CHAPTER 1
Introduction ................................................................................................. 1
  About Adaptive Server .............................................................................. 1
  System-specific issues ............................................................................ 2
  System user roles ................................................................................... 3
  Environment variables ............................................................................ 3
  Adaptive Server devices and system databases .................................. 6
    The master device .............................................................................. 5
    The subsystemdb device ............................................................... 6
    The sysprocsdev device ............................................................... 6
    Optional devices and databases ....................................................... 6
  Using UNIX operating system files as database devices .............. 8
  Determining the location, type, and size of a database device .... 10
  Client/server communication ............................................................. 12
  Changing Adaptive Server configuration ........................................... 13
  Languages other than U.S. English ....................................................... 14
  Adaptive Server specifications ............................................................. 14

CHAPTER 2
Starting and Stopping Servers ............................................................... 19
  Overview ............................................................................................... 19
  Requirements for starting servers ....................................................... 19
  Starting servers ..................................................................................... 20
    Server start-up parameters ............................................................ 20
    Using a RUN_server_name file ...................................................... 21
    Using the startserver command .................................................... 21
  Starting servers when the operating system restarts ...................... 22
    For HP-UX ...................................................................................... 22
    For IBM RS/6000 ........................................................................... 22
    For Sun Solaris and Linux ........................................................... 23
  Starting XP Server after initial installation ..................................... 25
  Stopping servers .................................................................................... 25
    Stopping Adaptive Server ........................................................... 26
    Stopping Backup Server .............................................................. 26
    Using the kill command ................................................................. 27
    Shutdown and shared memory files .............................................. 28
Contents

Huge pages on Linux .............................................................. 29

CHAPTER 3 Configuring the Operating System ................................. 31
  Verifying environment variables ........................................... 31
  Configuring new servers with srvbuild or srvbuildres ............... 32
  Using the stty setting .................................................................. 35
  Restoring correct permissions .................................................. 36
  File descriptors and user connections ....................................... 36
    For HP-UX ........................................................................... 36
    For AIX .............................................................................. 36
    For Linux ............................................................................ 37
    For Sun Solaris .................................................................... 37
  Displaying current soft and hard limits .................................... 37
  Increasing the soft limit ......................................................... 37
  Increasing the hard limit ......................................................... 38
  Sample program ...................................................................... 39
  Enabling asynchronous disk I/O............................................... 40
  Adjusting the client connection timeout period ....................... 43
    For HP-UX 11 and later ....................................................... 43
    For IBM RS/6000 .................................................................. 43
    For Sun Solaris .................................................................... 43
    For Linux ............................................................................ 44
  Checking for hardware errors .................................................. 44
    For HP-UX ........................................................................... 44
    For IBM RS/6000 .................................................................. 45
    For Sun Solaris .................................................................... 45
    For Linux ............................................................................ 45
  Monitoring the use of operating system resources .................... 45
    For HP-UX ........................................................................... 45
    For IBM RS/6000 .................................................................. 46
    For Sun Solaris and Linux .................................................... 46
  Checking the integrity of databases ......................................... 47

CHAPTER 4 Adaptive Server Default Configuration ............................. 49
  Default settings ....................................................................... 49

CHAPTER 5 Setting Up Communications Across the Network .................. 51
  How Adaptive Server determines which directory service entry to use 52
    How a client uses directory services ..................................... 53
    Creating a directory services entry ....................................... 53
    Supported directory drivers .................................................. 54
## Contents

- Contents of an interfaces file .......................................................... 54
- Heterogeneous and homogeneous environments ............................. 55
- Understanding the format of the interfaces file ............................. 57
- Components of an interfaces file entry ........................................ 58
- Creating a master interfaces file .................................................... 60
- Using dsedit or dsck to create a master interfaces file .................. 60
- Using a text editor to create a master interfaces file ...................... 61
- Configuring interfaces files for multiple networks ....................... 61
- Configuring the server for multiple network handlers ................. 62
- Configuring the client connections ............................................. 62
- Configuring for query port backup ............................................. 64
- IPv6 support ................................................................................... 65
- IPv6 infrastructure ....................................................................... 66
- Troubleshooting ............................................................................. 68
  - Server fails to start ................................................................ 68
  - Error when executing an ESP ................................................. 69

### CHAPTER 6
**Using the Lightweight Directory Access Protocol as a Directory Service** ................................................................. 71
- Overview ......................................................................................... 71
- LDAP directory services versus the Sybase interfaces file .......... 72
- The libtcl*.cfg file ........................................................................... 75
- Enabling LDAP directory services ................................................ 76
- Adding a server to the directory services .................................... 78
- Multiple directory services ............................................................ 79
- Encrypting the password .............................................................. 80
- Performance .................................................................................. 80
- Migrating from the interfaces file to LDAP .................................. 81

### CHAPTER 7
**Customizing Localization for Adaptive Server** ......................... 83
- Overview of localization support .................................................. 83
  - Language modules ................................................................. 84
  - Default character sets for servers .......................................... 85
  - Supported character sets ....................................................... 86
- Character set conversion ............................................................. 91
  - Conversions between server and client ............................... 91
- Sort orders ................................................................................... 92
  - Available sort orders ............................................................. 92
- Language modules ...................................................................... 95
  - Installing a new language module ....................................... 95
  - Message languages ............................................................... 95
- Localization .................................................................................. 96
  - Localization directories ......................................................... 96
CHAPTER 1

Introduction

Sybase® Adaptive Server® Enterprise for UNIX is a full-featured Adaptive Server that runs on the following platforms:

- HP-UX
- IBM AIX
- Sun Solaris
- Linux

Note The instructions in this book assume that Adaptive Server is installed and running. For information about installing and starting Adaptive Server, as well as for an overview of Adaptive Server, see the Adaptive Server Installation Guide for your platform.

This chapter provides an overview of how to configure Adaptive Server and the steps you need to take to customize it for your use.

About Adaptive Server

Adaptive Server performs data management and transaction functions, independent of client applications and user interface functions.
Adaptive Server also:

- Manages multiple databases and multiple users
- Keeps track of the data's location on disks
- Maintains the mapping of logical data description to physical data storage
- Maintains data and procedure caches in memory

Adaptive Server uses these auxiliary programs to perform dedicated tasks:

- Backup Server manages database load, dump, backup, and restoration activities.
- XP Server stores the extended stored procedures (ESPs) that allow Adaptive Server to run operating-system and user-defined commands.

System-specific issues

Adaptive Server runs on a variety of hardware and operating system platforms. System-specific issues do not affect the basic functionality of Adaptive Server, but there are differences among platform implementations. These differences may include:

- Adaptive Server configuration
- Changes to the operating system that enable or enhance Adaptive Server performance
- The structure of entries in the *interfaces* file
- Options for selecting database devices
- Operating system commands or utilities that simplify or automate routine system administration tasks
- Operating system utilities for monitoring Adaptive Server performance

System-specific issues are described in this document. For more information about system-specific issues, see the *Installation Guide* and release bulletin for your platform.
System user roles

The Adaptive Server installation and setup process defines various user roles. Different user roles have different responsibilities and privileges. These user roles clarify the way Adaptive Server is to be integrated into your system:

- Operating system administrator – the individual who maintains the operating system. This individual has superuser or “root” privileges.
- System Administrator – the individual in charge of Adaptive Server system administration, creating user accounts, assigning permissions on databases, and creating new databases. At installation time, the System Administrator’s login name is “sa”. The “sa” login is not a UNIX login. The “sa” login is specific to Adaptive Server and is used to log in to Adaptive Server using the `isql` command.
- “sybase” login – the “sybase” login is a UNIX login that owns all the Sybase installation directories and files, sets permissions on those directories and files, and performs the installation and upgrading of Adaptive Server. This login is preferred, but not required.

Environment variables

It is crucial to the operation of Sybase products that the system environment variables be set correctly.

Environment variables are set in the user’s environment either interactively or by including them in the user’s `.login` and `.cshrc` files (for C shell) or `.profile` file (for Bourne shell). The installation instructions in this guide explain when to set these variables.

- **Note**  As part of the installation, the installer setup these environment variables in `SYBASE.csh` and `SYBASE.sh` files. You can source the file to set their environment.

  - DLISTEN – defines the name Adaptive Server uses to listen for client connections if no name is given during the Adaptive Server start-up. If DLISTEN is not set, and no name is given during the Adaptive Server start-up, the Adaptive Server name defaults to SYBASE.
Environment variables

- **DSQUERY** – defines the Adaptive Server name that client programs attempt to connect to if no Adaptive Server is specified with a command line option. If DSQUERY is not set, and you do not supply the Adaptive Server name with a command line option, clients attempt to connect to SYBASE.

- **SYBASE** – defines the path of the Sybase installation directory. The installation program sets up the variable SYBASE to point to the release directory specified during installation.

- **SYBASE_ASE** – defines the subdirectory of the Adaptive Server component.

- **SYBASE_OCS** – defines the subdirectory to which Open Client™ is set.

- **SYBASE_SYSAM** – points to the license-management software directory.

- **PATH** – specifies directory path to search for executables. The Sybase executables are located in the **installed_components/bin** directory. When using the source files SYBASE.csh or SYBASE.sh, the following paths are prepended to PATH:
  
  ```
  $SYBASE/$SYBASE_ASE/bin:$SYBASE/$SYBASE_OCS/bin
  ```

- **LD_LIBRARY_PATH** – this variable specifies the directory to search for shared libraries. The Sybase shared libraries are located in the installed component’s **/lib** directory.

  When using the source files: SYBASE.csh or SYBASE.sh, the following paths are prepended to the LD_LIBRARY_PATH environment variable:

  ```
  $SYBASE/$SYBASE_ASE/lib:$SYBASE/$SYBASE_OCS/lib:$SYBASE/SYBASE_FTS/lib, etc.
  ```

- **LD_LIBRARY_PATH64** – this variable specifies the directory to search for shared libraries on 64-bit Sun platforms.

- **LIBPATH** – on the IBM RS/6000 platform, this variable specifies which directory to search for libraries.

  When using the source files: SYBASE.csh or SYBASE.sh, the following paths are prepended to the LIBPATH environment variable:

  ```
  $SYBASE/$SYBASE_ASE/lib:$SYBASE/$SYBASE_OCS/lib: $SYBASE/SYBASE_FTS/lib, etc.
  ```

- **SHLIB_PATH** – on HP-UX platforms, this variable specifies which directory to search for libraries.
When using the source files: SYBASE.csh or SYBASE.sh, the following paths are prepended to the LIBPATH environment variable:

$SYBASE/$SYBASE_ASE/lib:$SYBASE/$SYBASE_OCS/lib:$SYBASE/SYBASE_FTS/lib, etc.

Adaptive Server devices and system databases

Devices are files or portions of a disk that are used to store databases and database objects. You can initialize devices using raw disk partitions or operating system files.

Adaptive Server requires the following devices:

- **master** – to store system databases.
- **sybsystemdb** – to store information about distributed transaction.
- **sysprocsdev** – to store system procedures.

The master, sybsystemdb, and sysprocsdev devices are created when you create a new Adaptive Server.

The master device

The master device contains the following databases:

- **master** – controls the operation of Adaptive Server as a whole and stores information about all users, user databases, devices, objects, and system table entries. The master database is contained entirely on the master device and cannot be expanded onto any other device.
- **model** – provides a template for new user databases. The model database contains required system tables, which are copied into a new user database with the create database command.
- **tempdb** – the work area for temporary databases in Adaptive Server. Adaptive Server supports multiple tempdbs. See “create temporary database” in the *Transact-SQL Users Guide*. Each time Adaptive Server is started, the tempdb database is cleared and rebuilt from the model database.
Adaptive Server devices and system databases

- The sample databases, pubs2 and pubs 3, are stored on the master device at installation, but should be moved to a user-defined device after installation.

  **Note** For recovery purposes, Sybase recommends that you do not create other system or user databases or user objects on the master device.

The sybsystemdb device

For new installations the master device also contains the sybsystemdb database. The sybsystemdb device stores the sybsystemdb database, which stores information about transactions in progress, and which is also used during recovery.

The sybsystemdb database is required to support distributed transaction management (DTM) features. Before installation, make sure you have enough space available on the default segment to support sybsystemdb.

The sysprocsdev device

The syprocsdev devices stores the sybsystemprocs database, which contains most of the Sybase-supplied system procedures. System procedures are a collection of SQL statements and flow-of-control statements that perform system tasks, for example, sp_configure.

The system procedures that are needed during recovery situations are stored in the master database.

  **Note** sysprocsdev is the default system name for this device. However, it is frequently referred to as the sybsystemprocs device, since it stores the sybsystemprocs database.

Optional devices and databases

The devices and databases described in the following sections are optional.
The pluggable component interface (PCI) allows you to add pluggable libraries that provide different functionalities to the Adaptive Server. Java support (pluggable component adaptor/Java virtual machine) is included as a pluggable component with Adaptive Server 15.0.3.

The sybpcidb database stores necessary configuration information for the PCI and the pluggable component adaptor/Java virtual machine (PCA/JVM) plug-in.

To enable PCI in Adaptive Server by using srvbuildres, add PCI/Java related properties to the resource files used by these utilities. Enter these values:

```
sqlsrv.do_configure_pci: yes
sqlsrv.sybpcidb_device_physical_name: /device_path
sqlsrv.sybpcidb_device_size: USE_DEFAULT
sqlsrv.sybpcidb_database_size: USE_DEFAULT
```

The sample databases

The sample databases are:

- pubs2 and pubs3 – provided as learning tools for Adaptive Server. pubs2 is used for most of the examples in the Adaptive Server documentation; other examples use the pubs3 database. Both are available in U.S. English versions of Adaptive Server.
- interpubs – contains French and German data.
- jpubs – contains Japanese data.

For information about installing the sample databases, see the Installation Guide for your platform, Chapter 3, “Post-Installation Tasks.”

For information about the contents of the sample databases, see the Transact-SQL Users Guide.

The sybsecurity device and database

The sybsecurity device is created as part of the auditing installation process. The sybsecurity device stores the sybsecurity database and the auditing system procedures with which you can configure auditing for your system.

The auditing system records system security information in an Adaptive Server audit trail. You can use this audit trail to monitor the use of Adaptive Server or system resources.

For instructions on configuring Adaptive Server for auditing, see Chapter 10, “Adding Optional Functionality to Adaptive Server.” For information about installing and using the auditing system, see the chapter on “Auditing” in the Security Administration Guide.
Adaptive Server devices and system databases

**dbccdb database**

The database consistency checker (dbcc) provides commands for checking the logical and physical consistency of a database. The dbccdb database stores the results of dbcc when dbcc checkstorage or dbcc checkverify is used.

dbcc checkstorage records consistency information for the **target database**, operation activity, and the results of the operation in the dbccdb database. Stored in the database are dbcc stored procedures for creating and maintaining dbccdb and for generating reports on the results of dbcc checkstorage operations.

For information on installing and using dbccdb, see Chapter 25, “Checking Database Consistency,” in the *System Administration Guide: Volume 2*.

**sybmgmtdb database**

sybmgmtdb is a Job Scheduler database.

All the job, schedule, and scheduled job information, and data needed by the Job Scheduler task for internal processing is stored in the sybmgmtdb database. Most access to data in the sybmgmtdb database is via stored procedures. The stored procedures make the data available to the GUI, the JS Agent and the command-line interface. Only the Job Scheduler task accesses data directly from the sybmgmtdb database.

For more information about sybmgmtdb and Job Scheduler, see *Job Scheduler Users Guide*.

Using UNIX operating system files as database devices

For devices that are initialized on UNIX operating system files, choose one of these methods to ensure that writes on the device occur directly on the physical media:

- Use the directio parameter with disk init, disk reinit or sp_deviceattr
- Specify the dsync setting with disk init

directio and dsync parameters are mutually exclusive. If a device has dsync set to true, you cannot set directio to true for this device. To enable directio for a device, also reset dsync to false.
Using the **directio** parameter

The `directio` parameter for `disk init`, `disk reinit`, and `sp_deviceattr` lets you configure Adaptive Server to transfer data directly to disk, bypassing the operating system buffer cache. `directio` performs I/O in the same manner as raw devices and provides the same performance benefit, but has the ease of use and manageability of file system devices.

By default, the `directio` option is set to false (off) for all platforms.

For more information on using `directio`, see the *System Administration Guide, Volume 1*.

Using the **dsync** option

The `dsync` option ensures Adaptive Server can recover data from devices on the file systems. By default, Adaptive Server enables `dsync` for file system devices. However, `dsync` may cause performance degradation for file system devices that experience high write activity. You can set or reset `dsync` using the `disk init`, `disk reinit`, and `sp_deviceattr` commands.

*Note* The `dsync` option is ignored for raw devices.

By default, when you install a new Adaptive Server, `dsync` is set on for all file system devices.

When you are upgrading from ASE 12.0 or earlier versions on a UNIX server that stored databases on UNIX file system devices, by default, `dsync` is set:

- on for the master device
- off for all other devices

Immediately after upgrading, check to make sure that `dsync` or `directio` is set for the file system devices. See also `sp_deviceattr` and `sp_helpdevice` in the *Reference Manual: Procedures*

*Warning!* Data corruption may occur if `dsync` option is not set immediately after upgrade.
Determining the location, type, and size of a database device

Adaptive Server requires several database devices. Table 1-1 below shows baseline values for each of the devices. See the release bulletin for any last minute changes to these values.

**Table 1-1: Adaptive Server database devices**

<table>
<thead>
<tr>
<th>Device</th>
<th>Purpose</th>
<th>Minimum size</th>
<th>Minimum recommended size</th>
</tr>
</thead>
<tbody>
<tr>
<td>master</td>
<td>Stores system databases</td>
<td>24MB for 2K pages</td>
<td>30MB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>45MB for 4K pages</td>
<td>60MB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>89MB for 8K pages</td>
<td>120MB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>177MB for 16K pages</td>
<td>240MB</td>
</tr>
<tr>
<td>sysprocddev</td>
<td>Stores the sybsystemprocs database</td>
<td>136MB</td>
<td>140MB (plus any space for holding stored procedures that you have created)</td>
</tr>
<tr>
<td></td>
<td>(also called the sybsystemprocs device)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sybsystemdb</td>
<td>Transaction processing</td>
<td>3MB for 2K pages</td>
<td>5 – 24MB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6MB for 4K pages</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>12MB for 8K pages</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>24MB for 16K pages</td>
<td></td>
</tr>
<tr>
<td>syssecurity</td>
<td>Required for auditing</td>
<td>10MB or size of the model database (the larger of the two)</td>
<td>Depends on specific auditing needs; see “Adding auditing” on page 115</td>
</tr>
<tr>
<td>(optional)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For Solaris, HP-UX, IBM AIX, and Linux

For all databases you can use either a raw partition or a file.

Sybase Adaptive Server supports the database devices on NFS- and CIFS-mounted devices with Network Appliance Filers for storing data. Network appliance filers provide the same performance and data integrity as raw devices. There are no changes needed to the operating system or to Sybase Adaptive Server to use network appliance filers.

NFS mounted devices have been tested on Solaris, HP-UX, IBM AIX, and Linux.
For All Platforms

You may need to repartition the disk you choose. Contact your operating system administrator for assistance.

When preparing a raw partition device, follow these guidelines:

- Do not initialize a database device on the partition that contains your Sybase installation software. Doing so destroys all existing files on that partition.
- A raw partition designated for use by Sybase cannot be mounted for use by the operating system for any other purpose, such as for file systems or swap space.
- After a Sybase configuration utility or the `disk init` command has initialized a portion of a partition as a database device, you cannot use the entire partition for any other purpose. To dynamically increase the size of your database devices, rather than initializing a new device, use `disk resize`. See “Creating files for database devices” on page 112.
- For best performance, place the Sybase software and all disk devices, including the master device, on the same machine.
- To avoid any possibility of using a partition that contains the partition map, do not use cylinder 0.

Choosing a raw partition

1. Determine which raw partitions are available.
2. Determine the sizes of the raw partitions.
3. From the list of available raw partitions, select a raw partition for each device, based on the size recommendations in Table 1-1 on page 10.
4. Verify with the operating System Administrator that the partition you have chosen is available.
5. Make sure the “sybase” user has read and write privileges to the raw partition.

Note For more information on choosing a raw partition, see your operating system documentation.
Client/server communication

Adaptive Server communicates with other Adaptive Servers, Open Server™ applications (such as Backup Server), and client software on your network. Clients can talk to one or more servers, and servers can communicate with other servers by remote procedure calls.

For Sybase products to interact with one another, each product needs to know where the others reside on the network. Names and addresses of every known server are listed in a directory services file. This information can be stored in a directory services file two different ways:

- In an interfaces file, named `interfaces` on UNIX platforms, located in the `$SYBASE` installation directory, or
- In an LDAP server

After your Adaptive Server or client software is installed, it can connect with any server on the network that is listed in the directory services.

When you are using a client program, and you want to connect with a particular server, the client program looks up the server name in the directory services and connects to that server, as shown in Figure 1-1. You can supply the name of the server by using the DSQUERY environment variable.

On TCP/IP networks, the port number gives clients a way to identify the Adaptive Server, Open Server, or Backup Server to which they want to connect. It also tells the server where to listen for incoming connection attempts from clients. The server uses a single port for these two services (referred to as `query service` and `listener service`).
During installation, you use the `srvbuild` utility to create and configure a new server. The `srvbuild` process adds entries to the interfaces file for your new Adaptive Server, Backup Server, and XP Server.

For instructions on how to modify existing interfaces file entries using `dsedit` and `dscp`, see the *Utility Guide*. To create new interfaces file entries for existing servers, see Chapter 6, “Using the Lightweight Directory Access Protocol as a Directory Service.”

### Changing Adaptive Server configuration

Use `sp_configure` to change the configuration of Adaptive Server. To change the configuration of several servers, enter the `sp_configure` information in a script. For details on using `sp_configure`, see the *System Administration Guide: Volume 1* and the *Reference Manual: Procedures*.

For information about configuring languages, character sets, and sort orders, see Chapter 7, “Customizing Localization for Adaptive Server.”

For information about configuring Adaptive Server to use high availability features, see *Using Sybase Failover in a High Availability Environment*.

For information about configuring Adaptive Server for distributed transaction management (two-phase commit), see the *System Administration Guide: Volume 2*. 
Languages other than U.S. English

If you are running srvbuild in a language other than U.S. English, verify that any input you provide uses a character set that is supported by the us_english character set.

**Note** The us_english character set does not support accent marks, such as tildes (¨) and umlauts (ü). This prevents srvbuild from supporting the character sets that use these characters.

For more information about languages, character sets, and sort orders, see the *Installation Guide* for your platform. For more information on srvbuild, see the *Utility Guide*.

