Software through Pictures®

Millennium Edition Release 8

StP Guide to Sybase Repositories

UD/UG/ST0000-10128/002

Aonix

Software through Pictures
StP Guide to Sybase Repositories
Millennium Edition (UNIX)
March 2001

Aonix® reserves the right to make changes in the specifications and other information contained in this publication without prior notice. In case of doubt, the reader should consult Aonix to determine whether any such changes have been made. The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

Trademarks

Aonix

World Headquarters
5040 Shoreham Place
San Diego, CA 92122
Phone: (800) 97-AONIX
Fax: (858) 824-0212
E-mail: info@aonix.com
http://www.aonix.com

Table of Contents

Index

1 Introduction to StP Repositories

This chapter covers the following topics:

Overview of the Sybase Repository Manager

Each system in StP has a repository, which is a relational database made up of tables that hold interrelated data. Every object in an StP system maps to data in database tables. Databases are stored on database devices, which can be files or raw partitions (depending on the platform).

StP repositories (databases) are not visible at the operating system level and are managed by a relational database management system (RDBMS), called a "repository manager." Sybase Adaptive Server Enterprise (ASE) is a Sybase relational database management system that is supported in StP Release 8.x.

In general, StP users are not aware of the underlying database or the repository manager. For most routine StP administration tasks, StP provides an interface to the underlying relational database management system. However, there are times when it may be necessary for the StP administrator to interact directly with the Sybase servers for such tasks as:

- Server configuration

- Device management
- Transaction log management
- Troubleshooting

Every database on the server has a transaction log that records every update to the database. ASE uses the transaction log to recover data automatically following a system failure. StP does not make use of the Sybase transaction log, but the existence of this log requires that the user perform maintenance on it. For more information, see “Managing the Transaction Log” on page 2-30.

ASE has an error log, usually found in the ASE install directory. It is more fully described in “Error Log” on page 2-40.

Sybase Devices and Databases

Sybase databases are stored on Sybase server devices. A device is not necessarily a distinct physical device, such as a disk, although it can be. A device can be a disk partition or a file in a file system. A device has a logical name, by which it is known to the Sybase server, as well as a physical location.

ASE can support several devices, each of which is initialized with a specified amount of storage space. You cannot change the initialized size of a device, but you can add more devices if the server needs more space. When you no longer need a device, you can remove it.

The required devices inTable 1 (UNIX) and Table 2 (Windows NT) are initialized when StP is installed.:

Table 1: Required Devices (UNIX)

Logical Name	Stores	Default Physical Location	ASE Default Size
master	*master, model* and *tempdb* databases	<StP_base_dir>/Sybase/sybase_12.0.0/<platform_architecture>/data/master.dat	30MB
sysprocsdev	Sybase system procedures	<StP_base_dir>/Sybase/sybase_12.0.0/<platform_architecture>/data/sybprocs.dat	80MB
stp_device	StP repositories	<StP_base_dir>/Sybase/sybase_12.0.0/<platform_architecture>/data/stp_dev_0.dat	80 MB

Table 2: Required Devices (Windows NT)

Logical Name	Stores	Default Physical Location	ASE Default Size
master	*master, model* and *tempdb* databases	C:\StP8.x\SYBASE\DATA \MASTER.DAT	45 MB
sysprocsdev	Sybase system procedures	C:\StP8.x\SYBASE\DATA \SYBPROCS.DAT	80 MB
stp_device	StP repositories	C:\StP8.x\SYBASE\DATA \ STP_DEVICE.DAT	80 MB

See the Sybase *System Administration Guide* for a full explanation of data and log storage.

Databases on the *master* Device

The *master* device holds the *master, model,* and *tempdb* databases, and the *sysprocsdev* device holds the Sybase system procedures. The StP installation procedure creates the *stp_device* device, which is the default device for StP repositories.

Master Database

The *master* database controls user databases and the operation of Sybase servers through the maintenance of a collection of special system tables. A system table manages the user databases and Sybase servers as a whole. All ASE databases have system tables, and the *master* database has some additional system tables not found in the user databases. These system tables support the StP repositories.

It is essential to keep the *master* database current. You should back up the *master* database with the Sybase **dump database** command on a regular basis. If the *master* database is damaged, you must recover it using a special procedure. See the Sybase *System Administration Guide* for a detailed description of the *master* database and information on the **dump database** command, as well as the procedure for restoring the *master* database. Instructions for backing up the *master* database are in “Backing up the master Database” on page 2-38.

Model Database

The *model* database provides a template for new user databases. Every time the StP **File > New > System** command or sys_create QRL function is successfully executed, the Sybase server first makes a copy of the *model* database; then StP creates its own tables and other database objects to hold repository information.

Temporary Database

The temporary database, *tempdb,* provides a storage area for all the temporary tables and other temporary working storage needs for all the databases on the server. The StP installation procedure initializes *tempdb* to 2MB. This should be sufficient for most installations.

System Procedure Database on its Own Device

The system procedure device, *sysprocsdev*, contains a singular database named *sybprocs*.This database stores a collection of system procedures provided by Sybase.

A stored procedure, which users can create, is a collection of SQL statements that are predefined and can be invoked by name. A system procedure is a special stored procedure created by Sybase for use in the Sybase server's system administration. These procedures provide shortcuts for retrieving information and updating system tables on ASE.

StP Repositories on their Own Device

The StP installation procedure initializes the *stp_device* device with 80 MB for StP repositories. The *stp_device* device is the default device, but you can create and specify additional default devices. See "Adding a Device" on page 2-23 and "Designating Default Devices" on page 2-26 for more information.

Sybase Utilities

The Sybase installation includes several utilities for various ASE administrative tasks.

ASE Utilities for UNIX

The following commands invoke these ASE utilities:

In <SYBDIR>/ASE-12_0/install:

- **showserver**—Shows whether or not a Sybase server is running on the host machine
- **startserver**—Takes input from a runserver file to boot up ASE
- **srvbuild**—Allows you to create a Sybase server, edit or add a server to an interfaces file, configure an Adaptive Server, or upgrade an Adaptive Server

- **buildmaster**-creates new master device; creates new master database.

In *<SYBDIR>/OCS-12_0/bin:*

- **isql**—Allows you to log into ASE, where you can issue SQL commands directly
- **dsedit**

Refer to Sybase utility guides for a full description of features and functionality for each utility.

ASE Utilities for Windows NT

You can find the utilities provided with ASE in **Start** > **Programs** > **Software through Pictures** > **Sybase Repository** > **Sybase Central**. They provide a graphical user interface (GUI) for a number of ASE tasks. Refer to Sybase utility guides for a full description of features and functionality for each utility. You can get access to the Sybase utility guides from the Sybase website at *http://www.sybase.com*.

The utilities you will use the most often are as follows:

- Sybase Central—Lets you perform administrative tasks, including starting and stopping ASE.
- Dsedit—Lets you edit the interfaces file, which connects Sybase servers across a network.
- Server Config—Creates a new ASE, Backup Server, or Monitor Server. It also upgrades an earlier Adaptive Server to ASE.

If you have Sybase Open Client, you can find the dsedit utility in **Start** > **Programs** > **Software through Pictures** > **Sybase Repository** > **Edit Server definitions**.

From the command line, you can run the **isql** utility. This utility allows you to log into ASE, where you can issue SQL commands directly.

Sybase Naming Conventions

When you create objects in the Sybase servers, such as login and user names or database objects such as tables, you must be aware of the Sybase rules for named identifiers. In general, the following rules apply:

- ASE identifiers can be a maximum of 30 characters in length, whether or not multibyte characters are used. The first character of an identifier must be an alphabetic character as defined in the current character set in use on ASE or the _ symbol (underscore).
- After the first byte, you can include characters declared as alphabetic, numeric, and these symbols: #, @, and money symbols such as $(dollars) or ¥ (yen).
- You **cannot** begin an identifier with a number.
- You **cannot** have a hyphen in a server name.
- You **cannot** embed spaces in identifiers.
- You **cannot** use the names of Sybase reserved words

Use the following guidelines when you create Sybase logins or user accounts:

- Use 3–8 characters, all lowercase (the character set supplied with ASE is case sensitive).
- If possible, use an alphabetic first character and alphanumeric characters for the remainder.
- Avoid using Sybase reserved words, such as “read,” “grant,” “use,” “alter,” and so on.
- Avoid using the following characters anywhere within the login or user name, since they are significant to many shells: #, @, $, %, and &.

For more information on Sybase naming conventions, identifiers, and reserved words, consult the Sybase documentation, which is available on their website at *http://www.sybase.com*.

Additional References

In addition to this manual, you may want to consult other documentation and books on Sybase ASE.

Sybase Documentation

- *System Administration Guide*
- *Server Reference Manual*
- *Performance and Tuning Guide*

Some useful Sybase documentation for ASE on Windows NT is:

- *Installing ASE and OmniConnect for Windows NT*
- *Introducing Sybase Central for Adaptive Server Enterprise*
- *Managing and Monitoring Sybase Adaptive Server Enterprise Using Sybase Central*
- *Release Bulletin Adaptive Server Enterprise for Windows NT*

Note: More complete documentation is available on the Sybase web site at *http://www.sybase.com*.

2 Administering Sybase Repositories

This chapter describes some common StP administrative tasks that require an StP administrator to interact directly with Adaptive Server Enterprise (ASE). Topics covered are as follows:

The Sybase System Administrator

The installation of Sybase sets up an account for the system administrator with the following default settings:

Username = **sa**

Password = **welcome**

For security reasons, it is important to assign a new password for the system administrator's account, as well as unique passwords for the StP users.

For information on changing the sa password, see "Changing the sa Password" on page 2-9.

For more information on passwords, see *STP Administration*.

Determining if the Sybase Server is Running

To verify that a Sybase server is running:

For **UNIX**:

- Execute the **showserver** command from the ASE install directory.

For **Windows NT**:

- 1. From the **Sybase** program group, choose **Sybase Central**.

  2. Select the **Sybase Adaptive Server Enterprise** icon, which displays the available ASEs and indicates whether they are running or stopped.

