CHAPTER 27

Deploying Databases and Applications

About this chapter

This chapter describes how to deploy Adaptive Server Anywhere components. It identifies the files required for deploying client applications, and addresses related issues such as connection settings.

Check your license agreement
Redistribution of files is subject to your license agreement with Sybase. No statements in this document override anything in your license agreement. Please check your license agreement before considering deployment.

Contents

Deployment overview

The following deployment steps are examined in this chapter:

- Determining required files based on the choice of application platform and architecture.
- Configuring client applications.
- Understanding the file name convention to determine the function and platform of files.

Deployment models

The files you need to deploy depend on the deployment model you choose. Here are some possible deployment models:

- **Client deployment** You may deploy only the client portions of Adaptive Server Anywhere to your end-users, so that they can connect to a centrally located network database server.
- **Network server deployment** You may deploy network servers to offices, and then deploy clients to each of the users within those offices.
- **Embedded database deployment** You may deploy an application that runs with the personal database server. In this case, both client and personal server need to be installed on the end-user's machine.
- **SQL Remote deployment** Deploying a SQL Remote application is an extension of the embedded database deployment model.

Ways to distribute files

There are two ways to deploy Adaptive Server Anywhere:

- **Use the Adaptive Server Anywhere installation** You can make the Setup program available to your end-users. By selecting the proper option, each end-user is guaranteed of getting the files they need.

  This is the simplest solution for many deployment cases. In this case, you must still provide your end users with a method for connecting to the database server (such as an ODBC data source).

- **Develop your own installation** There may be reasons for you to develop your own installation program that includes Adaptive Server Anywhere files. This is a much more complicated option, and this chapter addresses the needs of those who are developing their own installation.

  If Adaptive Server Anywhere has already been installed for the server type and operating system required by the client application architecture, the required files can be found in the appropriately named subdirectory, located in the Adaptive Server Anywhere installation directory.

  For example, assuming the default installation directory was chosen, the *win32* subdirectory of your installation directory contains the files required to run the server for the Windows 95/Windows NT platform.

Of course, whichever option you choose, you must not violate the terms of your license agreement.

Where to install files

For a deployed application to work properly, the database server and client libraries must each be able to locate the files they need. The deployed files should be located relative to each other in the same fashion as your Adaptive Server Anywhere installation.

In practice, this means that on PC's, most files belong in a single directory.

For a full description of the places where the software looks for files, see "How Adaptive Server Anywhere locates files" on page 3 of the book *Adaptive Server Anywhere Reference Manual*.

UNIX deployment issues

UNIX deployments are different from PC deployments in some ways:

- **Directory structure** For UNIX installations, the directory structure is as follows:

Directory	Contents
/opt/SYBSasa6/bin	Executable files
/opt/SYBSasa6/lib	Shared objects and libraries
/opt/SYBSasa6/res	String files

  On AIX, the default root directory is */usr/lpp/SYBSasa6* instead of */opt/SYBSasa6*.

- **File extensions** In the tables in this chapter, the shared objects are listed with an extension *.so*. For HP-UX, the extension is *.sl*.

  On the AIX operating system, shared objects that applications need to link to are given the extension *.a*.

- **Symbolic links** Each shared object is installed as a symbolic link to a file of the same name with the additional extension *.1* (one). For example, the *libdblib6.so* is a symbolic link to the file *libdblib6.so.1* in the same directory.

  If patches are required to the Adaptive Server Anywhere installation, these will be supplied with extension *.2*, and the symbolic link must be redirected.

- **Threaded and unthreaded applications** Some shared objects are provided in two forms, one of which has the additional characters *_r* before the file extension. For example, in addition to *libdblib6.so*, there is a file named *libdblib6_r.so*. In this case, threaded applications must be linked to the *_r* shared object, while non-threaded applications must be linked to the shared object without the *_r* characters.

For a description of the places where the software looks for files, see "How Adaptive Server Anywhere locates files" on page 3 of the book *Adaptive Server Anywhere Reference Manual*.