Adaptive Server specifications

**Database specifications**

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Databases per Adaptive Server</td>
<td>A maximum of 32,767 databases per server</td>
</tr>
<tr>
<td>Maximum database size</td>
<td>• 2K page server – 4TB</td>
</tr>
<tr>
<td></td>
<td>• 4K page server – 8TB</td>
</tr>
<tr>
<td></td>
<td>• 8K page server – 16TB</td>
</tr>
<tr>
<td></td>
<td>• 16K page server – 32TB</td>
</tr>
<tr>
<td>Minimum allowable sybsystemprocs database</td>
<td>136MB Required for an upgrade</td>
</tr>
<tr>
<td>Maximum size of a database device (disk partition)</td>
<td>$2^{34}$ (4TB) If the Operating System supports file sizes up to 4TB, then Adaptive Server supports file system devices up to 4TB</td>
</tr>
<tr>
<td>Maximum number of database devices per server</td>
<td>$2^{31}$</td>
</tr>
<tr>
<td>Maximum number of devices or device pieces per database</td>
<td>Unlimited Limited by available memory</td>
</tr>
<tr>
<td>Maximum number of segments per database</td>
<td>31</td>
</tr>
<tr>
<td>Maximum number of login IDs per server</td>
<td>2147516416</td>
</tr>
</tbody>
</table>
### Maximum number of users per database
2,146,484,223

### Maximum number of groups per database
103,219,3

### Table specifications

<table>
<thead>
<tr>
<th>Specification</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>User objects per database</td>
<td>$2^{31} - 255$</td>
</tr>
<tr>
<td>Indexes per table</td>
<td>250 (one clustered index)</td>
</tr>
<tr>
<td>Rows per table</td>
<td>Limited by available storage</td>
</tr>
<tr>
<td>Columns per composite index</td>
<td>31</td>
</tr>
<tr>
<td>Creation of clustered index</td>
<td>$1.2 \times (x + y)$ $x =$ total data space in table, $y =$ sum of space of all nonclustered indexes on table, and 20 percent overhead for logging</td>
</tr>
<tr>
<td>Maximum size of object name</td>
<td>255</td>
</tr>
</tbody>
</table>

### Query specifications

<table>
<thead>
<tr>
<th>Specification</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum number of tables participating in a query, for a query without a union, or for each side of a union in a query</td>
<td>64</td>
</tr>
<tr>
<td>Maximum number of tables in a “union” query</td>
<td>256</td>
</tr>
<tr>
<td>Maximum number of databases participating in one transaction</td>
<td>Unlimited</td>
</tr>
</tbody>
</table>

- Maximum of 50 user tables, including result tables, tables referenced by views (the view itself is not counted) correlations and self-joins; maximum of 46 worktables
- Includes up to 50 user tables and 14 work tables on every side of the union, for a total of up to 256 tables across all sides of the union.
- Includes database where transaction began, all databases changed during transaction, and tempdb, if it is used for results or worktables
Adaptive Server specifications

<table>
<thead>
<tr>
<th>Practical number of databases participating in one query</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Includes each occurrence of each database queried and tempdb, if it is used for results or worktables</td>
<td></td>
</tr>
<tr>
<td>Maximum number of tables with referential integrity constraints for a query</td>
<td>192</td>
</tr>
</tbody>
</table>

Procedure specifications

<table>
<thead>
<tr>
<th>Number of buffers and procedure buffers</th>
<th>Configurable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limited by amount of RAM and maximum size of shared memory segment</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Minimum memory required per stored procedure</th>
<th>2K</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum number of parameters per stored procedure</td>
<td>2048</td>
</tr>
</tbody>
</table>

Adaptive Server extended-limit capabilities vary by type of table and the database logical page size. Table 1-2 lists the column and row limits for allpages-locked (APL) tables.

### Table 1-2: Allpages-locked (APL) tables

<table>
<thead>
<tr>
<th>Maximum APL table limits</th>
<th>Number of columns</th>
<th>Column size 2K page</th>
<th>Column size 4K page</th>
<th>Column size 8K page</th>
<th>Column size 16K page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed-length column</td>
<td>1024</td>
<td>1960 bytes</td>
<td>4008 bytes</td>
<td>8104 bytes</td>
<td>16296 bytes</td>
</tr>
<tr>
<td>Variable-length column</td>
<td>254</td>
<td>1948 bytes</td>
<td>3988 bytes</td>
<td>8068 bytes</td>
<td>16228 bytes</td>
</tr>
</tbody>
</table>

Table 1-3 lists the column and row limits for data-only-locked (DOL) tables.

### Table 1-3: Data row and data page tables.

<table>
<thead>
<tr>
<th>Maximum DOL table limits</th>
<th>Number of columns</th>
<th>Column size 2K page</th>
<th>Column size 4K page</th>
<th>Column size 8K page</th>
<th>Column size 16K page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed-length column</td>
<td>1024</td>
<td>1958 bytes</td>
<td>4006 bytes</td>
<td>8102 bytes</td>
<td>16294 bytes</td>
</tr>
<tr>
<td>Variable-length column</td>
<td>1024</td>
<td>1954 bytes</td>
<td>4002 bytes</td>
<td>8098 bytes</td>
<td>16290 bytes</td>
</tr>
</tbody>
</table>

Database space requirements depend upon the logical page size of the server. If your model database is larger than the minimum size listed below, then the minimum size of the database is equal to the model database. Table 1-4 lists the minimum size for each database.

### Table 1-4: Database requirements for varying page sizes

<table>
<thead>
<tr>
<th>Databases</th>
<th>2K page</th>
<th>4K page</th>
<th>8K page</th>
<th>16K page</th>
</tr>
</thead>
<tbody>
<tr>
<td>master database</td>
<td>13MB</td>
<td>26MB</td>
<td>52MB</td>
<td>104MB</td>
</tr>
</tbody>
</table>
Larger logical page sizes can contain more data. Table 1-5 lists the maximum data for each logical page size.

**Table 1-5: Data limits for tables according to page size**

<table>
<thead>
<tr>
<th>Databases</th>
<th>2K page</th>
<th>4K page</th>
<th>8K page</th>
<th>16K page</th>
</tr>
</thead>
<tbody>
<tr>
<td>model database</td>
<td>3MB</td>
<td>6MB</td>
<td>12MB</td>
<td>24MB</td>
</tr>
<tr>
<td>tempdb database</td>
<td>4MB</td>
<td>6MB</td>
<td>12MB</td>
<td>24MB</td>
</tr>
<tr>
<td>sybsystemdb database</td>
<td>3MB</td>
<td>6MB</td>
<td>12MB</td>
<td>24MB</td>
</tr>
<tr>
<td>sybpcidb database</td>
<td>24MB</td>
<td>48MB</td>
<td>96MB</td>
<td>192MB</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tables</th>
<th>2K page</th>
<th>4K page</th>
<th>8K page</th>
<th>16K page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bytes per index key</td>
<td>600</td>
<td>1250</td>
<td>2600</td>
<td>5300</td>
</tr>
<tr>
<td>User-visible row length</td>
<td>1958</td>
<td>4006</td>
<td>8102</td>
<td>16294</td>
</tr>
<tr>
<td>DOL table</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User-visible row length</td>
<td>1960</td>
<td>4008</td>
<td>8104</td>
<td>16296</td>
</tr>
<tr>
<td>APL table</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
CHAPTER 2

Starting and Stopping Servers

This chapter describes how to start and stop Adaptive Server, Backup Server, and XP Server.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>19</td>
</tr>
<tr>
<td>Starting servers</td>
<td>20</td>
</tr>
<tr>
<td>Starting servers when the operating system restarts</td>
<td>22</td>
</tr>
<tr>
<td>Starting XP Server after initial installation</td>
<td>25</td>
</tr>
<tr>
<td>Stopping servers</td>
<td>25</td>
</tr>
</tbody>
</table>

Overview

The methods described in this chapter are used to start Adaptive Server and Backup Server after a shutdown for database maintenance, because of an operating system crash, or for other reasons.

XP Server is not started by the installation process. XP Server is started by Adaptive Server when an XP command is issued through isql.

You can use Sybase Control Center to start and stop servers manually or automatically. For more information about Sybase Control Center, see the System Administration Guide: Volume 1 and the online help.

Requirements for starting servers

To start a server, your user account must have:

- Anyone with access to the database servers (execute privileges) and database devices (read/write privileges).
- Access to Adaptive Server distribution files
- The system environment variables discussed in Chapter 1, “Introduction”
Starting servers

- Access to SySAM licenses. For more information, see the Sybase Software Asset Management User’s Guide.

The installation program creates the interfaces file and system environment variables when you install servers on your computer.

Starting servers

You can start Adaptive Server or Backup Server from the command line by using RUN_server_name files and startserver commands. You can customize the starting options.

You can also customize the start up options by editing the RUN_server_name.

Server start-up parameters

The default start-up parameters are stored under $SYBASE/$SYBASE_ASE/install/RUN_server_name, where server_name is the name of the server you installed.

Backup Server server names are appended with “_back”.

Table 2-1 lists the default start-up parameters for Adaptive Server.

<table>
<thead>
<tr>
<th>Switch</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-d $SYBASE/data/master.dat</td>
<td>Location of the master device file</td>
</tr>
<tr>
<td>-s server_name</td>
<td>Name of the Adaptive Server</td>
</tr>
<tr>
<td>-e $SYBASE/$SYBASE_ASE/install/errorlog</td>
<td>Location and name of the error log file</td>
</tr>
<tr>
<td>-M $SYBASE/$SYBASE_ASE</td>
<td>Directory that stores shared memory files</td>
</tr>
<tr>
<td>-NSYBASE/$SYBASE_ASE/sysam/</td>
<td>Location and name of license cache file.</td>
</tr>
<tr>
<td>&lt;srv_name&gt;.properties</td>
<td></td>
</tr>
</tbody>
</table>

Changing start-up parameters

You cannot change any of these default start-up parameters unless you directly edit the $SYBASE/$SYBASE_ASE/install/RUN_server_name file. You can also specify additional start-up parameters within the $SYBASE/$SYBASE_ASE/install/RUN_server_name file.
Additional start-up parameters include any valid server command line options listed for the **databaserver** and **backupserver** descriptions in the *Utility Guide*.

### Using a *RUN_server_name* file

Each time a new Adaptive Server or Backup Server is created, the *srvbuild* program creates a *RUN_server_name* file that contains the information required to restart that server. *RUN_server_name* files are created in the `$SYBASE/$SYBASE_ASE/install` directory. For Adaptive Server, the *RUN_server_name* file is named *RUN_server_name*, where `servername` is the name of the server.

For example, the *RUN_server_name* file name for an Adaptive Server named TEST is *RUN_TEST*.

The *RUN_server_name* files is named *RUN_servername_back* for Backup Server, where `servername` is the name of the server.

**Warning!** Do not delete the *RUN_server_name* file that is created in `$SYBASE/$SYBASE_ASE/install`. This file is used to restart servers when you customize your installation. If you need the *RUN_server_name* file in another location, make a copy of the original *RUN_server_name* file in the new location.

### Using the *startserver* command

To start a server from the command line, enter:

```
$SYBASE/$SYBASE_ASE/install/startserver [ -f RUN_server_name file ]
```

where `$SYBASE/$SYBASE_ASE/install/startserver` is the full path name of the `startserver` utility, and `RUN_server_name file` is the full path name of the `RUN_server_name` file (usually `$SYBASE/$SYBASE_ASE/install/RUN_server_name`).

If your Adaptive Server is named SYBASE, the `RUN_server_name` file specification is optional.

You must have read and write permissions on the master device for an Adaptive Server to start it with `startserver`. For more information about `startserver`, see the Adaptive Server *Utility Guide*. 
Starting servers when the operating system restarts

This section describes how to configure your operating system for automatic restart of Adaptive Server and Backup Server.

On production systems, Adaptive Server and Backup Server should restart automatically when the UNIX operating system starts. This is accomplished by making an entry for the server in the operating system start-up script.

Note The startup script must set up all the required Sybase environment variables, or source SYBASE.csh or SYBASE.sh, before starting the server.

For HP-UX

In HP-UX versions 10.0 and later, you cannot edit commands in the /etc/rc file, so the System Administrator must create a script that executes when the operating system starts or shuts down.

For more information about rc (1M), see the HP-UX manual page. You can model your script after the HP-UX template file /sbin/init.d/template.

After creating the start-up and shutdown script, place it in the directory /sbin/init.d. Execution scripts placed in this directory have symbolic links to directories /sbin/rcn.d where n is the system run level. The linked scripts in /sbin/rcn.d are used to control the sequencing order of the execution scripts.

For IBM RS/6000

On production systems, you can restart Adaptive Server automatically whenever the UNIX operating system restarts. To have Adaptive Server restart automatically, place the startserver command in the /etc/inittab file.

Here is a suggested format for an Adaptive Server start-up command to add to /etc/inittab:

"sybase:2:wait:/release_directory/install/startserver -f \ RUN_servername /dev/console 2>&1"

Where release_directory is the full path to the Sybase installation directory (specified as the SYBASE environment variable), and RUN_servername is the RUN_server_name file for the server you are starting.
Make sure that the entry for starting Adaptive Server follows any entry for
/etc/rc.tcpip and /etc/rc.nfs in the /etc/inittab file. All network resources must
be available before you start Adaptive Server; otherwise, Adaptive Server does
not start. Also, your server does not start if the network is not running.

Slow start-up of network operations can also prevent Adaptive Server from
starting, even when the commands in the /etc/rc.tcpip file are in the correct
order. You can designate a period of time for Adaptive Server to wait before
starting by inserting a sleep command before the server start-up command in
the RUN_server_name file. The sleep command is in the form:

```
sleep seconds_of_rest
```

### Note
To enable restart of a Backup Server when the operating system starts,
add a Backup Server start-up command to /etc/inittab. Use the path of the
Backup Server RUN_server_name file in this command.

---

**For Sun Solaris and Linux**

Before you set Adaptive Server or Backup Server to start automatically, make
sure that all network resources are available. Your server does not start if the
network is not running. Be sure that the entry for restarting the server follows
any commands in the rc directory for starting network operations. Slow start-
up of network operations may prevent servers from starting, even if commands
are in the correct order in the rc directory. You may want to designate a period
of time for your server to wait before starting by inserting a sleep command at
the beginning of the linked RUN_server_name file you created.

To configure your operating system for automatic restart of Adaptive Server
and Backup Server:

1. Create a start-up script file with contents like the following:

   ```
   $SYBASE/$SYBASE_ASE/install/startserver
   -f
   $SYBASE/$SYBASE_ASE/install/RUN_servername
   ```

   where $SYBASE/$SYBASE_ASE/install/startserver is the full path name
   of the startserver utility, and
   $SYBASE/$SYBASE_ASE/install/RUN_servername is the full path name
   of the RUN_server_name file for the server.

2. Copy the script to the /etc/init.d directory, using syntax like the following:
Starting servers when the operating system restarts

3  Review the contents of the /etc/inittab file to determine the default run level for your operating system. The initdefault entry specifies the default run level, which is usually 2 or 3 for Sun Solaris and 5 for Linux.

4  Use the ln command to create a hard link from the RUN_server_name file to the appropriate run control (rc) directory, rc#, where # is the default run level obtained in step 2.

Use syntax like the following for Sun Solaris:

```
ln /etc/init.d/script_name     /etc/rc#.d/S##script_name
```

For Linux:

Use ln -s to create symbolic links from the script name to the appropriate run control (rc) directory, rc# where # is the run level. Use syntax like the following:

```
ln -s /etc/init.d/script_name  /etc/rc#.d/S##script_name
```

When you type the command to create this link, add an uppercase “S” and a two-digit sequence number preceding the name of the script file. “S” indicates a start-up file. The sequence number is necessary because files in the rc directory are executed in order of their numbers. This file should be executed last, so you should use a sequence number that follows all existing numbers in ASCII order.

You can perform an ls command on the rc directory to view existing sequence numbers. For example:

```
ls /etc/rc3.d/S*
```

returns:

```
/etc/rc3.d/S10syslog
/etc/rc3.d/S15nfs.server
/etc/rc3.d/S21rfs
```

If your script is named sybstart, type the following:

```
ln /etc/init.d/sybstart     /etc/rc3.d/S77sybstart
```

In this example, you can use any number greater than 27 in place of 77.

For Linux:

You can perform an ls command on the rc directory to view existing sequence numbers. For example:

```
ls /etc/rc5.d/S*
```
returns:

/etc/rc5.d/S12syslog
/etc/rc5.d/S14nfslock
/etc/rc5.d/S27ypbind

and so on. If your script is named sybstart, type the following:

ln -s /etc/init.d/sybstart    /etc/rc5.d/S99sybstart

In this example, you can use any number greater than 27 in place of 99.

**Starting XP Server after initial installation**

If you install Adaptive Server and XP Server in the same build session, srvbuild automatically adds information about the XP Server to the `sysservers` table of Adaptive Server. If you install XP Server in a different build session than the Adaptive Server installation, you are prompted during the XP Server installation process to supply the name of the related Adaptive Server and the System Administrator’s name and password. This information is required by Adaptive Server to start XP Server.

If you do not supply this information during the installation process, XP Server cannot run. When you execute an extended stored procedure (ESP), you see an error message similar to the following:

```
Msg 11018, Level 16, State 1:
Procedure 'xp_cmdshell', Line 2
XP Server must be up for ESP to execute.
```

To add this required information to the `sysservers` table manually, enter:

```
sp_addserver SERVERNAME_XP, NULL, SERVERNAME_XP
```

where `servername` is the name of the Adaptive Server.

**Stopping servers**

Only the System Administrator has permission to issue a shutdown command. Using a shutdown command minimizes the amount of work that automatic recovery needs to do when the servers are restarted.
Stopping servers

The preferred method of stopping Adaptive Server or Backup Server is to use the Transact-SQL shutdown command.

Stopping Adaptive Server

To shut down Adaptive Server:

1. Use isql to log in to an Adaptive Server account with System Administrator privileges:
   
isql -Usa -Ppassword -Sserver_name

2. Enter:
   
   1> shutdown
   2> go

The default for shutdown uses the with wait option, which allows Adaptive Server to finish executing SQL statements or procedures, perform a checkpoint in each database, disable new logins, and perform other shutdown tasks.

Issuing the shutdown command prints a message like this to the sterr file:

   Server SHUTDOWN by request. The SQL Server is terminating this process.
   CT-LIBRARY error:

This is normal behavior. If the message indicates that Adaptive Server is waiting for processes to complete, and you must stop Adaptive Server immediately, you can use shutdown with nowait which neither wait for currently executing statements to finish, nor performs checkpoints in every database.

Note Using the shutdown with nowait command is not recommended; use it only when necessary.

Stopping Backup Server

To shut down a Backup Server:

1. Use isql to log in to a server with System Administrator privileges.

2. Enter:

   1> shutdown SYB_BACKUP
After you shut down a Backup Server, you must wait at least 30 seconds before restarting it.

Issuing the `shutdown` command prints a message similar to the following to the `stderr` file:

```
Backup Server: 3.48.1.1: The Backup Server will go down immediately.
Terminating sessions.
```

This is normal behavior. If a message indicates that Adaptive Server or Backup Server is waiting for processes to complete, and you must stop Adaptive Server or Backup Server immediately, use `shutdown with nowait`, which neither waits for currently executing statements to finish, nor performs checkpoints in every database.

Using `shutdown with nowait` for Backup Server can cause inconsistent or incomplete dumps and loads. Use this command only when necessary.

For more information on the `shutdown` command, see the *Reference Manual: Commands*.

---

**Using the kill command**

**Warning!** Use the `kill` command to stop Adaptive Server and Backup Server only as a last resort.

When possible, use the Transact-SQL `shutdown` or `shutdown with nowait` command. Do not use `kill` with the `-9` flag, because it exits the server without running a checkpoint to ensure that all database changes are written to the database device. Adaptive Server may also exit without removing associated shared memory files and network handlers.

Because Adaptive Server and Backup Server are background processes, they can be killed from the operating system by their owner or by “root” with the UNIX `kill` command. The syntax is:

```
kill pid
```

where `pid` is the process identification of any `dataserver` or `backupserver` process, as determined by the `showserver` command. Killing one engine for a particular Adaptive Server kills all engines for that server.
Stopping servers

If more than one Adaptive Server is running on the same system, be careful that
the engine you kill is associated with the correct Adaptive Server. If your
Adaptive Server is configured to use multiple engines (CPUs), each engine has
an associated operating system process. The correct way to kill a multi-engine
server is to specify the process ID for engine 0.

This showserver output shows the processes for a four-engine server:

```
showserver

UID PID PPID  C STIME  TTY  TIME   COMD
jorge 3320  1  80 10:31:40 pts/4 302:15 dataserver -dteamster
jorge 3321 3320 80 10:31:45 pts/4 324:47 dataserver -ONLINE:1
jorge 3322 3320 80 10:31:45 pts/4 326:02 dataserver -ONLINE:2
jorge 3323 3320 80 10:31:45 pts/4 328:56 dataserver -ONLINE:3
```

This example shows four running dataserver processes with operating system
process identifications (PID) 3320, 3321, 3322, and 3323 (dataserver is the
executable form of the Adaptive Server program.)

Child engine processes for the dataserver have the -ONLINE: argument.

Each child engine has a parent process identification (PPID) that is equal to the
process identification (PID) of the parent. In the example above, the PID of the
parent server is 3320. The other three engines spawned by the parent process
have the same PPID.

If the PPIDs appear to be unrelated, and there is more than one dataserver
process, then more than one Adaptive Server is running on the system.

Shutdown and shared memory files

When Adaptive Server starts, it creates SERVER_NAME.krg file in
$SYBASE/$SYBASE_ASE directory to store information about shared memory
segments that it uses.

If Adaptive Server is configured with a memory size that is greater than the
MAXSHMSEGSIZE parameter in the operating system then Adaptive Server
may create additional shared memory segments and for every additional shared
memory segment that it creates, an additional file with
SERVER_NAME.srg[N] (where N ranges from 0 – N), is created under
$SYBASE/$SYBASE_ASE.
When Adaptive Server is shut down in a normal manner, the shared memory files are automatically removed. If Adaptive Server crashes or is stopped with the `kill -9` command, these files are not deleted. You need read and write permissions on these files to restart Adaptive Server after a crash or a `kill -9` command, because Adaptive Server must be able to overwrite the previously created shared memory files.

If Adaptive Server or Backup Server is killed abnormally it also leaves shared memory segments. Use the `ipcs` and `ipcrm` commands to identify and remove these shared memory segments that have “NATTACH” count as “0”.

See the UNIX man pages for more information about `ipcs` and `ipcrm`.

### Huge pages on Linux

The CPU-Cache translation lookaside buffer (TLB) stores information about conversions from an virtual page address to the physical page address, and every byte access to physical memory requires a conversion (called a “cache miss”). Although these cache misses are very expensive, you can improve the TLB hits by enabling “huge pages” on Linux machines.

**Note** Adaptive Server supports huge pages for x64 and P-series Linux versions.

Huge pages use fewer pages to cover the physical address space, so the size of “book keeping” (mapping from the virtual to the physical address) decreases, requiring fewer entries in the TLB and improving the system performance.

Adaptive Server version 15.0.3 and later allocates shared memory using huge pages by default. However, if the system does not have enough huge pages—or is not configured for huge pages,—Adaptive Server uses regular pages, and writes this message to the error log:

```
Could not allocate memory using Huge Pages. Allocated using regular pages. For better performance, reboot the server after configuring enough Huge Pages
```

Adaptive Server adjusts its shared memory up to the nearest multiple of 256MB. For example, if you configure Adaptive Server with 800MB of shared memory, it is rounded off to 1GB (some versions of Linux do not allow you to allocate huge pages if the size is not a multiple of `Hugepagesize`).

Before starting Adaptive Server, check `/proc/meminfo` to make sure Linux already has huge pages configured:
Stopping servers

cat /proc/meminfo

....
HugePages_Total: 32
HugePages_Free: 32
Hugepagesize: 16384 kB

**Note** Memory you allocate for huge pages is used for the shared memory only. If you allocate too many huge pages, this may lead to Adaptive Server excessively swapping physical pages. You should only allocate the required number of huge pages.
CHAPTER 3

Configuring the Operating System

This chapter discusses the operating system configuration settings that you can adjust after installing or upgrading Adaptive Server. Unless stated otherwise, the information pertains to all supported UNIX platforms. The installer on Adaptive Server 15.0.3 and later lets you configure some Adaptive Server settings during the installation process. See the Installation Guide for more details.

Verifying environment variables

You may want to verify that the Adaptive Server environment variables discussed in “Environment variables” on page 3 are set correctly.

To check the current value of environment variables, enter the following command at the operating system prompt:

```
env
```
Configuring new servers with *srvbuild* or *srvbuildres*

Use *srvbuild* to configure a new server with default or user-specified values for key configuration attributes. Use *srvbuildres*, in non-GUI mode, to configure a new server using a resource file. For more information on *srvbuildres*, see the Utility Commands Reference chapter in the *Utility Guide*.

