

## CHAPTER 1

# Running the Database Server

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

### Contents

<b>Topic</b>	<b>Page</b>
Introduction	4
Starting the server	8
Some common command-line switches	9
Stopping the database server	15
Loading and unloading databases	17
Running the server outside the current session	18
Troubleshooting server startup	29

## Introduction

Adaptive Server Anywhere provides two versions of the database server:

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

On Windows 95 and Windows NT the name of the personal server executable is *dbeng6.exe*. On Windows 3.x the name of the personal server is *dbeng6w.exe*, on UNIX operating systems is *dbeng6*.

- ◆ **The network database server** This executable does support client/server communications across a network, and is provided for multi-user use.

On Windows 95 and Windows NT the name of the network server executable is *dbsrv6.exe*. On Novell NetWare the name is *dbsrv6.nlm*, and on UNIX operating systems it is *dbsrv6*.

### Server differences

The request-processing engine is identical in the two servers. Each supports exactly the same SQL, and exactly the same database features. The main differences are as follows:

- ◆ **Network protocol support** Only the network server supports communications across a network
- ◆ **Number of connections** There is a limit of ten simultaneous connections for the personal server. The limit for the network server depends on your license.
- ◆ **Number of CPUs** The personal database server uses no more than two CPUs for request processing.
- ◆ **Default number of internal threads** The number of requests that the server can process at one time is configurable using the `-gn` command-line switch. The default number is different, however.

☞ For information on database server command-line switches, see "The database server" on page 12 of the book *Adaptive Server Anywhere Reference Manual*.

- ◆ **Startup defaults** The startup defaults are slightly different for the two servers, to reflect their use as a personal server and a server for many users.

## First steps

You can start a personal server running on a single database very simply. For example, on most operating systems you can start a personal server and load the sample database by running the following command in the Adaptive Server Anywhere installation directory:

```
dbeng6 asademo
```


The personal server is the executable *dbeng6.exe*, and the sample database is the file *asademo.db*.

Where commands are entered

You can enter a command such as the one above in several ways, depending on your operating system:

- ◆ You can type it at a system command prompt.
- ◆ You can place the command in a shortcut or desktop icon.
- ◆ You can run the command in a batch file.
- ◆ You can include the command as a **startline** parameter in a connection string.

There are slight variations in the basic command from platform to platform, described in the following section.

 You can also start a personal server using a database file name in a connection string. For more information, see "Connecting to an embedded database" on page 35.

## Platform-specific instructions

The way you start the database server varies slightly depending on the operating system you are using. This section describes how to enter command lines in for the simple case of running a single database with default settings, on each of supported operating system.

Notes

- ◆ The commands here start the personal server (**dbeng6**). To start a network server, simply replace **dbeng6** with **dsrv6**. The network server is not available on Windows 3.x.
- ◆ If the starting directory contains the database file, you do not need to specify the *path*.
- ◆ If you do not specify a file extension in *database-file*, the extension *.db* is assumed.

Starting the server for Windows NT

You can use a Program Manager icon to hold a command line, or enter the following command at the command prompt:

```
dbeng6 path\database-file
```

	<p>To start the server in a separate session, use the Windows NT <b>start</b> command:</p> <pre>start dbeng6 path\database-file</pre>
Starting the server for Windows 95	<p>You can use a Program Manager icon to hold a command line, or enter the following command at the command prompt:</p> <pre>dbeng6 path\database-file</pre>
Starting the server for Windows 3.x	<p>You can use a Program Manager icon to hold a command line, or enter the following command line in the Run dialog box:</p> <pre>dbeng6w path\database-file</pre> <p>You can run only one Windows 3.x server on a given computer at one time. There is no network server for Windows 3.x.</p>
Starting the server for UNIX	<p>The database server can be started with the following command:</p> <pre>dbeng6 path/database-file</pre>
Starting the server for NetWare	<p>There is no personal server for Novell NetWare, just a network server. The database server for NetWare is a NetWare Loadable Module (<i>dbsrv6.nlm</i>). An NLM is a program that you can run on your NetWare server.</p> <p>You can load a database server on your NetWare server as follows:</p> <pre>load dbsrv6.nlm path\database-file</pre> <p>The database file must be on a NetWare volume. A typical filename is of the form <i>DB:\database\sales.db</i>.</p> <p>You can load the server from a client machine using the Novell remote console utility. See your Novell documentation for details.</p> <p>You can put the command line into your Novell <i>autoexec.ncf</i> file so that Adaptive Server Anywhere is loaded automatically each time you start the NetWare server.</p>