Windows CE deployment issues

For Win CE 2.0, all DLLs must go in the *\Windows* directory. To facilitate finding the Adaptive Server Anywhere DLLs in this directory, the DLLs for Windows CE all have the prefix *ASA_*:

- *ASA_dblgen6.dll*
- *ASA_dblib6.dll*
- *ASA_dbodbc6.dll*
- *ASA_dbuodbc6.dll*
- *ASA_dbfile.dll*
- *ASA_dbsmtp.dll*
- *ASA_dbwtsp6.dll*

In Windows CE 2.1, the operating system searches the directory containing the executable that loads the DLL, so the SQL Remote DLLs (dbfile and so on) could go into the folder that will contain Adaptive Server Anywhere. However, *dblib6.dll* and *dbodbc6.dll*, since they are loaded from other applications, must still go in *\Windows*, as must the language DLL *dblgen6.dll*.

Deploying client applications

In order to deploy a client application that runs against a network database server, you must provide each end user with the following items:

- **Client application** The application software itself is independent of the database software, and so is not described here.
- **Database interface files** The client application requires the files for the database interface it uses (ODBC, JDBC, Embedded SQL, or Open Client).
- **Connection information** Each client application needs database connection information.

The interface files and connection information required varies with the interface your application is using. Each interface is described separately in the following sections.

Deploying ODBC clients

Each ODBC client machine must have the following:

- **A working ODBC installation** ODBC files and instructions for their redistribution are available for redistribution from Microsoft Corporation. They are not described in detail here.

  Microsoft provides their ODBC Driver Manager for 32-bit Windows and for Windows 3.x. Third party vendors such as Intersolv provide ODBC Driver managers for UNIX.

  ODBC applications can run without the driver manager, but except on platforms for which an ODBC driver manager is not available, this is generally not recommended.
- **The Adaptive Server Anywhere ODBC driver** This is the file *dbodbc6.dll* together with some additional files.
- **Connection information** The client application must have access to the information needed to connect to the server. This information is typically included in an ODBC data source.

ODBC driver required files

The following table shows the files needed for a working Adaptive Server Anywhere ODBC driver. These files should be copied into a single directory. The Adaptive Server Anywhere installation places them all in the operating-system subdirectory of your Adaptive Server Anywhere installation directory.

Description	32-bit Windows	Windows 3.x	UNIX
ODBC driver	*dbodbc6.dll*	*dbodbc6wdll*	*libdbodbc6.so*
ODBC translator	*dbodtr6.dll*	*dbodtr6w.dll*	
Language-specific string library	*dblgen6.dll*	*dblgen6w.dll*	*dblgen6.res*
Network ports	*dbport6.dll*	*N/A*	*libdbport6.so*
Connection Dialog	*dbcon6.dll*	*dbcon6w.dll*	

Notes

- Your end user must have a working ODBC installation, including the driver manager. Instructions for deploying ODBC are included in the Microsoft ODBC SDK.
- The network port library handles network communications. It is required only if the client is working with the network database server. No network port is available for Windows 3.x; instead, the network ports are part of the ODBC driver DLL.
- The Connection dialog is needed if your end users are to create their own data sources, if they need to enter user IDs and passwords when connecting to the database, or if they need to display the Connection dialog for any other purpose.
- The ODBC translator is required only if your application relies on OEM to ANSI character set conversion.
- On HP-UX, all files listed with extension *.so* instead have extension *.sl*. On AIX, the files have extension *.so* or *.a*.

Configuring the ODBC driver

In addition to copying the ODBC driver files onto disk, your Setup program must also make a set of registry entries to install the ODBC driver properly.

Windows NT and Windows 95

The Adaptive Server Anywhere Setup program makes changes to the Windows NT and Windows 95 system Registry to identify and configure the ODBC driver. If you are building a setup program for your end users, you should make the same settings.

You can use the Windows *regedit* utility to inspect registry entries.