To configure new servers with *srvbuild*:

1. Source *SYBASE.csh* or *SYBASE.sh* file in $SYBASE to setup the environment variables.
2. Run $SYBASE/$SYBASE_ASE/bin/srvbuild.
3. The Select Servers to Build window displays. Choose the servers you want to configure by clicking the boxes on the left. When you select each of the boxes, the server name text field is enabled where you can enter the server name.

   **Note** The list of “Server type” available for you to choose depends on what you have installed in $SYBASE.

4. When you have selected the servers you want to build, click OK. The next screen or screens displays information depending on what you chose to configure.

5. On the Adaptive Server type screen, select the:
   - Adaptive Server Application type – Based on how you plan to use the new server, choose:
     - MIXED – for both OLTP and DSS.
     - OLTP – for online transaction processing. This is generally used for a high rate of smaller, less complex transactions.
     - DSS – for decision support systems. Generally these systems have less update activity with large complex queries.
   - Server page size – Depending on the Adaptive Server Application type, select:
     - 4K – for MIXED and OLTP
     - 8K – for DSS
   - Master device path
   - Master device size
CHAPTER 3 Configuring the Operating System

- Master database size
- Sybsystemprocs device path
- Sybsystemprocs device size
- Sybsystemprocs database size
- Error log
- Transport type
- Host name
- Port number

In the Edit Advance Adaptive Server Attributes tab, enter:
- Adaptive Server configuration file
- Sybsystemdb device path
- Sybsystemdb device size
- Sybsystemdb database size
- Share memory file directory
- Default backup server
- tempdb information – Adaptive Server 15.0.3 ESD #1 and later versions create tempdb on a specified separate device. The default tempdb device and database size is 100MB. Specify these attributes for tempdb:
  - tempdb device path
  - tempdb device size
  - tempdb database size
- Enable PCI in Adaptive Server
  If you enable PCI in Adaptive Server, also set:
  - sybpcidb device path
  - sybpcidb device size
  - sybpcidb database size
- Optimize Adaptive Server configuration values:
  After configuring a server, srvbuild prompts you for these values:
• Available physical memory for Adaptive Server
• Available CPU for Adaptive Server

When a new server is created using configuration tools, the default configuration that it is configured with may not optimally use all the resources available on the system. Input new values based on the system resource usage. If the value specified is larger than the available resource for allocation to the server then the optimization may fail, causing the server to not start. The current default input values for available physical memory and available CPU are 80% of the physical memory and CPU number.

Note The Adaptive Server 15.0.3 installer allows you to tune basic configuration settings during installation, instead of as a postinstallation task. See the Installation Guide for more details.

6 To configure Backup Server, in the Backup Server configuration screen, select the:
• Error log
• Tape configuration file
• Language
• Character set
• Maximum number of network connections
• Maximum number of server connections
• Transport type
• Host name
• Port number

7 In the XP Server type screen, select:
• Transport type
• Host name
• Port number

8 To configure Job Scheduler, enter the following information in the Job Scheduler type screen:
• Sybmgmtdb device path
CHAPTER 3 Configuring the Operating System

- Sybmgmtdb device size
- Sybmgmtdb database size
- Transport type
- Host name
- Port number

9 In the “Configure Self Management...” screen, you can enter:
- Enable Self Management
- Self Management user name
- Self Management user password

If you change the Self Management user password after configuring the Self Management, you need to rerun:

```
sp_addexternlogin loopback, <Self Management user name>, <Self Management user name>, <new Self Management user password>
```

10 Click Build Servers! to proceed with the servers configuration. srvbuild displays the configuration status as it builds the servers you selected.

11 After the server configuration completes, click Exit to quit srvbuild.

Using the `stty` setting

Setting the `stty tostop` option causes a background Adaptive Server to stop as soon as it tries to write to the terminal. To avoid this error, execute the following command before starting Adaptive Server:

```
stty -tostop
```

If you are redirecting all Adaptive Server output to files, you do not have to change the `stty` setting.
Restoring correct permissions

Sybase software files and directories are installed with the correct access permissions. If you notice that the permissions are no longer correct, you can restore the correct permissions with the script `setperm_all`, located in the `$SYBASE/$SYBASE_ASE/install` directory.

File descriptors and user connections

The number of user connections used by Adaptive Server cannot exceed the number of file descriptors available to Adaptive Server on the operating system. When configuring user connections on Adaptive Server, the System Administrator should take into account the number of file descriptors available per process. Although most of the open file descriptors are available for user connections, a few are used by Adaptive Server for opening files and devices.

For HP-UX

The kernel parameters `maxfiles` and `maxfiles_lim` control the number of file descriptors available to any one process. The limit is 10,000 for HP-UX on a 32-bit system, and 60,000 on a 64-bit system.

To obtain the current values on the file descriptors use the Korn or Bourne shell `ulimit` command:

```
ulimit -n
```

For AIX

The number of file descriptors per process is determined by the operating system parameter `open_max`. The default value of `open_max` is 32767. Adaptive Server can use a maximum of 2000 file descriptors per engine, regardless of the value of `open_max`. For more information on setting `open_max`, see the AIX operating system documentation.

To obtain the current value of the `open_max` parameter, use the Korn or Bourne shell `ulimit` command:

```
ulimit -n
```
For Linux

The number of file descriptors per process is limited to 10,000. You can set the number of file descriptors using ulimit.

For Sun Solaris

For Sun Solaris, you can set both soft and hard limits for file descriptors. The soft limit can be increased up to the hard limit by the user, but the hard limit can be increased only by someone with “root” permissions. The soft limit determines the number of open file descriptors available to an Adaptive Server engine. The limit is 10,000.

Although most of the open file descriptors are available for user connections, a few are used by Adaptive Server engines for opening files and devices.

See the System Administration Guide for additional information on user connections.

Displaying current soft and hard limits

To display the current soft limit, for C shells, enter:

```
limit descriptors
```

For Bourne shells, enter:

```
ulimit -n
```

To display the current hard limit for C shells, enter:

```
limit -h descriptors
```

For Bourne shells, enter:

```
ulimit -Hn
```

Increasing the soft limit

To increase the soft limit for C shells, enter:

```
limit descriptors n
```

For Bourne shells, enter:
ulimit -Sn new_value

where $n$ is the current value for the soft limit, and $new\_value$ is the value to which you want to increase the soft limit.

**Note** You can use the preceding commands in your $RUN\_server\_name$ file to increase the hard and soft limits. The $RUN\_server\_name$ file is a Bourne shell script; be sure to use the Bourne shell versions of these commands in the $RUN\_server\_name$ file.

## Increasing the hard limit

To increase the hard limit, use a program like the sample program shown in “Sample program” on page 39.

- **Setting up the sample program to increase the hard limit**

1. Create $file\_name.c$ (where $file\_name$ is the name you give the file), by using an ASCII text editor. Type the text shown in the sample in “Sample program” on page 39.

2. Compile the file:

   ```
   cc file\_name.c -o program\_name
   ```

   where $file\_name$ is the name of the source file you created, and $program\_name$ is the name you want to give the program.

3. Change the program’s permissions and ownership so that it will execute as “root”:

   ```
   chmod 755 program\_name
   chown root program\_name
   ```

   where $program\_name$ is the name of the compiled program.

4. The “root” user can use the program to start Adaptive Server with increased user connections by typing the following command at the operating system prompt:

   ```
   # program\_name dataserver -d master\_device\_name
   ```
where \textit{program\_name} is the name of the compiled program, and 
\textit{master\_device\_name} is the full path of Adaptive Server’s master device. 
Instead of typing the command at the operating system prompt, you can add \textit{program\_name} preceding the \texttt{dataserver} command line in the 
Adaptive Server \texttt{RUN\_server\_name} file.

\section*{Sample program}

\textbf{Note} This is an sample script; modify it as necessary.

The following example shows the source code that you can use to increase the 
hard limit:

\begin{verbatim}
#include <sys/time.h>
#include <sys/resource.h>
#include <sys/types.h>
/*
 ** define MAX_CONNECTIONS to a number less than 
 ** 10000. The number defined will then become the maximum 
 ** number of connections allowed by an Adaptive Server.
 */
#define MAX_CONNECTIONS 9999
extern int errno;
main(argc,argv)
char **argv;
{
    struct rlimit rlp;
    uid_t uid;

    rlp.rlim_cur = MAX_CONNECTIONS;
    rlp.rlim_max = MAX_CONNECTIONS;
    /* set the number of open file descriptors to 
    MAX_CONNECTIONS */
    if (setrlimit (RLIMIT_NOFILE,&rlp) == -1)
    {
        perror("setrlimit");
        exit(1);
    }

    /* reset the user id to disable superuser 
     privileges */
    uid = getuid();
}
\end{verbatim}
Enabling asynchronous disk I/O

Sybase recommends that you do not use block devices on UNIX and Linux systems. A system crash may lead to loss of data integrity if you use block devices as database devices.

If you are using raw devices on a UNIX platform, you cannot:

- Set the disk init...directio or dsync parameter to true
- Set the sp_deviceattr...directio or dsync parameter to true

If either of these are set to true, Adaptive Server returns a similar to:

```
You cannot set directio option for raw device
'/dev/raw/raw235' or You cannot set attribute dsync for raw device
```

Linux

Verify there are sufficient system resources for kernel asynchronous I/O before starting Adaptive Server.

Linux requires that the total number of system-wide reservable I/O descriptors (aio-max-nr), minus the current number of descriptors reserved by all processes (aio-nr), must be sufficient to accommodate the value for max async i/os per engine multiplied by the value for max online engines.

To determine the values for aio-max-nr and aio-nr:

```
cat /proc/sys/fs/aio-max-nr
```

```
cat /proc/sys/fs/aio-nr
```

Modify the number of reservable descriptors by setting fs.aio-max-nr with the sysctl operating system command.

See your operating system documentation.
Adaptive Server may issue messages similar to these in the errorlog if there are insufficient number of I/O descriptors during startup, or when an engine is brought online:

```
kernel KAIO not initialized because the requested number of async I/Os (%d) will exceed the resources available on the operating system.
```

```
kernl Kernel asynchronous I/O not initialized. The io_setup() system call returned %d.
```

See max async i/os per engine in the *System Administration Guide, Volume 1*.

**HP-UX 11.31 and later using file system devices**

**Enabling asynchronous I/O on HP-UX running with file system devices**

Before enabling asynchronous I/O for file system devices, Sybase recommends you to adjust below OS parameters to their maximum values:

```
# kctune aio_max_ops=0x100000
# kctune aio_proc_threads=2048
```

1. Enable the `allow sql server async i/o` configuration parameter.
2. Enable the `enable hp posix async i/o` configuration parameter.

```
sp_configure 'enable hp posix async i/o', 1
```

3. This parameter is static: restart Adaptive Server.

See “Setting Configuration Parameters” in the *System Administration Guide, Volume 1*.

**Enabling Concurrent I/O on HP-UX running with file systems**

Sybase recommends that you to enable VxFS Concurrent I/O (CIO) on HP-UX file systems to take advantage of its performance benefits. VxFS Concurrent I/O is available with OnlineJFS (VxFS-Full) version 5.0.1 and later for HP-UX 11.31.

- To verify that OnlineJFS is installed and enabled, enter:

```
# vxlicrep | grep -i onlinejfs
```

Or,

```
# swlist -l product | grep -i onlinejfs
```

See your operating system documentation.

**HP-UX on raw partitions**

It is mandatory for HP-UX users to enable asynchronous I/O.
Enabling asynchronous disk I/O

To improve I/O performance on character or raw and block devices, enable asynchronous I/O by installing the HP asynchronous I/O driver from SAM. For help with installing this driver, contact your operating system administrator or HP technical support.

**Note** Before executing the following instructions, shut down Adaptive Server (or SQL Server).

To enable asynchronous I/O:

1. From the SAM Kernel Configuration menu, choose Drivers and set the Pending State for `asyncdisk` to In by adding the driver.
   
   An alternate step could be done by adding the `asyncdsk` subsystem keyword to `/stand/system`.

2. Rebuild the kernel, and reboot the system.

3. Execute the following commands using the `userid` root:
   ```
   #/etc/mknod/dev/async c 101 4
   #chmod 0660/dev/async
   #chown <uid> /dev/async
   #/etc/setprivgrp <ugrp> MLOCK
   ```

   Where:

   `<uid>` is the identification (user ID) used by the user that is booting Adaptive Server.

   `<ugrp>` is the user group for the `<uid>` user ID.

4. At the UNIX prompt, execute the following statements as "root". The user ID of the user who is starting Adaptive Server and Backup Server must be the owner of the `/dev/async` directory.

   **For IBM AIX**

   Enable asynchronous disk I/O.

   This step is mandatory for IBM users.

   Enable asynchronous I/O by adjusting the kernel parameters, using the System Management Interface Tool (SMIT):

   1. Enter “smit” at the UNIX prompt.

   2. From the Devices menu, select Asynchronous I/O.

   3. Select Change/Show Characteristics of Asynchronous I/O.
Adjusting the client connection timeout period

Adaptive Server uses the KEEPALIVE option of the TCP/IP protocol to detect clients that are no longer active. When a connection to a client is inactive for a period of time (the timeout period), the operating system sends KEEPALIVE packets at regular intervals. If it does not receive a response from the client machine for any of these packets, the operating system notifies Adaptive Server that the client is no longer responding. Adaptive Server then terminates the client’s connection.

The KEEPALIVE default timeout period is 2 hours (7,200,000 ms). To display the current time value, use the command for your platform as shown in the following sections.

For HP-UX 11 and later

To display the current timeout period enter:

```
/ndd -get/set /dev/tcp tcp_keepalive_interval
```

The `tcp_keepalive_interval` parameter specifies the length of time (measured in seconds) to keep an idle connection active before the system checks to see if the connection died.

To change the timeout period, use the `nettune -s` or `ndd -set` command.

For IBM RS/6000

To display the current timeout value, enter:

```
/usr/sbin/no -o tcp_keepidle
```

The `tcp_keepidle` parameter specifies the length of time (measured in half seconds) to keep an idle connection active before the system checks to see if the connection died. The default is 14,400 half seconds (7200 seconds, or 2 hours).

IBM recommends a value of at least 15 minutes.

For Sun Solaris

To display the timeout value, enter:
Checking for hardware errors

```
/usr/sbin/ndd -get /dev/tcp tcp_keepalive_interval
```

To reduce the timeout period to 15 minutes (900,000 ms.), enter:

```
/usr/sbin/ndd -set /dev/tcp tcp_keepalive_interval 900000
```

**For Linux**

To display the timeout value, enter:

```
/sbin/sysctl -e net.ipv4.tcp_keepalive_time
```

To reduce the timeout period to 15 minutes (900 seconds,) enter:

```
/sbin/sysctl -w net.ipv4.tcp_keepalive_time=900
```

**Checking for hardware errors**

The following types of hardware error messages indicate problems that may lead to database corruption:

- Disk read, write, or retry errors
- Timeouts
- System panics
- Memory problems of any type

**For HP-UX**

Check the `/var/adm/syslog/syslog.log` file on a regular basis. You can view the file directly, or you can use the HP-UX `dmesg` command. See your operating system documentation for more information.
For IBM RS/6000

The `errpt` command includes several options for limiting the report to events that match certain criteria. Use the `errpt` command on a regular basis. If errors appear, use the diagnostic tool `diag` to check your memory and disks. Or use the System Management Interface Tool (SMIT) to run the `errpt` command. This command may produce a lot of output.

For Sun Solaris

Check the `/var/adm/messages` file on a regular basis. If any of the types of hardware errors described in the beginning of this section appear, use the Sun Microsystems diagnostic tool, `sundiag`, to check memory and disks. See the operating system documentation for more information.

For Linux

Check the `/var/log/messages` file on a regular basis. See the operating system documentation for more information.

Monitoring the use of operating system resources

The *System Administration Guide: Volume 2* discusses maintaining the optimal number of Adaptive Server engines for your workload and system configuration. To determine the optimal number, monitor system and CPU usage.

For HP-UX

HP-UX supplies many tools to help monitor performance, some of which are described as follows:

- The `sar` command reports relative and absolute I/O throughput rates to each disk and controller.
- The `vmstat` command monitors virtual memory usage.
Monitoring the use of operating system resources

- The netstat command monitors network status.
- The ps command provides a snapshot of accumulated CPU time and usage for individual processes.
- The time command can be useful in determining the various user, system, and real-time resources used over a complete run.

For details about these tools, see your operating system documentation.

For IBM RS/6000

IBM RS/6000 supplies the following tools for monitoring performance:
- The iostat command reports the amount of I/O on terminals and hard disks and how CPU time is spent.
- The vmstat command monitors virtual memory usage.
- The netstat command monitors network status.
  - netstat -v displays Transmit/Receive Statistics. It is also used to determine if enough buffers have been configured for network traffic.
  - no -a command displays current network options. It is also used for tuning mbuf pools.
- The ps command provides a snapshot of accumulated CPU time and usage for individual processes.
- The time command determines the various user, system, and real-time resources during a complete run.

For details about these tools, see your operating system documentation.

For Sun Solaris and Linux

Sun Solaris and Linux supplies the following tools to help monitor performance:
- The iostat command reports the amount of I/O on terminals and hard disks and how CPU time is spent.
- The vmstat command monitors virtual memory usage.
- The netstat command monitors network status.
• The *ps* command gives you an accurate snapshot of accumulated CPU time and usage for individual processes. This can be very helpful in determining the dataserver-, engine-, and process-specific loading.

• The *time* command can be useful in determining the various user, system, and real-time resources used over a complete run.

For details about these tools, see your operating system documentation.

### Checking the integrity of databases

Running *dbcc* checks and performing database backups protects the integrity and recoverability of your Adaptive Server databases.

Use *dbcc checkstorage* to run regular consistency checks on large databases when the amount of time you require to perform consistency checks is less than the amount of time required for *dbcc checkalloc* or *dbcc checkdb* to complete. *dbcc checkstorage* performs a fast consistency check, but it is not as thorough as *dbcc checkalloc* or *dbcc checkdb* See “Preparing to use *dbcc checkstorage*” in Chapter 10, “Checking Database Consistency,” in the System Administration Guide, Volume 2, and the Reference Manual: Commands.

You can also run the following sample C shell script which calls several *isql* scripts to help you do this:

```bash
#!/bin/csh -f
if ( -e dbcc_mail.out) then
  rm dbcc_mail.out
endif
foreach i (*.dbcc)
isql -Usa -Ppassword < $i > dbcc_out
if ( 'grep -c 'Msg 25[0-9]' dbcc_out' ) then
  echo "There are errors in" $i >> dbcc_mail.out
cat dbcc_out >> dbcc_mail.out
else
  echo "Backing up " $i:r >> dbcc_mail.out
  isql -Usa -Ppassword < $i:r.backup
endif
end
mail -s "Backup Report" jjones < dbcc_mail.out
```

The first set of scripts (one for each database with a file name appended with ".dbcc") runs *dbcc checkalloc* and *dbcc checkdb* for each database and sends the messages to an output file called *dbcc_out*. 

For example, the script `master.dbcc` runs `dbcc` to check the `master` database:

```
dbcc checkalloc (master)
go
dbcc checkdb (master)
go
```

The C shell script then runs the `grep` command to find 2500-level error messages in the `dbcc` output. The results of the `grep` command go into an output file called `dbcc_mail.out`.

Next, the script invokes an `isql` backup script for each database for which no 2500-level errors occurred and adds the “Backing up `database_name`” line to `dbcc_mail.out`. For example, the script `master.backup` backs up the `master` database:

```
use master

go
dump database master to master_dump

go
```

You may want to add appropriate `dump transaction` commands to your scripts.

If there are 2500-level error messages, the script does not back up the database. At the end of the script, `dbcc_mail.out` is mailed to the System Administrator “jjones,” who then has a record of fatal `dbcc` errors and successful backups.

You can tailor the sample shell and `isql` scripts to suit the needs of your installation.

To have the scripts execute automatically, edit the `crontab` file, and add an entry similar to this:

```
00 02 * * * /usr/u/sybase/dbcc_ck 2>&1
```

This example executes a C shell script called `dbcc_ck` every morning at 2:00 a.m.
CHAPTER 4

Adaptive Server Default Configuration

When you install or upgrade Adaptive Server, it includes some default parameter settings and a few of its auxiliary programs.

After installing and testing this “default” Adaptive Server, you must configure it to your system’s needs and install other optional features.

For information about configuring Adaptive Server and Backup Server, see the *System Administration Guide Volume 1* and *Volume 2*.

For information about configuring languages, character sets, and sort orders, as well as optional features, see the *System Administration Guide: Volume 1*.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default settings</td>
<td>49</td>
</tr>
</tbody>
</table>

Default settings

After installation, Adaptive Server default settings are as listed in Table 4-1. You may need to configure these settings to suit your computer and database needs.

*Table 4-1: Defaults for Adaptive Server parameter settings*

<table>
<thead>
<tr>
<th>Item</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Servername</td>
</tr>
<tr>
<td>Transport Type</td>
<td>TCP/IP</td>
</tr>
<tr>
<td>Port number</td>
<td>5000</td>
</tr>
<tr>
<td>Error log path</td>
<td>$SYBASE/$SYBASE_ASE/install/servername.log</td>
</tr>
<tr>
<td>Event logging</td>
<td>Not configured</td>
</tr>
<tr>
<td>International support (Localization)</td>
<td></td>
</tr>
<tr>
<td>• Language</td>
<td>us_english</td>
</tr>
</tbody>
</table>
**Default settings**

Table 4-2 lists the default settings for the Backup Server and XP Server. For more information about these servers, see “Introduction” on page 1.

### Table 4-2: Defaults for Backup and XP Servers

<table>
<thead>
<tr>
<th>Server</th>
<th>Item</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backup Server</td>
<td>Name</td>
<td>AdaptiveServername_back</td>
</tr>
<tr>
<td></td>
<td>Network support</td>
<td>(TCP/IP)</td>
</tr>
<tr>
<td></td>
<td>Socket number</td>
<td>5001</td>
</tr>
<tr>
<td></td>
<td>Error log path</td>
<td>SSYBASE/SSYBASE_ASE/install/AdaptiveServername_back.log</td>
</tr>
<tr>
<td>XP Server</td>
<td>Name</td>
<td>SSYBASE/SSYBASE_ASE/ADAPTIVESERVERNAME_XP</td>
</tr>
<tr>
<td></td>
<td>Network support</td>
<td>(TCP/IP)</td>
</tr>
<tr>
<td></td>
<td>Socket number</td>
<td>5002</td>
</tr>
<tr>
<td></td>
<td>Error log path</td>
<td>N/A</td>
</tr>
</tbody>
</table>

- **Character set**
  - HP – Roman8
  - IBM – ISO 8859-1
  - Sun – ISO 8859-1
  - Linux – iso_1

- **Sort order**
  - Linux – Binary ordering

- **Login security mode**
  - Standard
Setting Up Communications Across the Network

Adaptive Server can communicate with other Adaptive Servers, Open Server applications, and client software across a network. Clients can communicate with one or more servers, and servers can communicate with other servers via remote procedure calls.

Directory services contain information about the network locations of servers. Directory services contain entries for all Adaptive Servers, Backup Servers, and other server products on the network.

In the Sybase client/server environment, a client can connect with Adaptive Server if it knows where the server resides on the network and if the server supports the client’s language or character set. When a client initiates a connection, it looks in its directory services for the network location of the target server.

Directory services list the name and address of every server, including Backup Server and XP Server. When you are using a client program, and you want to connect with a particular server, the client program looks up the server name in the directory services and connects to that server.
How Adaptive Server determines which directory service entry to use

Servers also need network information. When a server starts up, it looks in its interfaces file to determine where to listen for client connection requests. In addition, Adaptive Server can take on a client role when it makes remote procedure calls to other Adaptive Servers.