-or-

- To test an ASE's connections, start **dsedit:**

  1. Go to the **Sybase** program group and selecting **Dsedit**.
  2. Select the interfaces file you want, and click **OK**.
  3. In the **Interfaces Driver** dialog box, select the server you want.
  4. From the **Server Object** menu, choose **Ping Server**.

-or-

- In the Windows NT task manager, look for the ASE process.

  This process is the ASE executable **sqlsrvr.exe**

Note: If the server process is not running, you can restart it manually, as described under in "Starting the Sybase Server (UNIX)" on page 2-3 or"Starting the Sybase Server (Windows NT)" on page 2-7.

Starting the Sybase Server (UNIX)

You can restart the ASE process by either of these methods:

- Manually-specify the name of the runserver file in the **startserver** command.

-or-

- Arrange to have the ASE process start automatically whenever the server machine is rebooted.

Starting ASE Manually

When the machine on which ASE is installed is rebooted, you can restart the server process manually. Before restarting the server process, set the environment variables and **SYBASE** (ASE directory path to the location of the interfaces file).

Note: You must have permissions on the file system on which ASE is installed.

To restart the server manually:

1. Change the directory to the ASE install directory, usually */Sybase/sybase_<version>/<PLATFORM>/ASE-12_0/install.*
2. Execute the **startserver** command with the **–f** flag and the name of the runserver file, where <servername> is the name of the StP server (see "The Runserver File" on page 2-4):

   ```
   ./startserver –f ./RUN_<servername>
   ```

For example, the following commands start a Sybase Server named STP_SERVER from an install directory named: */StP8.x/Sybase/sybase_12.0.0/SOL24SN4 / ASE-12_0 / install:*

```
cd /StP8.x/Sybase/sybase_12.0.0/SOL24SN4/ASE-12_0/install
./startserver –f ./RUN_STP_SERVER
```

The Runserver File

When Sybase ASE is installed, a file known as the runserver file is created in the ASE install directory. The runserver file is an executable file that contains the **dataserver** command for starting the **dataserver** process, which is the ASE executable.

The StP installation process automatically assigns the runserver filename by appending the name of the ASE Server to *RUN_*. For example, if the name of the server is STP_SERVER, the name of the corresponding runserver file is *RUN_STP_SERVER*. You can have multiple ASE Servers, each with its own runserver file.

The arguments to the **dataserver** command are described in Table 1:

Table 1: dataserver Command Arguments

Argument	Description	For details, see:
-d	The master device.	"Sybase Devices and Databases" on page 1-2

Table 1: dataserver Command Arguments

Argument	Description	For details, see:
-e	The errorlog, which contains the backtrace from fatal ASE messages and error messages from the kernel. If the error log does not exist, it is created when the server is started.	"Error Log" on page 2-40 and the Sybase Server system administration documentation
-i	The location of the ASE interfaces file.	"Interfaces File" on page 2-32
-s	The name of the ASE, exactly as it appears in the interfaces file.	The example of a runserver file, which follows this table

Additional information about the locations of ASE files may be included in the comments in the runserver file.

The following is an example of a runserver file:

```
!/bin/sh
#
# Master device path:
/usr2zazu/stpqa/Stp_8.0_16/Sybase/sybase_12.0.0/SOL24SN4/data/master.dat
# Error log path:
/usr2zazu/stpqa/Stp_8.0_16/Sybase/sybase_12.0.0/SOL24SN4/install/errorlog
# Directory for shared memory files:
/usr2zazu/stpqa/Stp_8.0_16/Sybase/sybase_12.0.0/SOL24SN4
# Adaptive Server name: CALVIN_SD_AONIX_COM
#
/usr2zazu/stpqa/Stp_8.0_16/Sybase/sybase_12.0.0/SOL24SN4/ASE-12_0/bin/dataserver
\
-d/usr2zazu/stpqa/Stp_8.0_16/Sybase/sybase_12.0.0/SOL24SN4/data/master.dat \
-e/usr2zazu/stpqa/Stp_8.0_16/Sybase/sybase_12.0.0/SOL24SN4/install/errorlog \
-M/usr2zazu/stpqa/Stp_8.0_16/Sybase/sybase_12.0.0/SOL24SN4 \
-sCALVIN_SD_AONIX_COM \
```

Automatic Restart on Reboot

You can arrange for the server process to restart automatically whenever the server machine is rebooted. This is accomplished by making an entry for the server in the operating system start-up script.

To set up automatic restart:

1. Login as root.
2. Execute the following commands:

   cd <SYBDIR>

   . /sybase.startconfig

   where <SYBDIR> is the home directory of the Sybase server software, usually *StP/Sybase/Sybase-12.0/<platform>*.
3. Add the following line to the end of the .etc/rc.local or /etc/rc file on SunOS or HP-UX machines.

   /etc/sybstart
4. If the sa account has a non-default password, modify the /etc/sybstart file so that the password line sets the password to the actual sa password:

   PASSWORD = <sa_password>

See the Sybase system administration documentation for additional information about starting the server automatically on reboot. In addition, if the server is running Solaris, refer to your operating system's administration manual for platform-specific instructions on automatic start-up of processes on reboot.

Starting the Sybase Server (Windows NT)

You can restart ASE by either of these methods:

- Manually, either from Sybase Central, or as a Windows NT service
- Arrange to have the Sybase server processes start automatically whenever the server machine is rebooted

Starting the Sybase Server Manually

When the machine on which the Sybase server is installed is rebooted, you can restart the server process manually, using any of the following methods.

Starting Adaptive Server Enterprise from Sybase Central

Sybase Central lets you log onto and administer ASE running on the local PC or on a remote PC or UNIX host on the same network (see "Accessing a Remote Sybase Server" on page 2-32).

To start and log onto ASE from Sybase Central:

1. Click the **Start** button.
2. Choose **Programs > Aonix Software through Pictures > Sybase Repository** > **Sybase Central**.
3. Double-click the **Sybase Adaptive Server Enterprise** icon, which lists the servers installed on your machine.
4. Right-click the server you want, and from the menu, choose **Start**.

5. To log onto the server, double-click its name, and in the **Sybase Adaptive Server Enterprise Logon** dialog box, specify your username and password.
6. Click **OK**.

   For more information on logging onto a Sybase server, see “Using Transact-SQL in Sybase” on page 2-8.

Starting the Sybase Server as a Windows NT Service

To start the Sybase server as a Windows NT service:

1. Click the **Start** button.
2. Choose **Settings > Control Panel**.
3. Double click the **Services** icon.
4. Select **Sybase Server_<servername>**.

   The Services Manager lists ASE servers as “Sybase SQL Server.”
5. Click **Start**.

Automatic Restart on Reboot

To arrange for the Sybase server to restart automatically whenever the server machine is rebooted:

1. Click the **Start** button.
2. Choose **Settings > Control Panel**.
3. Double-click the **Services** icon.
4. Select a service and click **Startup**.
5. In the **Startup Type** box, select **Automatic**.
6. Click **OK**.

Using Transact-SQL in Sybase

Once you start and log onto ASE, you can use Transact-SQL to query the server

Connecting to the Sybase Server through isql

To connect to ASE using **isql**:

1. In a shell or command prompt window, type `isql`, using the `–U` option to connect to the server as *sa* and the `-P` option to specify your password:

   ```
   isql –Usa
   ```

2. When prompted for the password, enter the *sa* password.

   When connected to the server, the **isql** prompt appears as a number followed by a greater than symbol:

   ```
   1>
   ```

3. To send an **isql** command to the server, type the command at the **isql** prompt and then type `go` on a line by itself. For example, to find help information for the databases in the current server, you enter:

   ```
   1> sp_helpdb
   2> go
   ```

   If you make a mistake, just use the **reset** command, for example:

   ```
   1> sp_bozo
   2> reset
   ```

4. When you are finished administering the server, disconnect by typing `quit` at the **isql** prompt (you do not have to enter `go` afterwards).

   ```
   1>  quit
   ```

   This returns you to the shell or command prompt.

Changing the sa Password

By default, when ASE is installed, “welcome” is set as the *sa* password. If you leave the *sa* password unset, you are operating in an unsecure environment. To operate in a secure environment, change the *sa* password to a setting other than “welcome.”

You can change the sa password using one of these methods:

- From StP, use the StP Desktop or ToolInfo variables (see *StP Administration*).
- You can also use **isql**.
- **Windows NT**—For ASE, use **Sybase Central** in the **Sybase** program group.

Changing sa Password with isql

To change the password for ASE user sa, login as “sa” and then issue the following command:

```
1> sp_password welcome, <new password>
2> go
```

Changing sa Password with Sybase Central (Windows NT)

For ASE, to change the sa password:

1. Start ASE and log onto the server (see “Starting Adaptive Server Enterprise from Sybase Central” on page 2-7.
2. In the current ASE, open the **Logins** folder.
3. Select the sa login icon.
4. From the **File** menu, choose **Change Password**.
5. Enter the current password and the new password, using at least six characters, and then confirm the new password.
6. Choose **OK**.

Configuring the Sybase Server

Configuration parameters, which you set by the system procedure **sp_configure**, control how the Sybase server uses memory, and let you manage a variety of administrative operations, from basic configuration operations to performance tuning.

Memory Considerations

Of the memory the Sybase server takes when it starts, some is used for executable code, static memory, non-cached data structures and user-configurable variables, such as databases, devices, and user connections. The remainder is used for procedure and data caches. Because accessing data in cache is much faster than accessing data from disk, it is important to have as much cache available as possible. The larger the databases, the more complicated the queries, and the greater the response-time requirements, the more memory is needed. See the Sybase *System Administration Guide* for advice on how much memory to configure for your system.

The error log shows how much memory was allocated to cache when you booted the server.

When the Sybase server is installed, it is configured for a default amount of dedicated RAM from which memory is allocated to default numbers of user connections, open databases and database devices.

