Configuration Guide

# Adaptive Server® Enterprise

# 15.7 ESD #2

UNIX

DOCUMENT ID: DC35823-01-1572-01
LAST REVISED: July 2012

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

# Contents

# CHAPTER 1 About Adaptive Server

Adaptive Server® Enterprise 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.

**Note:** These instructions 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.

Adaptive Server Enterprise is a full-featured Adaptive Server that runs on:

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

## 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

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.

## Environment Variables

The operation of Sybase products relies on system environment variables being 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).

**Note:** The installer configures these environment variables in the `SYBASE.csh` and `SYBASE.sh` files. Source these file to set the user's environment.

- DSLISTEN – defines the name Adaptive Server uses to listen for client connections if no name is given during the Adaptive Server start-up. If DSLISTEN is not set, and no name is given during the Adaptive Server start-up, the Adaptive Server name defaults to SYBASE.
- 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 the directory path to search for executables. The Sybase executables are located in the installed componen'ts `/bin` directory. When using the source files `SYBASE.csh` or `SYBASE.sh`, these 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`, these 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`, these 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`, these paths are prepended to the LIBPATH environment variable:

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

### See also

- *Verifying Environment Variables* on page 35

# CHAPTER 2 Adaptive Server Devices and System Databases

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

These devices are created when you create a new Adaptive Server:

- `master` – stores system databases.
- `sybsystemdb` – stores information about distributed transaction.
- `sysprocsdev` – stores system procedures.

## The master Device

The master device contains the `master`, `tempdb`, `model`, and sample databases.

- `master` – controls the operation of Adaptive Server 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 `tempdb` databases. 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.
- The sample databases, `pubs2` and `pubs3`, 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

The `sybsystemdb` device stores the `sybsystemdb`database, which stores information about transactions in progress, and which is also used during recovery.

For new installations, the `master` device also contains the `sybsystemdb` database.

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 `sybprocsdev` 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. The system procedures that are needed, for example `sp_configure`, 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

Optional devices and databases include the pluggable component interface (PCI) database, the sample databases, the `sybsecurity` device and database, and the `dbccdb` database.

## Pluggable Component Interface (PCI) Database

The pluggable component interface (PCI) lets you add 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 and later.