Table 5-1 shows where to find more information on server and client interfaces file tasks and topics.

<table>
<thead>
<tr>
<th>Type of interfaces file</th>
<th>Task or topic</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNIX server or client</td>
<td>Adding entries for multiple Adaptive Server installations</td>
<td>Chapter 7, “Customizing Localization for Adaptive Server”</td>
</tr>
<tr>
<td></td>
<td>Creating a master interfaces file for multiple installations</td>
<td>“Creating a master interfaces file” on page 60</td>
</tr>
<tr>
<td></td>
<td>Configuring for multiple networks</td>
<td>“Configuring interfaces files for multiple networks” on page 61.</td>
</tr>
<tr>
<td></td>
<td>Reference information</td>
<td>“Understanding the format of the interfaces file” on page 57.</td>
</tr>
<tr>
<td>PC-client</td>
<td>Configuring a client</td>
<td>Installation Guide for your platform</td>
</tr>
<tr>
<td></td>
<td>Reference information and instructions for advanced tasks</td>
<td>Open Client and Open Server Programmer’s Supplement for your PC-client platform, or the appropriate Open Client documentation</td>
</tr>
<tr>
<td>Client platforms not listed</td>
<td>Configuring, reference information, and instructions for advanced tasks</td>
<td>Open Client and Open Server Programmer’s Supplement for your PC-client platform, or the appropriate Open Client documentation</td>
</tr>
</tbody>
</table>

How Adaptive Server determines which directory service entry to use

Adaptive Server uses directory services to determine the address at which it should listen for clients. When you start Adaptive Server, it performs the following steps:

1. It looks for the server name supplied in the command line -s option. If the server name is not supplied in the command line:

2. It determines its own name by checking the value of the DSLISTEN environment variable. If the DSLISTEN environment variable is not set, then it assumes that the server name is SYBASE.
3 Looks in directory services for an entry whose name matches the name found in the steps above.

4 It uses the network information provided by the directory services entry it has found to listen for client connections.

**How a client uses directory services**

When a client connects to a server it:

- Determines the name of the server either programmatically or by referring to the DSQUERY environment variable. If the application user has not set DSQUERY, the runtime value for the server name defaults to the SYBASE environment variable.

- Looks in directory services for an entry whose name matches the name of the server.

- Uses the network information provided by the directory services entry to connect to the server. If the client cannot connect the first time, it makes additional attempts according to the delay and retry numbers indicated in directory services. If no matching entry is found, an error message is written to the client’s standard error file. If multiple networks are supported, the client attempts to connect using the information in the second network address entry for the server.

The Open Client documentation discusses client connections in much greater detail. See the *Open/Client Programmer’s Supplement* for your client platform or the appropriate Open/Client documentation.

**Creating a directory services entry**

The installation program, srvbuild, automatically creates a directory services entry for each server installation. You can also use the following Sybase utilities to edit the network information in directory services:

- dsedit – an X-Windows GUI utility.
- dscp – a UNIX command line utility.

For details on using these utilities, see the Adaptive Server *Utility Guide*.
Supported directory drivers

There are three supported drivers:

- interfaces driver
- Lightweight Directory Services driver (LDAP)
- Cell Directory Service (CDS) provided by Distributed Computing Environment (DCE)

This remainder of this chapter describes the interfaces file and provides specific configuration information for each supported UNIX platform. For information about LDAP drivers, Cell Directory Services, and for a comparison between interfaces files and LDAP directory services, see the Open Client/Server Configuration Guide for your platform.

Contents of an interfaces file

An interfaces file contains network information about all servers on your network, including Adaptive Server, Backup Server, and XP Server, plus any other server applications, such as Replication Server, and any other Open Server applications.

The network information in the file includes the server name, network name or address of the host machine, and the port, object, or socket number (depending on the network protocol) on which the server listens for queries. See “Understanding the format of the interfaces file” on page 57 for the specific makeup of the interfaces file entry.

Each entry in an interfaces file can contain two types of lines:

- Master lines, which are used by server applications to listen for queries over the network. This information is called a listener service.
- Query lines, which are used by client applications to connect to servers over the network. This information is called a query service.

The network information contained in the master and query lines for a server is identical because a server listens for connection requests on the same port that clients use to request connections.

A server needs both master and query lines in its interfaces file, since servers sometimes act as clients to other servers.
A client’s interfaces file does not need a master line. It functions correctly with only a query line.

If you have more than one Adaptive Server installation, each server’s interfaces file should contain information about all servers on the network.

If all of your server products are running on the same platform, you can create one master interfaces file and copy that file to each machine. For more information, see “Creating a master interfaces file” on page 60.

If the host machine supports multiple networks, see “Configuring interfaces files for multiple networks” on page 61.

**Heterogeneous and homogeneous environments**

You can run Adaptive Server and clients on the same platform or on different platforms.

If the platforms are different, each platform may require a different format and configuration for its interfaces file. Figure 5-1 illustrates how a PC client uses network information in its interfaces file (sql.ini) to connect to Adaptive Server running under UNIX, and how Adaptive Server uses its interfaces file to connect to another server during a remote procedure call.
If both a client and a server are running under UNIX, the same interfaces file is valid for both. Figure 5-2 illustrates how clients and Adaptive Servers running in a homogeneous environment can use copies of the interfaces file to establish connections. Because the two Adaptive Servers are running under the same operating system, they can use the same interfaces file or exact copies of the same file.
Understanding the format of the *interfaces* file

The following rules apply to the format of interfaces file entries:

- Each Adaptive Server has only one entry, although there may be multiple lines in the entry.
- Each line that follows the *servername* line must begin with a space or a tab.
- Each element on the line must be separated by a single space.
- Each entry is separated by a blank line.
- You can add comments to an interfaces file by adding a pound sign (#) at the beginning of the line and a line break at the end.

There are two interfaces file entry formats, TLI and TCP.

A TLI style entry has this format:

```
Harpo SQL server
Harpo
xxxx
Get Harpo's address and connect.
```

```
Chico SQL server
Chico
xxxx
Get Chico's address and connect.
```

```
Get Chico's address and make an RPC.
```
Understanding the format of the interfaces file

```
servername retry_attempts delay_interval<newline>
<tab>service_type api protocol device address filter<newline>
<tab>ha_failover servername<newline>
```

For example:

```
harpo_server1
  master tli tcp /dev/tcp 0x0002333337f00001
  query tli tcp /dev/tcp 0x0002333337f00001
```

A TCP style entry has this format:

```
servername retry_attempts delay_interval<newline>
<tab>service_type protocol network machine port filter<newline>
<tab>ha_failover servername<newline>
```

For example:

```
chico_server1
  master tcp ether chico 5678 ssl
  query tcp ether chico 5678 ssl
```

Components of an interfaces file entry

Table 5-2 describes the components of an interfaces file entry.

<table>
<thead>
<tr>
<th>Component</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>servername</code></td>
<td>Name of Adaptive Server or Backup Server. Requirements for a server name are:</td>
</tr>
<tr>
<td></td>
<td>• The name cannot be more than 30 characters long.</td>
</tr>
<tr>
<td></td>
<td>• The first character must be a letter (ASCII a through z, A through Z).</td>
</tr>
<tr>
<td></td>
<td>• The characters that follow must be letters, numbers, or underscores (_).</td>
</tr>
<tr>
<td><code>retry_attempts</code> (optional)</td>
<td>Number of times you want the client to try to connect to a server after initial failure. Default is 0.</td>
</tr>
<tr>
<td><code>delay_interval</code> (optional)</td>
<td>Number of seconds between connection attempts. Default is 0.</td>
</tr>
<tr>
<td><code>service_type</code></td>
<td>Type of service defined by entry. Must be one of the following:</td>
</tr>
<tr>
<td><code>api</code></td>
<td>Application programming interface available to the network. The supported value is tli.</td>
</tr>
</tbody>
</table>
### Chapter 5  Setting Up Communications Across the Network

<table>
<thead>
<tr>
<th>Component</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>protocol</td>
<td>Name of the network protocol. Protocol types available are:</td>
</tr>
<tr>
<td></td>
<td>• TCP/IP, represented by the letters “tcp”</td>
</tr>
<tr>
<td>network</td>
<td>Name of the network; not currently used by Adaptive Server. srvbuild enters “ether” as a placeholder.</td>
</tr>
<tr>
<td>host</td>
<td>Network name or address of server’s host machine.</td>
</tr>
<tr>
<td></td>
<td>• For TCP/IP, use either the host name or Internet address. Maximum size of entry is 32 bytes.</td>
</tr>
<tr>
<td></td>
<td>To determine the host name of a machine, log in to that machine and enter:</td>
</tr>
<tr>
<td></td>
<td>/bin/hostname</td>
</tr>
<tr>
<td>machine</td>
<td>Network name or address of server’s host machine.</td>
</tr>
<tr>
<td></td>
<td>You can use either the host name or Internet address. Maximum size of entry is 32 bytes.</td>
</tr>
<tr>
<td></td>
<td>To determine the host name of a machine, log in to that machine and enter:</td>
</tr>
<tr>
<td></td>
<td>/bin/hostname</td>
</tr>
<tr>
<td>device</td>
<td>The network device endpoint.</td>
</tr>
<tr>
<td></td>
<td>For TCP networks, the device varies according to the vendor supplying the networking software. Check the vendor-specific documentation for the name of the device. Your network may provide multiple stream devices corresponding to different protocols in the TCP protocol suite. Choose the TCP streams device.</td>
</tr>
<tr>
<td></td>
<td>Common TCP streams devices is /dev/tcp.</td>
</tr>
<tr>
<td>address for a TLI protocol entry</td>
<td>Address consists of:</td>
</tr>
<tr>
<td></td>
<td>• Address prefix, “x” for TLI.</td>
</tr>
<tr>
<td></td>
<td>• Network type, always 0002.</td>
</tr>
<tr>
<td></td>
<td>• Port number converted to four digits, hexadecimal. Must be a unique number between 1025 and 65535. Check the /etc/services file on each machine on the network to see what port numbers are in use. Enter the Adaptive Server port number in a new section of /etc/services labeled “Sybase specific services.” You do not have to make this entry for the operating system to function properly, but the presence of the port number in the file may prevent other users from using the port number.</td>
</tr>
<tr>
<td></td>
<td>• IP network node address of the host machine converted to 8 digits, hexadecimal.</td>
</tr>
<tr>
<td></td>
<td>• Trailing zeros, optional, 16 digits.</td>
</tr>
<tr>
<td>port</td>
<td>A unique port number between 1025 and 65535. Check the /etc/services file on each machine on the network to see what port numbers are in use. Enter the Adaptive Server port number in a new section of /etc/services labeled “Sybase specific services.” You do not have to make this entry for the operating system to function properly, but the presence of the port number in the file may prevent other users from using that port number.</td>
</tr>
<tr>
<td>ha failover</td>
<td>An entry created in directory services or the interfaces file for high availability.</td>
</tr>
</tbody>
</table>
Creating a master interfaces file

A master *interfaces* file contains entries for all Sybase servers on the network. It can be used with every server and client connected to the network. By distributing copies of a master *interfaces* file, you can ensure that all Sybase products on the network interact with one another.

Distributing copies of one interfaces file (a master file) with entries for all Adaptive Servers is the easiest way to maintain consistency in the interfaces files in a homogeneous environment on a network.

You can make all changes to one version of the file and then copy the updated master file to all appropriate Sybase directories.

You can make a master file in one of two ways:

- Using `dsedit` or `dscp`
- Using a text editor

Using `dsedit` or `dscp` to create a master interfaces file

You can use the `dsedit` or `dscp` utility to create a master interfaces file, which you can then distribute to all servers. If you are not an experienced Sybase user, you may find that using `dsedit` or `dscp` is easier than using a text editor. Using `dsedit` or `dscp` also ensures that your interfaces file is consistent in format.

To create a master interfaces file with `dsedit` or `dscp`:

1. Select the interfaces file that contains the most complete, up-to-date information.
2. Begin a `dsedit` or `dscp` session in your latest Sybase installation to edit this interfaces file.
3. Add entries for any Adaptive Servers or Backup Servers that are not listed in this file.

<table>
<thead>
<tr>
<th>Component</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>filter</code></td>
<td>Adaptive Server supports Secure Socket Layers (SSL) as a filter which is appended to the master and query lines of the directory services. SSL is the standard for securing the transmission of sensitive information.</td>
</tr>
</tbody>
</table>
For details on using these utilities, see the Adaptive Server Utility Guide.

Using a text editor to create a master interfaces file

To construct a single master interfaces file from several individual interfaces files:

1. Concatenate the individual interfaces files.
2. Make a copy of the file.
3. Use an ASCII text editor to modify the copy of the concatenated file.

**Note** When you manually edit an interfaces file, be sure that, for each entry, each line following the first line begins with a `<tab>` character.

The following elements must be correct and unique in the resulting file:

- `servername` – each server entry in the interfaces file must be unique. During the `srvbuild` session, you had the choice of entering a server name or accepting the default server name, SYBASE, for those servers. If you accepted the default name, find any duplicate SYBASE entries in your merged file, and rename them.

- A combination of the host machine’s network name or address and Adaptive Server’s port or object number.

- If the original interfaces file was created when there was only one machine on the network, its entries may have the word “loghost” in place of the machine name (address). If `loghost` is present, replace it with the machine name.

Configuring interfaces files for multiple networks

On some platforms, Adaptive Server can accommodate multiple networks. This allows Adaptive Server to listen for clients over multiple network interfaces. You must add an entry for each network interface to the interfaces file.
Configuring the server for multiple network handlers

To configure multiple network listeners:

1. Define a unique host name for each network interface in your operating system’s host database.

2. In your interfaces file, use a text editor to add copies of the “master” line for your Adaptive Server; one for each additional interface you want the server to listen on.

3. Include a unique host name on each line to configure a network handler for each network interface.

4. Port numbers within the interface need not be the same, but they can be. They fall under the same rules for naming and numeric range as the primary network interface.

Sample interfaces files for multiple network handlers

The following example shows an interfaces file for an Adaptive Server with two network interfaces. The server host machine is known as SERV_CORPNET on the corporate network and SERV_ENGNET on the engineering network.

```plaintext
# PRODUCTION server with two network listeners
PRODUCTION    3    3
    master tcp ether SERV_CORPNET 4559
    master tcp ether SERV_ENGNET 5479
    query tcp ether SERV_CORPNET 4559
```

When Adaptive Server restarts, it spawns a network handler process for each master line in the entry that corresponds to the server’s DSLISTEN value. Connections made on each interface are handled equally, as peers.

Configuring the client connections

When an Adaptive Server client scans the interfaces file for a server name, the client uses the first “query” entry it encounters for the server’s entry. This makes configuring clients to use multiple network connections less straightforward than configuring the server ports. You have two choices:

- Use the same DSQUERY name for all clients. The interfaces files on the different machines contain different network names.
• Use different DSQUERY names for the clients. The interfaces files on all the machines are the same, but they contain multiple DSQUERY names.

Using one network-independent DSQUERY name

If uniform client DSQUERY naming is important, you can make the necessary changes in the network addressing of the clients in the interfaces file. You can install separate Sybase installation directories and distinct interfaces files on client file servers on each network to allow users to connect to the correct network address. Instead of altering the DSQUERY name the clients use, you maintain one DSQUERY name for all clients, on all networks, and alter each network’s interfaces file accordingly.

This method assumes that:

• You have complete control over what the Sybase installation clients see on each network.
• The interfaces file (at the very least) is not shared or copied among Sybase installations on different networks.

The interfaces file looks like the following example on the “engineering” network:

```
PRODUCTION	3	3
	query tcp ether SERV_ENGNET 5470
	master tcp ether SERV_ENGNET 5479
```

The interfaces file looks like the following example on the “corporate” network:

```
PRODUCTION	3	3
	query tcp ether SERV_CORPNET 4559
	master tcp ether SERV_CORPNET 4559
```

The “query” line in each file name is different, depending on the network to be used.

The full “master” entry is present in both files. This is allowed because only Adaptive Server will use the “master” lines. Assuming that the server host machine can see both networks (both host names are interchangeable), it does not matter which interfaces file is used for Adaptive Server start-up.
Using different DSQUERY names

To use different DSQUERY names for each network listener:

1. Choose an additional server name.
   
   You can concatenate the original server name and the network name. For example, if your server is named PRODUCTION, you could choose the names PRODUCTION_network1 and PRODUCTION_network2.

2. Do one of the following:
   
   • For PC clients, use sqledit to create multiple sql.ini file entries for the server, one for each network. In the following example, you create one entry for PRODUCTION_network1 and one for PRODUCTION_network2. For more information, see the Open Client documentation for your client platform.

   • For UNIX clients, you can edit the interfaces files with an ASCII text editor. From the server’s interfaces files, copy the server name line and the “master” line for each network into the client interfaces file. Add the appropriate server name for each entry, and change “master” to “query.”

   Clients on each network must use the DSQUERY value that corresponds to the network the client is on. In the following example, either PRODUCTION_network1 or PRODUCTION_network2 can be used.

   ```
   # Client entry for PRODUCTION on network1
   PRODUCTION_network1<tab>3<tab>3<newline>
   <tab>query tcp ether serv_corpnet 4559
   # Client entry for PRODUCTION on network2
   PRODUCTION_network2<tab>3<tab>3<newline>
   <tab>query tcp ether serv_engnet 5479
   ```

Configuring for query port backup

Another use for multiple network interfaces is to provide a backup in case of network failure. If a client is connected to a server via two networks, the client can establish a connection via the second network if the first one goes down.

To configure Adaptive Server for query port backup:

1. Install multiple “master” and “query” lines in a server entry in the interfaces file.
CHAPTER 5 Setting Up Communications Across the Network

2 Adaptive Server listens for connections at both ports. Clients looking for a host name and a port number for a connection to an Adaptive Server try the port on each “query” line in order, until they establish a connection.

The following example shows how to configure a backup network that will be used only if the normal connection fails. The primary network is “corporate network” and backup is “engineering network.”

```
# PRODUCTION server with two network listeners
PRODUCTION
<tab>master tcp ether SERV_CORPNET 4559
<tab>master tcp ether SERV_ENGNET 5479
<tab>query tcp ether SERV_CORPNET 4559
<tab>query tcp ether SERV_ENGNET 5479
```

3 Configure PC-client interfaces files with the appropriate multiple “query” entries, as described in the Open Client documentation. For client interfaces files in a homogeneous environment, you can copy the entire interfaces file entry for the Adaptive Server into the client interfaces file.

4 A connection on the secondary port occurs only if the corporate network is disabled, or if the corporate network interface on the host machine fails or is shut down due to a network-related failure.

IPv6 support

Adaptive Server supports IPv6 technology.

IPv6 terminology:
- Link-local address – an IPv6 address that is usable only over a single link.
- Site-local address – an IPv6 address that can be used within a single site.
- Global address – an IPv6 address that can be used across the Internet.

Interfaces files also provide IPv6 support. An example `interfaces` file entry is:

```
RHAM0_10778_XP
```

IPv6 application types:
IPv6 support

- IPv6-unaware – an application that cannot handle IPv6 addresses.
- IPv6-aware – an application that can communicate with nodes that do not have IPv4 addresses. In some cases, this might be transparent to the application, for instance when the API hides the content and format of the actual addresses.
- IPv6-enabled – an application that, in addition to being IPv6-aware, takes advantage of some IPv6 features.
- IPv6-required – an application that requires some IPv6 features and cannot operate over IPv4.

IPv6 infrastructure

A dual-stack infrastructure implements both IPv4 and IPv6. This is the recommended infrastructure implementation for using Adaptive Server Enterprise as an IPv6-aware server.

Sybase applications are IPv6-aware. Table 5-3 lists the platform runtime requirements and the specific product and its release version:

<table>
<thead>
<tr>
<th>Platform</th>
<th>Adaptive Server IPv6 awareness</th>
<th>Open Client/Server IPv6 awareness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sun Solaris 8 32- and 64-bit</td>
<td>12.5.3a and 15.0</td>
<td>12.5 and 15.0</td>
</tr>
<tr>
<td>HP-UX 11i(v1) 32- and 64-bit</td>
<td>12.5.3a and 15.0</td>
<td>12.5 and 15.0</td>
</tr>
<tr>
<td>Microsoft Server 2003</td>
<td>12.5.3a and 15.0</td>
<td>12.5 and 15.0</td>
</tr>
<tr>
<td>Linux RHEL 3.0</td>
<td>15.0</td>
<td>12.5 and 15.0</td>
</tr>
</tbody>
</table>

Many Sybase products that are Open Client/Server based such as XP Server, Backup Server, Replication Server, and Open Switch became automatically IPv6-aware due to the layered Open Client Transport Control Layer (DB-Library-based) which is IPv6-aware for network-socket operations. Open Client products are not IPv6-aware.

For Adaptive Server Enterprise, being IPv6-aware is a complex issue because some third-party components within Adaptive Server are not yet IPv6-aware. The functional mechanisms of Adaptive Server Enterprise that are IPv6-aware with respect to the supported platforms and versions include:

- Connection handler
CHAPTER 5 Setting Up Communications Across the Network

- RPC mechanisms
- Job Scheduler task / agent session connection
- Network Host API
- UDP message support for sybsendmsg
- Component Integration Services connectivity
- Host / name resolving
- XML URL connection handler
- Auditing for client address data

The following functional mechanisms in Adaptive Server Enterprise do not currently support IPv6, but are scheduled to do so in future versions:

- Java support
- License Management Server
- LDAP driver

**Note** By default, Adaptive Server is IPv6-unaware.

Before starting Adaptive Server for IPv6-aware operations, make sure that your infrastructure is correctly set up. Once your operating system is correctly configured, you can configure and enable an IPv6 connection handler by adding an additional DCL entry. A single Adaptive Server configuration can typically carry up to 32 connection handler assignments within the DCL.

For example, if you have a site-local setup with two domains administrated under the names:

sybase.com - being responsible for all IPv4 networking applications
v6.sybase.com - being responsible for all IPv6 networking applications

The DCL entry named “SYBASE” on the host “revival” for port 17100 looks similar to:

```
SYBASE
master tcp ether revival.sybase.com 17100
query tcp ether revival.sybase.com 17100
master tcp ether revival.v6.sybase.com 17100
query tcp ether revival.v6.sybase.com 17100
```
In this example, when Adaptive Server is started with IPv6-awareness, it creates two connection handlers. One listens on port 17100 for incoming IPv4 client connection requests, and the other listens on port 17100 for incoming IPv6 client connection requests.

**Note** When you start Adaptive Server, set trace flag 7815 to capture and log IPv4 and IPv6 client address connection requests and host and name lookups.

---

**Troubleshooting**

This section describes how to correct some common situations that may cause a server to not start.

**Server fails to start**

If a server fails to start with the following message, the port number specified in the interfaces file may be in use:

```
00:0000:00002:2003/09/22 12:37:23.63 kernel network name SERV_CORPNET, type ether, port 4559, filter NONE
00:0000:00002:2003/09/22 12:37:23.65 kernel ninit: bind, Address already in use
00:0000:00002:2003/09/22 12:37:23.68 server Unable to initialize network 0
00:0000:000016:2003/09/22 16:11:35.46 server SQL Server shutdown by request.
```

❖ **Investigating the port assignment**

1. Look in the interfaces file to identify the port number assigned to the server.
2. Determine whether another process is using the same port number by entering:
   ```
   netstat -a
   ```
If the port number is presented as a local address in the `netstat` output, you cannot use that port for the server. Another process is already using that port.

3 To verify that the server port is in use, start the server manually. The server does not start if its assigned port number is already in use.

For information on starting servers manually, see the installation documentation for your platform and the Adaptive Server Utility Guide.

❖ If a stale server process is retaining use of the port number

1 Do one of the following:
   • Use the operating system `kill` command to terminate the process.
   • Use another port number for the server by modifying the interfaces file.