## Starting the server for a Java-enabled database

If you are running a server with a Java-enabled database, and you wish to carry out Java operations, you must allow additional cache memory for the Java VM. You can do this by setting a cache size of 8 M for development work:

```
dbeng6 -c 8M path/database-file
```

☞ For more information on cache size, see "Controlling performance and memory from the command line" on page 10.

☞ For information on memory management for Java, see "Configuring memory for Java" on page 501.

## What else is there to it?

Although you can start a personal server in the simple way described above, there are many other aspects to running a database server in a production environment, and these are the subject of the remainder of the chapter.

- ◆ You can choose from many **command-line options** or **switches** to specify such features as how much memory to use as cache, how many CPU's to use (on multi-processor machines), and the network protocols to use (network server only). The command-line switches are one of the major ways of tuning Adaptive Server Anywhere behavior and performance.
- ◆ You can run the server as a **service** under Windows NT. This allows it to keep running even when you log off the machine.
- ◆ You can start the personal server from an application, and shut it down when the application has finished with it. This is typical when the database server is employed as an **embedded database**.

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

## Starting the server

The general form for the server command line is as follows:

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

The elements of this command line are as follows:

- ◆ **Executable** This can be either the personal server or the network server. For the file names on different operating systems, see "Introduction" on page 4.

**Can use either personal or network server**

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

- ◆ **Server switches** These options control the behavior of the database server, for all databases that are running.
- ◆ **Database file** You can enter zero, one, or more database file names on the command line. Each of these databases is loaded and available for applications.
- ◆ **Database switches** For each database file you start, you can provide database switches that control certain aspects of its behavior.

☞ In this section, we look at some of the more important and commonly-used options. For full reference information on each of these switches, see "The database server" on page 12 of the book *Adaptive Server Anywhere Reference Manual*.

**Commands must be entered on one line**

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

### Case sensitivity

Command-line parameters are generally case sensitive. You should enter all parameters in lower case.

## Some common command-line switches

This section describes some of the most common command-line switches, and points out when you may wish to use them.

It covers the following areas:

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

These command-line switches may be used in the **StartLine** connection parameter, or typed at a command prompt.

### Using configuration files

If you use an extensive set of command-line options, you can store them in a configuration file, and invoke that file on a server command line. The configuration file can contain switches on several lines. For example, the following configuration file starts the sample database, with a cache of 10 Mb, and starts the personal server named **Elora**.

```
-n Elora
-c 10M
path\asademo.db
```


where *path* is the name of your Adaptive Server Anywhere installation directory.

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

```
dbeng6 @sample.cfg
```