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
```

## Sample Databases

`pubs2`, `pubs3`, `interpubs`, and `jpubs` are the available sample databases.

- `pubs2` and `pubs3` are 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 US English versions of Adaptive Server.

- `interpubs` database contains French and German data.
- `jpubs` contains Japanese data.

For information about installing the sample databases, see *Post-Installation Tasks* in the *Installation Guide* for your platform.

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

## sybsecurity Device and Database

The `sybsecurity` device is created as part of the auditing installation process, storing the `sybsecurity` database and the system procedures with which to configure auditing.

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.

See *Auditing* in the *Security Administration Guide*.

### See also

- *Install Auditing* on page 108
- *Chapter 18, Database Management System Auditing* on page 107
- *Determine the Size of a Database Device* on page 9

## dbccdb Database

Database consistency checker (**dbcc**) commands lets you check the logical and physical consistency of a database. The `dbccdb` database stores the results of **dbcc checkstorage** or **dbcc checkverify**.

**dbcc checkstorage** records configuration 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.

See *Checking Database Consistency* in the *System Administration Guide: Volume 2*.

## sybmgmtdb 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 procedure, which make 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.

See the *Job Scheduler Users Guide*.

## Using Operating System Files as Database Devices

For devices that are initialized on operating system files, ensure that device writes occur directly on the physical media.

Use:

- **directio** with **disk init**, **sp_deviceattr**, and **disk reinit**
- **dsync** 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 the same device. To enable **directio** for a device, also reset **dsync** to false.

### The directio Parameter

The **directio** parameter for **disk init**, **disk reinit**, and **sp_deviceattr** lets you bypass the operating system buffer cache and transfer Adaptive Server data directly to disk.

**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 true (on) for all platforms. **directio** and **dsync** are ignored for raw devices.

See the *System Administration Guide, Volume 1.*

#### See also

- *The dsync Parameter* on page 8
- *Preinstallation Recommendations for Auditing Devices* on page 108

### The dsync Parameter

The **dsync** parameter ensures Adaptive Server can recover data from devices on file systems.

By default, Adaptive Server disables **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. When **dsync** is set to false (off), cached I/O is used.

**Note: dsync** and **directio** are ignored for raw devices.

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

#### See also

- *The directio Parameter* on page 8

- *Preinstallation Recommendations for Auditing Devices* on page 108

## Determine the Size of a Database Device

Adaptive Server requires baseline values for each database device.

See the release bulletin for any last-minute changes to these values.

**Table 1. Adaptive Server Database Devices**

| Device | Purpose | Minimum size | Minimum recommended size |
|---|---|---|---|
| `master` | Stores system databases | 24MB for 2K pages<br>45MB for 4K pages<br>89MB for 8K pages<br>177MB for 16K pages | 30MB<br>60MB<br>120MB<br>240MB |
| `sysprocsdev` (also called the `sybsystemprocs` device) | Stores the `sybsystemprocs` database | 160MB | 172MB (plus any space for holding stored procedures you have created) |
| `sybsystemdb` | Transaction processing | 3MB for 2K pages<br>6MB for 4K pages<br>12MB for 8K pages<br>24MB for 16K pages | 5 – 24MB |
| `sybsecurity` (optional) | Required for auditing | 10MB or size of the `model` database, whichever is larger | Depends on specific auditing needs |

### See also

- *sybsecurity Device and Database* on page 7
- *Install Auditing* on page 108
- *Chapter 18, Database Management System Auditing*  on page 107

## Support for Raw Partitions or Files

You can use either a raw partition or a file system for all databases.

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. No operating system or Adaptive Server changes are required.

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

## Preparation of a Raw Partition Device

Increase performance and avoid file damange by using best practices when preparing a raw partition device.

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

- Do not initialize a database device on the partition that contains your Sybase installation software. Doing so destroys all existing files on that partition.
- You cannot mount a raw partition designated for use by Sybase 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**.
- 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

Determine the size and availability of raw devices.

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.
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.

   For more information on choosing a raw partition, see your operating system documentation.

# CHAPTER 3 Client/Server Communication

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

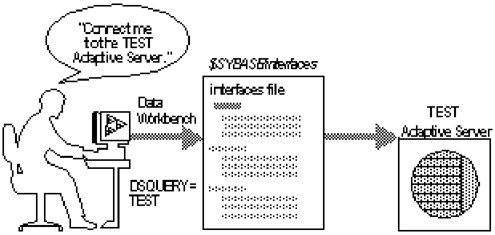
- In an interfaces file, named `interfaces` on UNIX platforms, located in the `$SYBASE` installation directory, or
- In an LDAP server.

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. 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*).

**Figure 1: Communicating with a Server Using the interfaces File**



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 about how to modify existing interfaces file entries using **dsedit** and **dscp**, see the *Utility Guide*.

### See also

- *Chapter 14, Use the Lightweight Directory Access Protocol as a Directory Service* on page 69

# CHAPTER 4 About Changing Adaptive Server Configuration

Use **sp_configure** to change Adaptive Server configuration.

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*.

You can also:

- Configure languages, character sets, and sort orders.
- Configure Adaptive Server to use high availability features. See *Using Sybase Failover in a High Availability Environment*.
- Configure Adaptive Server for distributed transaction management (two-phase commit). See the *System Administration Guide: Volume 2*.

### See also

- *Changing Adaptive Server Localization Configuration* on page 98

# CHAPTER 5 Languages Other Than US English

If you are running **srvbuild** in a language other than US 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*.

# CHAPTER 6 Adaptive Server Specifications

Adaptive Server specifications include databases, tables, queries, procedures, and extended-limit capabilities information about Adaptive Server requirements.

## Database Specifications

Database specifications define requirements for items such as database size, and the number of database devices per server.

| Item | Requirement | Notes |
|---|---|---|
| Databases per Adaptive Server | A maximum of 32,767 databases per server | |
| Maximum database size | • 2K page server – 8TB<br>• 4K page server – 16TB<br>• 8K page server – 32TB<br>• 16K page server – 64TB | Because Adaptive Server reserves 256 logical page IDs—which cannot be allocated or used—at the high end of the logical page range, the sizes listed above are slightly higher than the actual amount of space available for use. This overhead reduces the actual amount of space available by 256 times the logical page size for each listed page size (for example, the actual available size for a 2K server is 8TB – (256 x 2K)). |
| Minimum allowable `sybsystemprocs` database | 136MB | Required for an upgrade |
| Maximum size of a database device (disk partition) | $2^{42}$ (4TB) | If the operating system supports file sizes up to 4TB, then Adaptive Server supports file system devices up to 4TB |
| Maximum number of database devices per server | $2^{31}$ | |
| Maximum number of devices or device pieces per database | Unlimited | Limited by available memory |

| Item | Requirement | Notes |
|---|---|---|
| Maximum number of segments per database | 31 | |
| Maximum number of login IDs per serve | 2147516416 | |
| Maximum number of users per database | 2146484223 | |
| Maximum number of groups per database | 1032193 | |

## Table Specifications

Table specifications define requirements for items such as indexes, rows, and columns per table.

| Item | Requirement | Notes |
|---|---|---|
| User objects per database | $2^{31} – 255$ | |
| Indexes per table | 250 (one clustered index) | |
| Rows per table | Limited by available storage | Maximum $2^{32}$ |
| Columns per composite index | 31 | |
| Creation of clustered index | 1.2*(x + y)<br>x = total data space in table, y = sum of space of all nonclustered indexes on table, and 20 percent overhead for logging | For sorted data, approximately 20 percent of the table size needed |
| Maximum size of object name | 255 | |

## Procedure Specifications

Procedure specifications list items such as number of buffers and procedure buffers, and the required minimum memory per stored procedure.

| Item | Requirement | Notes |
|---|---|---|
| Number of buffers and procedure buffers | Configurable | Limited by amount of RAM and maximum size of shared memory segment |

| Item | Requirement | Notes |
|---|---|---|
| Minimum memory required per stored procedure | 2K | |
| Maximum number of parameters per stored procedure | 2048 | |

## Query Specifications

Query specifications define requirements for items such as maximum number of tables in a "union" query.

| Item | Requirement | Notes |
|---|---|---|
| Maximum number of tables participating in a query, for a query without a union, or for each side of a union in a query | 64 | 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 |
| Maximum number of tables in a “union” query | 256 | Includes up to 50 user tables and 14 worktables on every side of the union, for a maximum of 256 tables across all sides of the union |
| Maximum number of databases participating in one transaction | Unlimited | Includes database where transaction began, all databases changed during transaction, and `tempdb`, if it is used for results or worktables |
| Practical number of databases participating in one query | 16 | Includes each occurrence of each database queried and `tempdb`, if it is used for results or worktables |
| Maximum number of tables with referential integrity constraints for a query | 192 | |

## Maximum Column Sizes

Adaptive Server extended-limit capabilities vary by type of table and the database logical page size

The column and row limits for allpages-locked (APL) tables are:

| Maximum APL Table Limits | Number of Columns | Column Size 2K Page | Column Size 4K Page | Column Size 8K Page | Column Size 16K Page |
|---|---|---|---|---|---|
| Fixed-length column | 1024 | 1960 bytes | 4008 bytes | 8104 bytes | 16296 bytes |
| Variable-length column | 254 | 1948 bytes | 3988 bytes | 8068 bytes | 16228 bytes |

The column and row limits for data-only-locked (DOL) tables are:

| Maximum DOL Table Limits | Number of Columns | Column Size 2K Page | Column Size 4K Page | Column Size 8K Page | Column Size 16K Page |
|---|---|---|---|---|---|
| Fixed-length column | 1024 | 1958 bytes | 4006 bytes | 8102 bytes | 16294 bytes |
| Variable-length column | 1024 | 1954 bytes | 4002 bytes | 8098 bytes | 16290 bytes |

## Database Requirements for Varying Page Sizes

Database space requirements depend on 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.

The database requirements for varying page sizes for each database are:

| Databases | 2K page | 4K page | 8K page | 16K page |
|---|---|---|---|---|
| `master` database | 13MB | 26MB | 52MB | 104MB |
| `model` database | 3MB | 6MB | 12MB | 24MB |

| Databases | 2K page | 4K page | 8K page | 16K page |
|---|---|---|---|---|
| `tempdb` database | 4MB | 6MB | 12MB | 24MB |
| `sybsystemdb` database | 3MB | 6MB | 12MB | 24MB |
| `sybpcidb` database | 24MB | 48MB | 96MB | 192MB |

## Data Limits for Tables According to Page Size

Larger logical page sizes can contain more data.

The data limits for tables according to page size:

| Tables | 2K page | 4K page | 8K page | 16K page |
|---|---|---|---|---|
| Bytes per index key | 600 | 1250 | 2600 | 5300 |
| User-visible row length DOL table | 1958 | 4006 | 8102 | 16294 |
| User-visible row length APL table | 1960 | 4008 | 8104 | 16296 |

# CHAPTER 7 Start and Stop Servers

Start and stop Adaptive Server, Backup Server, and XP Server.

XP Server is not started by the installation process. XP Server is started by Adaptive Server when you issue an XP command 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.

### See also

- *Migrating from the interfaces File to LDAP* on page 77
- *Adding a Server to the Directory Services* on page 74
- *Set the Adaptive Server Error Log Path* on page 102

## Start Servers

Start Adaptive Server or Backup Server from the command line.

You can start Adaptive Server or Backup Server using `RUN_server_name` files and **startserver** commands. You can also customize the start up options by editing `RUN_server_name`.

To start a server, your user account must have:

- Access to the database servers (execute privileges) and database devices (read/write privileges).
- Access to Adaptive Server distribution files.
- System environment variables set.
- Access to SySAM licenses. See the *Sybase Software Asset Management User Guide*.

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

### Server Start-up Parameters

Set the start-up parameters to the correct location and device names for starting an Adaptive Server or Backup Server.

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.

You cannot change any of these default start-up parameters unless you 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.

Backup server names are appended with "_BAK".

**Table 2. Default Adaptive Server Start-Up Parameters**

| Switch | Description |
|---|---|
| **-d** `$SYBASE/data/master.dat` | Location of the master device file |
| **-s** `server_name` | Name of the Adaptive Server |
| **-e** `$SYBASE/$SYBASE_ASE /install/errorlog` | Location and name of the error log file |
| **-M** `$SYBASE/$SYBASE_ASE` | Directory that stores shared memory files |
| **-N**`$SYBASE/$SYBASE_ASE/sysam/<srv_name>.properties` | Location and name of license cache file. |

Additional start-up parameters include any valid server command line options listed for the **databaserver** and **backupserver** descriptions in the *Utility Guide*.

## 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_`*`servername`* file is named `RUN_`*`servername`* , where *servername* is the name of the server. For example, the `RUN_`*`servername`* file name for an Adaptive Server named TEST is `RUN_TEST`.

For Backup Server, the `RUN_`*`servername`* files is named `RUN_`*`servername`*`_back`.

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

## Using the startserver Command

Use **startserver** to start a server from the command line.

### Prerequisites

Read and write permissions on the master device.

**Task**

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.

For more information about **startserver**, see the *Adaptive Server Utility Guide*.

# Start Servers When the Operating System Restarts

Configure your operating system to automatically restart Adaptive Server and Backup Server.

On production systems, set up Adaptive Server and Backup Server to restart automatically when the UNIX operating system starts. Do this by making an entry for the server in the operating system start-up script.

**Note:** The start-up script must set up all the required Sybase environment variables, or source `SYBASE.csh` or `SYBASE.sh,` before starting the server.

## Creating a System Restart Script for HP-UX

Create a script that executes when the operating system starts or shuts down.

1. As system administrator must create a script that executes when the operating system starts or shuts down.
   You can model your script after the HP-UX template file `/sbin/init.d/template`. You cannot edit commands in the `/etc/rc` file in HP-UX versions 10.0 and later. For more information about **rc (1M)**, see the HP-UX manual page.
2. After creating the start-up and shutdown script, place it in the directory `/sbin/init.d`.
   Execution scripts 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` control the sequencing order of the execution scripts.

## Creating a System Restart Script for IBM RS/6000

On production systems, you can restart Adaptive Server automatically whenever the UNIX operating system restarts.

### Prerequisites

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.

### Task

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 the entry for starting Adaptive Server follows any entry for `/etc/rc.tcpip` and `/etc/rc.nfs` in the `/etc/inittab` file.

Slow start-up of network operations may 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.

## Creating a System Restart Script for Sun Solaris and Linux

Create a script that automatically starts or shuts down Adaptive Server or Backup Server when the operating system starts or shuts down.

### Prerequisites

Make sure all network resources are available. Your server does not start if the network is not running. Be sure 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.

### Task

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.

1. Create a start-up script file with contents such as:

   ```
   $SYBASE/$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 such as:

   ```
   cp script_name /etc/init.d/script_name
   ```
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 Sun Solaris syntax such as:

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

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

   Use Linux syntax such as:

   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 such as:

   ```
   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 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**, enter:

**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`, enter:

```
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

Add information about the XP Server to the `sysservers` table.

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 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 installation, XP Server cannot run. When you execute an extended stored procedure (ESP), you see an error message similar to:

```
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.

# Stop Servers

Only the system administrator can issue a **shutdown** command. Using a **shutdown** command minimizes the amount of work for automatic recovery when servers are restarted.

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

## Stopping Adaptive Server

Only a system administrator can shut down an 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 such as 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 waits for currently executing statements to finish, nor performs checkpoints in every database.

   **Note:** Sybase recommends that you use the **shutdown with nowait** command only when necessary.

## Stopping Backup Server

Only a system administrator can shut down a Backup Server.

1. Use **isql** to log in to a server with system administrator privileges.
2. Enter:

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

   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 `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.

**Note:** Sybase recommends that you use the **shutdown with nowait** command only when necessary.

See the *Reference Manual: Commands*.

## Using the kill Command

You can use the **kill** command to stop Adaptive Server and Backup Server processes.

**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.

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 multiengine server is to specify the process ID for engine 0.

This `showserver` output shows the processes for a four-engine server when Adaptive Server runs in process kernel mode. In the default threaded mode each engine is not a process, therefore only one process is listed in `showserver`.

```
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

The method you select to shut down Adaptive Server or Backup Server may affect shared memory segments.

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 greater than the MAXSHMSEGSIZE parameter in the operating system, 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 fails 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 failure or a **kill -9** command, because Adaptive Server must be able to overwrite the previously created shared memory files.

Killing Adaptive Server or Backup Server abnormally also leaves shared memory segments. Use the **ipcs** and **ipcrm** commands to identify and remove these shared memory segments that show a “NATTACH” count of 0.