2 Start the server manually to confirm that the port number is available.

For information on starting servers manually, see the installation documentation for your platform and the Utility Guide.

Error when executing an ESP

If you attempt to execute an ESP (extended stored procedure), you may see the following error:

```
00:00000:00008:1997/09/10 12:52:53.03 kernel XP Server failed to start. Try bringing up XP Server manually. Check SQL Server documentation for more information on how to bring XP Server up.
```

XP Server cannot start because the port number may be in use by another process. Use the `netstat` command described in the previous section to determine if the port number specified for XP Server is in use.

If you find no processes using the same port number:

1 Restart Adaptive Server.

2 Execute the ESP that you attempted earlier.

   XP Server should start automatically.

If you find a process using the same port number, you can do one of the following:

   • Change the interfaces file to use a new port number for the XP Server.
Troubleshooting

- Stop the process using the port number allotted to XP Server.

  Restart Adaptive Server, and execute the ESP that you attempted earlier. XP Server should start automatically.
CHAPTER 6

Using the Lightweight Directory Access Protocol as a Directory Service

Adaptive Server uses directory services to establish client and RPC connections over the Internet. This chapter provides information about using LDAP directory services to establish connections. LDAP is a licensed feature of Adaptive Server.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>71</td>
</tr>
<tr>
<td>LDAP directory services versus the Sybase interfaces file</td>
<td>72</td>
</tr>
<tr>
<td>The libtcl*.cfg file</td>
<td>75</td>
</tr>
<tr>
<td>Enabling LDAP directory services</td>
<td>76</td>
</tr>
<tr>
<td>Adding a server to the directory services</td>
<td>78</td>
</tr>
<tr>
<td>Multiple directory services</td>
<td>79</td>
</tr>
<tr>
<td>Encrypting the password</td>
<td>80</td>
</tr>
<tr>
<td>Performance</td>
<td>80</td>
</tr>
<tr>
<td>Migrating from the interfaces file to LDAP</td>
<td>81</td>
</tr>
</tbody>
</table>

Overview

Lightweight Directory Access Protocol (LDAP) is an industry standard for accessing directory services. Directory services allow components to look up information by a distinguished name (DN) from an LDAP server that stores and manages server, user, and software information that is used throughout the enterprise or over a network.

The LDAP server can be located on a different platform from the one on which Adaptive Server or the clients are running. LDAP defines the communication protocol and the contents of messages exchanged between clients and servers. Messages are operators, such as client requests for read, write and query, and server responses, including data-format information.
The LDAP server stores and retrieves information about:

- Adaptive Server, such as IP address, port number, and network protocol
- Security mechanisms and filters
- High availability companion server name

The LDAP server can be configured with these access restrictions:

- Anonymous authentication – all data is visible to any user.
- User name and password authentication – Adaptive Server uses the default user name and password for UNIX platforms:
  - $SYBASE/$SYBASE_OCS/config/libtcl.cfg on 32-bit platforms
  - $SYBASE/$SYBASE_OCS/config/libtcl64.cfg on 64-bit platforms

User name and password authentication properties establish and end a session connection to an LDAP server.

**Note** The user name and password that are passed to the LDAP server for user authentication purposes are distinct and different from those used to access Adaptive Server.

When an LDAP server is specified in the *libtcl.cfg* or *libtcl64.cfg* file (collectively the *libtcl*.cfg file), the server information is accessible only from the LDAP server. Adaptive Server ignores the interfaces file.

If multiple directory services are supported in a server, then the order in which they are searched is specified in *libtcl*.cfg. You cannot specify the search order with the dataserver command-line option. See “Multiple directory services” on page 79.

**LDAP directory services versus the Sybase interfaces file**

The LDAP driver implements directory services for use with an LDAP server. LDAP directories are an infrastructure that provide:

- A network-based alternative to the traditional Sybase interfaces file
• A single, hierarchical view of information, including users, software, resources, networks, files, and so on

Table 6-1 highlights the differences between the Sybase interfaces file and an LDAP server.

Table 6-1: interfaces file versus LDAP directory services

<table>
<thead>
<tr>
<th>interfaces file</th>
<th>Directory services</th>
</tr>
</thead>
<tbody>
<tr>
<td>Platform-specific</td>
<td>Platform-independent</td>
</tr>
<tr>
<td>Specific to each Sybase installation</td>
<td>Centralized and hierarchical</td>
</tr>
<tr>
<td>Contains separate master and query entries</td>
<td>One entry for each server that is accessed by both clients and servers</td>
</tr>
<tr>
<td>Cannot store metadata about the server</td>
<td>Stores metadata about the server</td>
</tr>
</tbody>
</table>

LDAP directory services support more attributes than the Sybase interfaces file. These attributes can include server version, server status, and so on. See Table 6-2 for a list of attributes.

Note LDAP is only supported with reentrant libraries. You must use isql\_r, instead of isql, when connecting to a server using LDAP directory services.

Table 6-2 lists the Sybase LDAP directory entries.

Table 6-2: Sybase LDAP directory definitions

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Value type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ditbase</td>
<td>interfaces file or libtcl.cfg</td>
<td>DIT base for object tree. If the libtcl.cfg file is specified, the interfaces file is ignored. The libtcl.cfg file can be overridden with ct_con_prop() for a specified connection.</td>
</tr>
<tr>
<td>dn</td>
<td>Character string</td>
<td>Distinguished name. Must be unique name that identifies the object.</td>
</tr>
<tr>
<td>sybaseVersion</td>
<td>Integer</td>
<td>Server version number.</td>
</tr>
<tr>
<td>sybaseServername</td>
<td>Character string</td>
<td>Server name.</td>
</tr>
<tr>
<td>sybaseService</td>
<td>Character string</td>
<td>Service type: Sybase Adaptive Server, or Sybase SQL Server.</td>
</tr>
<tr>
<td>sybaseStatus</td>
<td>Integer</td>
<td>Status: 1 = Active, 2 = Stopped, 3 = Failed, 4 = Unknown.</td>
</tr>
</tbody>
</table>
**LDAP directory services versus the Sybase interfaces file**

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Value type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sybaseAddress</td>
<td>String</td>
<td>Each server address includes:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Protocol: TCP, NAMEPIPE, SPX DECNET (entry is case sensitive).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Address: any valid address for the protocol type.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note</strong> dscp splits this attribute into Transport type and Transport address.</td>
</tr>
<tr>
<td>sybaseSecurity (optional)</td>
<td>String</td>
<td>Security OID (object ID).</td>
</tr>
<tr>
<td>sybaseRetryCount</td>
<td>Integer</td>
<td>This attribute is mapped to CS_RETRY_COUNT, which specifies the number of times that ct_connect retries the sequence of network addresses associated with a server name.</td>
</tr>
<tr>
<td>sybaseRetryDelay</td>
<td>Integer</td>
<td>This attribute is mapped to CS_LOOP_DELAY, which specifies the delay, in seconds, that ct_connect waits before retrying the entire sequence of addresses.</td>
</tr>
<tr>
<td>sybaseHAServername (optional)</td>
<td>String</td>
<td>A secondary server for failover protection.</td>
</tr>
</tbody>
</table>

The traditional interfaces file with TCP connection and a failover machine looks like:

```
master tcp ether huey 5000
query tcp ether huey 5000
hafailover secondary
```

An example of an LDAP entry with TCP and a failover machine looks like:

```
dn: sybaseServername=foobar, dc=sybase, dc=com
objectClass: sybaseServer
sybaseVersion: 1500
sybaseServername: foobar
sybaseService: ASE
sybaseStatus: 4
sybaseAddress: TCP#1#foobar 5000
sybaseRetryCount: 12
sybaseRetryDelay: 30
sybaseHAServername: secondary
```

All entries in the LDAP directory service are called entities. Each entity has a distinguished name (DN) and is stored in a hierarchical tree structure based on its DN. This tree is called the directory information tree (DIT). Client applications use a DIT base to specify where entities are stored. See “The libtcl*.cfg file” on page 75.
In the example above, the entry describes an Adaptive Server named “foobar” listening on a TCP connection with a port number of 5000. This entity also specifies a retry count of 12 (times) and a retry delay of 30 (seconds). Once a client has found an address where a server responds, the login dialog between the client and the server begins.

You can find a complete list of the Sybase LDAP directory schema in UNIX in a file called `sybase.schema` in the directory `$SYBASE/$SYBASE_OCS/config`.

In the same directory, there is also a file called `sybase-schema.conf`, which contains the same schema, but uses a Netscape-specific syntax.

Since LDAP supports multiple entries for each attribute, each address attribute must contain the address of a single server, including protocol, access type, and address. See `sybaseAddress` in Table 6-2.

For example, this is an LDAP entry for an Windows server listening on two addresses, with different connection protocols:

```plaintext
sybaseAddress = TCP#1#TOEJAM 4444
sybaseAddress = NAMEPIPE#1#\pipe\sybase\query
```

**Note** Each entry in the address field is separated by the # character.

You can edit these entries with `dsedit`. See “Adding a server to the directory services” on page 78.

To ensure cross-platform compatibility for all Sybase products, the protocol and address attribute fields should be in a platform- and product-independent format.

### The `libtcl*.cfg` file

You use the `libtcl*.cfg` file to specify the LDAP server name, port number, DIT base, user name, and password to authenticate the connection to an LDAP server.

The purpose of the `libtcl*.cfg` file is to provide configuration information such as driver, directory, and security services for Open Client/Open Server and Open Client/Open Server-based applications. 32-bit utilities such as `dsedit` and `srvbuild`, look up the `libtcl.cfg`, while 64-bit applications use the `libtcl64.cfg` file for configuration information.
Enabling LDAP directory services

You should edit both the *libtcl.cfg* and the *libtcl64.cfg* files to ensure compatibility between 32- and 64-bit applications.

The default *libtcl.cfg* file is located in `$SYBASE/$SYBASE_OCS/config`.

If LDAP is specified in the *libtcl.cfg* file, the interfaces file is not used.

**Note** Open Client/Open Server applications that use the `-I` option at start-up override the *libtcl.cfg* file and use the interfaces file.

In its simplest form, the *libtcl.cfg* file is in this format:

```
[DIRECTORY]
ldap=libsybdldap.dll ldapurl
```

where the *ldapurl* is defined as:

```
ldap://host:port/ditbase
```

The following LDAP entry, using these same attributes, is an anonymous connection and only works only if the LDAP server allows read-only access.

```
ldap=libsybdldap.dll ldap://seashore/d=sybase,dc=com
```

You can specify a user name and password in the *libtcl.cfg* file as extensions to the LDAP URL to enable password authentication at connection time.

---

Enabling LDAP directory services

To use a directory service, you must:

1. Install the “Security and directory services” license package ASE_SEC DIR.
2. Add the location of the LDAP libraries to the Unix load library path environment variable for your platform.
3. Configure the *libtcl.cfg* file to use directory services.

   Use any standard ASCII text editor to:

   - Remove the semicolon (;) comment markers from the beginning of the LDAP URL lines in the *libtcl.cfg* file under the `[DIRECTORY]` entry.
Add the LDAP URL under the [DIRECTORY] entry. See Table 6-3 for supported LDAP URL values.

**Warning!** The LDAP URL must be on a single line.

```
libtcl.cfg
ldap=libsydblad.so ldap://host:port/ditbase??scope??bindname=username password

libtcl64.cfg
ldap=libsydbldap64.so ldap://host:port/ditbase??scope??bindname=username password
```

For example:

```
[DIRECTORY]
ldap=libsydbldap.so ldap:///huey:11389/dc=sybase,dc=com??one??
bindname=cn=Manager,dc=sybase,dc=com secret
```

“one” indicates the scope of a search that retrieves entries one level below the DIT base.

Table 6-3 defines the keywords for the `ldapurl` variables.

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>host (required)</td>
<td>The host name or IP address of the machine running the LDAP server</td>
<td>None</td>
</tr>
<tr>
<td>port</td>
<td>The port number that the LDAP server is listening on</td>
<td>389</td>
</tr>
<tr>
<td>ditbase (required)</td>
<td>The default DIT base</td>
<td>None</td>
</tr>
<tr>
<td>username</td>
<td>Distinguished name (DN) of the user to authenticate</td>
<td>NULL (anonymous authentication)</td>
</tr>
<tr>
<td>password</td>
<td>Password of the user to be authenticated</td>
<td>NULL (anonymous authentication)</td>
</tr>
</tbody>
</table>

4 Verify that the appropriate environment variable points to the required third-party libraries. The Netscape LDAP SDK libraries are located in `$SYBASE/$SYBASE_OCS/lib3p` or `lib3p64`. The Unix load library path environment variable must point to this directory.

5 Add your server entry to the LDAP server using dscp or dsedit. See “Adding a server to the directory services” on page 78.
Adding a server to the directory services

Warning! Most LDAP servers have an `ldapadd` utility for adding directory entries. Sybase recommends you use `dsedit` instead since it has built-in semantic checks that generic tools do not provide.

Each server entry is made up of a set of attributes. When you add or modify a server entry, you are prompted for information about server attributes. Some attributes are provided by default, others require user input. When a default value is provided, it appears in brackets “[ ]”. See Table 6-2 for accepted values.

You can use `srvbuild` to add entries, but not modify or delete them.

❖ Adding a server entry to the directory service using `dsedit`

Before you can add, delete, or modify an LDAP server entry, you must add the LDAP URL to the `libtcl.cfg` file. See “The libtcl*.cfg file” on page 75.

Use `dsedit` to add a server to the directory service:

1. Source `SYBASE.csh` or `SYBASE.sh` to set the environment variables.
2. `cd` to `$SYBASE/$SYBASE_OCS/bin`.
3. Execute `dsedit`.
4. Select LDAP from the list of servers, and click OK.
5. Click Add New Server Entry.
6. Enter:
   - The server name – this is required.
   - The security mechanism – optional. This is the name of the high-availability failover server, if you have one.
7. Click Add New Network Transport and:
   - Select the transport type from the drop-down list.
   - Enter the host name.
   - Enter the port number
8. Click OK two times to edit `dsedit`.

To view the server entries, enter the following URL in Netscape:

```
http://host:port/ditbase??one.
```
For example:

```
ldap://huey:11389/dc=sybase,dc=com
```

**Note** Microsoft Internet Explorer does not recognize LDAP URLs.

For more information about `dscp`, see the *Open Client/Server Configuration Guide*, in the 11.1.x Generic Collection at http://www.sybase.com/support/manuals.

### Multiple directory services

Any type of LDAP service, whether it is an actual server or a gateway to other LDAP services, is called an LDAP server.

You can specify multiple directory services for high-availability failover protection. Not every directory service in the list needs to be an LDAP server.

For example:

```
[DIRECTORY]
ldap=libsydbladp.so ldap://test:389/dc=sybase,dc=com
ldap=libsybldldap.so ldap://huey:11389/dc=sybase,dc=com
```

In this example, if the connection to `test:389` fails, the connection fails over to the DCE driver with the specified DIT base. If this also fails, a connection to the LDAP server on `huey:11389` is attempted. Different vendors employ different DIT base formats.

Encrypting the password

Entries in the libtcl.cfg file are in human-readable format. Sybase provides a pwdcrypt utility for basic password encryption. pwdcrypt is a simple algorithm that, when applied to keyboard input, generates an encrypted value that can be substituted for the password. pwdcrypt is located in $SYBASE/SSYBASE_OCS/bin.

From the $SYBASE/SSYBASE_OCS directory, enter:

```
bin/pwdcrypt
```

Enter your password twice when prompted. pwdcrypt generates an encrypted password. For example:

```
0x01312a775ab9d5c71f99f05f7712d2cdded218d0ae1ce78868d0e8669313d1bc4c706
```

Copy and paste the encrypted password into the libtcl.cfg file using any standard ASCII-text editor. Before encryption, the file entry appears as:

```
ldap=libsybdldap.so
ldap://seashore/dc=sybase,dc=com??one??bindname=uid=Manager,dc=sybase,
dc=com?password
```

Replace the password with the encrypted string:

```
ldap=libsybdldap.so
ldap://seashore/dc=sybase,dc=com??one??bindname=uid=Manager,dc=sybase,dc=com?
0x01312a775ab9d5c71f99f05f7712d2cdded218d0ae1ce78868d0e8669313d1bc4c706
```

**Warning!** Even if your password is encrypted, you should still protect it using file-system security.

---

Performance

Performance when using an LDAP server may be slower than when using an interfaces file because the LDAP server requires time to make a network connection and retrieve data. Since this connection is made when Adaptive Server is started, changes in performance will be seen at login time, if at all. During normal system load, the delay should not be noticeable. During high system load with many connections, especially repeated connections with short duration, the overall performance difference of using an LDAP server versus the traditional interfaces file might be noticeable.
Migrating from the *interfaces* file to LDAP

There is no direct method to upgrade an existing server using the *interfaces* file to one that uses lightweight directory services. To upgrade a previous release of Adaptive Server to Adaptive Server version 15.0, see the *Installation Guide*.

Once you have upgraded the server, you can configure your server to use LDAP service.

2. Edit the `$SYBASE/$SYBASE_OCS/config/libtcl.cfg` or `libtcl64.cfg` file to add the directory service. See “Enabling LDAP directory services” on page 76.
3. Use `dsedit` and add the server entry to directory service. See “Adding a server to the directory services” on page 78.
4. Restart your server.
Migrating from the interfaces file to LDAP
CHAPTER 7
Customizing Localization for Adaptive Server

This chapter provides information about Sybase localization support for international installations, including configuring languages, character sets, and sort order. For more information, see the System Administration Guide: Volume 1.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview of localization support</td>
<td>83</td>
</tr>
<tr>
<td>Character set conversion</td>
<td>91</td>
</tr>
<tr>
<td>Sort orders</td>
<td>92</td>
</tr>
<tr>
<td>Language modules</td>
<td>95</td>
</tr>
<tr>
<td>Localization</td>
<td>96</td>
</tr>
<tr>
<td>Changing the localization configuration</td>
<td>99</td>
</tr>
</tbody>
</table>

Overview of localization support

Localization is the process of setting up an application to run in a particular language or country environment, including translated system messages and correct formats for date, time, and currency. Adaptive Server supports localization for international customers and for customers with heterogeneous environments.

This support includes:

- Data processing support – Adaptive Server comes with character set and sort order definition files it uses to process the characters used in different languages.

  Sybase provides support for the major languages in:

  - Western Europe
  - Eastern Europe
  - Middle East
Overview of localization support

- Latin America
- Asia
- Translated system messages – Adaptive Server includes language modules for:
  - Brazilian Portuguese
  - Chinese (Simplified)
  - French
  - German
  - Japanese
  - Korean
  - Polish
  - Spanish
  - Thai
- Translated documentation – translated documentation is available in:
  - Chinese (Simplified)
  - French
  - German
  - Japanese
  - Korean

Language modules

Adaptive Server stores its localized software messages in separate language modules.

When you install a language module, the installation program loads the messages, character set, and sort-order files that support the new language in the correct locations.

When you install Adaptive Server and Backup Server, system messages in English are installed by default.
Default character sets for servers

The default character set is the character set in which data is encoded and stored on the Adaptive Server databases.

Changing the default language and character set

Warning! Make all changes to the character set and sort order for a new Adaptive Server before creating any user databases or making any changes to the Sybase-supplied databases. Changing the character set and sort order after data or data structures have been added to Adaptive Server may require additional, time-consuming steps. To change the character set or sort order after you have added data, see the System Administration Guide: Volume 1.

After srvbuild configures a new Adaptive Server, it displays a message box asking if you want to localize your Adaptive Server to a language other than us_english and if you want to use a character set or sort order other than the default.

srvbuild creates an Adaptive Server with the following defaults:

- us_english language
- iso_1 character set (on HP-UX platforms, use Roman8)
- Binary sort order

Valid language options depend on what language modules were unloaded from the distribution media onto your system.

All character sets are copied from the distribution media by default.

You can:

- Click No to accept the defaults.
- Click Yes to change the defaults.

The sqlloc menu is displayed. sqlloc is the GUI utility used to change default languages, character sets, and sort orders.

By default, when Adaptive Server and Backup Server are installed on IBM, and SUN Solaris systems, the installation installs the character set files for ISO 8859-1, which supports the Western European languages.
Overview of localization support

By default, when Adaptive Server and Backup Server are installed on HP systems, the installation installs the character set files for ROMAN8, which supports the Western European languages.

Changing the default character set for servers

You can select any character set as the default on Adaptive Server, including character sets that are not the platform default character sets. Keep the following guidelines in mind when selecting a new default character set:

- To avoid conversion errors or overhead, determine the default character set based on the character set used by your clients.
  
  For example, if most of your clients use ISO 8859-1, you can minimize the amount of data conversion that has to occur by specifying ISO 8859-1.

- If your server is operating in a heterogeneous language environment, choose a character set that works with all the character sets needed. Often, this is Unicode (UTF-8).

Warning! Make all changes to the default character set and sort order for a new Adaptive Server before creating any user databases or making any changes to the Sybase-supplied databases. Changing the character set and sort order after data or data structures have been added to Adaptive Server can cause incorrect behavior. To change the character set or sort order after you have added data, see the System Administration Guide: Volume 1.

Supported character sets

The following language, scripts and character sets are supported by Adaptive Server:

- Arabic – see Table 7-1 on page 87.
- Baltic – see Table 7-2 on page 87.
- Chinese, Simplified – see Table 7-3 on page 88.
- Chinese, Traditional – see Table 7-4 on page 88
- Cyrillic – see Table 7-5 on page 88.
- Eastern European – see Table 7-6 on page 88.
- Greek – see Table 7-7 on page 89.
CHAPTER 7  Customizing Localization for Adaptive Server

- Hebrew – see Table 7-8 on page 89.
- Japanese – see Table 7-9 on page 89.
- Korean – see Table 7-10 on page 89.
- Thai – see Table 7-11 on page 89.
- Turkish – see Table 7-12 on page 90.
- Unicode (which supports over 650 languages) – see Table 7-13 on page 90.
- Vietnamese – see Table 7-14 on page 90.
- Western European – see Table 7-15 on page 90.

The tables define each character set and indicate information on whether it requires Unilib conversion (Unilib Required column).

- Check mark (x) – the character set requires Unilib conversion.
- No check mark – the character set may use either the Unilib conversion or the built-in conversion.

For more information see “Character set conversion” on page 91.