### Naming the server and the databases

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

	<p>The server and database names are among the connection parameters that client applications may use when connecting to a database. The server name appears on the desktop icon and on the title bar of the server window.</p>
Default names	<p>If no database name is provided, the default database name is the root of the database file name (the file name without the <i>.db</i> extension). For example, in the following command line the first database is named <b>asademo</b>, and the second <b>sample</b>.</p> <pre>dbeng6 asademo.db sample.db</pre> <p>If no server name is provided, the default server name is the name of the first database started. For example, with the above command the server is named <b>asademo</b>.</p>
Naming databases	<p>You can name databases by supplying a <code>-n</code> switch following the database file. For example, the following command line starts a sample database and names it:</p> <pre>dbeng6 asademo.db -n MyDB</pre>
Naming the server	<p>You can name the server by supplying a <code>-n</code> switch before the first database file. For example, the following command line starts a server named <b>Cambridge</b> on the <b>asademo</b> database:</p> <pre>dbeng6 -n Cambridge asademo.db</pre> <p>If a server name is used, you can start a database server with no database loaded. The following command starts a server named <b>Galt</b> with no database loaded:</p> <pre>dbeng6 -n Galt</pre> <p> For information about loading databases onto a running server, see "Loading and unloading databases" on page 17.</p>
Case sensitivity	<p>Server names and database names are case insensitive.</p>

## Controlling performance and memory from the command line

Several command-line options can have a major impact on database server performance. The following are some of these:

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



Generally speaking, the more memory made available to the database server, the faster it performs. The cache holds information that may be required more than once. Accessing information in cache is many times faster than accessing it from disk. The default cache size is 2 Mb, or 8 Mb if a Java-enabled database is started on the command line. On machines dedicated to running Adaptive Server Anywhere, you may wish to increase this value to match the available physical memory on your machine.

☞ For a detailed description of performance tuning, see "Monitoring and Improving Performance" on page 623.

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

By default, the database server uses all available processors.

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

☞ For a full list of startup options, see "The database server" on page 12 of the book *Adaptive Server Anywhere Reference Manual*.

## Controlling permissions from the command line

Some command-line options that control the permissions required to carry out certain global operations. These include the following:

- ◆ **Loading and unloading databases** By default, any user can load an extra database onto a running server. The `-gd` option allows you to limit the users who can do this to users with a certain level of permission in the database to which they are already connected. The permissible values are **dba**, **all**, or **none**.
- ◆ **Creating and deleting databases** By default, any user can use the CREATE DATABASE statement to create a database file. The `-gu` option allows you to limit the users who can do this to users with a certain level of permission in the database to which they are connected. The permissible values are **dba**, **all**, **none**, or **utility\_db**.

☞ For information about this option, "The database server" on page 12 of the book *Adaptive Server Anywhere Reference Manual*.

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

## Setting a maximum page size

The database server cache is arranged in **pages**—fixed-size areas of memory. As the server uses a single cache for its lifetime (until it is shut down), all pages must have the same size.

A database file is also arranged in pages, of size 1024, 2048, or 4096 bytes. Every database page must fit into a cache page. By default, the server page size is set to be the same as the largest page size of the databases on the command line. Once the server is started, you cannot load a database that has a larger page size than the server.

If you wish to allow databases of larger page size to be loaded after startup, you can force the server to start with a specified page size using the `-gp` option, which can take the values 1024, 2048, or 4096.

## Running in special modes

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

- ◆ **Bulk load** This is useful when loading large quantities of data into a database if it is being done through the Interactive SQL INPUT command. The `-b` option should not be used if you are using LOAD TABLE to bulk load data.

☞ For more information, see "The database server" on page 12 of the book *Adaptive Server Anywhere Reference Manual*, and "Importing and Exporting Data" on page 277.

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

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

For more information, see "The database server" on page 12 of the book *Adaptive Server Anywhere Reference Manual*.

## Selecting communications protocols

Any communications between a client application and a database server require a communications protocol. Adaptive Server Anywhere supports a set of communications protocols for communications across networks and for same-machine communications.

By default, the database server starts up all available protocols. You can limit the protocols available to a database server by using the `-x` command-line switch. At the client side, many of the same options can be controlled using the **CommLinks** connection parameter.

Available protocols for the personal server

The personal database server (*dbeng6.exe*) supports the following protocols:

- ◆ **Shared memory** This is used for same-machine communications, and is always loaded.
- ◆ **TCP/IP** This is provided for same-machine communications only, from TDS clients. Open Client or the jConnect JDBC driver. You must run TCP/IP if you wish to connect from Open Client or jConnect.