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

# CHAPTER 8 Huge Pages on Linux

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

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

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

By default, Adaptive Server versions 15.0.3 and later allocate shared memory using huge pages. 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:

```
cat /proc/meminfo
....
HugePages_Total:    32
HugePages_Free:     32
Hugepagesize:    16384 kB
```

**Note:** Memory you allocate for huge pages is used only for the shared memory. Allocating too many huge pages may lead to Adaptive Server excessively swapping physical pages. Allocate only the required number of huge pages.

# CHAPTER 9 Configure the Operating System

After installing or updating Adaptive Server, adjust configuration settings.

You can configure some Adaptive Server settings during installation. Unless stated otherwise, configuration information pertains to all supported UNIX platforms. See the *Installation Guide* for more details.

## Verifying Environment Variables

Use **env** at the operating system prompt to verify that the Adaptive Server environment variables are set correctly.

**See also**

- *Environment Variables* on page 2

## Using the stty Setting

Set **stty tostop** to stop a background Adaptive Server from writing output to the terminal.

To stop background output, enter this command before starting Adaptive Server:

```
stty -tostop
```

If you are redirecting all Adaptive Server output to files, you need not change the **stty** setting.

## Restore Correct Permissions

Sybase software files and directories are installed with the correct access permissions. If necessary, you can restore the correct permissions with the script `setperm_all`, located in `$SYBASE/$SYBASE_ASE/install`.

## 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.

## File Descriptors and User Connections for HP-UX

Find or change the current file descriptor values.

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

To find the current file descriptor values, use:

```
ulimit -n
```

## File Descriptors and User Connections for AIX

Find or change the current file descriptor values.

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 65534 file descriptors, regardless of the value of **open_max**. See the AIX operating system documentation.

To obtain the current **open_max** value, use:

```
ulimit -n
```

## File Descriptors and User Connections for Linux

Find or change the current file descriptor values.

The number of file descriptors per process is limited to 10,000. You can set the number of file descriptors using **ulimit**. Adaptive Server can use a maximum of 40000 file descriptors.

## File Descriptors and User Connections for Sun Solaris

Find or change the current file descriptor values.

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 for opening files and devices.

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

## Displaying Current Soft and Hard Limits

You can display the current soft and hard limits for C and Bourne shells.
To display the current soft or hard limits:

- For soft C shells, enter:

  ```
  limit descriptors
  ```

- For soft Bourne shells, enter:

  ```
  ulimit -n
  ```

- For hard C shells, enter:

  ```
  limit -h descriptors
  ```

- For hard Bourne shells, enter:

  ```
  ulimit -Hn
  ```

## Increasing the Soft Limit

Increase the soft limits for C and Bourne shells.

- To increase the soft limit for C shells, enter:

  ```
  limit descriptors n
  ```

- To increase the soft limit 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 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, so be sure to use the Bourne shell versions of these commands in the `RUN_server_name` file.

## Increasing the Hard Limit

Set up a program to increase the hard limit.

1. Create `file_name.c`, (where `file_name` is the name you give the file.
2. Enter the text shown below, making the appropriate modifications for your situation. .

   **Note:** This is an sample script; modify it as necessary.

   ```
   #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();
     setuid(uid);
   /* run the program indicated as arguments to
     this program */
     execv(*++argv, argv);
 }
```

3. Compile the file:

   ```
   cc file_name.c -o program_name
   ```

   where *program_name* is the compiled file name.
4. Change the program's permissions and ownership:

   ```
   chmod 755 program_name
   chown root program_name
   ```

5. The root user can start Adaptive Server with increased user connections by entering this command at the operating system prompt.

   ```
   # program_name dataserver -d master_device_name
   ```

   where *master_device_name* is the full path of the Adaptive Server master device.Alternatively, you can add *program_name* preceding the **dataserver** command line in the Adaptive Server `RUN_server_name` file.

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

## Enable Asynchronous Disk I/O

Enable asynchronous disk I/O on Linux, IBM, and HP-UX.

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

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 message similar to:

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

## Enabling Asynchronous Disk I/O on Linux

Enable asynchronous I/O on Linux.

### Prerequisites

Before starting Adaptive Server, verify there are sufficient system resources for kernel asynchronous I/O.

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**.

### Task

1. To determine the values for `aio-max-nr` and `aio-nr`, use:

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

2. Modify the number of reservable descriptors by setting `fs.aio-max-nr` with the **sysctl** operating system command. See your operating system documentation for information.

   Adaptive Server may issue messages similar to these in the error log if there are insufficient number of I/O descriptors during start-up, 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.

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

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

## Enabling Asynchronous Disk I/O on IBM AIX

On IBM AIX, 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.**

## Enabling Asynchronous Disk I/O HP-UX

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.

1. Shut down Adaptive Server.
2. From the SAM Kernel Configuration menu, choose **Drivers** and set the pending state for *asyncdisk* to In by adding the driver. Alternatively, add the *asyncdisk* subsystem keyword to */stand/system*.
3. Rebuild the kernel, and restart the system.
4. Using the `userid` root, execute:

   ```
   #/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 who is restarting Adaptive Server. This user must be the owner of the `/dev/async` directory.
   - *<ugrp>* is the user group for the *<uid>* user ID.

## Enabling Asynchronous I/O for File System Devices

Enable asynchronous I/O for file system devices on HP-UX.

### Prerequisites

Sybase recommends that you adjust these operating system parameters to their maximum values:

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

### Task

1. Enable the **allow sql server async i/o** configuration parameter.
2. Enable the **enable hp posix async i/o** configuration parameter. This parameter is static.

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

3. Restart Adaptive Server.
   See *Setting Configuration Parameters* in the *System Administration Guide, Volume 1*.

## Enabling Asynchronous I/O for File Systems

Enable asynchronous I/O for file systems on HP-UX.

### Prerequisites

Sybase recommends that you 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.

### Task

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.

# Adjust the Client Connection Timeout Period

Adaptive Server uses the **KEEPALIVE** option of the TCP/IP protocol to detect inactive clients.

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 appropriate command for your platform.

## Adjusting the Client Connection Timeout for HP-UX

Display or change the current timeout period.

1. 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.
2. Use the **nettune -s** or **ndd -set** commands to change the timeout period.

## Adjusting the Client Connection Timeout for IBM RS/6000

Display, and if necessary, adjust the current timeout period.
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 you use a value of at least 15 minutes.

### Adjusting the Client Connection Timeout for Linux

Display or change the current timeout period.

1. To display or change the timeout value, enter:

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

2. If necessary, adjust the timeout perod.

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

   ```
   /sbin/sysctl -w net.ipv4tcp_keepalive_time=900
   ```

## Check for Hardware Errors

Check for hardware errors on a regular basis.

These 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 you see any of the types of hardware errors described above, 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.

# Monitor the Use of Operating System Resources

Monitor operating system performance.

*Managing Multiprocessor Servers* in the *System Administration Guide: Volume 2* discusses the optimal number of Adaptive Server engines for your workload and system configuration.

## Monitor Operating System Resources for HP-UX

Use HP-UX tools to help monitor performance.

- **sar** – reports relative and absolute I/O throughput rates to each disk and controller.
- **vmstat** – monitors virtual memory usage.
- **netstat** – monitors network status.
- **ps** – provides a snapshot of accumulated CPU time and usage for individual processes.
- **time** – 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.

## Monitor Operating System Resources for IBM RS/6000

Use IBM RS/6000 tools for monitoring performance.

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

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

## Monitor Operating System Resources for Sun Solaris and Linux

Use Sun Solaris and Linux tools to help monitor performance.

- **iostat** – reports the amount of I/O on terminals and hard disks and how CPU time is spent.
- **vmstat** – monitors virtual memory usage.
- **netstat** – monitors network status.

- **ps** – 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.
- **time** – 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.

## Check Database Integrity

Run **dbcc** checks and perform database backups to protect 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* and *Checking Database Consistency* in the *System Administration Guide, Volume 2*, and the *Reference Manual: Commands*.

You can also run this sample C shell script, which calls several **isql** scripts:

```
#!/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][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 **dbcc** output. The results of the **grep** command are sent to 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, which executes a C shell script called dbcc_ck daily at 2:00 a.m.:

```
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 10 Configuring New Servers with srvbuild

Use **srvbuild** to configure a new server with default or user-specified values for key configuration attributes.

**Note:** Use **srvbuildres** in non-GUI mode, to configure a new server using a resource file. See *Utility Commands Reference* in the *Utility Guide*.

1. Source `SYBASE.csh` or `SYBASE.sh` file in `$SYBASE` to set up the environment variables.
2. Run `$SYBASE/$SYBASE_ASE/bin/srvbuild`.
3. Click the server types to configure. When you select each type, the server name is enabled where you can enter the server name. Click **OK**.

   The available server types depend on what you have installed in `$SYBASE`.

   The screen you see next depends on the server types you selected.
4. 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
   - 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 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

  The default configuration 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, 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 and later installerThe installer from versions 15.0.3 and later allows you to tune basic configuration settings during installation, instead of as a postinstallation task. See the *Installation Guide*.

5. 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

6. In the XP Server type screen, select:
   - Transport type
   - Host name
   - Port number
7. To configure Job Scheduler specify:
   - `sybmgmtdb` device path
   - `sybmgmtdb` device size
   - `sybmgmtdb` database size
   - Transport type
   - Host name
   - Port number
8. In Configure Self Management, enter:
   - Enable Self Management
   - Self Management user name
   - Self Management user password

   If you change the Self Management user password after configuring Self Management, execute:

   ```
   sp_addexternlogin loopback, <Self Management user name>, <Self
   Management user name>, <new Self Management user password>
   ```
9. Click **Build Servers!** to proceed with configuration. **srvbuild** displays the configuration status as it builds the servers you selected. After the server configuration completes, click **Exit** to quit **srvbuild**.

# CHAPTER 11 Adaptive Server Default Configuration

Installing or updating Adaptive Server automatically sets some default parameters and settings, and includes several 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 *Setting Configuration Parameters* in the *System Administration Guide, Volume 1*.

For information about configuring languages, character sets, and sort orders, as well as optional features, see *Configuring Character Sets, Sort Orders, and Languages* in the *System Administration Guide, Volume 1*.

## Default Settings

You may need to configure the default settings to suit your computer and database needs.

**Table 3. Defaults for Adaptive Server Parameter Settings**

| Item | Default Value |
|---|---|
| Name | *Servername* |
| Transport Type | TCP/IP |
| Port number | 5000 |
| Error log path | *$SYBASE/$SYBASE_ASE/install/servername.log* |
| Event logging | Not configured |
| Language | us_english |
| Character set | HP – Roman8<br>IBM – ISO 8859-1<br>Sun – ISO 8859-1<br>Linux – iso_1 |
| Sort order | Linux – binary ordering |
| Login security mode | Standard |

**Table 4. Defaults for Backup and XP Servers**

| Server | Item | Default Value |
|---|---|---|
| Backup Server | Name | *AdaptiveServername*_back |
| | Network support | (TCP/IP) |
| | Socket number | 5001 |
| | Error log path | `$SYBASE/$SYBASE_ASE/install/Adaptive-Servername_back.log` |
| XP Server | Name | $SYBASE/$SYBASE_ASE/ADAPTIVESERVERNAME_XP |
| | Network support | (TCP/IP) |
| | Socket number | 5002 |
| | Error log path | N/A |

# CHAPTER 12 Set 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 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.

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.

Servers also need network information. When a server starts, 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.

## How Adaptive Server Determines Which Directory Service Entry to Use

Adaptive Server uses directory services to determine the address at which to listen for clients.

When you start Adaptive Server, it:

1. Looks for the server name supplied in the command line **-s** option. If the server name is not supplied in the command line, it determines its own name by checking the value of the DSLISTEN environment variable. If the DSLISTEN environment variable is not set, Adaptive Server assumes that the server name is SYBASE.
2. Looks in directory services for an entry that matches the name found in the steps above.
3. Uses the network information provided by the directory services entry it has found to listen for client connections.

## How a Client Uses Directory Services

Clients use directory services to determine how to connect to a server.

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 that 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.

See the *Open/Client Programmer's Supplement* for your client platform or the appropriate Open/Client documentation for a more detailed discussion about client connections.

## Create a Directory Services Entry

Use Sybase utilities to edit the network information in directory services.

The installation program, **srvbuild**, automatically creates a directory services entry for each server installation. Use these 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

Sybase supports several directory drivers: interfaces, Lightweight Directory Services (LDAP), and Cell Directory Services (CDS).

- interfaces driver
- LDAP driver
- CDS provided by Distributed Computing Environment (DCE)

See the *Open Client/Server Configuration Guide* for your platform.

## Contents of the 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 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.

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

- Master lines – used by server applications to listen for queries over the network. This information is called a *listener service*.
- Query lines – 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 Your Site has Multiple Installations*

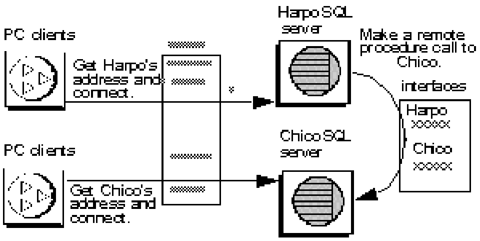
If you have more than one Adaptive Server installation, each server's interfaces file should contain information about all servers on the network.

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.

### *Heterogeneous and Homogeneous Environments*

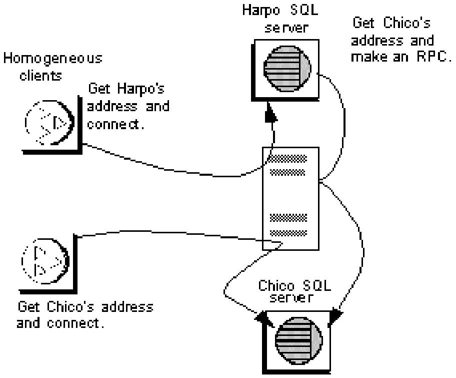
If the platforms are different, each platform may require a different format and configuration for its `interfaces` file. For example, a PC client uses network information in its interfaces file (`sql.ini`) to connect to Adaptive Server running under UNIX, and how Adaptive Server uses its `interfaces` file to connect to another server during a remote procedure call.

**Figure 2: Establishing Network Connections in a Heterogeneous Environment**


Harpo SQL server
PC clients
Make a remote procedure call to Chico.
Get Harpo's address and connect.
interfaces
Harpo xxxxx
Chico SQL server
Chico xxxxx
PC clients
Get Chico's address and connect.

If both a client and a server are running under UNIX, the same interfaces file is valid for both. For example, 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.

**Figure 3: Establishing Network Connections in a Homogeneous Environment**


Harpo SQL server
Get Chico's address and make an RPC.
Homogeneous clients
Get Harpo's address and connect.
Chico SQL server
Get Chico's address and connect.

## interfaces File Format

The format for the `interfaces` file supports the TLI and TCP style entry.

These rules apply:

- 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.

**Note:** Adaptive Server does not support the TLI interface in threaded mode.

A TLI-style entry uses this format:

```
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 uses 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

The `interfaces` file consists of entries that define server attributes.

| Component | Value |
|---|---|
| *servername* | Name of Adaptive Server or Backup Server. Requirements for a server name are:<br>• The name cannot be more than 30 characters long.<br>• The first character must be a letter (ASCII a through z, A through Z).<br>• The characters that follow must be letters, numbers, or underscores (_). |
| *retry_attempts* (optional) | Number of times you want the client to try to connect to a server after initial failure. Default is 0. |
| *delay_interval* (optional) | Number of seconds between connection attempts. Default is 0. |

| Component | Value |
|---|---|
| *service_type* | Type of service defined by entry. Must be either:<br>- master<br>- query |
| *api* | Application programming interface available to the network. The only supported value is "tli". |
| *protocol* | The only available protocol type is:<br>- TCP/IP, represented by the letters “tcp”. |
| *network* | Name of the network; not currently used by Adaptive Server. **srvbuild** enters “ether” as a placeholder. |
| *host* | Network name or address of server’s host machine.<br>- For TCP/IP, use either the host name or Internet address. Maximum size of entry is 32 bytes.<br>To determine the host name of a machine, log in to that machine and enter:<br>`/bin/hostname` |
| *machine* | Network name or address of server’s host machine.<br>You can use either the host name or Internet address. The maximum size of entry is 32 bytes.<br>To determine the host name of a machine, log in to that machine and enter:<br>`/bin/hostname` |
| *device* | The network device endpoint.<br>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. Common TCP streams devices is `/dev/tcp`. |

| Component | Value |
|---|---|
| *address* for a TLI protocol entry | Address consists of:<br>• Address prefix, “\x” for TLI.<br>• Network type, always 0002.<br>• 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.<br>• IP network node address of the host machine converted to 8 digits, hexadecimal.<br>• Trailing zeros, optional, 16 digits. |
| *port* | 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. |
| *ha_failover* | An entry created in directory services or the interfaces file for high availability. |
| *filter* | Adaptive Server supports Secure Sockets Layer (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. |

## Create a Master interfaces File

A master `interfaces` file contains entries for all Sybase servers on the network.

The master `interfaces` file 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 using either:

- **dsedit** or **dscp**, or,
- 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.

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

## Using a Text Editor to Create a Master interfaces File

Use an ASCII text editor to modify a concatenated `interfaces` file.

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

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

   These 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. Find any duplicate SYBASE entries in your merged file, and rename them.
   - A combination of the host machine’s network name or address and the Adaptive Server 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.

# Configure the interfaces File 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

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.
   This 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.

```
# PRODUCTION server with two network listeners
 PRODUCTION<tab>3<tab>3<newline>
 <tab>master tcp ether SERV_CORPNET 4559
 <tab>master tcp ether SERV_ENGNET 5479
 <tab>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.

## Configure 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 options:

- 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.

### Manage One Network-Independent DSQUERY Name

You can maintain one DSQUERY name for all clients, on all networks, and alter each network's interfaces file accordingly.

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 this on the “engineering” network:

```
PRODUCTION<tab>3<tab>3<newline>
<tab>query tcp ether SERV_ENGNET 5470
<tab>master tcp ether SERV_CORPNET 4559
<tab>master tcp ether SERV_ENGNET 5479
```

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