Determining Data and Procedure Cache

Look in the *<SYBDIR>/ASE-12_0/install/errorlog* for the following information:

```
2000/08/22 15:38:14.39 server  Number of proc buffers allocated: 3086.
2000/08/22 15:38:14.89 server  Number of blocks left for proc headers: 3048
2000/08/22 15:38:14.96 server  Proc header memory allocated 1524 pages for
each per engine cache
2000/08/22 15:38:14.97 server   Memory allocated for the default data cache
cachelet 1: 23188 Kb
```

- *proc buffers* indicates how many compiled objects you can have in procedure cache;
- *proc headers* are the number of 2-K pages available for compiled objects in the procedure cache. Divide each of these figures by 512 to calculate their size in megabytes.
- *default data cache* is the amount of data cache available for processing. The *default data cache* is reported in kilobytes. Divide this figure by 1024 to calculate the size of data cache in megabytes.

Viewing and Changing Configuration Values

You can view and set configuration values from Sybase Central or **isql**.

Viewing/Changing Configurations from Sybase Central (Windows NT)

If you are using ASE, you can view configuration values from Sybase Central. Only an sa user can change the values.

1. Start ASE and log onto the server (see "Starting Adaptive Server Enterprise from Sybase Central" on page 2-7.
2. Select the server to configure.
3. From the **File** menu, choose **Configure**.
4. In the **Server Configuration** dialog box, select the parameter you want from the list.

   The default value, minimum value, maximum value, explanation, and whether the server needs to be restarted afterwards displays. If you want, you can show configuration parameters for a specific group.
5. Reset the **Value** setting as necessary.

6. Click **OK**.

Viewing/Changing Configurations with isql

In **isql**, the Sybase system administrator, sa, can view or change server configuration values in the Sybase server by using the **sp_configure** system procedure. All users can execute **sp_configure** and **sp_helpconfig** to examine the state of the configuration parameters. However, only user sa can change the values.

The syntax for **sp_configure** is:

```
sp_configure ["<config_name>" [,<config_value>]
    |<group_name> | <non_unique_parameter_fragment>]
```

where:

- **`<config_name>`** is the configuration parameter’s name.
- **`<config_value>`** is the configuration parameter’s value.
- **`<group_name>`** refers to a particular group of configuration parameters, such as **`user environment`** or **`meta-data caches`**.
- **`<non_unique_parameter_fragment>`** refers to parameter names that match a name fragment. For example, entering **`identity`** displays information about the **identity burning set factor**, **identity grab size**, and **size of auto identity column** parameters.

Using the **sp_configure** command with no parameters displays a list of all the configuration parameters and their current and permissible range of values, sorted by group name.

This list is rather long, so it is better to specify parameters with **sp_configure** or use **sp_helpconfig**. Using either of these system procedures with the parameter name specified displays the value of the named configuration parameter.

The value in the Run Value column for each parameter is the configuration value with which the Sybase server was started. For many parameters, if you change a configuration the new value is stored in the Config Value column and becomes the new Run Value when the Sybase server is restarted.

If you are using ASE, you can find help information, including memory usage, for a particular configuration parameter by using the **sp_helpconfig** system procedure, for example:

```
1> sp_helpconfig "number of remote connections"
2> go
```

```
number of remote connections controls the limit on active connections
initiated to and from this Adaptive Server. The default is 20.

Minimum Value   Maximum Value   Default Value   Current Value   Memory Used
-------------   -------------   -------------   -------------   -----------
            0           32767              20              20            33
```

Sybase Configuration Parameters

Table 2 summarizes each of the Sybase server configuration parameters you may need to change. When changing any of these parameters, be sure to configure or reconfigure the memory or total memory parameter with the additional amount of memory recommended in the table. Additional information on each parameter follows the table.

Table 2: Adaptive Server Configuration Parameters

Parameter	isql Configuration	Sybase Central Configuration	Additional Memory Requirement
Open Databases	sp_configure "open databases", <config_value>	Group: Sybase Server Administration	17 K (eight-and-a-half 2-K blocks) per additional database (round up)
Devices	sp_configure devices, <config_value>	Group: Disk I/O	4 K (two 2-K blocks) per additional device
User Connections	sp_configure “user connections”, <config_value>	Group: User Environment	52 K (twenty-six 2-K blocks) per additional user connection
Memory	sp_configure “total memory”, <config_value>	Group: Physical Memory	Sum of all memory increases for other parameters, plus any additional increase

Table 2: Adaptive Server Configuration Parameters (Continued)

Parameter	isql Configuration	Sybase Central Configuration	Additional Memory Requirement
Multiple CPUs	sp_configure “max online engines”, <config_value>	Group: Processors	None

Configuring Databases

The configuration value should equal to or exceed the actual number of repositories (user databases) in the server, whether or not they are open. On very active installations with many users, make periodic checks to see if the number of repositories created exceeds the number for which the Sybase server is configured.

You can use the **List Repositories** StP Desktop command to see how many repositories exist in the server (see *StP Administration*). If your installation has many systems, it may be quicker to use one of the methods described in this section.

When you use the Sybase server to find or reconfigure the number of databases in the server, remember that it takes into account the system databases: *master*, *sybsystemprocs*, *model*, and *tempdb*.

For example, if you have 20 StP repositories, the **List Repositories** command will list 20, but Sybase will list 24.

Note: When reconfiguring databases, be sure also to reconfigure the memory or total memory parameter with an additional 17 K (nine 2-K blocks) per additional database.

Configuring Databases with Sybase Central (Windows NT)

To configure databases:

1. Start ASE and log onto the server (see “Starting Adaptive Server Enterprise from Sybase Central” on page 2-7.
2. Select the server to configure.
3. From the **File** menu, choose **Configure**.

4. In the **Server Configuration** dialog box, select the **number of open databases** parameter.
5. Reset the **Value** setting as necessary.
6. Click **OK**.
7. Restart ASE.

Configuring Databases with isql

If you are using ASE, you can use the **sp_countmetadata** system procedure in **isql** to determine to determine how many databases there are in the server, and how much memory they use:

```
1> sp_countmetadata databases
2> go
```

```
There are 12 databases, requiring 396 Kbytes of memory. The
'open databases' configuration parameter is currently set to
12.
```

To see what the configured value is:

```
1> sp_configure "open databases"
2> go
```

The number in the Config Value column in the results shows the number of databases for which the server was configured:

```
 Parameter Name                 Default     Memory Used Config Value Run Value
------------------------------ ----------- ----------- ------------ -----------
number of open databases                12         396           12          12
(return status = 0)
```

- The *Run Value* column shows the number of databases for which the server is currently configured.
- The *Config Value* will take the place of the *Run Value* the next time the Sybase server is restarted.

To increase the configured number of open databases from 12 to 20:

```
1> sp_configure "open databases", 20
2> go
```

Configuring Devices

The configured number of devices must be equal to or exceed the actual number in the Sybase server. There is no StP Desktop command for determining number of devices. Use one of the procedures described in this section, as appropriate.

Note: When reconfiguring devices, be sure also to reconfigure the memory or total memory by an additional 4 K (two 2-K blocks) per additional device to maintain the current amount of cache. Otherwise, the memory needed for the additional devices is allocated from total memory, reducing the memory allocated to cache.

Configuring Devices with Sybase Central (Windows NT)

For ASE, to determine the number of existing devices:

1. Start ASE and log onto the server (see “Starting Adaptive Server Enterprise from Sybase Central” on page 2-7.
2. Select the server to configure.
3. From the **File** menu, choose **Configure**.
4. In the **Server Configuration** dialog box, select the **number of devices** parameter.
5. Reset the **Value** setting as necessary.
6. Click **OK**.

The change will take place the next time you restart ASE.

Configuring Devices with isql

To determine the number of existing devices, using **isql**:

```
1> select count(*) from sysdevices
2> go
```

If you are using ASE, you can use the **sp_helpconfig** system procedure to find the configured value:

```
1> sp_helpconfig "devices"
2> go
```

To increase the configured number of devices from the default of 10 to 15:

```
1> sp_configure "devices", 15
2> go
```

Configuring User Connections

The number of user connections recommended for the typical StP user is up to 10 connections per user. ASE uses one user connection for each device, Backup Server, mirror device and active site handler.

Note: Each connection requires 52 K of configured memory. If you add user connections, you need to increase the memory value by 52 K, or twenty-five (25) 2-K pages, per additional user connection.

Configuring User Connections with Sybase Central (Windows NT)

For ASE, to configure user connections:

1. Start ASE and log onto the server (see "Starting Adaptive Server Enterprise from Sybase Central" on page 2-7.
2. Select the server to configure.
3. From the **File** menu, choose **Configure**.
4. In the **Server Configuration** dialog box, select the **number of user connections** parameter.
5. Reset the **Value** setting as necessary.
6. Click **OK**.

The change will take place the next time you restart ASE.

Configuring User Connections with isql

To determine the maximum number of user connections available, using **isql**:

```
1> select @@max_connections
2> go
```

If you need to configure more user connections than are available, see “File Descriptors (UNIX Only)” on page 2-22.

You can use **sp_helpconfig**:

```
1> sp_helpconfig "user connections"
2> go
```

To configure user connections for 30 concurrent StP users:

```
1> sp_configure "user connections", 30
2> go
```

Configuring Memory

In addition to adding memory to the Sybase server for increased user-configurable parameters, we recommend increasing the configured memory value to allocate more data cache for processing data. Memory is configured in 2-K blocks (also referred to as 2-K pages).

Because the procedure cache is allocated as a percentage of total memory, when you increase total memory, the procedure cache also increases. If you increase total memory by a sizable amount, you may want to reduce the percentage allocated to procedure cache, for example, from 20 percent to 15 percent.

Configuring Memory with Sybase Central (Windows NT)

For ASE, to configure the amount of memory:

1. Start ASE and log onto the server (see “Starting Adaptive Server Enterprise from Sybase Central” on page 2-7.
2. Select the server to configure.
3. From the **File** menu, choose **Configure**.

4. In the **Server Configuration** dialog box, select the **total memory** parameter.
5. Reset the **Value** setting as necessary.
6. Click **OK**.

The change will take place the next time you restart ASE.

Configuring Memory with isql

You can use sp_helpconfig to determine total memory:

```
1> sp_helpconfig “total memory”
```

```
2> go
```

To allocate 64 MB of physical memory to the Sybase server memory, using **isql**:

```
1> sp_configure "total memory", 32768
2> go
```

To decrease the percentage of the reconfigured memory used for procedure cache from 20 percent to 15 percent:

```
1> sp_configure "procedure cache", 15
2> go
```

Configuring for Multiple CPUs

If more than one CPU is available, ASE can be configured accordingly. Configure one less than the total number of available CPUs on the machine to be dedicated to the Sybase server.