For more information on TDS clients, see "Adaptive Server Anywhere as an Open Server" on page 815.

- ◆ **Named Pipes** Provided on Windows NT only. Named Pipes is provided for same machine communications to from Windows 3.x client applications using ODBC or Embedded SQL.
- ◆ **DDE** Provided on Windows 95 only. DDE is used to communicate with a Windows 3.x client application running on the same computer.

Available protocols for the network server

The network database server (*dsrv6.exe*) supports the following protocols:

- ◆ **Shared memory** This is used for same-machine communications, and is always loaded.
- ◆ **TCP/IP** This is supported on all platforms.
- ◆ **IPX** This is supported on all platforms except for UNIX.
- ◆ **NetBIOS** This is supported on all platforms except for NetWare and UNIX.
- ◆ **Named Pipes** On Windows NT only. Named Pipes is provided for same machine communications to from Windows 3.x client applications using ODBC or Embedded SQL.

Specifying protocols

- ◆ **DDE** On Windows 95 only. DDE is used to communicate with a Windows 3.x client application running on the same computer.

☞ For more information on running the server using these options, see "Supported network protocols" on page 694.

You can instruct a server to use only some of the available network protocols when starting up, by using the `-x` command-line switch. The following command starts a server using the TCP/IP and IPX protocols:

```
dbsrv6 -x "tcpip,ipx"
```

The quotes are not strictly required in this example, but are needed if there are spaces in any of the arguments to `-x`.

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

```
dbsrv6 -x  
tcpip{MyIP=192.75.209.12:2367,192.75.209.32}  
path\asdemo.db
```

☞ For detailed descriptions of the available network communications parameters that can serve as part of the `-x` switch, see "Network communications parameters" on page 54 of the book *Adaptive Server Anywhere Reference Manual*.

## Stopping the database server

You can stop the database server in the following ways:

- ◆ Click SHUTDOWN on the database server display
- ◆ Using the *dbstop* command-line utility.

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

- ◆ If the server is a personal server started by an application connection string, it shuts down automatically by default when the application disconnects.

### Examples

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

```
dbsrv6 -n Ottawa asademo.db
```

- 2 Stop the server using *dbstop*:

```
dbstop -c "eng=Ottawa;uid=dba;pwd=sql"
```

☞ For information on *dbstop* command-line switches, see "The DBSTOP command-line utility" on page 103 of the book *Adaptive Server Anywhere Reference Manual*.

### Who can stop the server?

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

### Shutting down operating system sessions

If you close an operating system session where a database server is running, or if you use an operating system command to stop the database server, the server will not shut down cleanly. Next time the database is loaded, recovery will be required, and will happen automatically (see "Backup and Data Recovery" on page 553).

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

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

- ◆ Stopping the process in the Windows NT or Windows 95 Task Manager Processes window.
- ◆ Using a UNIX **slay** or **kill** command.

## Loading and unloading databases

A database server can have more than one database loaded at a time. You can load databases at the time the server is started, as follows:

```
dbeng6 asademo.db sample.db
```

### Loading a database onto a running server

You can also load databases after a server is started in the following ways:

- ◆ While connected to a server, connect to a database using a DBF parameter. This parameter specifies a database file for a new connection. The database file is loaded onto the current server.

☞ For more information, see "Connecting to an embedded database" on page 35.

- ◆ From Interactive SQL, use the START DATABASE statement.

☞ For a description, see "START DATABASE statement" on page 561 of the book *Adaptive Server Anywhere Reference Manual*.

### Limitations

- ◆ The server holds database information in memory using **pages** of a fixed size. Once a server has been started, you cannot load a database that has a larger page size than the server.
- ◆ The permissions required to load databases is determined by the `-gd` server command-line option.