The Adaptive Server Anywhere ODBC driver is identified to the system by a set of registry values in the following registry key:

```
HKEY_LOCAL_MACHINE\
    SOFTWARE\
        ODBC\
            ODBCINST.INI\
                Adaptive Server Anywhere 6.0
```

The values are as follows:

Value name	Value type	Value data
Driver	String	*path\dbodbc6.dll*
Setup	String	*path\dbodbc6.dll*

There is also a registry value in the following key:

```
HKEY_LOCAL_MACHINE\
    SOFTWARE\
        ODBC\
            ODBCINST.INI\
                ODBC Drivers
```

The value is as follows:

Value name	Value type	Value data
Adaptive Server Anywhere 6.0	String	Installed

Windows 3.x

For Windows 3.x, the ODBC driver information is held in the *odbcinst.ini* file, instead of the registry. The individual entries are analogous to the registry entries above.

Third party ODBC drivers

If you are using a third-party ODBC driver on an operating system other than Windows, consult the documentation for that driver on how to configure the ODBC driver.

Deploying connection information

ODBC client connection information is generally deployed as an ODBC data source. You can deploy an ODBC data source in one of the following ways:

- **Programatically** Add a data source description to your end-user's Registry or ODBC initialization files.
- **Manually** Provide your end-users with instructions, so that they can create an appropriate data source on their own machine.

You create a data source manually using the ODBC Administrator, from the User DSN tab or the System DSN tab. The Adaptive Server Anywhere ODBC driver displays the configuration dialog for entering settings. Data source settings include the location of the database file, the name of the database server, as well as any start up parameters and other options.

This section provides you with the information you need to know for either approach.

Types of data source

There are three kinds of data sources: User data sources, System data sources, and File data sources.

User data source definitions are stored in the part of the registry containing settings for the specific user currently logged on to the system. System data sources, however, are available to all users and to Windows NT or Windows 95 services, which run regardless of whether a user is logged onto the system or not. Given a correctly configured System data source named MyApp, any user can use that ODBC connection by providing DSN=MyApp in the ODBC connection string.

File data sources are not held in the registry, but are held in a special directory. A connection string must provide a FileDSN connection parameter to use a File data source.

Data source registry entries

Each user data source is identified to the system by registry entries.

You must enter a set of registry values in a particular registry key. For User data sources the key is as follows:

```
HKEY_CURRENT_USER\
   SOFTWARE\
      ODBC\
         ODBC.INI\
            userdatasourcename
```

For System data sources the key is as follows:

```
HKEY_LOCAL_MACHINE\
   SOFTWARE\
      ODBC\
         ODBC.INI\
            systemdatasourcename
```

The key contains a set of registry values, each of which corresponds to a connection parameter. For example, the ASA 6.0 Sample key corresponding to the ASA 6.0 Sample data source contains the following settings:

Value name	Value type	Value data
Autostop	String	Yes
DatabaseFile	String	*path*\asademo.db
Description	String	Adaptive Server Anywhere Sample Database
Driver	String	*path*\win32\dbodbc6.dll
PWD	String	sql
Start	String	*path*\win32\dbeng6.exe -c 8m
UID	String	dba

In these entries, *path* is the Adaptive Server Anywhere installation directory.

In addition, you must add the data source to the list of data sources in registry. For User data sources, you use the following key:

```
HKEY_CURRENT_USER\
   SOFTWARE\
      ODBC\
         ODBC.INI\
            ODBC Data Sources
```

For System data sources, use the following key:

```
HKEY_LOCAL_MACHINE\
   SOFTWARE\
      ODBC\
         ODBC.INI\
            ODBC Data Sources.
```

The value associates each data source with an ODBC driver. The value name is the data source name, and the value data is the ODBC driver name. For example, the User data source installed by Adaptive Server Anywhere is named ASA 6.0 Sample , and has the following value:

Value name	Value type	Value data
ASA 6.0 Sample	string	Adaptive Server Anywhere 6.0

Caution: ODBC settings are easily viewed
User data source configurations can contain sensitive database settings such as a user's ID and password. These settings are stored in the registry in plain text, and can be view using the Windows registry editors regedit.exe *or* regedt32.exe, *which are provided by Microsoft with the operating system. You can choose to encrypt passwords, or require users to enter them on connecting.*

Required and optional connection parameters

You can identify the data source name in an ODBC configuration string in this manner,

```
DSN=userdatasourcename
```

... identifies which user data source or system data source from the Registry is to be used for the ODBC connection.