Configuring for Multiple CPUs with Sybase Central (Windows NT)

For ASE, to configure for multiple CPUs:

1. Start ASE and log onto the server (see “Starting Adaptive Server Enterprise from Sybase Central” on page 2-7.
2. Select the server to configure.

3. From the **File** menu, choose **Configure**.
4. In the **Server Configuration** dialog box, select the **max online engines** parameter.
5. Reset the **Value** setting as necessary.
6. Click **OK**.

The change will take place the next time you restart ASE.

Configuring for Multiple CPUs with isql

To configure the Sybase server to use three out of four available CPUs, using **isql**:

```
1> sp_configure "max online engines", 3
2> go
```

Recovering from Badly Configured Values

If you increase a parameter's value without allocating more memory for it, or if you set memory higher than what the operating system can allocate, the Sybase server will not restart. To remedy this problem, do the following:

1. Go to the ***<SYBDIR>*** directory.
2. Copy the ***<servername>.bak*** file to ***<servername>.cfg.***

- The <*servername*>*.cfg* file (which you are overwriting) contains the current values that are causing the problem.
- The <*servername*>*.bak* file (which you copy to <*servername*>*.cfg)* contains the last set of values with which the Sybase server booted successfully.

  These values were written to the *.bak* file when you made the configuration change that is preventing the Sybase server from starting up.

Resetting Parameters to Default Values

To set all the parameters back to default values:

- Either remove or rename the *<servername>.cfg* file in *<SYBDIR>* and restart the server.

  This will set the configuration back to Sybase default values, which may differ from those that resulted from installing the Sybase server as part of StP.

File Descriptors (UNIX Only)

File descriptors are made available by the operating system. To display the per-process limit for file descriptors you have configured in your operating system kernel, type the following command at the operating system prompt.

C shell:

```
% limit descriptors
```

Bourne shell:

```
$ ulimit -n
```

ASE is a single process. From the file descriptors allocated to it by the operating system, it uses one file descriptor for:

- Each master network listener (one for every “master” line in the interfaces file entry for that Adaptive Server)
- Standard output
- The errorlog file
- Each data device, including the master device and mirrored devices
- Each active site handler (including Backup Server)
- Each user connection
- Each online engine

To increase the number of descriptors the Adaptive Server process takes when it starts:

- Edit the runserver file by adding the following line after the comments but before the dataserver line:

  ```
  ulimit -S<n> <new_value>
  ```

  where <n> is the current value and <new_value> is what you want to change it to. The runserver file is a Bourne shell script.

The number of user connections ASE is configured for can never exceed the number of file descriptors available to an ASE process from the operating system. The number of currently available file descriptors is stored in the global variable @@max_connections.

You can report the maximum number of available file descriptors in your server, after logging in through **sybisql,** with this command:

```
1> select @@max_connections
2> go
```

Adding a Device

The StP install process creates the *stp_device* device for StP repositories. You can add more devices when *stp_device* is filled.

Each database device must be configured for the Sybase server before it can be used for database storage. This process is called initialization, and you perform it by using the Sybase **disk init** command. After you create a device, use the *syscreate_device* ToolInfo variable to control device placement for StP systems. For more information, see the discussion on Sybase database placement in *StP Administration*.

UNIX—You must reinitialize the repository manager with the **repos_maint** QRL function after adding a device with **disk init** (see "Initializing a Device with isql" on page 2-24).

After a database device has been initialized, it can be:

- Allocated to the pool of space available for StP repositories
- Used to store a database's transaction logs
- Designated as a default device for new and expanded systems

Initializing a Device with Sybase Central (Windows NT)

For ASE, to initialize a device:

1. Start ASE and log onto the server (see “Starting Adaptive Server Enterprise from Sybase Central” on page 2-7.
2. Select the server that needs the device.
3. Double-click the **Database Devices** folder for that server.
4. Double-click the **Add Database Device** wizard, and follow the instructions. Specify the database device name, its path, device number (Sybase Central displays the next available one), and size. You can specify to mirror the device if you want.
5. Click **Finish**.

Initializing a Device with *isql*

The Sybase system administrator executes the **disk init** command from **isql** in the *master* database. The **disk init** command names a physical device and formats it for storage of databases or transaction logs. This command:

- Maps the specified physical disk device or operating system file to a logical device name
- Prepares the device for database storage
- Adds a row to the *sysdevices* system table in the *master* database. The *sysdevices* table maps the Sybase server logical names to the physical device names

Syntax and examples of the **disk init** command are provided here for your convenience. For additional information on **disk init**, see the Sybase ASE or SQL Server documentation, which is available at their website at *http://www.sybase.com*.

The syntax for **disk init** is:

```
disk init
name = "<device_name>",
physname = "<physical_name>",
vdevno = <virtual_device_number>,
size = <number_of_blocks>
```

Table 3 describes the arguments for **disk init**.

Table 3: Arguments for disk init

Parameter	Description
<device_name>	The logical name that you assign to the device, used in all references to this device from the Sybase server
<physical_name>	The name of a raw partition or disk file (the disk file cannot already exist)
<virtual_device_number>	A unique identifier for the device, which can be any number from 1 to the number configured for the server, within a limit of 255 (see “Specifying a Virtual Device Number,” following this table for more information)
<number_of_blocks>	The size of the device in 2-K pages: 1 page = 2 K 1 MB = 512 pages

Specifying a Virtual Device Number

To specify virtual device number for **disk init**, you need to know how many devices have been configured for the server and which numbers are already in use. If the server has been configured for 10 devices, device numbers 0 through 9 will be in use.

Use the **sp_configure** system procedure with the **devices** option to find the number of devices for which the Adaptive Server has been configured.

To list device numbers already in use:

```
1> select distinct low/16777216 from sysdevices
2> order by low
3> go
```

You can also use the **sp_helpdevice** command and look at the values for device_number.

Sybisql Device Initialization Example (UNIX)

The following example creates a device of 200 MB named *stpdev* on a physical device named */db/sybase_databases/stp_device.dat*. The system administrator must have write permission on the physical file */db/sybase_sunos/stp_device.dat*.

```
1> use master
2> go
1> disk init
2> name = "stpdev",
3> physname = "/db/sybase_sunos/stp_device.dat",
4> vdevno = 4,
5> size = 102400
6> go
```

Designating Default Devices

Once you have initialized a device (see “Adding a Device” on page 2-23), you should designate it as a default device. You can designate multiple default devices for use by StP; when one device is filled, Sybase will use the next available one listed in the *sysdevices* system table. Unless another device is specified, StP creates the repository and transaction log for a new system on any default device that has space available. For more information about creating repositories on default and specified devices, see *StP Administration*.

Designating a Default Device with Sybase Central (Windows NT)

For ASE, to specify a device as the default:

1. Start ASE and log onto the server (see “Starting Adaptive Server Enterprise from Sybase Central” on page 2-7.

2. Select the server that has the device you need.
3. Double-click the **Database Devices** folder for that server.
4. Double-click the device name.
5. In the **Properties** dialog box, select the **Parameters** tab, and then select the **Default device** option.
6. Click **OK**.

Designating a Default Device in *isql*

To designate a default device, using **sp_diskdefault**:

```
1> sp_diskdefault <device_name>, defaulton
2> go
```

Removing a Device

You may need to remove a device's default designation, or remove the device entirely.

Removing a Device's Default Designation

You may want to remove a device's default designation if:

- You want to reserve the device for a particular system
- You are preparing to remove the device and do not want users to create repositories on it

Removing a Device's Default Designation in Sybase Central (Windows NT)

For ASE, to remove a device from the default:

1. Start ASE and log onto the server (see "Starting Adaptive Server Enterprise from Sybase Central" on page 2-7.
2. Select the server that has the device.
3. Double-click the **Database Devices** folder for that server.

4. Double-click the device name.
5. In the **Properties** dialog box, select the **Parameters** tab, and then deselect the **Default device** option.
6. Click **OK**.

Removing a Device's Default Designation in isql

In **isql**, you can remove a device's default designation with the following command:

```
1> sp_diskdefault <device_name>, defaultoff
2> go
```

Removing a Device with Sybase Central (Windows NT)

For ASE, first determine if the device has any databases on it:

1. Start ASE and log onto the server (see "Starting Adaptive Server Enterprise from Sybase Central" on page 2-7.
2. Select the server that has the device to remove.
3. Double-click the **Database Devices** folder for that server.
4. Right-click on the name of the device to remove, and from the menu, choose **Properties**.
5. In the **Properties** dialog box, select the **Databases** tab.
6. If there are any databases on that device, you need to delete the database.
7. Click **OK**.

To delete the device:

1. Double-click the **Database Devices** folder.
2. Right-click on the device to delete, and from the menu, choose **Delete**.

Removing a Device with *isql*

You cannot remove a device from the Sybase server if there are any databases on it. Before attempting to remove a device, determine the identity of any databases that have fragments on it:

```
1> select distinct dbid from sysdevices, sysusages
2> where vstart between low and high
3> and name = "<device_name>"
4> go
```

To determine the name of each database, for each dbid returned by the above query:

```
1> select name, dbid from sysdatabases
2> where dbid between <low_dbid> and <high_dbid>
3> go
```

If you would like to sort the output by dbid, you could enter:

```
1> select name, dbid from sysdatabases
2> where dbid between <low_dbid> and <high_dbid>
3> order by dbid
4> go
```

If appropriate, you can remove an obsolete database either from the StP Desktop (see the section on the **Destroy System Repository** command in *StP Administration*) or with the following **isql** command:

```
1> drop database <database_name>
2> go
```

Once all the databases that had fragments on that device have been removed, you can remove the device from Adaptive Server:

```
1> sp_dropdevice <device_name>
2> go
```

Managing the Transaction Log

Every database on the server has a transaction log that records each update to the database. ASE uses the transaction log to recover data automatically following a system failure. The transaction log is actually a system table named *syslogs*, which exists in every database. StP does not make use of the Sybase transaction log, since you can regenerate repositories from the ASCII files in your project/system directories. There is no way to turn off logging, so you will need to perform periodic administrative tasks, such as dumping or expanding the transaction log.