### Unloading a database

You can unload a database in the following ways:

- ◆ Disconnect from a database loaded by a connection string. Unless you explicitly set the AUTOSTOP connection parameter to NO this happens automatically.

☞ For information, see "AutoStop connection parameter" on page 41 of the book *Adaptive Server Anywhere Reference Manual*.

- ◆ From Interactive SQL or Embedded SQL, use the STOP DATABASE statement.

☞ For a description, see "STOP DATABASE statement" on page 564 of the book *Adaptive Server Anywhere Reference Manual*.

## Running the server outside the current session

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

It is common to require database servers to be available all the time. To make this easier, you can run Adaptive Server Anywhere for Windows NT and for UNIX in such a way that, when you log off the computer, the database server does not terminate. The way you do this depends on the operating system you are using.

- ◆ **Windows NT service** You can run the Windows NT database server as a **service**. This has many convenient properties for running high availability servers.
- ◆ **UNIX daemon** You can run the UNIX database server as a **daemon** by using the `-ud` command-line option. This enables the database server to run in the background, and to continue running after you log off.

### Running the UNIX database server as a daemon

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

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

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

- ❖ **To run the UNIX database server as a daemon:**
  - ◆ Use the `-ud` command-line option when starting the database server. For example:

```
dbsrv6 -ud asademo.db
```

### Understanding Windows NT services

Although the database server can be run like any other Windows NT program rather than as a service, there are limitations to running it as a standard program, particularly in multi-user environments.



Limitations of running as a standard executable	When you start a program, it runs under your Windows NT login session: if you log off the computer, the program terminates. Only one person can be logged on to Windows NT (on any one computer) at one time. This restricts the use of the computer if you wish to keep a program running much of the time, as is commonly the case with database servers. You must stay logged in to the computer running the database server in order for the database server to keep running. This can also present a security risk as the Windows NT computer must be left in a logged on state.
Advantages of services	<p>Installing an application as a Windows NT service enables it to run even when you log off.</p> <p>When you start a service, it logs on using a special system account called LocalSystem (or using another account you specify). The service is not tied to the user ID of the person starting it, and therefore is not stopped when that person logs off. A service can also be configured to start automatically when the NT computer is started, before a user logs on.</p>
Managing services	Sybase Central provides a more convenient and comprehensive way of managing Adaptive Server Anywhere services than the Windows NT services manager.

## Programs that can be run as Windows NT services

The following programs can be run as services:

- ◆ The network server (*dsrv6.exe*).
- ◆ The personal server (*dbeng6.exe*).
- ◆ The Adaptive Server Anywhere Replication Agent (*dblrm.exe*: available separately).
- ◆ The SQL Remote Message Agent (*dbremote.exe*).
- ◆ A sample application.

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

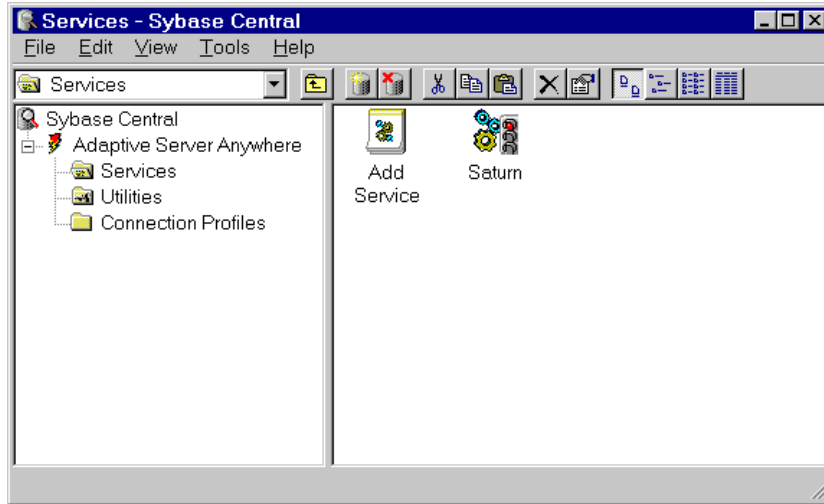
## Managing services

This section describes how to manage services. You can carry out the following service management tasks from Sybase Central:

- ◆ Add, edit, and remove services.
- ◆ Start, stop, and pause services.