Table 7-1 lists the Arabic character set:

**Table 7-1: Arabic character sets**

<table>
<thead>
<tr>
<th>Character set</th>
<th>Unilib required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cp864</td>
<td>X</td>
<td>PC Arabic</td>
</tr>
<tr>
<td>cp1256</td>
<td>X</td>
<td>Microsoft Windows Arabic</td>
</tr>
<tr>
<td>iso88596</td>
<td>X</td>
<td>ISO 8859-6 Latin/Arabic</td>
</tr>
</tbody>
</table>

Table 7-2 lists the Baltic character set:

**Table 7-2: Baltic character sets**

<table>
<thead>
<tr>
<th>Character set</th>
<th>Unilib required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cp1257</td>
<td>X</td>
<td>Microsoft Windows Baltic</td>
</tr>
</tbody>
</table>

Table 7-3 lists the simplified Chinese character set:
### Overview of localization support

Table 7-3: Simplified Chinese character sets

<table>
<thead>
<tr>
<th>Character set</th>
<th>Unilib required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eucgb</td>
<td>X</td>
<td>EUC GB encoding = Simplified Chinese character sets</td>
</tr>
<tr>
<td>cp936</td>
<td>X</td>
<td>Microsoft Simplified Chinese character sets</td>
</tr>
<tr>
<td>gb18030</td>
<td>X</td>
<td>PRC 18030 standard</td>
</tr>
</tbody>
</table>

Table 7-4 lists the traditional Chinese character set:

Table 7-4: Traditional Chinese character set

<table>
<thead>
<tr>
<th>Character set</th>
<th>Unilib required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cp950</td>
<td>X</td>
<td>PC (Microsoft) Traditional Chinese</td>
</tr>
<tr>
<td>euccns</td>
<td>X</td>
<td>EUC CNS encoding = Traditional Chinese with extensions</td>
</tr>
<tr>
<td>big5</td>
<td>X</td>
<td>Big 5 Traditional Chinese</td>
</tr>
<tr>
<td>big5hk</td>
<td>X</td>
<td>Big 5 with HKSCS extensions</td>
</tr>
</tbody>
</table>

Table 7-5 lists the Cyrillic character set:

Table 7-5: Cyrillic character sets

<table>
<thead>
<tr>
<th>Character set</th>
<th>Unilib required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cp855</td>
<td></td>
<td>IBM PC Cyrillic</td>
</tr>
<tr>
<td>cp866</td>
<td></td>
<td>PC Russian</td>
</tr>
<tr>
<td>cp1251</td>
<td></td>
<td>Microsoft Windows 3.1 Cyrillic</td>
</tr>
<tr>
<td>iso88595</td>
<td></td>
<td>ISO 8859-5 Latin/Cyrillic</td>
</tr>
<tr>
<td>koi8</td>
<td></td>
<td>KOI-8 Cyrillic</td>
</tr>
<tr>
<td>kz1048</td>
<td></td>
<td>Kazakhstan Cyrillic</td>
</tr>
</tbody>
</table>

Table 7-6 lists the Eastern European character set:

Table 7-6: Eastern European character sets

<table>
<thead>
<tr>
<th>Character set</th>
<th>Unilib required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cp852</td>
<td></td>
<td>PC Eastern Europe</td>
</tr>
<tr>
<td>cp1250</td>
<td></td>
<td>Microsoft Windows 3.1 Eastern European</td>
</tr>
<tr>
<td>iso88592</td>
<td></td>
<td>ISO 8859-2 Latin-2</td>
</tr>
</tbody>
</table>

Table 7-7 lists the Greek character set:
Table 7-7: Greek character sets

<table>
<thead>
<tr>
<th>Character set</th>
<th>Unilib required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cp869</td>
<td></td>
<td>IBM PC Greek</td>
</tr>
<tr>
<td>cp1253</td>
<td></td>
<td>MS Windows Greek</td>
</tr>
<tr>
<td>greek8</td>
<td></td>
<td>HP GREEK8</td>
</tr>
<tr>
<td>iso88597</td>
<td></td>
<td>ISO 8859-7 Latin/Greek</td>
</tr>
</tbody>
</table>

Table 7-8 lists the Hebrew character set:

Table 7-8: Hebrew character sets

<table>
<thead>
<tr>
<th>Character set</th>
<th>Unilib required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cp1255</td>
<td>X</td>
<td>Microsoft Windows Hebrew</td>
</tr>
<tr>
<td>iso88598</td>
<td>X</td>
<td>ISO 8859-8 Hebrew</td>
</tr>
</tbody>
</table>

Table 7-9 lists the Japanese character set:

Table 7-9: Japanese character sets

<table>
<thead>
<tr>
<th>Character set</th>
<th>Unilib required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cp932</td>
<td>X</td>
<td>IBM J-DBCS:CP897 + CP301 (Shift-JIS)</td>
</tr>
<tr>
<td>eucjis</td>
<td>X</td>
<td>EUC-JIS encoding</td>
</tr>
<tr>
<td>sjis</td>
<td></td>
<td>Shift-JIS (no extensions)</td>
</tr>
</tbody>
</table>

Table 7-10 lists the Korean character set:

Table 7-10: Korean character sets

<table>
<thead>
<tr>
<th>Character set</th>
<th>Unilib required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eucksc</td>
<td>X</td>
<td>EUC KSC Korean encoding = CP949</td>
</tr>
</tbody>
</table>

Table 7-11 lists the Thai character set:

Table 7-11: Thai client character sets

<table>
<thead>
<tr>
<th>Character set</th>
<th>Unilib required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tis620</td>
<td>X</td>
<td>TIS-620 Thai standard</td>
</tr>
<tr>
<td>cp874</td>
<td>X</td>
<td>Microsoft Windows Thai</td>
</tr>
</tbody>
</table>

Table 7-12 lists the Turkish character set:
Overview of localization support

Table 7-12: Turkish character sets

<table>
<thead>
<tr>
<th>Character set</th>
<th>Unilib required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cp857</td>
<td></td>
<td>IBM PC Turkish</td>
</tr>
<tr>
<td>cp1254</td>
<td></td>
<td>Microsoft Windows Turkish</td>
</tr>
<tr>
<td>iso88599</td>
<td></td>
<td>ISO 8859-9 Latin-5 Turkish</td>
</tr>
<tr>
<td>turkish8</td>
<td></td>
<td>HP TURKISH8</td>
</tr>
</tbody>
</table>

Table 7-13 lists the Unicode character set:

Table 7-13: Unicode character set

<table>
<thead>
<tr>
<th>Character set</th>
<th>Unilib required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>utf8</td>
<td>X</td>
<td>Unicode UTF-8 encoding</td>
</tr>
</tbody>
</table>

Table 7-14 lists the Vietnamese character set:

Table 7-14: Vietnamese character set

<table>
<thead>
<tr>
<th>Character set</th>
<th>Unilib required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cp1258</td>
<td>X</td>
<td>Microsoft Windows Vietnamese</td>
</tr>
</tbody>
</table>

Table 7-15 lists the Western European character set:

Table 7-15: Western European character set

<table>
<thead>
<tr>
<th>Character set</th>
<th>Unilib required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ascii8</td>
<td>X</td>
<td>US ASCII, with 8-bit data, ISO 646</td>
</tr>
<tr>
<td>cp437</td>
<td></td>
<td>IBM CP437 - U.S. code set</td>
</tr>
<tr>
<td>cp850</td>
<td></td>
<td>IBM CP850 - European code set</td>
</tr>
<tr>
<td>cp860</td>
<td>X</td>
<td>PC Portuguese</td>
</tr>
<tr>
<td>cp858</td>
<td>X</td>
<td>cp850 with Euro support</td>
</tr>
<tr>
<td>cp1252</td>
<td>X</td>
<td>Microsoft Windows US (ANSI)</td>
</tr>
<tr>
<td>iso_1</td>
<td></td>
<td>ISO 8859-1 Latin-1</td>
</tr>
<tr>
<td>roman8</td>
<td></td>
<td>HP ROMAN8</td>
</tr>
<tr>
<td>iso15</td>
<td>X</td>
<td>ISO 8859-15 Latin-1 with Euro support</td>
</tr>
<tr>
<td>roman9</td>
<td>X</td>
<td>HP ROMAN8 with Euro support</td>
</tr>
</tbody>
</table>
Character set conversion

Backup Server passes messages to Adaptive Server in the client’s language and in the Adaptive Server character set. Adaptive Server then converts the messages and issues them in the client’s language and character set. Keep the following requirements in mind when selecting a character set:

- In a heterogeneous environment, Adaptive Server and Backup Server may need to communicate with clients running on different platforms and using different character sets. To maintain data integrity, the server converts the code between the character sets.
- To use the built-in conversion, you must install the character set definition files on the server for all the character sets being used by your clients. Built-in conversion support is available for many character sets.
- Unilib conversion support is available for all character sets supported by Sybase. To enable Unilib conversion, you must use `sp_configure` and turn enable unicode conversions on. For more information, see the `System Administration Guide: Volume 1`.

If either Adaptive Server or Backup Server does not support a client’s language or character set, that server issues a warning message. Errors also occur when the Backup Server character set is not compatible with the Adaptive Server character set. By default, unicode conversion is enabled.

Character set conversion is supported only between character sets for the same language or between character sets in the same language group.

For example, automatic character set conversion is supported between the character sets for the Western European languages: ASCII 8, CP 437, CP 850, CP 860, CP 863, CP 1252, ISO 8859-1, ISO 8859-15, and ROMAN8. Similarly, conversion is supported between the character sets for Japanese: CP 932, EUC-JIS, Shift-JIS, and DEC-Kanji.

However, code conversion is not supported between any of the Western European language character sets and the Japanese character sets. For more information about supported conversions, see the `System Administration Guide: Volume 1`.

Conversions between server and client

If Adaptive Server does not support the client’s language or character set, the client can connect with the server, but no character conversions occur.
When a localized client application connects to Adaptive Server, the server checks to see if it supports the client’s language and character set.

- If Adaptive Server supports the language, it automatically performs all character set conversions and displays its messages in the client’s language and character set.
- If Adaptive Server does not support the language, it uses the user’s default language or Adaptive Server’s default language.
- If Adaptive Server does not support the character set, it issues a warning to the client, turns conversion off, and sets the language to U.S. English.

Sort orders

Each character set comes with one or more sort orders (collating sequences), which are located in the sort-order definition files (.srt files). These files accompany the character set definition files and can be found in the same directory.

You can select a sort order for your data according to the needs at your site. However, the server can support only one sort order at a time, so select a sort order that will work for all of your clients.

**Warning!** Make all changes to the default character set and sort order for a new Adaptive Server before creating any user databases or making any changes to the Sybase-supplied databases. Changing the character set and sort order after data or data structures have been added to Adaptive Server may cause incorrect behavior. To change the character set or sort order after you have added data, see the *System Administration Guide: Volume 1*.

Available sort orders

The sort order determines the collating sequence Adaptive Server uses to order, compare, and index character data. Each character set comes with one or more sort orders.
Sort orders are located in sort order definition files (.srt files) that accompany your character set definition files.

**Note**  Available sort orders vary according to the character set installed on Adaptive Server.

You can see the available sort orders for your character set by looking in the .srt file for your language. Sort orders are stored in:

```
$SYBASE/charsets/<charset_name>/*.srt
```

For more information about localization files, see “Localization directories” on page 96.

Table 7-16 describes the sort orders that you can specify at installation time or at a later time using the sqlloc utility.

**Table 7-16: Sort orders available in Adaptive Server**

<table>
<thead>
<tr>
<th>Sort order name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary order</td>
<td>Sorts all data according to numeric byte values for that character set. Binary order sorts all ASCII uppercase letters before lowercase letters. Accented or ideographic (multibyte) characters sort in their respective standards order, which may be arbitrary. All character sets have binary order as the default. If binary order does not meet your needs, you can specify one of the other sort orders either at installation or at a later time by, using the sqlloc utility.</td>
</tr>
<tr>
<td>Dictionary order, case sensitive, accent sensitive</td>
<td>Case sensitive. Sorts each uppercase letter before its lowercase counterpart, including accented characters. Recognizes the various accented forms of a letter and sorts them after the associated unaccented letter.</td>
</tr>
<tr>
<td>Dictionary order, case insensitive, accent sensitive</td>
<td>Case-insensitive dictionary sort order. Uppercase letters are equivalent to their lowercase counterparts and are intermingled in sorting results.</td>
</tr>
<tr>
<td>Dictionary order, case insensitive, accent insensitive</td>
<td>Case-insensitive dictionary sort order. Diacritical marks are ignored.</td>
</tr>
</tbody>
</table>
### Sort orders

<table>
<thead>
<tr>
<th>Sort order name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dictionary order, case insensitive with preference</td>
<td>Case-insensitive dictionary sort order, with case preference for collating purposes. A word written with uppercase letters is equivalent to the same word written with lowercase letters. Uppercase and lowercase letters are distinguished only when you use an ORDER BY clause. The ORDER BY clause sorts uppercase letters before it sorts lowercase.</td>
</tr>
<tr>
<td>Note</td>
<td>Do not select this sort order unless your installation requires that uppercase letters be sorted before lowercase letters in otherwise equivalent strings for ORDER BY clauses. Using this sort order may reduce performance in large tables when the columns specified in an ORDER BY clause match the key of the table’s clustered index.</td>
</tr>
<tr>
<td>Alternate dictionary order, case sensitive</td>
<td>Case-sensitive alternate dictionary sort order with lowercase variants sorted before uppercase. Use with several of the Western European languages.</td>
</tr>
<tr>
<td>Alternate dictionary order, case insensitive, accent insensitive</td>
<td>Case-insensitive and accent-insensitive alternate dictionary sort order. Use with several of the Western European languages.</td>
</tr>
<tr>
<td>Alternate dictionary order, case insensitive, uppercase preference</td>
<td>Case-insensitive alternate dictionary sort order with uppercase preference. Use with several of the Western European languages.</td>
</tr>
<tr>
<td>Spanish dictionary order, case sensitive</td>
<td>Case-sensitive Spanish dictionary sort order. Use with Spanish and for most Latin American locales.</td>
</tr>
<tr>
<td>Spanish dictionary order, case insensitive</td>
<td>Spanish case-insensitive dictionary sort order. Use with Spanish and for most Latin American locales.</td>
</tr>
<tr>
<td>Spanish dictionary order, case insensitive, accent insensitive</td>
<td>Spanish case-insensitive and accent-insensitive dictionary sort order. Use with Spanish and for most Latin American locales.</td>
</tr>
<tr>
<td>Scandinavian dictionary order, case sensitive</td>
<td>Case-sensitive dictionary sort order. Use with Scandinavian languages.</td>
</tr>
<tr>
<td>Scandinavian dictionary order, case insensitive, uppercase preference</td>
<td>Case-insensitive and accent-insensitive dictionary sorting, with uppercase preference. Use with Scandinavian languages.</td>
</tr>
</tbody>
</table>

To see the sort orders that are available, use `sqlloc` to display the sort orders for the character sets you plan to use. For more information on unicode sort orders for utf_8, see Chapter 7, “Configuring Character Sets, Sort Orders, and Languages” in the *System Administration Guide: Volume 1*. 

94    Adaptive Server Enterprise
Language modules

If you want Adaptive Server error messages to be displayed in a language other than U.S. English (us_english), you must install the appropriate language module.

When you install a new language module, installation automatically loads the language into the Sybase installation directory to support the new language. For information about directories, see “Localization directories” on page 96.

Installing a new language module

A full Adaptive Server installation includes all the language components automatically. If you did not select a full install, you must install additional language modules manually.

To install a new language module:

1 Load the language module software from the distribution media. You must load this software into the same directory in which you loaded Adaptive Server.

2 Reconfigure the language and, if necessary, the character set and sort order for Adaptive Server. For instructions, see “Changing the localization configuration” on page 99.

Message languages

For messages, U.S. English is installed as the default language in Adaptive Server. The following rules apply to language modules:

- During Adaptive Server installation or reconfiguration, you can specify a default language other than U.S. English. However, you must have installed the language module for the language you specify.

- If your clients require Adaptive Server messages in a language other than U.S. English, you must load the language module for those languages. Then, you can configure Adaptive Server to the language used by your clients.

- If Adaptive Server does not support messages in a client’s language, these clients receive messages in the server default language.
Localization

For example, if your client’s language is Latin, the Spanish language module is installed, and Spanish is specified as the Adaptive Server default language, the client receives messages in Spanish.

Localization

By default, the Adaptive Server and Backup Server configurations use the English locale settings, which include:

- Character set definition files for Western European character sets
- Sort-order definition files for Western European character sets
- U.S. English system message files

During the installation process or through reconfiguration, you can specify a different language, character set, and sort order.

Localization directories

Sybase localization configuration involves the following directories:

- locales
- charsets

The table below illustrates the structure of the localization files. It does not show a complete list of all the files.

<table>
<thead>
<tr>
<th>%SYBASE% or $SYBASE/</th>
<th>charsets</th>
<th>*srt files</th>
<th>*uct files</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>charset_name...</td>
<td>charset.loc</td>
<td></td>
</tr>
<tr>
<td></td>
<td>charset_name...</td>
<td>charset.loc</td>
<td>UCode</td>
</tr>
<tr>
<td></td>
<td>ucode</td>
<td></td>
<td></td>
</tr>
<tr>
<td>locales</td>
<td>language_name</td>
<td>charset_name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>language_name...</td>
<td>charset_name...</td>
<td></td>
</tr>
<tr>
<td></td>
<td>locales.dat</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>message</td>
<td>language_name</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>language_name...</td>
<td></td>
</tr>
</tbody>
</table>
About the directory

The $SYBASE/locales directory contains a subdirectory for each available language. Each language subdirectory contains a subdirectory for each character set available with that language.

- The .loc files in these subdirectories enable Adaptive Server or Backup Server to report errors in a specific language, encoded in a specific character set.

  There are a variety of .loc files in each subdirectory. Most of these files contain translated error messages for a specific product or utility.

- The common.loc file in the “utf8” subdirectory for each language contains localized information, such as local date, time, and currency formatting, that is used by all products.

- The locales.dat file contains entries that associate platform-specific locale names with Sybase language and character set combinations.

About the charsets directory

The files in $SYBASE/charsets/charset_name contain information related to each particular character set, such as the definition of the character set and any sort orders available for that character set.

About the locales.dat file

You can edit the locales.dat file to:

- Change the default language or character set for a platform, or
- Add new associations between platform locale names and Sybase language and character set names.

Format of locales.dat file entries

Each entry in the locales.dat file links a platform-specific locale definition to a Sybase language and character set combination. Each entry has the following format:

```
locale = platform_locale, syb_language, syb_charset
```

where:
• *platform_locale* is the platform-specific keyword for a locale. For acceptable values, see your operating system documentation.

When the locale being defined as the default for the site, *platform_locale* is “default.”

• *syb_language* is the name of the language directory to be used from within $SYBASE/locales/language_name.

• *syb_charset* is the character set name that determines the character set conversion method and identifies the directory location of the message files for clients from within $SYBASE/locales/language_name/charset_name.

For example, the following entry specifies that the default locale uses us_english for the language and iso_1 for the character set:

```
locale = default, us_english, iso_1
```

### How client applications use locales.dat

Client applications use the *locales.dat* file to identify the language and character set to use. The connection process follows these steps:

1. When a client application starts, it checks the operating system locale setting and then checks the *locales.dat* file to see if that setting is appropriate for Adaptive Server. For example, a locale entry for French can look like the following:

```
locale = fr_FR, french, iso_1
```

2. When the client connects to Adaptive Server, the language and character set information is passed to Adaptive Server in the login record.

3. Adaptive Server then uses:

   • The character set information, for example, iso_1, to identify the client’s character set and verify whether it can convert character data to this character set
   
   • The language (in the preceding example, French) and character set information to see if it has messages in the client’s language

**Note** Adaptive Server software includes some locale entries already defined in the *locales.dat* file. If these entries do not meet your needs, you can either modify them or add new locale entries.
Editing the `locales.dat` file

Before beginning the edit, make a copy of the original file, in case you have problems with the resulting edited version.

To edit the `locales.dat` file:

1. Open the `locales.dat` file copy in a text editor.
2. Find the section enclosed in brackets:
   - For Sun Solaris, `[sun_svr4]`
   - For HP, `[hp ux]`
   - For IBM, `[aix]`
3. Make sure the section contains an entry for the language (`syb_language`) and character set (`syb_charset`) combination that you want to use.

   **Note** The value for `platform_locale` must match the value required by your operating system. If the locales definitions in your system configuration files do not match the Sybase locale definitions, your applications will not run properly.

For example, if you want your Open Client messages to appear in French, and Adaptive Server is using the ROMAN8 character set, you would check the `locales.dat` entries for your platform and look for the following entry:

```
locale = fr_FR, french, roman8
```

4. Add the required entry or modify an existing entry.
5. Save the changes, if any, and exit the text editor.

Changing the localization configuration

By default, the Adaptive Server and Backup Server configurations uses the English locale settings localization, which include:

- Character set definition files for Western European character sets
- Sort order definition files for Western European character sets
- `us_english` system message files
During the installation process and through reconfiguration, you can specify a different language, character set, and sort order.

**Adaptive Server localization**

Each language uses about 2MB of database space per module. If necessary, use the `alter database` command to increase the size of the master database before adding another language.

---

**Note** If you want to install more than one language on Adaptive Server, and the master database is not large enough to manage more than one language, the transaction log may become too full. You can expand the master database only on the master device. For more information, see the *System Administration Guide: Volume 2*.

---

1. Source `SYBASE.csh` or `SYBASE.sh` if you have not set up the Sybase environment variables.
2. To configure localization for Adaptive Server on the server, start `sqlloc`:
   ```
   $SYBASE/$SYBASE_ASE/bin/sqlloc
   ```
3. Select Localize an Existing Server.
4. From the Adaptive Server selection window, select the server.
5. Supply the user name and password. The user must have “sa” privileges.
6. Next, supply the:
   - Default language
   - Default character set
   - Default sort order
7. Select any other languages you want to install. You may select only languages that are supported by the default character set.
   The Add and Remove Languages window lists all Sybase supported languages.
8. The Localization Summary window summarizes the configuration options you selected. Click OK to confirm your selections.
   The Status Output window notifies you upon completing the installation.
Backup Server localization

You can change the Backup server language and character set by modifying the `RUN_<backup_server_name>` file. See the Utility Guide for more information on the backupserver command arguments.

Configuring Adaptive Server for other character sets

To configure Adaptive Server with the character set and sort order for your language, complete the following steps. Your system messages appear in the default language, English.

1. Use the charset utility to load the default character set and sort order.

   To use `charset`, the server must be running and you must have System Administrator privileges. Use the file name of the sort order:
   ```
   $SYBASE/$SYBASE_ASE/bin/charset -Usa -P password -S server_name sort_order_file character_set
   ```
   Replace `sort_order_file` with the name of the sort order file. See Table 7-17 on page 102. Replace `character_set` with the Sybase name for your character set. See Table 7-18 on page 104.

2. Use charset utility to load any additional character sets. See “charset utility” on page 105 for more about this utility.

   To use the Adaptive Server built-in character set conversions, you must load the character set definition files for all the characters set on your client platforms. If you are using the Unilib character set conversions, you do not need to do this.

3. Using `isql`, log in to your server as “sa” and select the master database.
   ```
   1> use master
   2> go
   ```

4. Use the ID of the sort order to configure your server for the new character set and sort order.
   ```
   1> sp_configure "default sortorder_id", "sort_order_id", "character_set"
   3> go
   ```
   Replace `sort_order_id` with the ID for your sort order. See Table 7-17 on page 102. Replace `character_set` with the Sybase name for your character set. See Table 7-18 on page 104.

5. Shut down the server to start the reconfiguration process.
Changing the localization configuration

6 Use your normal process on your UNIX system to reboot the server, usually by invoking one of the $RUN_xxx$ scripts from $SYBASE/$SYBASE_ASE/install.

7 The server starts, rebuilds all the system indexes, then shuts down. Restart a second time to bring the server up in a stable state.

Sort orders

Table 7-17 describes the available sort orders. If your language does not appear, then there is no language-specific sort order for your language—use a binary sort order.