The transaction log can be stored on the same device as the repository data or on a separate device. The transaction log device is specified by the *syscreate_log_device* ToolInfo variable.

Dumping the Transaction Log

Transaction logs can become very large in a heavily updated repository. The **Perform Manager Maintenance** StP Desktop command or the **r**epos_maint QRL function dumps the transaction log as part of routine maintenance (for details, see *StP Administration* and *Query and Reporting System*). Running either of these frequently is the simplest way of ensuring that the transaction log does not grow too large. These commands require that the transaction log be able to record the transaction of dumping the log.

If the transaction log is too full to record the dump of itself, you can truncate it from **isql** using the **dump transaction** command with the **no_log** option. The following command truncates the transaction log for the *sys1* database:

```
1> dump transaction sys1 with no_log
2> go
```

Expanding the Log

All transactions are written to the transaction log before being committed to the database tables. Expanding the transaction log is necessary when issuing a batch of transactions that require more space than is available in the transaction log. An example is loading an ASCII dump of a large database when moving a system. If the number of transactions required to load an ASCII dump exceeds the amount of space available in the transaction log, expand the log and try the load again.

Expanding the Log with Sybase Central (Windows NT)

For ASE, to expand the transaction log:

1. Start ASE and log onto the server (see “Starting Adaptive Server Enterprise from Sybase Central” on page 2-7.
2. Double-click the **Databases** folder.
3. Right-click the database whose log you want to expand, and from the menu, choose **Properties**.
4. Click the **Devices** tab.
5. From the **Name** menu, choose the name of the device on which the new database allocation is to occur, and click **Edit**.
6. In the **Device Size** dialog box, increase the space in the **Add space** field.
7. Click **OK**.
8. In the **Properties** dialog box, click **OK**.

Expanding the Log with isql

To expand the transaction log from **isql**, use the **alter database** command. The following example adds 3 MB to the database’s transaction log on the default device:

```
1> use master
2> go
1> alter database <database_name>
2> log on default=3
3> with override
4> go
```

Accessing a Remote Sybase Server

You can install StP on a different machine from the machine on which the Sybase server is installed, and access that server over a network. You can also configure StP and the Sybase server to run on different platforms.

Client-Server Architecture

StP is a client application to an ASE application. In a client-server environment, client applications request service from server applications, which process the requests and return the appropriate information to the clients for their use. Network-based clients and servers can be on the same or separate hosts. For a client and server to communicate, the client must know where the server is on the network.

Interfaces File

The interfaces file contains a list of ASEs with connection information that uniquely identifies each one on a network. When a Sybase server is installed, an interfaces file is created in the *ini* directory of the Sybase home directory. It contains information about the connection the Sybase server listens on, as well as connection information for clients querying the server. StP clients must include the same connection information in their interfaces file. An StP client on UNIX looks at a file named *interfaces*. An StP client on Windows NT looks at a file named *sql.ini*.

Adding a Sybase Server to the ASE *sql.ini* File

The ASE interfaces file on Windows NT is named *sql.ini*. It resides in the Sybase *ini* directory, usually *<SYBDIR>/ini/sql.ini*. Do not edit this file manually; instead, use **Dsedit**.

To add a new server entry to *sql.ini*:

1. Click the **Start** button.
2. Choose **Programs > Software through Pictures > Sybase Repository > Edit Server Definitions**.

3. In the **Select Directory Service** dialog box, select InterfacesDriver and click **OK**.
4. From the **Server Object** menu, choose **Add**.
5. In the **Input Server Name** dialog box, enter the name of the server.
6. In the **Server** area, select the name of the new server and in the **Attributes** area, select **Server Address**.
7. From the **Server Object** menu, select **Modify Attribute**.
8. In the **Network Address Attribute** dialog box, select **Add**.
9. From the **Protocol** list, select **NLWNSCK**.
10. In the **Network Address** box, enter the server name, followed by a comma and a unique socket number, for example:

    ```
    hawk,6438
    ```

    Valid numbers are in the range of 1025 to 65535. By default, the first Sybase server you add to the interfaces file uses 5000.
11. Click **OK**.
12. In the **Network Address Attribute** dialog box, click **OK**.
13. To exit Dsedit, from the **File** menu, choose **Exit**.

Adding a Sybase Server to the Interfaces file in UNIX

1. From the **“Select a directory service to open**” dialog, **select** “Sybase interfaces file” and **click** OK.
2. A session dialog will appear prompting you to select an action. **Click** the “**Add new server entry**” button. The **Server Entry Editor** will appear.
3. **Enter** the name of the server in the **Server Name** field.
4. Click on the **“Add new network transport**” button. The **Network Transport Editor** will appear.
5. Select the Transport type. This is generally “**tcp**”. On Solaris, it’s “**tli tcp**”.
6. In the **Host Name** field, **enter** the **host name** of the machine where the server is running
7. In the **Port Number** field, **enter** the **port number** for the server.
8. **Click OK** to exit.

Environment Variables

Each machine running StP must also have at least the Sybase ASE/Open Client software installed. For StP to connect to the Sybase server, there are a few environment variables that need to be properly set. Generally, these variables are set automatically when StP and ASE are installed on the same machine. However, if you want to connect to a Sybase server on a separate machine, StP has to know how to find it. For each machine, make sure the following environment variables have been set:

- DSQUERY—Set DSQUERY to point to the server used for StP systems. DSQUERY must be set before StP users can perform most server maintenance functions. However, once an StP system has been created, the server the repository was created on is recorded in that system's *.repinfo* file. At this point, connecting to that system does not require that DSQUERY be set.
- SYBASE—Set this environment variable so that StP can find the location of the file that lists the available Sybase servers.
- PATH—Set PATH (or path) to locate the executables to which StP needs to connect.

For more information on setting environment and ToolInfo variables for system administration purposes, see the discussion on user environments in *StP Administration*.

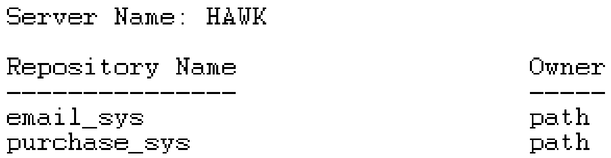
Confirming the Server to Which StP is Connected

In a multi-server environment it may not be clear which server you are logged into.

To confirm which server StP is connected to:

1. Start StP.
2. From the **Repository** menu, choose **List Repositories**.

   The **List Repositories** window appears, listing repository names and owners, for example:



```
Server Name: HAWK

Repository Name                    Owner
---------------                    -----
email_sys                          path
purchase_sys                       path
```

Dedicated Sybase Server Operation (Windows NT)

You can install the Sybase server for Windows NT on a PC that is:

- Also used as a file server
- Dedicated to the Sybase server application

Installing the Sybase server on a dedicated PC greatly improves performance, because the software does not have to share system resources with file and print serving applications. If you install the Sybase server on a dedicated PC, set the default Windows NT tasking option to give the foreground application (in this case, the Sybase server) the best application response time.

Do not perform this procedure on a PC that doubles as a file or print server. Doing so will degrade the performance of all other applications that run on the PC.

To install the Sybase server on a dedicated PC:

1. Click the **Start** button.
2. Choose **Settings > Control Panel**.
3. Double-click the System icon and click the **Performance** tab.
4. Set the performance boost for the foreground application to **Maximum**.
5. Click **OK**.

Using the Sybase Performance Option

If you install the Sybase server on a dedicated PC, you can also use **-P** command line option to improve server performance. The **-P** option tells the Sybase server not to surrender system resources to other applications as much as it normally would. This option serves as a safeguard to maintain high performance even if users log into or start applications on the dedicated server.

To use the **-P** option:

1. Click **Start** and choose **Programs**.
2. Start Server Config as follows:
   - Choose the **Software through Pictures** program group, and then choose **Sybase Repository > Sybase Central**.
   - Click on the Utilities folder and choose **Server Config**.
3. Choose the server you want to configure and click **Continue**. When prompted, enter the sa login and password and click **Continue** again.
4. In the **Configure** dialog box, choose **Command Line**.
5. Enter **-P** in the **Command Line Parameters** box and click **OK**.
6. Choose **Save**.
7. Exit Server Config and restart the Sybase server.

Maintaining the Sybase Server

There are a few maintenance tasks that you should perform regularly (on UNIX, you can include them in a nightly cron job). You can use the **Perform Manager Maintenance** StP Desktop command or the repos_maint QRL function to:

- Dump transaction logs for all StP databases in the Sybase server
- Update statistics about the distribution of key values in specified indexes on all StP databases to optimize query and sorting procedures
- Drop any StP-created temporary tables in *tempdb* that are no longer in use

For more information on these commands, see *StP Administration* and *Query and Reporting System*.

In addition to performing these maintenance activities, you should regularly make backups of StP repositories and their flat files. For more information, see the discussion of backup and recovery in *StP Administration*.

You also need to make periodic backups of the *master* database. There are no StP Desktop commands that enable you to dump the *master* database to files in the same manner that you can dump repositories to files. To back up the *master* database, you can use either **isql**, or Sybase Central. The Backup Server must be installed to back up the *master* database (see "Installing and Starting Backup Server," which follows).

Installing and Starting Backup Server

The Backup Server backs up and restores the Sybase server databases. The Backup Server should be installed and configured on the same machine as the Sybase server. Both the Sybase server, and Backup Server must be running in order to back up the *master* database.

On UNIX, you need to install Backup Server separately from the Sybase server. On Windows NT, Backup Server is installed by default when you install the Sybase server.

Installing Backup Server (UNIX)

To install Backup Server on UNIX:

1. Set the SYBASE environment variable to the location of the *sybase_12.0.0/<platform>* directory.
2. Set the **SYBASE_ASE** environment variable to **ASE-12_0**
3. Set the **SYBASE_OCS** environment variable to **OCS-12_0**.
4. Change to the *$SYBASE/$SYBASE_ASE/bin* directory. Execute **srvbuild**.
5. In the "**Select Servers to Build**" window, check the box for **"Backup Server**" and enter the name **STP_SERVER_12_BCK**. Click **OK.**
6. In the "Server Attribute Editor" enter the appropriate information into the field with an *.