- ◆ Modify the parameters governing a service.
- ◆ Add databases to a service, so that you can run several databases at one time.

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



## Adding a service

This section describes how to set up services using Sybase Central.

### ❖ To add a new service:

- 1 In Sybase Central, open the Services folder.
- 2 Double-click Add Service. The Service Creation Wizard is displayed.
- 3 Follow the instructions in the wizard to define the service you are creating.

### Notes


- ◆ Service Names must be unique within the first eight characters.
- ◆ If you choose to start a service automatically, it will start whenever the computer starts Windows NT. If you choose to start manually, you need to start the service from Sybase Central each time. You may want to select Disabled if you are setting up a service for future use.

- ◆ In the window, enter command-line switches for the executable, without the executable name itself. For example, if you wish a network server to run using the sample database with a cache size of 20Mb, and a name of **myserver**, you would enter the following in the Parameters box:

```
-c 20M  
-n myserver c:\asa6\asademo.db
```

Line breaks are optional. For information on valid command-line switches, see the description of each program in "Database Administration Utilities" on page 63 of the book *Adaptive Server Anywhere Reference Manual*.

- ◆ Choose the account under which the service will run: the special LocalSystem account or another user ID. For more information about this choice, see "Setting the account options" on page 23.
- ◆ If you want the service to be accessible from the Windows NT desktop, check Allow Service to Interact with Desktop. If this option is unchecked, no icon or window appears on the desktop.

 For more information on the configuration options, see "Configuring services" on page 21.

## Removing a service

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

If you wish to re-install a service that you have removed, you will need to reenter the command-line switches.

### ❖ To remove a service:

- 1 In Sybase Central, open the Services folder.
- 2 Right-click the icon of the service you wish to remove, and select Delete from the popup menu

## Configuring services

A service runs a database server or other application with a set of command-line switches. For a full description of the command-line switches for each of the administration utilities, see "Database Administration Utilities" on page 63 of the book *Adaptive Server Anywhere Reference Manual*.

In addition to the command-line switches, services accept other parameters that specify the account under which the service will run and the conditions under which it will start.

❖ **To change the parameters for a service:**

- 1 In Sybase Central, open the Services folder.
- 2 Right-click the icon of the service whose parameters you wish to change in the right panel of Sybase Central, and select Properties from the popup menu.
- 3 Alter the parameters as needed in the tabs of the Property sheet.
- 4 When you have altered the parameters as you wish, click OK to accept and save the changes.

Changes to a service configuration do not take effect immediately. They take effect next time the service is started. The Startup option is applied the next time Windows NT is started.

### Setting the startup option

The following options govern startup behavior for an Anywhere service:

- ◆ **Automatic** If you choose the **Automatic** setting, the service starts whenever the Windows NT operating system is started on the computer. This setting is appropriate for database servers and other applications that are run all the time.
- ◆ **Manual** If you choose the **Manual** setting, the service starts only when a user with Administrator permissions starts it. For information about Administrator permissions, see your Windows NT documentation.
- ◆ **Disable** If you choose the **Disable** setting, the service will not start.

### Entering command-line switches

The Configuration tab of the service property sheet provides a text box for entering command-line switches for a service. Do not enter the name of the program executable in this box.