<table>
<thead>
<tr>
<th>Language or script</th>
<th>Sort orders</th>
<th>File name</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>All languages</td>
<td>Binary order</td>
<td>binary.srt</td>
<td>50</td>
</tr>
<tr>
<td>Cyrillic</td>
<td>Dictionary order, case sensitive, accent sensitive</td>
<td>cyrdict.srt</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td>Dictionary order, case sensitive, accent sensitive</td>
<td>cyrnocs.srt</td>
<td>64</td>
</tr>
<tr>
<td>English</td>
<td>Dictionary order, case sensitive, accent sensitive</td>
<td>dictiona.srt</td>
<td>51</td>
</tr>
<tr>
<td>French</td>
<td>Dictionary order, case insensitive, accent sensitive</td>
<td>nocase.srt</td>
<td>52</td>
</tr>
<tr>
<td>German</td>
<td>Dictionary order, case insensitive, accent sensitive, with preference</td>
<td>nocasepr.srt</td>
<td>53</td>
</tr>
<tr>
<td></td>
<td>Dictionary order, case insensitive, accent insensitive</td>
<td>noaccent.srt</td>
<td>54</td>
</tr>
<tr>
<td>English</td>
<td>Alternate dictionary order, case sensitive</td>
<td>altdict.srt</td>
<td>45</td>
</tr>
<tr>
<td>French</td>
<td>Alternate dictionary order, case sensitive, accent insensitive</td>
<td>altnoacc.srt</td>
<td>39</td>
</tr>
<tr>
<td>German</td>
<td>Alternate dictionary order, case sensitive, with preference</td>
<td>altnocsp.srt</td>
<td>46</td>
</tr>
<tr>
<td>Greek</td>
<td>Dictionary order, case sensitive, accent sensitive</td>
<td>elldict.srt</td>
<td>65</td>
</tr>
<tr>
<td>Hungarian</td>
<td>Dictionary order, case sensitive, accent sensitive</td>
<td>hundict.srt</td>
<td>69</td>
</tr>
<tr>
<td></td>
<td>Dictionary order, case insensitive, accent sensitive</td>
<td>hunnoc.srt</td>
<td>70</td>
</tr>
<tr>
<td>Russian</td>
<td>Dictionary order, case insensitive, accent insensitive</td>
<td>hunnocs.srt</td>
<td>71</td>
</tr>
<tr>
<td></td>
<td>Dictionary order, case sensitive, accent sensitive</td>
<td>rusdict.srt</td>
<td>58</td>
</tr>
<tr>
<td></td>
<td>Dictionary order, case insensitive, accent sensitive</td>
<td>runnocs.srt</td>
<td>59</td>
</tr>
</tbody>
</table>
## Character sets

Table 7-18 lists the supported character sets and their Sybase names.

<table>
<thead>
<tr>
<th>Language or script</th>
<th>Sort orders</th>
<th>File name</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scandinavian</td>
<td>Dictionary order, case sensitive</td>
<td>scandict.srt</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>Dictionary order, case insensitive, with preference</td>
<td>scannocp.srt</td>
<td>48</td>
</tr>
<tr>
<td>Spanish</td>
<td>Dictionary order, case sensitive</td>
<td>espdict.srt</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td>Dictionary order, case insensitive, accent sensitive</td>
<td>espnoacsrt</td>
<td>56</td>
</tr>
<tr>
<td></td>
<td>Dictionary order, case insensitive, accent insensitive</td>
<td>espnoacsrt</td>
<td>57</td>
</tr>
<tr>
<td>Thai</td>
<td>Dictionary order</td>
<td>dictionary.srt</td>
<td>51</td>
</tr>
<tr>
<td>Turkish</td>
<td>Dictionary order, case sensitive</td>
<td>turdict.srt</td>
<td>72</td>
</tr>
<tr>
<td></td>
<td>Dictionary order, case insensitive, accent insensitive</td>
<td>turnoacsrt</td>
<td>73</td>
</tr>
<tr>
<td></td>
<td>Dictionary order, case insensitive, accent sensitive</td>
<td>turnoacsrt</td>
<td>74</td>
</tr>
</tbody>
</table>
### Table 7-18: Sybase character set names

<table>
<thead>
<tr>
<th>Character sets</th>
<th>Sybase name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASCII 8</td>
<td>acsii_8</td>
</tr>
<tr>
<td>Big 5</td>
<td>big5</td>
</tr>
<tr>
<td>Big 5HK</td>
<td>big5hk</td>
</tr>
<tr>
<td>CP 437</td>
<td>cp437</td>
</tr>
<tr>
<td>CP 850</td>
<td>cp850</td>
</tr>
<tr>
<td>CP 852</td>
<td>cp852</td>
</tr>
<tr>
<td>CP 855</td>
<td>cp855</td>
</tr>
<tr>
<td>CP 857</td>
<td>cp857</td>
</tr>
<tr>
<td>CP 858</td>
<td>cp858</td>
</tr>
<tr>
<td>CP 860</td>
<td>cp860</td>
</tr>
<tr>
<td>CP 864</td>
<td>cp864</td>
</tr>
<tr>
<td>CP 866</td>
<td>cp866</td>
</tr>
<tr>
<td>CP 869</td>
<td>cp869</td>
</tr>
<tr>
<td>CP 874</td>
<td>cp874</td>
</tr>
<tr>
<td>CP 932</td>
<td>cp932</td>
</tr>
<tr>
<td>CP 936</td>
<td>cp936</td>
</tr>
<tr>
<td>CP 950</td>
<td>cp950</td>
</tr>
<tr>
<td>CP 1250</td>
<td>cp1250</td>
</tr>
<tr>
<td>CP 1251</td>
<td>cp1251</td>
</tr>
<tr>
<td>CP 1252</td>
<td>cp1252</td>
</tr>
<tr>
<td>CP 1253</td>
<td>cp1253</td>
</tr>
<tr>
<td>CP 1254</td>
<td>cp1254</td>
</tr>
<tr>
<td>CP 1255</td>
<td>cp1255</td>
</tr>
<tr>
<td>CP 1256</td>
<td>cp1256</td>
</tr>
<tr>
<td>CP 1257</td>
<td>cp1257</td>
</tr>
<tr>
<td>CP 1258</td>
<td>cp1258</td>
</tr>
<tr>
<td>DEC Kanji</td>
<td>deckanji</td>
</tr>
<tr>
<td>EUC-CNS</td>
<td>euccns</td>
</tr>
<tr>
<td>EUC-GB</td>
<td>eucgb</td>
</tr>
<tr>
<td>EUC-JIS</td>
<td>eucjis</td>
</tr>
<tr>
<td>EUC-KSC</td>
<td>eucksc</td>
</tr>
<tr>
<td>GB 18030</td>
<td>gb18030</td>
</tr>
<tr>
<td>GREEK8</td>
<td>greek8</td>
</tr>
<tr>
<td>ISO 8859-1</td>
<td>iso_1</td>
</tr>
<tr>
<td>ISO 8859-2</td>
<td>iso88592</td>
</tr>
<tr>
<td>ISO 8859-5</td>
<td>iso88595</td>
</tr>
</tbody>
</table>
CHAPTER 7 Customizing Localization for Adaptive Server

Use the `charset` utility to load character sets and sort orders into Adaptive Server. If you are using `charset` to load the default character set and sort order, this should be done only at the time of installation.

To change the default character set and sort order of Adaptive Server, see the `System Administration Guide: Volume 1`.

### Syntax

```
charset
[ -U username ]
[ -P password ]
[ -S server ]
[ -I interfaces ]
[ -v version ]
sort_order
[ charset ]
```

### Table 7-19: Keywords and options for charsets

<table>
<thead>
<tr>
<th>Character sets</th>
<th>Sybase name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISO 8859-6</td>
<td>iso88596</td>
</tr>
<tr>
<td>ISO 8859-7</td>
<td>iso88597</td>
</tr>
<tr>
<td>ISO 8859-8</td>
<td>iso88598</td>
</tr>
<tr>
<td>ISO 8859-9</td>
<td>iso88599</td>
</tr>
<tr>
<td>ISO 8859-15</td>
<td>iso15</td>
</tr>
<tr>
<td>Kazakhstan Cyrillic</td>
<td>kz1048</td>
</tr>
<tr>
<td>Koi8</td>
<td>koi8</td>
</tr>
<tr>
<td>ROMAN8</td>
<td>roman8</td>
</tr>
<tr>
<td>ROMAN9</td>
<td>roman9</td>
</tr>
<tr>
<td>Shift-JIS</td>
<td>sjis</td>
</tr>
<tr>
<td>TIS 620</td>
<td>tis620</td>
</tr>
<tr>
<td>TURKISH8</td>
<td>turkish8</td>
</tr>
<tr>
<td>UTF-8</td>
<td>utf8</td>
</tr>
</tbody>
</table>

### charset utility

Use the `charset` utility to load character sets and sort orders into Adaptive Server. If you are using `charset` to load the default character set and sort order, this should be done only at the time of installation.

To change the default character set and sort order of Adaptive Server, see the `System Administration Guide: Volume 1`.

```plaintext
Syntax

charset
[ -U username ]
[ -P password ]
[ -S server ]
[ -I interfaces ]
[ -v version ]
sort_order
[ charset ]
```

### Table 7-19: Keywords and options for charsets

<table>
<thead>
<tr>
<th>Keywords and options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-U</td>
<td>If you are not already logged in to your operating system as “sa”, you must specify “-Usa” or “/username = sa” in the command line.</td>
</tr>
<tr>
<td>-P</td>
<td>Specifies the “sa” password on the command line. If not specified, the user is prompted for the “sa” password.</td>
</tr>
</tbody>
</table>
# Changing the localization configuration

<table>
<thead>
<tr>
<th>Keywords and options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-S</td>
<td>Specifies the name of the server. If not specified, charset uses the DSQUERY environment variable to identify the server name. If there is no DSQUERY environment variable, charset attempts to connect to a server named &quot;SYBASE.&quot;</td>
</tr>
<tr>
<td>-I</td>
<td>Specifies the interfaces file to use. If not specified, charset uses the interfaces file in the SYBASE directory.</td>
</tr>
<tr>
<td>-v</td>
<td>Causes the Sybase version string to be printed, then exits. Use with no other options specified.</td>
</tr>
<tr>
<td>sort_order</td>
<td>When charset is used to load the default character set and sort order, sort_order is a mandatory parameter specifying the name of the sort order file to be used by Adaptive Server. When loading additional character sets, use charset.loc to indicate the name of the character set files.</td>
</tr>
<tr>
<td>charset</td>
<td>Specifies the directory of the character set to be used by Adaptive Server.</td>
</tr>
</tbody>
</table>
CHAPTER 8

Logging Error Messages and Events

This chapter describes how to use the error logging features of Adaptive Server.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaptive Server error logging</td>
<td>107</td>
</tr>
<tr>
<td>Setting error log paths</td>
<td>108</td>
</tr>
<tr>
<td>Managing messages</td>
<td>109</td>
</tr>
</tbody>
</table>

Adaptive Server error logging

Each time Adaptive Server starts, it writes information to a local error log file, called the Adaptive Server error log:

$SYBASE/$SYBASE_ASE/install/server_name.log

This file:

- Stores information about the success or failure of each start-up attempt
- Logs error and informational messages generated by the server during its operations
- Remains open until you stop the server process
- Contains startup messages from Adaptive Server

If the error log become too large, you can:

- Dynamically change the path of the error log using sp_errorlog. Once the older errorlog is not being user by Adaptive Server, you can move it, and make space available.
- Stop the Adaptive Server and delete logged messages.

Enabling and disabling error logging

Logging to the Adaptive Server error log is always enabled. However, when you create or modify a specific user-defined message, you can set it to be omitted from the log. See “Logging user-defined messages” on page 109.

Setting error log paths

The installation program sets the error log location in the Sybase installation directory when you configure a new Adaptive Server. Backup Server has its own error log.

The default location for each server’s error log is:
• Adaptive Server: $SYBASE/$SYBASE_ASE/install/servername.log
• Backup Server: $SYBASE/$SYBASE_ASE/install/servername_back.log

At start-up, you can reset the name and location of the Adaptive Server error log file from the command line. Use the -e start-up parameter and value in the dataserver command to start Adaptive Server.

Note: Multiple Adaptive Servers cannot share the same error log. If you install multiple Adaptive Servers, specify a unique error log file name for each server.

Setting the Adaptive Server error log path

You can change the error log path by editing the $SYBASE/$SYBASE_ASE/install/RUN_server_name file.

For example, to change the error log path from
$SYBASE/$SYBASE_ASE/bin/dataserver -d/Devices/ASE_2K.dat -sASE_2K -i/ASE_150 -e$SYBASE/$SYBASE_ASE/install/ASE_2K.log -M/ASE_150 to the $SYBASE directory, type:


For information about using the RUN_server_name file, see “Starting and Stopping Servers” on page 19, and the Utility Guide.
Managing messages

When event logging is enabled, you can manage its functions in the following ways:

- Use `sp_addmessage` to add a user message, or `sp_altermessage` to control whether a specific message is logged in the Adaptive Server error log. For the complete syntax for `sp_addmessage` and `sp_altermessage`, see the Reference Manual: Procedures.
- Use configuration parameters to specify whether auditing events are logged. Auditing events pertain to a user’s success, `log audit logon success`, or failure, `log audit logon failure`, in logging in to Adaptive Server.

Logging user-defined messages

You can specify whether a user-defined message is logged to the Adaptive Server error log. Adaptive Server lets you make this determination for:

- New messages (`sp_addmessage`).
- Existing messages (`sp_altermessage`).

For more information about these commands and their parameters, see `sp_addmessage` and `sp_altermessage` in the Reference Manual: Procedures.

New messages

Include the `with_log` option in `sp_addmessage` when you add a new user-defined message to `sysusermessages`. This parameter sets the Adaptive Server to log the message each time that the message appears.

Existing messages

Include the `with_log` option in `sp_altermessage` to change an existing user-defined message. This parameter alters the reporting status of that message:

- TRUE – to enable logging.
- FALSE – to disable logging.
Logging auditing events

By default, Adaptive Server does not log auditing events. However, you can use `sp_configure` parameters to specify whether Adaptive Server is to log auditing events, such as logins, to the Adaptive Server error log.

Possible parameters and values are:

- `log audit logon success` at 1 – to enable logging of successful Adaptive Server logins:
  
  ```
  sp_configure "log audit logon success", 1
  ```

- `log audit logon failure` at 1 – to enable logging of unsuccessful Adaptive Server logins:
  
  ```
  sp_configure "log audit logon failure", 1
  ```

- Either parameter at 0 – to disable logging of that message type:

  ```
  sp_configure "log audit logon success", 0
  sp_configure "log audit logon failure", 0
  ```

For more information about `sp_configure`, see the *System Administration Guide: Volume 1*. 
Managing Adaptive Server Databases

The administration of Adaptive Server databases includes both routine tasks and performance and tuning considerations.

- The *System Administration Guide Volume 1* and *Volume 2* discuss most of the administrative tasks in detail.
- The *Performance and Tuning Series* provides in-depth explanations of performance issues.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Managing database devices</td>
<td>111</td>
</tr>
</tbody>
</table>

Managing database devices

The term *database device* refers to a disk or a portion of a disk that stores Adaptive Server databases and database objects.

Device requirements

Adaptive Server devices and databases are subject to the following limits:

- The maximum device size is 4TB.
- The minimum usable device size depends on the servers logical page size. Databases manage space in groups of 256 logical pages, and the smallest specifiable disk piece size is 1MB, so the smallest usable device is the larger of 1MB or 256 logical pages:
  - The minimum device size for a 2K page server is 1MB
  - The minimum device size for a 4K page server is 1MB
  - The minimum device size for an 8K page server is 2MB
Managing database devices

- The minimum device size for a 16K page server is 4MB

- The maximum number of database devices is 2,147,483,647. However, Adaptive Server must retain a description of each device in memory, so in practice this number is limited by your system’s memory. Your operating system also limits how many devices one program can open simultaneously.

- A database can contain up to 2,147,483,648 logical pages, so its maximum size depends on its logical page size:
  - The maximum database size on a 2K page server is 4TB.
  - The maximum database size on a 4K page server is 8TB.
  - The maximum database size on an 8K page server is 16TB.
  - The maximum database size on a 16K page server is 32TB.

- The minimum database size is the size of the installation’s model database.

- Each database is stored on one or more database devices, in one or more disk pieces. The maximum number of disk pieces in one database is 8,388,608. However, Adaptive Server must retain a description of all active databases, so in practice this number is also limited by your operating system memory.

**Note** By default, `srvbuild` creates the devices in `$SYBASE/data` directory.

Creating files for database devices

You can create new database devices using the `disk init` command. You can specify a raw partition or operating system file. When using a raw partition, you must specify the full path to the partition. When using an operating system file, you may use the full path or a relative path. Path names are relative to your server’s current working directory.

Sybase recommends that you specify the full path to all database devices. Do not use environment variables when specifying path names with `disk init`.

Here is an example of creating a database device using `disk init`:

```
disk init name = "user_device1",
physname = "/work/data/device1.dat",
size = 2048
```
In this example, "size = 2048" tells the command to allocate 2048 “virtual” pages to the device. A virtual page is 2048 bytes, so this command creates a 4MB device.

The example command does not specify a device number, instead letting the server choose one. Unless you need to assign a specific number to a given device, Sybase recommends you use this method. If you do need an explicit device number, use the parameter "$vdevno = N $", where $N$ is the device number you want to use. If you specify a device number, that number must not be in use by any other device on this server. Use sp_helpdevice to see what device numbers have already been used.

If you find that an existing database device is too small, use the disk resize command to make the device larger. This command takes the same “name” and “size” parameters as disk init, except the size parameter specifies how much larger you want the device to be.

disk resize allows you to dynamically increase the size of your database devices, rather than initializing a new device. You can use disk resize to increase the size for devices on raw partitions and file systems. The minimum increase on a device is 1MB or an allocation unit, whichever is greater.

Note  Operating system constraints limit how much larger you can make any given device. For example, you cannot make a device on a UNIX raw partition larger if you have already allocated the full defined size of that partition.

For more information about sp_helpdevice and disk init command, see the System Administration Guide, Volume 1 and the Reference Manual: Commands.

For more information about device files, see the Performance and Tuning Series: Physical Database Tuning.
CHAPTER 10
Adding Optional Functionality to Adaptive Server

This chapter provides instructions for adding optional functionality to Adaptive Server:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adding auditing</td>
<td>115</td>
</tr>
<tr>
<td>Installing online help for Transact-SQL syntax</td>
<td>123</td>
</tr>
</tbody>
</table>

After you have installed the Sybase products on your system, see the product documentation for configuration and administration issues.

Adding auditing

Auditing is an important part of security in a database management system. Security-related system activity is recorded in an audit trail, which can be used to detect penetration of the system and misuse of resources. By examining the audit trail, the System Security Officer can inspect patterns of access to objects in databases and can monitor the activity of specific users. Audit records can be traced to specific users, enabling the audit system to act as a deterrent to users who are attempting to misuse the system.

A System Security Officer manages the audit system and is the only user who can start and stop auditing, set up auditing options, and process audit data.

Audit system devices and databases

The audit system includes several components. The main components are:

- The sybsecurity device and the sybsecurity database, which stores audit information
Adding auditing

- The audit trail, which consists of several audit devices and tables that you determine at configuration time
- The syslogs transaction log device, which stores transaction logs

The `syssecurity` device and database

The `syssecurity` device stores the `syssecurity` database. The `syssecurity` database is created as part of the auditing configuration process. It contains all the system tables in the `model` database, as well as a system table for keeping track of server-wide auditing options and system tables for the audit trail.

Tables and devices for the audit trail

Adaptive Server stores the audit trail in system tables, named `sysaudits_01` through `sysaudits_08`. At any given time, only one of the audit tables is current. Adaptive Server writes all audit data to the current audit table. A System Security Officer can use `sp_configure` to set or change which audit table is current.

When you configure Adaptive Server for auditing, you determine the number of audit tables for your installation. You can specify up to eight system tables (`sysaudits_01` through `sysaudits_08`). Plan to use at least two or three system tables for the audit trail and to put each system table on its own device, separate from the master device. If you do this, you can use a threshold procedure that archives the current audit table automatically, before it fills up and switches to a new, empty table for subsequent audit records.

Device for syslogs systems table

During auditing configuration, you must specify a separate device for the `syslogs` system table, which contains the transaction log. The `syslogs` table, which exists in every database, contains a log of transactions that are executed in the database.

Overview of audit installation

There are two methods for installing auditing for the first time in Adaptive Server:

- Use the `installsecurity` script. For information, see the Security Administration Guide.

- Use the `auditinit` utility. Tasks that you must perform before installing auditing and instructions on using the `auditinit` utility follow.
Preinstallation tasks for auditing devices

Determine the location of the devices for the sybsecurity, syslogs, and sysaudits table devices. You will need to provide this information later.

Sybase recommends that you:

- Configure your system with the minimum number of auditing devices you require—you must configure at least three devices. You can add more auditing devices later with sp_addaudittable. For information, see the Reference Manual: Procedures.

- Install auditing tables and devices in a one-to-one ratio. Tables that share the same device will share the same upper threshold limit. These tables cannot be used sequentially when a device fills up, because they both reside on the same device.

- Install each auditing table on its own device. This enables you to set up a smoothly running auditing system with no loss of auditing records. With two auditing tables, when one fills up, you can switch to the other. With a third auditing table, if one device fails, the System Security Officer can install a new threshold procedure that changes the device rotation to skip the broken device until the device is repaired.

- Make the device larger than the table. When you use only three auditing tables and devices, the size of the table and the size of the device can be similar, because you can obtain more auditing capacity by adding more auditing tables and devices (up to eight). When you are working toward the upper table and device limit (six to eight), you may want to make the device considerably larger than the table. Then, you can expand the table size later towards the upper size of the device when a larger auditing capacity is desired, and few or no device additions are available.

If you are using a file system device, either set the dsync attribute on, or use the directio attribute with that device. For more information, see “Using UNIX operating system files as database devices” on page 8.

Installing auditing

- Configuring Adaptive Server for auditing
  1. Source SYBASE.csh or SYBASE.sh file if you have not setup the Sybase environment variables.
  2. Start auditinit at the UNIX prompt:
Adding auditing

$SYBASE/$SYBASE_ASE/install/auditinit

auditinit displays the following menu:

AUDITINIT
1. Release directory: /usr/u/sybase
2. Configure a Server product
5. Select Configure an Existing Sybase Server.
6. Select the server to configure.
7. Provide the SA password for the server you selected.
8. From the Sybase Server Configuration screen, select Configure Auditing.

As you proceed through the menus in auditinit, you can change any default values that appear. As you finish each menu, press Ctrl+A to accept the defaults or changed values and move to the next menu.

CONFIGURE AUDITING
1. Configure auditing: no
2. Add a device for audit table(s)
3. Add a device for the audit database transaction log
4. Delete a device entry
5. Change a device entry

List of devices for the audit tables:
Logical name  Physical name  Segment name  Table name  Size

Device for the audit database transaction log:
Logical name  Physical name  Segment name  Table name  Size

9. From the Configure Auditing screen, select Configure Auditing.

auditinit redisplays the Configure Auditing menu with the value “yes” displayed for Configure Auditing.

10. Restart Adaptive Server for the changes to take effect.

Creating a device for an audit table

1. From the Configure Auditing screen, select Add a Device for Audit Table(s).

auditinit displays the following menu:
ADD/CHANGE A NEW DEVICE FOR AUDITING
1. sybsecurity physical device name:
2. Logical name of the device:
3. Size of the device (Meg):
4. Device size for auditing:

2 Select Sybsecurity Physical Device Name.

To create a device for an audit table:

1 Enter the full path of the physical device (file system or raw partition) that you located in “Preinstallation tasks for auditing devices” on page 117.

   Enter the physical name of the device to use for the audit database (default is " "):

   
   /dev/path_to_partition

   where path_to_partition is the path to the raw partition or filename for the device.

2 Press Return to acknowledge the warning.

auditinit redisplays the Add/Change a New Device for Auditing menu, which displays the physical name of the device:

ADD/CHANGE A NEW DEVICE FOR AUDITING
1. sybsecurity physical device name: /secret1/sybase_dr/install/aud1.dat
2. Logical name of the device:
3. Size of the device:
4. Device size for auditing:

3 Proceed through the remaining items on this menu.

Note The Size of the Device value must be equal to or greater than the Device Size for Auditing value. The Device Size for Auditing must be equal to the device size. If you are following Sybase auditing guidelines, you do not need to change the value displayed in Device Size for Auditing.

4 Press Ctrl+A to accept the settings. auditinit returns to the Configure Auditing menu and displays the device you have created.

