engine or server character translation.	
–d	Disable asynchronous I/O.	
–e	Encrypt communications messages.	
–f	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.
–ga	Automatically shutdown after last database closed.	
–gc *num*	Set checkpoint timeout period to *num* minutes.	
–ge *size*	Set external DLL thread stack size.	
–gf	Disable firing of triggers.	
–gm *num*	Allow maximum *num* connections, if possible.	
–gn *num*	Use *num* engine or server threads.	
–gp *size*	Set maximum page size of *size* bytes.	
–gr *num*	Set maximum recovery time to *num* minutes.	
–gt *num*	Allow *num* OS threads to run concurrently.	
–gw *num*	Background process every *num* milliseconds. Default 500 milliseconds.	

Switch	Description	Restrictions
–gx *num*	Use *num* OS threads.	
–m	Truncate transaction log after checkpoint.	Note that this also truncates the audit log after checkpoint.
–n *name*	Name the database engine or server.	
–o *file*	Filename for copy of message window.	
–os *size*	Maximum size for the file specified by `-o`.	
–p *size*	Set maximum communication packet size.	
–q	Quiet mode—suppress output.	
–r	Read-only mode—database modifications not allowed.	
–ti *min*	Client idle time before disconnect. Default 240 minutes.	
–tl *sec*	Client liveness timeout in seconds.	Has no effect in certified configuration.
–tq *time*	Set quitting time.	
–u	Use buffered disk I/O.	
–v	Display product version information.	
–z	Display debugging information.	
–zo *file*	Redirect request logging information to file.	
–zr *level*	Set request logging level. *Level* may be ALL, SQL, or NONE.	
–zs *size*	Maximum size for file specified by `-zo`.	

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	
–c	Case sensitivity on all string comparisons	
–e	Encrypt database	
–m *name*	Set transaction log mirror name	Full path must be specified; file must reside in your C2 database folder.
–o *file*	Log output messages to file	
–p *size*	Set page size	
–q	Quiet: do not print messages	
–t *name*	Transaction log file name	Full path must be specified; file must reside in your C2 database folder.
–z *cs*	Specify collation sequence	

Service creation utility

Syntax 1 **dbsvc** [*optional-switches*] **-d** *svc name*

Syntax 2 **dbsvc** [*optional-switches*] **-a sybase** [*creation-switches*] **-w** *svc-name Details*

Syntax 3 **dbsvc** [**-q**] **-d** *svc name*

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 *passwd*	Specify the password for the *sybase* account.	
–s *startup*	Startup option. *Startup* must be Automatic, Manual, or Disabled. Default is Manual.	
–t *type*	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 41.

Transaction log utility

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

Optional switches:

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

Interactive SQL utility

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

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

Optional switches:

Switch	Description	Restrictions
–c *conn_str*	Use connection string *conn_str*.	*conn_str* must contain "INT=YES;LINKS=namedpipes" and must not contain "UID=" or "PWD="
–d *delimiter*	Specify command delimiter.	
–q	Silent mode, no window.	
–x	Syntax check only, no commands executed.	

APPENDIX B

Integrated Logins

About this appendix

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

Contents

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 10.

For more information on integrated login, see "Connecting to a Database" on page 37 of the book *ASA Database Administration Guide*.

APPENDIX C

Connecting to the Adaptive Server Anywhere Service

About this appendix

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

Contents

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.

APPENDIX D

The Adaptive Server Anywhere C2 Patch

About this appendix

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

Contents

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-certifierd environment.

File	Description
dblgen7.dll	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.
dbtool7.dll	Used by all of the database utilities, as well as *dbisqlc.exe*. This file contains a fix to dbbackup to audit the truncation of the transaction log.
mdac1.bat	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.
mdac2.bat	This batch file copies the files from the temporary directory created by *mdac1.bat* 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.
mdac3.bat	This batch file deletes the file that was renamed by *mdac2.bat*, as well as the temporary directory created by *mdac1.bat*.
readme.txt	This file contains instructions for installing the *.dll* files included in the patch.

APPENDIX E

More Information

About this appendix

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

Contents

Where to look for more information

Subject	Source
Auditing	"Keeping Your Data Secure" on page 387 of the book *ASA Database Administration Guide*
Connection parameters	For a list, run `dbdsn -cl` or see "Client/Server Communications" on page 91 of the book *ASA Database Administration Guide*.
Database options	"Database Options" on page 535 of the book *ASA Database Administration Guide*
dbinit, dblog, dbtran, dbisqlc, dbbackup and other administrative utilities	"Database Administration Utilities" on page 435 of the book *ASA Database Administration Guide*
dbsvc utility	"The Service Creation utility" on page 499 of the book *ASA Database Administration Guide*
Engine / server switches	"The Database Server" on page 119 of the book *ASA Database Administration Guide*
Integrated login	"Connecting to a Database" on page 37 of the book *ASA Database Administration Guide*
Java in the Database	"Introduction to Java in the Database" on page 49 of the book *ASA Programming Guide* and "Using Java in the Database" on page 85 of the book *ASA Programming Guide*
Procedures, Functions, Triggers	"Using Procedures, Triggers, and Batches" on page 507 of the book *ASA SQL User's Guide*
Security tips	"Keeping Your Data Secure" on page 387 of the book *ASA Database Administration Guide*
Tables, Views	"Working with Database Objects" on page 27 of the book *ASA SQL User's Guide*
The GRANT and REVOKE SQL statements	"SQL Statements" on page 199 of the book *ASA SQL Reference Manual*
The transaction log file	"Backup and Data Recovery" on page 299 of the book *ASA Database Administration Guide*
User IDs and permissions	"Managing User IDs and Permissions" on page 351 of the book *ASA Database Administration Guide*

Index

E

F

I

L

M

N

O

P

R

S

T

U

W