#### Examples

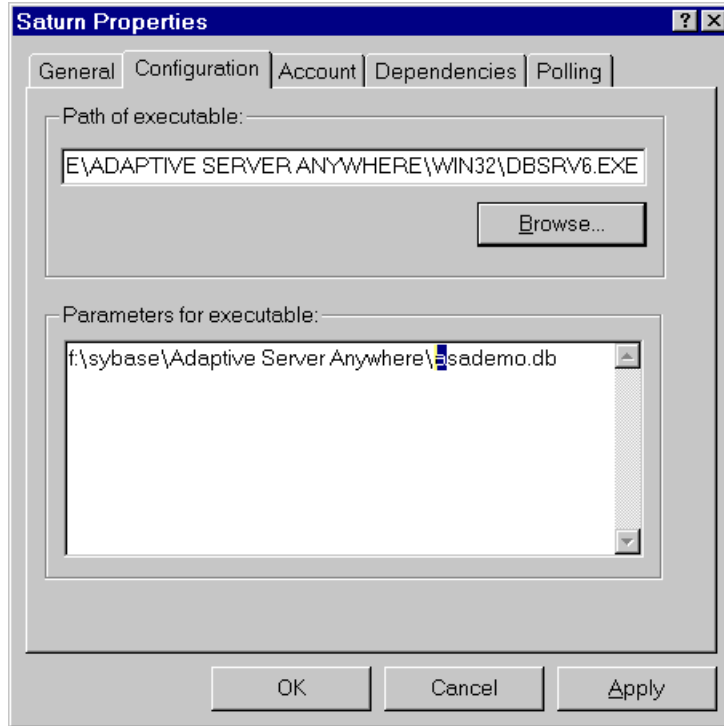
- ◆ To start a network server service running two databases, with a cache size of 20 Mb, and with a name of **my\_server**, enter the following in the Parameters box:

```
-c 20M  
-n my_server  
c:\asa6\db_1.db  
c:\asa6\db_2.db
```

- ◆ To start a SQL Remote Message Agent service, connecting to the sample database as user ID **DBA**, enter the following:

```
-c "uid=dba;pwd=sql;dbn=asademo"
```

The following figure illustrates a sample Property sheet.



✎ The command-line switches for a service are the same as those for the executable. For a full description of the command-line switches for each program, see "The Database Server" on page 11 of the book *Adaptive Server Anywhere Reference Manual*.

### Setting the account options

You can choose under which account the service will run. Most services run under the special LocalSystem account, and this is the default option for services. You can set the service to log on under another account by clicking **Account** and entering the account information.

If you choose to run the service under an account other than LocalSystem, that account must have the "log on as a service" privilege. This can be granted from the Windows NT User Manager application, under Advanced Privileges.

When an icon appears on the taskbar

Whether or not an icon for the service appears on the taskbar or desktop depends on the account that is selected, and whether Allow Service to Interact with Desktop is checked, as follows:

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

### **Changing the executable file**

To change the program executable file associated with a service, click the Configuration tab on the service property sheet and enter the new path and file name in the Path of Executable box.

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

### **Adding new databases to a service**

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


#### **❖ To add a new database to a service:**

- 1 Double-click the server to change the parameters for the service, and click the configuration tab.
- 2 Add the filename of the new database to the end of the list of parameters.

- 3 Click OK to save the changes.

The new database will not be added to the service immediately. It is loaded the next time the service is started.

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

 For a description of how to start a database on a server from Interactive SQL, see "START DATABASE statement" on page 561 of the book *Adaptive Server Anywhere Reference Manual*. For a description of how to implement this function in an Embedded SQL application, see the **db\_start\_database** function in "The Embedded SQL Interface" on page 7 of the book *Adaptive Server Anywhere Programming Interfaces Guide*. Starting a database from an application does not attach it to the service. If the service is stopped and restarted, the additional database will not be started automatically.

## Setting the service polling frequency

Sybase Central polls at specified intervals to check the state (started, stopped, paused, removed) of each service, and updates the icons to display the current state.