When a DSN parameter is provided in the connection string, the current User data source definitions in the Registry are searched, followed by the System data source. File data sources are searched only when FileDSN is provided in the ODBC connection string.

The following table illustrates the implications to the user and developer when a DSN exists, when it is provided and the effects of including different parameters in the DSN.

To connect a client application to a database …	The connection string must identify…	The user must supply…
When the ODBC connection string identifies: a fully-configured DSN that contains the ODBC driver name and location; the name of the database file/server; startup parameters; and the user ID and password.	No additional information	No additional information.
When the OBDC connection string identifies a DSN containing only the name and location of the ODBC driver.	The name of the database file/ server; and, optionally, the user ID and the password.	User ID and password if not provided in the DSN or ODBC connection string.
When the system has no appropriately configured DSN.	The name of the ODBC driver to be used, in the following format: `Driver = {ODBCdrivername}` Also, the name of the database, the database file or the database server; and, optionally, other connection parameters such as user ID and password.	User ID and password if not provided in the ODBC connection string.

For more information on ODBC connections and configurations, see the following:

- "Connecting to a Database" on page 31.
- The Open Database Connectivity (ODBC) SDK, available from Microsoft.

Deploying Embedded SQL clients

Deploying Embedded SQL clients involves the following:

- **Installed files** Each client machine must have the files required for an Adaptive Server Anywhere Embedded SQL client application.
- **Connection information** The client application must have access to the information needed to connect to the server. This information may be included in an ODBC data source.

Installing files for Embedded SQL clients

The following table shows which files are needed for Embedded SQL clients.

Description	32-bit Windows	Windows 3.x	UNIX
Interface library	*dblib6.dll*	*dblib6w.dll*	*libdblib6.so*
Language-specific string library	*dblgen6.dll*	*dblgen6w.dll*	*dblgen6.res*
Network ports	*dbport6.dll*	*N/A*	
Connection Dialog	*dbcon6.dll*	*dbcon6w.dll*	

Notes

- The network ports DLL is not required if the client is working only with the personal database server.
- If the client application uses an ODBC data source to hold the connection parameters, your end user must have a working ODBC installation. Instructions for deploying ODBC are included in the Microsoft ODBC SDK.

  For more information on deploying ODBC information, see "Deploying ODBC clients" on page 714.

- The Connection dialog is needed if your end users will be creating their own data sources, if they will need to enter user IDs and passwords when connecting to the database, or if they need to display the Connection dialog for any other purpose.
- On HP-UX, all files listed with extension *.so* instead have extension *.sl*. On AIX, the files have extension *.so* or *.a*.
- No network port library is provided for Windows 3.x. Instead, the network ports are linked into the interface library.

Connection information

You can deploy Embedded SQL connection information in one of the following ways:

- **Manual** Provide your end-users with instructions for creating an appropriate data source on their machine.
- **File** Distribute a file that contains connection information in a format that your application can read.
- **ODBC data source** You can use an ODBC data source to hold connection information. In this case, you need a subset of the ODBC redistributable files, available from Microsoft. For details see "Deploying ODBC clients" on page 714.
- **Hard coded** You can hard code connection information into your application. This is an inflexible method, which may be limiting, for example when databases are upgraded.

Deploying JDBC clients

In addition to a Java Runtime Environment, each JDBC client requires the Sybase jConnect JDBC driver. Instructions on deploying jConnect can be found on the Sybase Web site at the following location:

```
http://www.sybase.com/products/internet/jconnect
```

Your Java application needs a URL in order to connect to the database. This URL specifies the driver, the machine to use, and the port on which the database server is listening.

For more information on URLs, see "Supplying a URL for the server" on page 525.

Deploying Open Client applications

In order to deploy Open Client applications, each client machine needs the Sybase Open Client product. You must purchase the Open Client software separately from Sybase. It contains its own installation instructions.

☞ Connection information for Open Client clients is held in the interfaces file. For information on the interfaces file, see the Open Client documentation and "Configuring Open Servers with DSEDIT" on page 821.