```
PRODUCTION<tab>3<tab>3<newline>
<tab>query tcp ether SERV_CORPNET 4559
<tab>master tcp ether SERV_CORPNET 4559
<tab>master tcp ether SERV_ENGNET 5479
```

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 uses 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

You can 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_network 1 and one for PRODUCTION_network2. For more information, see the Open Client documentation for your client platform.
   - For UNIX clients, you can use an ASCII text editor to edit the `interfaces` files. 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

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.

1. Install multiple “master” and “query” lines in a server entry in the `interfaces` file.
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.

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

   ```
   # PRODUCTION server with two network listeners
   PRODUCTION<tab>3<tab>3<newline>
   <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 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  query tcp sun-ether fd77:55d:
459d9:169:250:56ff:feb3:4246 19560  master tcp sun-ether
fd77:55d:59d9:169:250:56ff:feb3:4246 19560
```

IPv6 application types:

- 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 Sybase-recommended infrastructure implementation for using Adaptive Server as an IPv6-aware server.

Sybase applications are IPv6-aware.

**Table 5. IPv6 Support**

| Platform | Adaptive Server IPv6 Awareness | Open Client/Server IPv6 Awareness |
|---|---|---|
| Sun Solaris 8 32- and 64- bit | 12.5.3a and 15.0 | 15.7 |
| HP-UX 11i(v1) 32- and 64-bit | 12.5.3a and 15.0 | 15.7 |
| Microsoft Server 2003 | 12.5.3a and 15.0 | 15.7 |
| Linux RHEL 3.0 | 15.0 | 15.7 |

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, 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
- 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

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

- Java support
- License Management Server
- LDAP driver

By default, Adaptive Server is IPv6-unaware.

Before starting Adaptive Server for IPv6-aware operations, make sure 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.

# CHAPTER 13 Troubleshoot

Troubleshoot common situations that may cause a server to not start.

## Troubleshooting when the Server Fails to Start

If the server fails to start, check that the port number specified in the interfaces file is available.

When a server fails to start, and issues this message, the port number specified in the interfaces file may be in use:

```
00:00000:00002:2003/09/22 12:37:23.63 kernel  network name
SERV_CORPNET, type ether, port 4559, filter NONE
00:00000:00002:2003/09/22 12:37:23.65 kernel  ninit: bind, Address
already in use
00:00000:00002:2003/09/22 12:37:23.68 server  Error: 1602, Severity:
18, State: 2
00:00000:00002:2003/09/22 12:37:23.68 server  Unable to initialize
network 0
00:00000:00002:2003/09/22 12:37:23.68 kernel  ninit: All master
network
listeners have failed.  Shutting down.
00:00000:00002:2003/09/22 12:37:23.68 kernel  ueshutdown: exiting
00:00000:00016:2003/09/22 16:11:35.46 server  SQL Server shutdown by
request.
```

1. Investigate the port assignment:
   a) Look in the interfaces file to identify the port number assigned to the server.
   b) 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.
   c) 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.

   See the installation documentation for your platform and the *Adaptive Server Utility Guide*.
2. If a stale server process is retaining use of the port number, either:
   - Use the operating system **kill** command to terminate the process, or,
   - Use another port number for the server by modifying the interfaces file.
3. Start the server manually to confirm that the port number is available.

   See the installation documentation for your platform and the *Utility Guide*.

## Troubleshooting an Error when Executing an ESP

If you receive an error when executing an ESP, check for processes that might be using the same port number.

### Prerequisites

Use the **netstat** command to determine if the port number specified for XP Server is in use.

### Task

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

```
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.
```

1. If you find no processes using the same port number:
   a) Restart Adaptive Server.
   b) Execute the ESP that you attempted earlier.
   XP Server should start automatically.
2. If you find a process using the same port number, you can:
   a) Change the interfaces file to use a new port number for the XP Server.
   b) Stop the process using the port number allotted to XP Server.
   c) Restart Adaptive Server, and execute the ESP that you attempted earlier. XP Server should start automatically.

# CHAPTER 14 Use the Lightweight Directory Access Protocol as a Directory Service

*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.

Adaptive Server uses directory services to establish client and RPC connections over the Internet. LDAP directory services are used to establish connections. LDAP is a licensed feature of Adaptive Server.

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

You can Configure the LDAP server 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, 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 the *Security Administration Guide*.

### See also

- *Chapter 3, Client/Server Communication* on page 11

## LDAP Directory Services Versus the Sybase Interfaces File

The LDAP driver implements directory services for use with an LDAP server.

LDAP directories 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

| interfaces File | Directory Services |
|---|---|
| Platform-specific | Platform-independent |
| Specific to each Sybase installation | Centralized and hierarchical |
| Contains separate master and query entries | One entry for each server that is accessed by both clients and servers |
| Cannot store metadata about the server | Stores metadata about the server |

LDAP directory services support more attributes than the Sybase interfaces file. These attributes can include server version, server status, and so on.

**Note:** LDAP is supported only with reentrant libraries. Use **isql_r** instead of **isql**.

**Table 6. Sybase LDAP Directory Definitions**

| Attribute name | Value type | Description |
|---|---|---|
| ditbase | `interfaces` file or `libtcl.cfg` | DIT base for object tree. If the `libtcl.cfg` file is specified, the `interfaces` file is ignored. You can use **ct_con_prop**() to override the `libtcl.cfg` file for a specified connection. |

| Attribute name | Value type | Description |
|---|---|---|
| dn | Character string | Distinguished name. Must be a unique name that identifies the object. |
| sybaseVersion | Integer | Server version number. |
| sybaseServername | Character string | Server name. |
| sybaseService | Character string | Service type: Sybase Adaptive Server |
| sybaseStatus | Integer | Status: 1 = Active, 2 = Stopped, 3 = Failed, 4 = Unknown. |
| sybaseAddress | String | Each server address includes:<br>• Protocol: TCP, NAMEPIPE, SPX DECNET (entry is case sensitive).<br>• Address: any valid address for the protocol type.<br>`dscp` splits this attribute into Transport type and Transport address. |
| sybaseSecurity (optional) | String | Security OID (object ID). |
| sybaseRetryCount | Integer | 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. |
| sybaseRetryDelay | Integer | 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. |
| sybaseHAservername (optional) | String | A secondary server for failover protection. |

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
sybaseHAServernam: 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 call the *directory information tree* (DIT). Client applications use a DIT base to specify where entities are stored.

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.

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

```
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**.

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

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.