CONFIGURE AUDITING
1. Configure auditing: yes
2. Add a device for audit table(s)
3. Add a device for the audit database transaction log
4. Delete a device entry
5. Change a device entry
Adding auditing

List of devices for the audit tables:
Logical name   Physical name   Segment name   Table name   Size

6. Audit_01' secret1/sybase_dr/install/aud1.dat' sysaudits_01  5

5. To add multiple audit devices, repeat steps 1–6.

You can add as many as eight devices. Sybase recommends adding three or more audit table devices.

After adding a device, auditinit returns to the Configure Auditing menu and displays all the devices you have created.

CONFIGURE AUDITING
1. Configure auditing: yes
2. Add a device for audit table(s)
3. Add a device for the audit database transaction log
4. Delete a device entry
5. Change a device entry

List of devices for the audit tables:
Logical name   Physical name   Segment name   Table name

6. Audit_01' /secret1/sybase_dr/install/aud1.dat' sysaudits_01  5
7. Audit_02' /secret1/sybase_dr/install/aud2.dat' sysaudits_02  5

❖ Creating a device for the audit database transaction log

1. From the Configure Auditing menu, select Add a Device for the Audit Database Transaction Log.

   auditinit displays the Add/Change a New Device for Auditing menu.

   ADD/CHANGE A NEW DEVICE FOR AUDITING
   1. sybsecurity physical device name:
   2. Logical name of the device:
   3. Size of the new device (Meg):
   4. Device size for auditing:

2. Select Sybsecurity Physical Device Name.

   auditinit prompts for the physical name and supplies you with a default, if available:

   Enter the physical name of the device to use for the sybsecurity database (default is''):
   /dev/path_to_partition
where `path_to_partition` is the path to the raw partition for the device.

3 Enter the full path name of a physical device.

4 Press Return to acknowledge this warning.

`auditinit` displays the Add/Change a New Device for Auditing menu and the value you selected for the physical name of the device.

```
ADD/CHANGE A NEW DEVICE FOR AUDITING
1. sybsecurity physical device name: /secret1/sybase_dr/install/auditlog.dat
2. Logical name of the device:
3. Size of the device:
4. Device size for auditing:
```

5 Proceed through the remaining items on this menu. As you do so, be aware of the following:

- Sybase recommends a minimum size of 2MB for the size of the transaction log.
- `auditinit` displays the size in both Size of the Device and in Device Size for Auditing in the Add/Change a New Device for Auditing menu.
- The Device Size for Auditing default value is equal to the size of the device, based on the assumption that you may want to devote the entire device to log for the auditing task. If you want to use only a subset of the device, you can edit the Size of the Device value.

6 Press Ctrl+A to accept the settings displayed in the Add/Change a New Device for Auditing menu.

`auditinit` returns to the Configure Auditing menu and displays all the devices you have created.

```
CONFIGURE AUDITING
1. Configure auditing: yes
2. Add a device for audit table(s)
3. Add a device for the audit database transaction log
4. Delete a device entry
5. Change a device entry

List of devices for the audit tables:
Logical name      Physical name      Segment name      Table name
size
6. Audit_01'   /secret1/sybase_ dr/install/audit1.dat' sysaudits_01 5
7. Audit_02'   /secret1/sybase_ dr/install/audit2.dat' sysaudits_02 5
8. auditlog    /secret1/.../auditlog.dat logsegment syslogs 2
```
Adding auditing

7 When you are ready to execute the audit configuration, press Ctrl+A. auditinit returns you to the Sybase Server Configuration screen.

8 Press Ctrl+A again. auditinit prompts with:

Execute the Sybase Server Configuration now?

9 Enter “y” (yes).

auditinit executes the tasks to install auditing. When the installation completes successfully, the following messages are displayed:

Running task: install auditing capabilities.
.................Done
Auditing capability installed.
Task succeeded: install auditing capabilities.
Configuration completed successfully.
Press <return> to continue.

Enabling auditing

After auditing is installed, no auditing occurs until a System Security Officer enables auditing using:

sp_configure 'auditing', 1

For more information, see the Security Administration Guide.

❖ Deleting a device entry

1 Select Delete a Device Entry from the Configure Auditing menu.

2 Enter the number of the device to delete.

3 Press return.

❖ Changing a device entry

1 Select Change a Device Entry from the Configure Auditing menu.

2 Enter the number of the device to change.

auditinit displays the Add/Change a New Device for Auditing menu with information on the device you selected:

ADD/CHANGE A NEW DEVICE FOR AUDITING
1. sybsecurity physical device name: 
   /secret1/sybase_dr/install/audlog
2. Logical name of the device: aud.log
3. size of the new device (Meg): 5
4. Device size for auditing: 5

3 Select each remaining entry you want to change.
4 Press Ctrl+A to save the new entries.

## Installing online help for Transact-SQL syntax

This section provides instructions for installing online help for Transact-SQL syntax.

## Online syntax help: `sp_syntax`

The $SYBASE/$SYBASE_ASE/scripts directory contains scripts for installing the syntax help database, sybsyntax. You can retrieve this data with `sp_syntax`. For more information on `sp_syntax`, see the Reference Manual: Procedures.

The `scripts` directory contains one or more of the `sp_syntax` scripts shown in Table 10-1, depending on which Sybase products are included with your server:

### Table 10-1: `sp_syntax` installation scripts

<table>
<thead>
<tr>
<th>Script</th>
<th>Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>ins_syn_cl</td>
<td>Open Client Client-Library™</td>
</tr>
<tr>
<td>ins_syn_esql</td>
<td>Embedded SQL™</td>
</tr>
<tr>
<td>ins_syn_os</td>
<td>Open Server</td>
</tr>
<tr>
<td>ins_syn_sql</td>
<td>Transact-SQL</td>
</tr>
</tbody>
</table>

All Adaptive Server installations include the `ins_syn_sql` script. This script includes syntax information for Transact-SQL, the system procedures, and the Sybase utilities. When you execute this script, you install the SQL portion of the sybsyntax database.

You can install any of these scripts, depending on the need for Sybase information on your server. The first script you execute creates the sybsyntax database and the needed tables and indexes. Any scripts that you execute after the first one add to the existing information in the database. If you execute a script that was executed previously, the previously installed rows of information are deleted from the table in the database and then reinstalled.

**Warning!** The `ins_syn_cl` and `ins_syn_os` scripts conflict. If you execute both scripts, errors occur.
Default device for the `sybsyntax` database

The `sybsyntax` database requires space on the device that is at least as large as the model database. By default, the `sybsyntax` installation scripts install the `sybsyntax` database on the device that is designated as the default database device.

If you have not used `sp_diskdefault` to change the status of the master device (which is installed as the default disk) or to specify another default device, the scripts install `sybsyntax` on the master device. Sybase does not recommend this configuration because `sybsyntax` uses valuable space, which is best left available for future expansion of the master database.

To avoid installing `sybsyntax` on the master device, do one of the following:

Use `sp_diskdefault` to specify a default device other than the master device. For information about `sp_diskdefault`, see the Reference Manual: Procedures.

- Modify each `sybsyntax` installation script that you plan to execute to specify a different device, as explained in the following section.

Installing `sybsyntax`

For each `sybsyntax` installation script you want to execute:

1. Determine the type (raw partition, logical volume, operating system file, and so on) and location of the device where you plan to store the `sybsyntax` database. You will need to provide this information later.

2. Make a copy of the original script. Be sure you can access this copy, in case you experience problems with the edited script.

3. Use a text editor to edit the script, if necessary, to change the default device from the master device to the device created in step 1. For information on the default device, see “Default device for the `sybsyntax` database” on page 124.

- Comment out the following section, which specifies the default device:

```sql
/* create the database, if it does not exist */
if not exists (select name from sysdatabases
where name = "sybsyntax")
begin

/* create the `sybsyntax` table if it doesn’t exist */
/* is the space left on the default database
devices > size of model? */
```
if (select sum (high-low +1) from sysdevices where status & 1 = 1) - (select sum(size) from sysusages, sysdevices where vstart >= sysdevices.low and vstart <= sysdevices.high and sysdevices.status &1 = 1) >
(select sum(sysusages.size) from sysusages where dbid = 3)
begin
  create database sybsyntax
end
else
begin
  print "There is not enough room on the default devices to create the sybsyntax database."
  return
end

• After you have commented out this entire section, add a line like this to the script:

create database sybsyntax on device_name

where device_name is the name of the device where you want to install sybsyntax.

4 Execute the script with a command like the following:

```
isql -Usa -Ppassword -Sservername < $SYBASE/$SYBASE_ASE/scripts/ins_syn_sql
```

where sa is the user ID of the System Administrator, password is the System Administrator’s password, and servername is the Adaptive Server where you plan to install the database.

If you have set the DSQUERY environment variable to the servername, you can replace the server name with $DSQUERY.

5 To ensure that you have installed the sybsyntax database and that it is working correctly, use isql to log in to the server on which you installed the database, and execute sp_syntax. For example:

```
isql -Usa -Ppassword -Sservername

1> sp_syntax "select"
2> go
```

Adaptive Server displays a list of commands that contain the word or word fragment “select.”
Installing online help for Transact-SQL syntax
Index

A
accented letters 14, 93
Adaptive Server
  character set, changing 85
  character sets 91
  client communications with 51
  conversions between, and clients 91
  customizing features 49
  database device requirements 10
  default character set 85
  default configuration 49, 50
  default sort order 85
  error log path 108
  language, changing 85
  naming in interfaces file 58
  shutting down 26
  sort order 85
  starting from UNIX command line 20
  starting with operating system 22
  start-up script 23
adding a server, LDAP 78
address component in interfaces files
  TCP protocol entry 59
administrator
  operating system 3
  Sybase system 3
API component in interfaces file described 58
Arabic character sets 87
asynchronous I/O (AIO)
  enabling 42
  HP driver 42
audit system 115
audit trail
  overview 115
  system audit tables 116
auditing
  database for 116
  device for 116

B
Backup Server
  character sets 91, 101
  configuring 96, 99
  default configuration 50
  default, for Adaptive Server 50
  starting from UNIX command line 20
  starting with operating system 22
binary sort order 93
buffer specifications 16

C
changing
  character sets 85, 99
  languages 99
  sort order 99
character devices
  improving I/O performance on 42
character sets 91
  accented letters in 14
  changing 85, 86, 99
  client selection of 86
  code conversions and 91
  configuring 101
  converting between 91
  databases and 92
  default 85
  in a heterogeneous environment 91
  global options 116
  installing using the auditinit utility 116
  installing using the installsecurity script 116
  process 116
  recommended database device size 10
  tables for tracking 116
  auditinit utility 7
Configuration Guide 127
Index

sort orders and 92
U.S. English 14
charsets directory 93, 96
about the 97
Chinese character sets 88
client interfaces files
difference between client and server versions 54
heterogeneous 55
homogeneous 55
clients
Adaptive Server communications with 51
applications and locales.dat file 98
conversion between, and server 91
default character set 86
DSQUERY and 63
file servers 63
code conversion
between character sets 91
collating sequences. See sort orders 92
commands
disk init 11
iostat 46
maxfiles 36
maxfiles_lim 36
netstat 46
netstat -v 46
no -a 46
ps 46, 47
sar 45
setenv 31
setperm_all 36
time 46, 47
vmstat 45, 46
common.loc file 97
communications between client and Adaptive Server 51
configurations, default 50
configuring
Backup Server 96, 99
character sets 101
conversions, Unicode character 87
converting between character sets 91
create database command, system tables created by 5
creating
interfaces files 53, 61
interfaces files automatically 52
interfaces files for beginners 60
master interfaces files with dscp utility 60
master interfaces files with dsedit 60
master interfaces files with text editor 60
customizing installations 49
Cyrillic character sets 88

D

data translation 83
database devices
master 5
preparing raw partitions for 11
sybsyntax 124
sybsystemdb 5
sybsprocsdev 5, 6
databases 92
dbccdb 8
devices 111
master 5, 6
model 5
pcidb 7
sample 7
sizes of 14
specifications 14
sybsecurity 7
sybsystemprocs 6, 14
tempdb 5
dataserver utility 28
dbcc checkstorage, database for 8
dbcc error messages 48
dbccdb database 8
debug service type 58
Dec-Kanji character set 91
default
character set for Adaptive Server 85
character set, changing 85
character sets installed by 85
language for Adaptive Server 85
language, changing 85
sort order 85
delay_interval component in interfaces files 58
device component in interfaces files 59
devices
files 111
dictionary sort orders 93
<table>
<thead>
<tr>
<th>Page</th>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>94</td>
<td>Scandinavian</td>
</tr>
<tr>
<td>94</td>
<td>Spanish</td>
</tr>
<tr>
<td>9</td>
<td>directio</td>
</tr>
<tr>
<td>93, 97</td>
<td>charsets</td>
</tr>
<tr>
<td>96</td>
<td>localization</td>
</tr>
<tr>
<td>24</td>
<td>run control</td>
</tr>
<tr>
<td>75</td>
<td>directory schema, LDAP</td>
</tr>
<tr>
<td>11</td>
<td>disk init command</td>
</tr>
<tr>
<td>36</td>
<td>displaying current file descriptors</td>
</tr>
<tr>
<td>84</td>
<td>Adaptive Server translated</td>
</tr>
<tr>
<td>13</td>
<td>dscep utility</td>
</tr>
<tr>
<td>60</td>
<td>creating master interfaces files with</td>
</tr>
<tr>
<td>78</td>
<td>adding an LDAP server</td>
</tr>
<tr>
<td>13</td>
<td>dsedit utility</td>
</tr>
<tr>
<td>60</td>
<td>creating master interfaces files with</td>
</tr>
<tr>
<td>3</td>
<td>DSLISTEN environment variable</td>
</tr>
<tr>
<td>4</td>
<td>DSQUERY environment variable</td>
</tr>
<tr>
<td>63</td>
<td>client connections and</td>
</tr>
<tr>
<td>53</td>
<td>described</td>
</tr>
<tr>
<td>64</td>
<td>naming in</td>
</tr>
<tr>
<td>59</td>
<td>error log paths</td>
</tr>
<tr>
<td>108</td>
<td>configuring</td>
</tr>
<tr>
<td>48</td>
<td>errors in dbcc messages</td>
</tr>
<tr>
<td>84</td>
<td>error logging</td>
</tr>
<tr>
<td>2</td>
<td>extended stored procedures (ESPs)</td>
</tr>
<tr>
<td>59</td>
<td>/etc/services file</td>
</tr>
<tr>
<td>59</td>
<td>ether placeholder in interfaces files</td>
</tr>
<tr>
<td>91</td>
<td>EUC-JIS character set</td>
</tr>
<tr>
<td>2</td>
<td>extended stored procedures (ESPs)</td>
</tr>
</tbody>
</table>

**F**

<table>
<thead>
<tr>
<th>File descriptors</th>
<th>file servers</th>
<th>63</th>
</tr>
</thead>
<tbody>
<tr>
<td>36</td>
<td>36</td>
<td></td>
</tr>
<tr>
<td>97</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>91</td>
<td>91</td>
<td></td>
</tr>
<tr>
<td>92</td>
<td>92</td>
<td></td>
</tr>
<tr>
<td>97</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>

**G**

<table>
<thead>
<tr>
<th>German sample database</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Globalization support, Sybase</td>
<td>49, 83, 96, 99</td>
</tr>
<tr>
<td>Greek character sets</td>
<td>89</td>
</tr>
</tbody>
</table>

**H**

<table>
<thead>
<tr>
<th>Hardware error messages</th>
<th>44</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM RS/6000</td>
<td>44</td>
</tr>
<tr>
<td>UNIX</td>
<td>44</td>
</tr>
<tr>
<td>Hebrew character sets</td>
<td>89</td>
</tr>
<tr>
<td>Heterogeneous environments</td>
<td>86, 91</td>
</tr>
<tr>
<td>described</td>
<td>55</td>
</tr>
<tr>
<td>Interfaces files and</td>
<td>55</td>
</tr>
<tr>
<td>Homogeneous environments</td>
<td>55</td>
</tr>
<tr>
<td>described</td>
<td>55</td>
</tr>
<tr>
<td>Interaces files and</td>
<td>55</td>
</tr>
<tr>
<td>Host component in ...</td>
<td>59</td>
</tr>
<tr>
<td>Host name</td>
<td>59</td>
</tr>
</tbody>
</table>
Index

HP-UX
  netstat command  46
  network protocol  59
  ps command  46
  sar command  45
  time command  46
  timeout period  43
  vmstat command  45

I
  I/O monitoring  45
  IBM RS/6000
    hardware error messages  44
    iostat command  46
    monitoring systems  46
    netstat command  46
    netstat -v command  46
    network protocol  59
    no -a command  46
    ps command  46
    time command  46
    timeout period  43
    vmstat command  46
  interfaces file  12
  interfaces files
    Adaptive Server, naming in  58
    Adaptive Server, used by  55
    address component for SPX  59
    API component in  58
    automatic creation of  52
    client and server versions, differences in  54
    clients, used by  51
    contents of  53
    creating automatically  52
    creating master files with dscp  60
    creating master files with dsedit  60
    creating master files with text editor  60
    creating, for beginners  60
    debug service type  58
    default location  52
    delay_interval component  58
    described  51
    device component  59
    ether placeholder  59
    heterogeneous environments and  53
    homogeneous environments and  53
    host component  59
    location  52
    loghost placeholder  61
    machine component  59
    master service type  58
    multiple network listeners  62
    multiple networks  53, 61
    network component  59
    port component  59
    protocol component  59
    query port backup configuration  64
    query service type  58
    retry_attempt component  58
    servername component  58
    service_type component  58
    spaces in  57
    tab characters in  57
    unique elements in entries  61
    used by clients  53
    international systems
      support for  83
      Sybase support for  83
    interpubs sample database  7
    iostat command
      IBM RS/6000  46
      Sun Solaris  46
    iso-Latin1 character set  85

J
  Japanese sample database  7
  jpubs sample database  7

K
  KEEPALIVE option, TCP/IP  43
  kill command  27, 29
  Korean character sets  89
Index

L
language modules 95
default 49
installing new 95
localization files 84
memory requirements for 100
translated system messages 84
languages
changing 99
error reporting in specific 97
selecting message 95
translation support 83
Latin character sets 88
LD_LIBRARY_PATH environment variable 4
LD_LIBRARY_PATH64 environment variable 4
LDAP
access restrictions 72
adding a server 78
defined 71
directory definitions 73
directory schema 75
enabling 76
multiple directory services 79
sample entry 74
specifying in libtcl.cfg 75
versus the interfaces file 72
LDAP libraries
environment variables 77
location of 77
LDAP server
using dsedit to add and modify 78
ldapurl
defined 76
example 76
keywords 77
letter case in sort orders 93
LIBPATH environment variable 4
libtcl*.cfg file 75
format of 76
location of 76
purpose of 75
libtcl*.cfg
password 80
limits for file descriptors 37
listener service 54
loc files 97

local date, time, and currency formatting 97
locales directory 96
locales.dat file 97
localization 83
changing the configuration 99
common, information 97
localization support 49
loghost in interfaces files 61
login
root 3
sa 3
superuser 3
sybase 3

M
machine component in interfaces files 59
master
database 5
device 5, 10
interfaces file 53, 61
service type 58
maxfiles kernel parameters 36
maxfiles_lim kernel parameters 36
mbuf pools 46
messages
hardware errors 44
selecting language for 95
model database 5
monitoring
I/O 45
network status 46
operating system resources 45
virtual memory usage 45
monitoring systems
IBM RS/6000 46
NCR 46
multiple directory services
LDAP 79
multiple installations
affecting interfaces files 55
creating one interfaces file for 53, 61
multiple networks
interfaces files and 53
interfaces files for 61
Index

used as a backup in case of network failure 64

N
named pipes
  default pipe 50
  naming requirements for servers 58
NCR
  monitoring systems 46
netstart -v command (IBM RS/6000) 46
netstat command
  HP-UX 46
  IBM RS/6000 46
  Sun Solaris 46
network component in interfaces files 59
network protocols
  Digital UNIX 59
  HP-UX 59
  IBM RS/6000 59
  SPX 12
  Sun Solaris 59
  TCP/IP 12
  UnixWare 59
network support
  default configuration 49, 50
networks
  backup connections and 65
  DSQUERY and 63
  failure of 65
  interfaces files 51
  monitoring status 46
  multiple 53
no -a command (IBM RS/6000) 46

O
online syntax help 123
operating system
  administrator 3
  resources 45

P
parent process identification (PPID) 28
partition map
  avoiding damage to 11
password encryption
  for libtcl*.cfg 80
  pwdcrypt 80
PATH environment variable 4
paths, error log 108
permissions 36
  restoring of 36
PID. See process identification
platform-specific locale names 97
pluggable component interface (PCI) 7
port component in interfaces files 59
port numbers and interfaces files 62
PPID. See parent process identification
privileges 11
procedure specifications 16
procedures, Sybase extended stored
  process identification (PID) 28
  protocol
    component in interfaces files 59
    SPX 59
    TCP/IP 59
ps command
  HP-UX 46
  IBM RS/6000 46
  Sun Solaris 47
pubs2 sample database 7
pubs3 sample database 7
pwdcrypt
  location of 80
  password encryption 80

Q
query
  port backup configuration 64
  service type 54, 58
  specifications 15
Index

R
raw block devices
  See character devices
raw partitions
  availability of 11
  minimum size for database devices 11
referential integrity constraint 16
retry_attempts component in interfaces files 58
roman8 character set 85
run control directory 24
runserver file 21

S
sar command
  HP-UX 45
Scandinavian dictionary sort orders 94
scripts
  C shell 47
  maintenance 47
  sample maintenance 47
  start-up 22, 23
security. See auditing
servername component in interfaces files 58
servers
  naming requirements 58
service types
  debug 58
  listener 54
  master 58
  query 54, 58
service_type component in interfaces file 58
setenv command 31
setperm_all command 36
shared memory files 28
Shift-JIS character set 91
shutdown command 26, 27
size
  sybsystemprocs database, minimum required for upgrade 14
slloc utility 93
SMIT
  See System Management Interface Tool
sort orders 92
  binary 93
  changing 86, 99
  character sets and 92
databases and 92
  default for Adaptive Server 85
definition files 92
dictionary 93
  letter case in 93
spaces in interfaces files 57
Spanish dictionary sort orders 94
SPX network protocol 12, 59
srt files 92
svrbuild utility 13
starting servers
  from UNIX command line 20
  startserver utility 21
start-up script 22
stty settings 35
Sun Solaris
  iostat command 46
  netstat command 46
  network protocol 59
  ps command 47
time command 47
timeout period 43
  vmstat command 46
sundiag system diagnostic tool 45
SYBASE environment variable 4
$SYBASE environment variable as default server name 61
Sybase globalization support 83, 96, 99
SYBASE_ASE environment variable 4
SYBASE_OCS environment variable 4
SYBASE_SYSAM environment variable 4
sybsecurity
  database 7, 116
  device 7, 10
sybsyntax database 123
sysystemdb
  device for 10
  purpose of 5
sysystemprocs database 6
sysprocsdev device
  minimum size for 10
  purpose of 5, 6
system audit tables 116
system databases
Index

where stored 10
System Management Interface Tool (SMIT) 42
system messages, translated 84
system procedures, storage location of 6

T
tab characters in interfaces files 57
table specifications 15
TCP/IP 43, 59
    KEEPALIVE option 43
TCP/IP network protocol 12
tempdb database 5
Thai character sets 89
time command
    HP-UX 46
    IBM RS/6000 46
    Sun Solaris 47
TLI protocol 58
translated messages
    error (.loc files) 97
    system 84
Turkish character sets 90

U
Unicode
    character conversion 87
UNIX
    hardware error messages 44
    network protocol 59
UnixWare
    network protocol 59
    us_english language 85
    user connections 36
    user-defined message 109
    utilities
        slloc 93

V
vmstat command
    HP-UX 45

IBM RS/6000 46
Sun Solaris 46

W
Windows Sockets
    default socket 49, 50

X
XP Server
    default configuration 50
XP Server, starting 19
xp_cmdshell command 19