7. Click on the **“Build Server!**” button.

Starting Backup Server (UNIX)

To start the backup server, follow the instructions in “Starting ASE Manually” on page 2-3.

Starting Backup Server (Windows NT)

Backup Server has the same name as the Sybase server with the extension _BS.

On ASE, to start Backup Server:

1. Start ASE and log onto the server (see “Starting Adaptive Server Enterprise from Sybase Central” on page 2-7.
2. Under the **Sybase Adaptive Server Enterprise** icon, select the Backup Server to start.
3. In the **Sybase Adaptive Server Enterprise Logon** dialog box, specify your user name and password, and the server you want.
4. Click **OK**.

Backing up the *master* Database

The *master* database contains information about the size and location of all the databases in the server. It is important to back up the *master* database often, on a regular basis. If you do not, you may have to reconstruct vital data, such as system tables, if *master* becomes corrupted. Bear in mind that while you can limit the creation of new objects in *master*, that certain system procedures, such as **sp_addlogin**, modify system tables.

The Sybase documentation recommends that you back up *master* after you use the following ASE commands:

- **disk init**, **sp_addumpdevice**, or **sp_dropdevice**
- Disk mirroring commands
- The segment system procedure **sp_addsegment**, **sp_dropsegment**, or **sp_extendsegment**

- **create procedure** or **drop procedure**
- **sp_logdevice**
- **sp_configure**
- **create database** (or, for StP, the **File > New > System** command)
- **alter database** (or for StP, the **Expand Current System Repository** command)

Furthermore, you should save any scripts that contain the **disk init**, **create database**, and **alter database** commands. When you back up the *master* database, you should also print the output from the following commands and save the hard copy in a safe place to aid in restoring your server:

```
select * from sysdatabases
select * from sysusages
select * from sysdevices
select * from syslogins
```

The **buildmaster** utility cannot recover changes that have occurred from these commands. If you keep your scripts using these commands, you can use them to recreate the changes.

The following methods back up the *master* database to a file. For more complete instructions on Backup Server, or for instructions on backing up the *master* database to tape, refer to the Sybase documentation.

Using the dump database Command in isql

With both the Sybase server and Backup Server running, log into the Sybase server through **isql** and issue the following command, where <device_name> is the path and name of the device to which you are dumping the database:

```
1> dump database master to "<device_name>"
2> go
```

Backing Up the master Database with Sybase Central (Windows NT)

For ASE, to back up the *master* database:

1. Start ASE and log onto the server (see “Starting Adaptive Server Enterprise from Sybase Central” on page 2-7.
2. Start both the ASE and Backup Server you want: right-click on the server name, and from the menu, choose **Start**.
3. Double click the Databases folder.
4. Right-click on **master** and from the menu, choose **Backup**.
5. In the **Create a backup command** dialog box, make sure that **master** is in the list box, and click **Next**.
6. In the **Select type of backup** dialog box, select the **Backup the entire database** option.
7. Click **Next**.
8. In the **Select dump devices** dialog box, click Add, and select a dump device.
9. Click **Next**.
10. In the **Select backup name** dialog box, specify a name for the backup, and then click **Next**.
11. In the **Ready to execute backup command** dialog box, click **Finish**.

Restoring the *master* Database

If your *master* database becomes corrupted or if you lose the *master* device, you can restore it from your dump file. Contact Aonix Technical Support for assistance.

Error Log

ASE maintains an error log that contains:

- Boot information when the server is started up
- Recovery information if the server is restarting after a system failure
- Backtraces on queries that generated fatal errors
- Error messages from the kernel

UNIX—The *errorlog* file is normally in the ASE install directory on the machine where ASE is running and is specified in the runserver file. If the runserver file does not specify the location of the error log, the log is created in the directory that was the current directory from which the server was started.

Windows NT—The *errorlog* file is normally located in the ASE install directory on the machine where the Sybase server is running. Although there is a *RUN_<servername>.bat* file in the Sybase install directory, this is not a true runserver file; it was created as an intermediate step during the installation process. The comparable values to a UNIX runserver file, including a specification for the location of the *errorlog*, are held in the Registry and should not be edited except by an experienced Windows NT administrator.

You can reduce the size of the error log by deleting unneeded messages. You can also remove the error log or archive it. The server must be shut down before you edit, remove, or archive the error log.

If you remove or archive the error log, a new error log is created when the server is restarted.

To create a machine-readable archive of the error log:

1. Shut down the server, according to the instructions in "Shutting Down the Server" on page 2-41.
2. Archive the error log.

   For UNIX, you can use the **ar** utility.
3. Restart the server.

Shutting Down the Server

You may need to shut down the Sybase server in preparation for rebooting the machine or to perform maintenance. When the server is shut down:

- All logins are disabled except that of sa.
- All users, except sa, are prevented from initiating database commands.

- Pages that have been updated since the last checkpoint are flushed from cache to disk.
- Currently executing SQL commands or stored procedures are allowed to finish.
- The the Sybase server process exits.

You should always use one of the following methods to shut down the server to avoid loss of data and possible corruption.

Shutting Down a Server with Sybase Central (Windows NT)

For ASE, to shut down the server:

1. Start ASE and log onto the server (see “Starting Adaptive Server Enterprise from Sybase Central” on page 2-7.
2. Right-click on the server to shut down, and from the menu, choose Stop.

Shutting Down a Server with isql

Before shutting down the server, you may want to check to see who is using it. You can use the **sp_who** system procedure in **sybiqsl** to do so:

```
1> sp_who
2> go
```

To shut down the server:

```
1> shutdown
2> go
```

If there are processes waiting to complete and you need to shut down the server immediately, you can invoke **shutdown** with the **nowait** option. Only use the **nowait** option in an emergency that requires the server to be shut down immediately.

```
1> use <repository_name>
2> go
1> checkpoint
2> go
1> shutdown with nowait
2> go
```

The **checkpoint** command writes uncommitted data to disk. After you perform the **shutdown**, the Sybase **dataserver** (UNIX) or **sqlsrvr** (Windows NT) process exits immediately. If you do not use the **checkpoint** command beforehand, the server does not wait for currently executing commands to complete and does not flush the updated pages in every database. In this case, there is no guarantee that all changes have been written to disk, so data can be lost. Normally, however, the uncommitted data will be committed the next time you start the server or perform a recovery operation.

Removing a Sybase Server

Before you remove a server, you need to also remove any remote users associated with that server. You can use the **droplogins** Sybase command to remove the remote users. If you are using ASE, the checks against the *sysattributes* system table apply when Component Integration Services is configured.

Removing a Server (UNIX)

To remove an Adaptive Server:

1. Remove the runserver file from the ASE install directory.
2. Remove the error log from the ASE install directory.

3. If you are using a UNIX file type device, remove the *master.dat* file from the Adaptive Server databases directory.

   If you are using a raw disk partition, remove the *master.dat* file from the partition.
4. Remove the *sybprocs.dat* file.
5. Edit the *interfaces* file in the ASE Server home directory, removing the lines that contain the name and connection information for that server. See "Interfaces File" on page 2-32 for more information about this file.

Removing a Server with Server Config (Windows NT)

Server Config drops remote logins as well as the server you want to remove.

To remove a server:

1. Shut down the server you want to remove.

   See "Shutting Down the Server" on page 2-41 for more information.
2. Click **Start** and choose **Programs**.
3. Start Server Config as follows:
   - Choose the **Software through Pictures** program group, and then choose **Sybase Repository** > **Sybase Central**.

     Click on the Utilities folder and choose **Server Config**.
4. Select the server type to remove (ASE Server, Backup Server, or Monitor Server).
5. Select **Remove Adaptive Server**.
6. In the **Existing Servers** list, choose the server you want to remove.
7. In the confirmation dialog box, click **Yes**.
8. In the **syconfig** dialog box, click **OK**.
9. In the **Configure Sybase Servers**, click **Exit**.
10. Restart your computer.

Removing a Server with isql

To remove a server named DIMADOZEN if it has no remote logins:

```
1> sp_dropserver DIMADOZEN
2> go
```

In **isql**, to remove the same server that has remote logins:

```
1> sp_dropserver DIMADOZEN, droplogins
2> go
```

Removing a Sybase ASE or Open Client Installation

If, during the StP installation procedure, the installation of the Sybase ASE or Open Client fails, the partially executed Sybase installation may leave files, as well as entries in the Windows NT Registry, that would prevent the success of a subsequent installation. Therefore, if the Sybase installation fails for any reason, you should remove it before attempting the installation again.

Removing ASE (Windows NT)

To remove an ASE or Open Client installation:

1. Make sure that ASE or Open Client, Backup Server and / or Monitor Server are no longer running, then reboot Windows NT to release any Sybase dlls.
2. From the **Start** menu, choose **Settings** > **Control Panel**.
3. Click on the Add / Remove Icon.
4. On the **Add/Remove Programs Properties** dialog box, select the Install / Uninstall tab.
5. Select **StP Sybase Components** from the list.
6. Click **Add/Remove**.
7. After each uninstall (if more than one) has completed, click **OK**.

8. If necessary, remove any Sybase directories that may not be needed.

3 Troubleshooting a Sybase Server

This chapter describes various troubleshooting procedures for determining possible Adaptive Server Enterprise (ASE) problems affecting StP operation. Topics covered are as follows:

Reading the Error Log

The ASE *errorlog* in the Adaptive Server install directory contains valuable information about the state of the server.

It includes:

- The boot sequence which shows the server’s devices and databases and whether the latter recovered successfully or not.
- Error messages and stack traces, which can help determine the source of the problem.

Note: If it is necessary to send error log information to Technical Support, send only the information from the point of the last good boot sequence to the end of the error log.

Determining the Sybase Server Version

To determine the version of the Sybase server:

1. Log into the Sybase server as user sa, using **isql.**
2. Enter the following command:

   ```
   1> select @@version
   2> go
   ```

   The output is similar to the following:

```
Adaptive Server Enterprise/12.0/P/SWR 8776 ESD 1/NT (IX86)/OS 4.0/1580/
32bit/OPT/Mon Dec 06 21:50:07 1999
```

In this example, the server’s version is 12.0 and it is running on Windows NT.