Deploying database servers

You can deploy a database server by making the SQL Anywhere Studio Setup program available to your end-users. By selecting the proper option, each end-user is guaranteed of getting the files they need.

In order to run a database server, you need to install a set of files. The files are listed in the following table. They should be installed in a single directory unless indicated with an asterisk. All redistribution of these files is governed by the terms of your license agreement. You must confirm whether you have the right to redistribute the database server files before doing so.

32-bit Windows	Windows 3.x	UNIX	NetWare
dbeng6.exe	*dbeng6w.exe*	*dbeng6*	
dbsrv6.exe	*N/A*	*dbsrv6*	*dbsrv6.nlm*
dbserv6.dll	*dbserv6w.dll*	*libdbtspt6.sol*	
dblgen6.dll	*dblgen6w.dll*	*dblgen6.res*	*dblgen6.res*
dbjava6.dll (1)	*dbjava6.dll* (1)	*libdbjava6.so* (1)	*dbjava6.nlm* (1)
dbctrs6.dll			
dbextf6.dll (2)	*dbextf6w.dll* (2)		*dbextf6.nlm* (2)
asajdbc.zip (1,3)	*asajdbc.zip* (1,3)	*asajdbc.zip* (1,3)	*asajdbc.zip* (1,3)
classes.zip (1,3)	*classes.zip* (1,3)	*classes.zip* (1,3)	*classes.zip* (1,3)

1 Required only if using Java in the database.
2 Required only if using system extended stored procedures and functions (xp_).
3 Install such that the CLASSPATH environment variable can locate classes in this file.

Notes

- Depending on your situation, you should choose whether to deploy the personal database server (*dbeng6*) or the network database server (*dbsrv6*).
- The Java DLL (*dbjava6.dll*) is required only if the database server is to use the Java in the Database functionality.
- The table does not include files needed to run utility command-line applications such as *dbbackup*.

  For information about deploying database utilities, see "Deploying database utilities and Interactive SQL" on page 725.
- The zip files are required only for applications that use Java in the database, and must be installed into a location so that they can be located in the user's CLASSPATH environment variable.

Deploying databases

You deploy a database file by installing the database file onto your end user's disk.

As long as the database server shuts down cleanly, you do not need to deploy a transaction log file with your database file. When your end-user starts running the database, a new transaction log is created.

For SQL Remote applications, the database should be created in a properly synchronized state, in which case no transaction log is needed. You can use the Extraction utility for this purpose.

Deploying databases on read-only media

Adaptive Server Anywhere databases distributed on read-only media such as a CD-ROM require the use of a Write File. This file records changes made to a database stored on a CD-ROM and is located on a read/write storage media such as a hard disk.

In this case, the database file is placed on the CD-ROM, while the write file is placed on disk. The connection is made to the write file, which maintains a transaction log file on disk.

Deploying embedded database applications

This section provides information on deploying embedded database applications, where the application and the database both reside on the same machine.

An embedded database application includes the following:

- **Client application** This includes the Adaptive Server Anywhere client requirements.

  For information on deploying client applications, see "Deploying client applications" on page 714.

- **Database server** The Adaptive Server Anywhere personal database server.

  For information on deploying database servers, see "Deploying database servers" on page 723.

- **SQL Remote** If your application uses SQL Remote replication, you must deploy the SQL Remote Message Agent.

- **The database** You must deploy a database file holding the data the application uses.

Deploying personal servers

When you deploy an application that uses the personal server, you need to deploy both the client application components and the database server components.

The language library (*dblgen6.dll*) is shared between the client and the server. You need only one copy of this file.

It is recommended that you follow the Adaptive Server Anywhere installation behavior, and install the client and server files in the same directory.

Remember to provide the Java zip files and the Java DLL if your application takes advantage of Java in the Database.