### ❖ To alter the Sybase Central polling frequency:

- 1 In Sybase Central, open the Services folder.
- 2 Right-click the icon of any service in the right panel of Sybase Central, and select Polling from the popup menu.
- 3 Set the polling frequency. The frequency applies to all services, not just the one selected.

The value you set in this window is saved for subsequent sessions, until you change it.

## Starting, stopping, and pausing services

### ❖ To start, stop, or pause a service:

- 1 In Sybase Central, open the Services folder.
- 2 Right-click the icon of the service.
- 3 Select Start, Stop, or Pause from the popup menu

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

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

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

## **The Windows NT Service Manager**

You can use Sybase Central to carry out all the service management for Adaptive Server Anywhere. You do not need to use the Windows NT Service Manager in the Control Panel.

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

You cannot install or configure parameters for Adaptive Server Anywhere services using the NT Service Manager. You must use Sybase Central.

## **Running more than one service**

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

### **Service dependencies**

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

In cases such as these, the services must start in the proper order. If a SQL Remote Message Agent service starts up before the server has started, it will fail because it cannot find the server.

You can prevent these problems using **service groups**. You can manage the service groups from Sybase Central.



## Service groups overview

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

Service	Default group
Network server	SQLANYServer
Personal server	SQLANYEngine
SQL Remote Message Agent	SQLANYRemote
Replication Agent	SQLANYLTM

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

### ❖ To check and change which group a service belongs to:

- 1 In Sybase Central, open the Services folder.
- 2 Right-click the icon of the service you wish to start, and select Properties from the popup menu.
- 3 Click the Dependencies tab. The text box displays the name of the group to which the service belongs.
- 4 Click Look Up to display a list of available groups on your system.
- 5 Select one of the groups, or type a name of a new group, and click OK to assign the service to that group.

## Managing service dependencies

With Sybase Central you can specify the following types of **dependency** for a service.

- ◆ You can ensure that at least one member of a each of a list of service groups has started before the current service
- ◆ You can ensure that each of a list of services, has started before the current service.

For example, you may wish to ensure that a particular network server has started before a SQL Remote Message Agent that is to run against that server starts.

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

- 1 In Sybase Central, open the Services folder.
- 2 Right-click the icon of the service, and select Properties from the popup menu.
- 3 Click the Dependencies tab. Click Add Service or Add Group to add a service or group to the list of dependencies.
- 4 Select one of the services or groups from the list, and click OK to add the service or group to the list of dependencies.

**Possible problems running more than one server as a service**

You may experience problems if you run more than one network server as a service at a time. Under some circumstances, it cannot be guaranteed that the correct service will respond to a client that sends a request to the database server as a broadcast. Broadcasts are typically used if you are using TCP/IP, or IPX on a Novell network without using the bindery. The broadcast is only used to find the server during startup.

We recommend that you not run more than one network server service at a time.

## Troubleshooting server startup

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

### Ensure that your transaction log file is valid

During development, you may replace a database file with a new version. If you do not delete the transaction log at the same time, the existing transaction log file will not match the database. The server does not start under these conditions.

### Ensure that you have sufficient disk space for your temporary file

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

If you do not have sufficient disk space on the drive where the temporary directory is located, you will have problems starting the server.

### Ensure that network communication software is running

Appropriate network communication software must be installed and running before you run the database server. If you are running reliable network software with just one network installed, this should be straightforward. If you experience problems, if you are running non-standard software, or if you are running multiple networks, you may want to read the full discussion of network communication issues in "Client/Server Communications" on page 685.

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

For example, if you are using NetBIOS under Windows 95, Windows for Workgroups, or Windows NT you may want to confirm that the *chat* or *winpopup* application is working properly between machines running client and database server software.

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

## **Debugging network communications startup problems**

If you are having problems establishing a connection across a network, you can use debugging options at both client and server to diagnose problems. On the server, you use the `-z` command-line option. The startup information is displayed on the server window: you can use the `-o` option to log the results to an output file.