Note: Sometimes the output includes “SWR,” also called EBF (Emergency Bug Fix). EBFs are patches Sybase makes available periodically. If a bug that impacts the functioning of StP has been fixed, Aonix makes the EBF available to customers on our ftp site.

For Adaptive Server running on Windows NT, you can determine your server's version with Sybase Central:

1. Start ASE and log onto the server (see "Starting Adaptive Server Enterprise from Sybase Central" on page 2-7.
2. Right-click on the server name, and from the menu, choose **Properties**.

If you cannot start the Sybase server, you can find out the current version by going to the *bin* subdirectory in the Sybase home directory and entering the following command:

For UNIX:

```
dataserver -v
```

For Windows NT:

```
sqlsrvr -v
```

The version of the Sybase server executable is displayed.

If the Environment Is Not Configured Correctly

If the Sybase server you are attempting to connect to has not been set in the interfaces file, the Message Log displays:

```
Error  : 10005: Unable to locate server in Sybase interfaces
file
Error  : 1009: Unable to initialize repository manager
Error  : Repository open failed
```

- For **UNIX**—

  Run **stp -debug** to check the values of DSQUERY and SYBASE.

  If your environment is either non-secure or partially secure, verify that the value of <server>_PASSWD is the same as the sa password.

- **Windows NT**—

  Verify that your shell environment is correctly set up to run StP and access the Sybase server by checking the variables in Control Panel / System / Environment: DSQUERY, SYBASE and Path.

  See "Environment Variables" on page 2-34 for more information.

If the Adaptive Server Does Not Start

Possible causes for the Adaptive Server not starting include:

- The *master* disk device is offline.
- The *master* database is damaged.
- Disk drives are powered off.
- Insufficient resources (such as memory and swap space).
- Use of illegal raw partitions (UNIX only).
- Badly configured values from **sp_configure**, such as more memory required than available with the configured value. See "Recovering from Badly Configured Values" on page 2-21.
- Inappropriate use of **-P** flag, described in "Dedicated Sybase Server Operation (Windows NT)" on page 2-35" (Windows NT only).
- The shared memory file still exists, although the server was shut down (look for <*servername*>.*krg* in the Sybase directory; if it exists, remove it).
- The server is already running. See "Determining if the Sybase Server is Running" on page 2-2.

Verifying the Server Process

To verify that an ASE Server process is running:

- For **UNIX**—

  Issue the **showserver** command from the Sybase install directory.

  If the server is running, the **dataserver** process appears in the command output.

- For **Windows NT**—

  Look for the Sybase server process, which is the executable **sqlsrvr.exe**, in the Windows NT task manager.

  If the Sybase server is already running, any attempt to start it with Sybase Central or the Services Manager will fail with a red light.

  You also can try entering **`sqlsrvr`** at a command prompt; if the Sybase server is running, an error message will let you know.

If the Adaptive Server Hangs

If you can do no work in StP and the editors seem frozen, it may be that the Adaptive Server is hung. Possible causes are:

- The Sybase server error log is full
- The transaction log is full
- The connection to the Sybase is not available

The Adaptive Server Error Log Is Full

If the error log is too full to be written to, the Adaptive Server hangs. You need to:

1. Shut down the server. (See “Shutting Down the Server” on page 2-41.)
2. Archive the error log.
3. Start the server. (See “Starting the Sybase Server (Windows NT)” on page 2-7.)

   A new log is created.

Transaction Log Is Full

If the transaction log is too full to be written to, the Adaptive Server hangs. You need to:

1. Check the error log for 1105 errors and note which database is mentioned in the error message.
2. In StP, perform repository maintanence on the server.
3. Log into the server and issue a dump transaction command (see “Dumping the Transaction Log” on page 2-30).

Connection to the Sybase Server Is Not Available

If the Sybase server has run out of user connections, the following message appears in the Message Log when you try to open a system.

For UNIX:

```
Maximum number of Sybase connections already allocated
10009: Can’t connect to Sybase (too many users?)
1009: Unable to initialize repository manager
Repository open failed
```

Additionally, error message 1601 usually appears in the Sybase server error log.

Two ways to clear this error are:

1. Free up connections to the Adaptive server by making sure users close unused StP editors.
2. Reconfigure the Adaptive server with more connections. To do so, use the **number of user configurations** configuration parameter. You can check the current number of connections as follows:

   ```
   1> sp_configure "number of user connections"
   2> go
   ```

   To configure for 35 connections, you would enter:

   ```
   1> sp_configure "number of user connections", 35
   2> go
   ```

   To configure remote connections, use the **number of user connections** configuration parameter.

For NT, use Sybase Central. For more information, see “Configuring User Connections” on page 2-18.

If StP Cannot Access the Adaptive Server

Error messages produced when you cannot access the Adaptive server include:

```
Invalid login/password
Not a valid login/password
```

Some possible explanations and remedies for StP's inability to access the Sybase server may be:

- The network is not configured
- The Sybase server is not accessible
- StP cannot locate the Sybase server
- The user has not been added as a repository manager user

Verifying the Network

- For **UNIX—**

  Use the shell command **ping** <*machine_name*> on the StP machine to elicit a response from the machine running the Adaptive server.

  Even if StP and the Adaptive server are on the same machine, if the network is configured properly, **ping** reports either "Reply from <ip address>" or responds that the Sybase server machine "is alive."
- **Windows NT—**

  Use the **ping** feature in Sybase Central to test the network connection. If the connection is not successful, there may be more than one copy of the **netlib.dll** on the StP machine.

  Search for *NLWNSCK.DLL* to confirm that there is only one.

To test the network connection for **ASE:**

1. From the **Sybase Repository** program group, choose **Sybase Central**.
2. In the **Sybase Central** dialog box, click on the Utilities folder.
3. Double click on Directory Services Editor.
4. In the **Select Directory Service** dialog box, select the interfaces file for the server to test.

5. In the **DSEDIT - Interfaces Driver** dialog box, select the name of the server to test in the **Server** area.
6. From the **Server Object** menu, choose **Ping Server**.
7. If the **Ping** dialog box appears, select the server address, and click **Ping**.

The Adaptive Server Is Not Accessible

If the Adaptive server is inaccessible, it may be that the Adaptive server is not running or is hung. To test server accessibility, use one of the following methods.

Testing Server Accessibility with Sybase Central

To test server accessibility:

1. Click on the server name.
2. Enter the password.

Testing Server Accessibility with isql

To test server accessibility, log into the server with **isql.**

```
isql -U <user_name> -P <user_password> -S <server_name>
```

- If 1> is returned, chances are that the Sybase server is running and you have connected to it successfully. To make sure, enter any SQL command, for example:

```
1> select * from sysusages
2> go
```

- If the command executes successfully (in this case, showing the column names for the *sysusages* system table), the connection is solid. If not, **isql** crashes and returns you to the command prompt.

The Sybase server may not be running if **isql** returns the following information:

```
Operating-system error:Connection refused
DB-LIBRARY error:Unable to connect: Adaptive Server is
unavailable or does not exist
```

StP Cannot Locate the Adaptive Server

There are a number of reasons why StP might not be able to locate the Adaptive Server:

- The **hostid** has been changed (either the IP address or the adapter) on the machine where the Sybase server resides and the *interfaces* file has not been updated to reflect the change.
- The environment variables **DSQUERY** and **SYBASE** are not set correctly.

  UNIX—Verify that the **SYBASE** environment variable is initialized to the path of the directory where the *interfaces* file resides and that the name of the server in the *interfaces* file matches exactly what **DSQUERY** is set to.

  Windows NT—Verify that the **SYBASE** environment variable is set to the Sybase home directory and that the name of the server in the *SQL.INI* file matches exactly what **DSQUERY** is set to.
- The *interfaces* file has been corrupted or incorrectly edited by hand.

The following messages are produced when **DSQUERY** is set incorrectly or the *.repinfo* file in the StP system directory is either missing or contains a bad entry for the server:

```
Error  : 1009: Unable to initialize repository manager
Error  : Repository open failed
```

The following messages are produced when **SYBASE** is not set correctly:

```
Error  : 10005: Unable to locate server in Sybase interfaces
file
Error  : 1009: Unable to initialize repository manager
Error  : Repository open failed
```

The following error messages are produced when Sybase cannot find or interpret the connection information in the *interfaces* file:

```
Operating-system error:
   There is no OS level error
DB-LIBRARY error:
Specified server name attribute could not be found
```

For more information, see the discussion on troubleshooting network connections in the following Sybase manuals, available from the Sybase website at *http://www.sybase.com*:

- For Adaptive Server, *Configuring Adaptive Server Enterprise for Windows NT*

Checking for Valid Users

If the following error messages are produced, check that the user is a valid user of the ASE Server:

```
Invalid login/password
Not a valid login/password
```

To check for valid users,:

- From the StP Desktop, choose **Repository** > **Manage Users** > **Repository Manager** > **List All Users** (for details, see *StP Administration*).

To add valid users of the Adaptive server:

- From the StP Desktop, choose **Repository > Manage Users > Repository Manager > Add Sybase Users**. (For details, see *StP Administration*.)

Adding valid users to an StP repository can be done only by the Sybase administrator (user sa) or by the repository owner.

If Environment and ToolInfo Variables Clash

Even though the ToolInfo variable settings are correct, there may be StP environment variables that are inconsistent with their ToolInfo variable equivalents.

The StP environment variables override the ToolInfo variable settings. It is better, however, to control user settings through ToolInfo variables because it is easier to manage these settings from a central location, the ToolInfo file. Furthermore, StP may not support the StP environment variables in future releases. To be safe, you should disable any StP-specific environment variables. (***Do not*** disable Sybase environment variables, however.)

For details, refer to *StP Administration* for appropriate documentation on ToolInfo variables.