Deploying database utilities and Interactive SQL

If you need to deploy database utilities (such as *dbbackup.exe*) along with your application, then you need the utility executable together with the following additional files:

Description	32-bit Windows	Windows 3.x	UNIX
Database tools library	*dbtools6.dll*	*dbtools6w.dll*	*dbtools.so*
Additional library	*dbwtsp6.dll*	*dbwtsp6w.dll*	*dbwtsp6.so*
Language library	*dblgen6.dll*	*dblgen6w.dll*	*dblgen6.res*
Connection dialog (Interactive SQL only)	*dbcon6.dll*	*dbcon6w.dll*	

Notes

- On HP-UX, all files listed with extension *.so* instead have extension *.sl*. On AIX, the files have extension *.so* or *.a*.
- The database tools are Embedded SQL applications, and you must supply the files required for such applications, as listed in "Deploying Embedded SQL clients" on page 720.

Deploying SQL Remote

If you are deploying the SQL Remote Message Agent, you need to include the following files:

Description	32-bit Windows	Windows 3.x	UNIX
Message Agent	*dbremote.exe*	*dbremotw.exe*	*dbremote*
Database tools library	*dbtools6.dll*	*dbtools6w.dll*	*dbtools.so*
Additional library	*dbwtsp6.dll*	*dbwtsp6w.dll*	*dbwtsp6.so*
Language library	*dblgen6.dll*	*dblgen6w.dll*	*dblgen6.res*
VIM message link library (1)	*dbvim.dll*	*dbvimw.dll*	
SMTP message link library (1)	*dbsmtp.dll*	*dbsmtpw.dll*	
FILE message link library (1)	*dbfile.dll*	*dbfilew.dll*	*libdbfile.so*
MAPI message link library (1)	*dbmapi.dll*	*dbmapiw.dll*	

1 Only deploy the library for the message link you are using.

Notes

- It is recommended that you follow the Adaptive Server Anywhere installation behavior, and install the SQL Remote files in the same directory as the Adaptive Server Anywhere files.
- On HP-UX, all files listed with extension *.so* instead have extension *.sl*. On AIX, the files have extension *.so* or *.a*.

File naming conventions

Adaptive Server Anywhere uses consistent file naming conventions to help identify and group system components.

These conventions include:

- **Version number** The Adaptive Server Anywhere version number is indicated in the filename of the main server components (*.exe* and *.dll* files).

  For example, the file *dbeng6.exe* is a Version 6 executable.

- **Operating system** In cases where a conflict is likely, the last character in the filename indicates the platform on which that file is intended to run. This mainly applies to Windows 3.x files, which have a final character of w.

- **Language** The language used in a language-specific string library is indicated by a two-letter code within its filename. These two-letter codes are specified by ISO standard. For example, *dblgen6.dll* is the string library for English, on the NT/Win95 platform. The language used in the library is indicated by the two characters before the version number.

Identifying other file types

The following table identifies the platform and function of Adaptive Server Anywhere files according to their file extension. Adaptive Server Anywhere follows standard file extension conventions where possible.

File extension	Platform	File type
.nlm	Novell NetWare	NetWare Loadable Module
.cnt, .ftg, .fts, .gid, .hlp	Windows 3.x, NT, Win95	Help system file
.lib	Varies by development tool	Static runtime libraries for the creation of Embedded SQL executables
.cfg, .cpr, .dat, .loc, .spr, .srt, .xlt	Windows 95/NT	Sybase Adaptive Server Enterprise components
.cmd .bat	Windows 95, NT, 3.x	Command files
.res	NetWare, UNIX	Language resource file for non-Windows environments
.dll	Windows 95, NT, 3.x	Dynamic Link Library
.so .sl .a	UNIX	Shared object (Sun Solaris and IBM AIX) or shared library (HP-UX) file. The equivalent of a DLL on PC platforms.

Database file names

Adaptive Server Anywhere databases are composed to two elements:

- **Database file** This is used to store information in an organized format. This file uses a *.db* file extension.
- **Transaction log file** This is used to record all changes made to data stored in the database file. This file uses a *.log* file extension, and is generated by Adaptive Server Anywhere if no such file exists and a log file is specified to be used.
- **Write file** If your application uses a write file, it typically has a *.wrt* file extension. A write file is used with read-only databases.
- **Compressed database file** If you supply a read-only compressed database file, it typically has extension *.cdb*.

Both files are updated, maintained and managed by the Adaptive Server Anywhere relational database management system.