If the User Cannot Access an StP Repository

Error messages reporting the failure to open a repository generally fall into three categories:

- Problem with the *.repinfo* file
- Problem with the database or server
- Problem with the user account

Problem with *.repinfo* File

These messages are the result of the wrong server name in the *.repinfo* file:

```
10020: SQL server is unavailable or does not exist
1009: Unable to initialize repository manager
Repository open failed
```

Problem with the Database or Server

These messages are the result of too few user connections in the server:

```
Error  : 1009: Unable to initialize repository manager
Error  : Repository open failed
```

Windows NT—These messages are the result of an StP user on Windows NT trying to open the project and system whose repository is in a UNIX Sybase server in which character set CP850 has not been installed:

```
Message  : Project/System: Matching pattern '*' in directory
'C:/StP/StP_V7.0/Examples/'
Message  : 9 matching directories, 2 containing StP systems
Error  : Msg 2409, Level 11, State 2
Error  : Server 'SS11',
Error  : Line 1
Error  :
        Cannot find the requested character set in
Syscharsets:  name = 'cp850'.
Error  : 10008: Unable to open current repository
Error  : 10006: Not a valid login/password
Error  : 1009: Unable to initialize repository manager
Error  : Repository open failed
```

These messages result from trying to open a repository that has been marked suspect, due to possible corruption:

```
Error  : Repository open failed
Error  : Msg 926, Level 14, State 1
Error  : Server 'SS11',
Error  : Line 1
Error  : Database 'test' cannot be opened. An earlier attempt
at recovery marked it 'suspect'. Check the ASE Server
errorlog for information as to the cause.
Error  : 1008: Unable to connect to system: test
Error  : 1009: Unable to initialize repository manager
Error  : Repository open failed
```

Problem with the User Account

These messages result if the user supplies the wrong password when prompted in a partially secure environment, in which the *<server>_password* ToolInfo variable is not set:

```
10008: Unable to open current repository
10006: Not a valid login/password
1009: Unable to initialize repository manager
Repository open failed
```

If the following error messages appear, check that the user is a valid user of the Sybase server (see “Checking for Valid Users” on page 3-10):

```
10007: You are not a valid user in the current system
1008: Unable to connect to repository:<database_name>
1009: Unable to initialize repository manager
```

Verifying Access to StP Repository

If you think you should be able to access an StP repository but cannot, you may want to verify your access privileges to the database from within **isql**.

```
1> use database_name
2> go
```

The following messages indicates that you have not been added as a valid user to the system repository:

```
Server user id <id> not a valid user in database
’<database_name>’.
```

If Creating a System Fails

If the creation of a system fails, whether from the StP Desktop **File > New > System** command or from the sys_create QRL function,any of the following error messages may be returned:

```
Repository creation failed for <database_name>
Repository <Database_name> exists on server <Server_name>
```

-or-

```
User "<user_name>" does not have the CREATE DATABASE
privilege
```

-or-

```
Not enough space in repository manager <Server_name> to
create system <System_name>
```

-or-

```
MODEL database in use: cannot create new database
```

Possible causes for an unsuccessful system creation include:

- User permissions are incorrect
- No space available for creating new system repositories
- Default device is not defined
- *model* database is in use
- There are problems with the Sybase server connection

See "If StP Cannot Access the Adaptive Server" on page 3-7 to troubleshoot the Sybase server connection problems.

User Permissions Incorrect

If permission to access the Sybase server is not set up correctly, you must correct the permissions.

Not A Valid User of the Adaptive Server

All users needing to read and write in the Adaptive server must be added as valid ASE users. See "Checking for Valid Users" on page 3-10 for instructions.

User Lacks System Repository Creation Privileges

ASE needs to be notified that you have privileges to create new StP repositories.

The StP administrator needs to grant the user system creation privileges, as described in *StP Administration*.

Not Enough Space on the Sybase Server Device

If the **File** > **New** > **System** command from the StP Desktop or the sys_create QRL function fails, try to create a database from within **isql**:

```
1> create database <test_name> on default=8
2> go
```

If **create database** fails, you either need to remove unused databases to free up space or you need to expand the Adaptive server by adding more devices (see “Adding a Device” on page 2-23).

If **create database** succeeds, there should be enough space in ASE to create an StP repository using the **sys_create** QRL function or the **File** > **New** > **System** StP Desktop command.

Be sure to remove the database <*test_name*> after you have verified that it can be created:

```
1> drop database <test_name>
2> go
```

Default Device Not Defined

If the **File** > **New** > **System** command from the StP Desktop or the sys_create QRL function fails, log into **isql** as the user attempting to create the StP repository and execute **create database**.

```
1> create database database_name on default = 6
2> log on default = 2 with override
3> go
resp=Msg 1808, Level 17, State 1:
Line 1:
Crdb_disk: Getnext SCAN_NOINDEX on sysdevices.status=DEFAULT
failed to find default rows
```

This message indicates that a default device is not configured on the Adaptive server.

When the server is created during the Adaptive install, the *stp_device* device is created and designated as a default device for the **create database** and **alter database** commands.

If an StP administrator has turned off all of the default device designations inadvertently with the **isql** command, then there will be no default device. Likewise, if all default devices have been dropped, there will be no default device (see “Removing a Device” on page 2-27 for more information).

To avoid this, either define an alternate device as a default device or set the ToolInfo variables *syscreate_device* and *syscreate_log_device* to valid device names (see *StP Administration*).

Model Database in Use

Only one database can be created at a time. If two users try to create a database at the same time, the database creation collides and one user gets the message:

```
MODEL database in use: cannot create new database
```

If Destroy System Repository Fails

If an StP repository is open and in use by the Adaptive server, it cannot be removed. Set your *project* and *system* to a different system name or use an invalid system name when setting *project* and *system* so the repository you want to destroy will no longer be in use. Make sure that all current users of the repository do the same.

If **Destroy System Repository** still fails, it may be that there is a sleeping process in the database. A user may have opened that repository and neglected to close it before leaving the premises for an extended period, or might have killed the StP process from the OS level. In either case, the process cannot be terminated from the application that started it.

To determine if there is a sleeping process in the repository to be destroyed, log into ASE with **isql** and enter the **sp_who** command:

```
1> sp_who
2> go
```

The output is similar to the following; it lists the status of each process:

```
 spid    status       loginame     hostname   blk   dbname     cmd
 ------ ------------ ------------ ---------- ----- ---------- -------------
      1 running      sa           snag       0     master     SELECT
      2 sleeping     NULL                    0     master     NETWORK HANDLER
      3 sleeping     NULL                    0     master     MIRROR HANDLER
      4 sleeping     NULL                    0     master     CHECKPOINT SLEEP
      5 recv sleep   sarahs       snag       0     foo        AWAITING COMMAND

(5 rows affected, return status = 0)
```

In this example, *foo* is the system repository that could not be destroyed because it was in use. After confirming with the user listed in the loginame column (in this case, sarahs) that he or she is no longer working in that system and has no open editors, enter the following command to terminate the process:

```
1> kill 5
2> go
```

Then run **sp_who** again to confirm that process 5 has been terminated.

```
1> sp_who
2> go
```

The results show that process 5 has been terminated:

```
 spid   status       loginame     hostname   blk   dbname     cmd
 ------ ------------ ------------ ---------- ----- ---------- -------------
      1 running      sa           snag       0     master     SELECT
      2 sleeping     NULL                    0     master     NETWORK HANDLER
      3 sleeping     NULL                    0     master     MIRROR HANDLER
      4 sleeping     NULL                    0     master     CHECKPOINT SLEEP

(4 rows affected, return status = 0)
```

If the Repository Is Full

The database can be increased in size from the Desktop with **Expand System Repository**, from a QRL script using **sys_expand_rep**, or from **isql** interface using **alter database**. The Desktop command and **sys_expand_rep** create additional space that is allocated to and shared by both the database’s data tables and transaction log. If you specifically want to increase the space allocated to only the database's data tables or to the transaction log separately, use **alter database**, which provides options to specify which type of segment to add.

Note: You cannot decrease the size of a database.

For more information on expanding a repository, see "Managing the Transaction Log" on page 2-30 and *StP Administration*.

Handling Crashes

A crash may be the result of:

- The Adaptive server stopping
- A machine crashing
- A network failure

In some cases, data in the repository or the *master* database can become corrupted after a crash. For this reason it is important to make periodic backups of both the repositories and the *master* database. See "Maintaining the Sybase Server" on page 2-36 for more information.

If the Adaptive Server Stops

If the Adaptive server stops:

1. Restart the server.
2. Read the error log.

Machine Crash

If a machine crashes, check the error log after system restart and server restart.

Network Fails

The network must be running for the server to run.

1. Shut the server down.
2. Start the network.
3. Start the server.

Capturing the SQL Generated by StP

It is sometimes useful to capture the SQL produced by StP. Viewing the SQL provides information that can be used for debugging purposes. You can set your machine to capture SQL by configuring your StP environment with the *oms_debug_sql* ToolInfo variable.

Set *oms_debug_sql* to a filename to capture output.

If you set *oms_debug_sql* and then start the StP Desktop from a shell command prompt (DOS prompt), SQL is printed to the window that spawned StP. However, because of the length of the SQL output, you may want to print it to a file rather than display it in a window.

The space required for the SQL that StP generates may be quite large. Make sure you unset *oms_debug_sql* after producing the SQL you want to capture.

Capturing StP-Generated SQL (UNIX)

To set the *oms_debug_sql* variable for screen or message window display:

1. At a shell prompt:

   ```
   setenv oms_debug_sql
   ```

2. To set the variable to a filename to capture output:

   ```
   setenv oms_debug_sql <file_name>
   ```

Capturing StP-Generated SQL (Windows NT)

To set the *oms_debug_sql* variable to output to a file, enter lines similar to the following in the user's ToolInfo file.

```
oms_debug_sql=<filename>
```

For example:

```
oms_debug_sql=debug.out
```

To set the *oms_debug_sql* variable to output to a window, use an invalid file name, for example:

```
oms_debug_sql=?
```

If You Cannot Resolve the Problem

If you cannot find a solution for the problem you are experiencing, contact Aonix Technical Support with the following information:

- The *errorlog* from the Sybase install directory (see "Reading the Error Log" on page 3-2)
- The version of the Sybase server and of StP (see "Determining the Sybase Server Version" on page 3-2)
- The hardware platform and operating system version
- If you can still log into the server, the output from the **sp_configure** system procedure
- The text of any error messages that were received
- A description of the circumstances under which the problem occurs

Index