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
```

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

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

To enable password authentication at connection time, you can specify a user name and password in the `libtcl.cfg` file as extensions to the LDAP URL.

## Enabling LDAP Directory Services

Enable LDAP directory service.

1. Install the “Security and directory services” license package ASE_SECDIR.
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.

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

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

   libtcl64.cfg
   ldap=libsybdldap64.so ldap://host:port/ditbase??scope??
   bindname=username
   password
   ```

   For example:

```
[DIRECTORY]
ldap=libsybdldap.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.

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**.

## Keywords for the LDAP URL Variables

Configure the `libtcl.cfg` file `DIRECTORY` section.

| Keyword | Description | Default |
|---|---|---|
| *host* (required) | The host name or IP address of the machine running the LDAP server | None |
| *port* | The port number that the LDAP server is listening on | 389 |
| *ditbase* (required) | The default DIT base | None |
| *username* | Distinguished name (DN) of the user to authenticate | NULL (anonymous authentication) |
| *password* | Password of the user to be authenticated | NULL (anonymous authentication) |

## Adding a Server to the Directory Services

Use **dsedit** to add a new server entry.

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 Sybase LDAP directory definitions for accepted values.

Use **srvbuild** to add entries, but not modify or delete them.

**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.

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 – 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.
   - Enter the host name.
   - Enter the port number.
8. Click **OK** twice to exit **dsedit**.
   To view the server entries, enter this URL in Netscape

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

   For example:

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

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

   For more information about `dscp`, see the *Open Client/Server Configuration Guide*.

### See also

- *Migrating from the interfaces File to LDAP* on page 77
- *Chapter 7, Start and Stop Servers* on page 23

## Multiple Directory Services

You can specify multiple directory services for high availability failover protection.

Not every directory service in the list must be an LDAP server. Any type of LDAP service, whether it is an actual server or a gateway to other LDAP services, is called an LDAP server.

For example:

```
[DIRECTORY]
```

```
ldap=libsybdladp.so ldap://test:389/dc=sybase,dc=com
ldap=libsybdldap.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.

**Note:** For more information, see the *Open Client Client-Library/C Programmer Guide* and the *Open Client Client-Library/C Reference Manual*

## 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 `$SYBASE/$SYBASE_OCS/bin`.

1. From the `$SYBASE/$SYBASE_OCS` directory, enter:

   ```
   bin/pwdcrypt
   ```

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

   ```
   0x01312a775ab9d5c71f99f05f7712d2cded2i8d0ae1ce78868d0e8669313d1bc
   4c706
   ```

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

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

4. Replace the password with the encrypted string:

   ```
   ldap=libsybdldap.so
   ldap://ldaphost/dc=sybase,dc=com??one??
   bindname=uid=Manager,dc=sybase,dc=com?
   0x01312a775ab9d5c71f99f05f7712d2cded2i8d0ae1ce78868d0e8669313d1bc
   4c706
   ```

   **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, any performance degradation is seen at login, 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

Once you have upgraded Adaptive Server, configure the server to use LDAP service.

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

1. Shut down the server.
2. Edit the `$SYBASE/$SYBASE_OCS/config/libtcl.cfg` or `libtcl64.cfg` file to add the directory service.
3. Use **dsedit** and add the server entry to directory service.
4. Restart your server.

### See also

- *Adding a Server to the Directory Services* on page 74
- *Chapter 7, Start and Stop Servers* on page 23

# CHAPTER 15 Localization Support

Adaptive Server supports localization for international customers and for customers with heterogeneous environments.

Localization is 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.

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
- US English system message files

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

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
  - 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
- Polish
- Spanish

## 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

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

### *Change 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, you are asked whether to localize your Adaptive Server to a language other than us_english and whether to use a character set or sort order other than the default.

The default Adaptive Server uses:

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

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

By default, all character sets are copied from the distribution media.

You can:

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

  You see **sqlloc** menu, which 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.

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.

### *Change 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 these 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 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

Adaptive Server supports many languages, scripts and character sets.

### Arabic Character Sets

Adaptive Server supports Arabic character sets.

- X – requires Unilib® conversion.
- No X – may use either the Unilib conversion or the built-in conversion.

| Character Set | Unilib Required | Description |
|---|---|---|
| cp864 | X | PC Arabic |
| cp1256 | X | Microsoft Windows Arabic |
| iso88596 | X | ISO 8859-6 Latin/Arabic |

## Baltic Character Set

Adaptive Server supports the Baltic character set.

- X – requires Unilib® conversion.
- No X – may use either the Unilib conversion or the built-in conversion.

| Character Set | Unilib Required | Description |
|---|---|---|
| cp1257 | X | Microsoft Windows Baltic |

## Simplified Chinese Character Sets

Adaptive Server supports Simplified Chinese character sets.

- X – requires Unilib® conversion.
- No X – may use either the Unilib conversion or the built-in conversion.

| Character Set | Unilib Required | Description |
|---|---|---|
| eucgb | X | EUC GB encoding = Simplified Chinese character sets |
| cp936 | X | Microsoft Simplified Chinese character sets |
| gb18030 | X | RC 18030 standard |

## Traditional Chinese Character Set

Adaptive Server supports Traditional Chinese character sets.

- X – requires Unilib® conversion.
- No X – may use either the Unilib conversion or the built-in conversion.

| Character Set | Unilib Required | Description |
|---|---|---|
| cp950 | X | PC (Microsoft) Traditional Chinese |
| euccns | X | EUC CNS encoding = Traditional Chinese with extensions |
| big5 | X | Big 5 Traditional Chinese |

| Character Set | Unilib Required | Description |
|---|---|---|
| big5hk | X | Big 5 with HKSCS extensions |

## Cyrillic Character Sets

Adaptive Server supports Cyrillic character sets.

- X – requires Unilib® conversion.
- No X – may use either the Unilib conversion or the built-in conversion.

| Character Set | Unilib Required | Description |
|---|---|---|
| cp855 | | IBM PC Cyrillic |
| cp866 | | PC Russian |
| cp1251 | | Microsoft Windows 3.1 Cyrillic |
| iso88595 | | ISO 8859-5 Latin/Cyrillic |
| koi8 | | KOI-8 Cyrillic |
| mac_cyr | | Macintosh Cyrillic |
| kz1048 | | Kazakhstan Cyrillic |

## Eastern European Character Sets

Adaptive Server supports Eastern European character sets.

- X – requires Unilib® conversion.
- No X – may use either the Unilib conversion or the built-in conversion.

| Character Set | Unilib Required | Description |
|---|---|---|
| cp852 | | PC Eastern Europe |
| cp1250 | | Microsoft Windows 3.1 Eastern European |
| iso88592 | | ISO 8859-2 Latin-2 |
| mac_ee | | Macintosh Eastern European |

## Greek Character Sets

Adaptive Server supports Greek character sets.

- X – requires Unilib® conversion.
- No X – may use either the Unilib conversion or the built-in conversion.

| Character Set | Unilib Required | Description |
|---|---|---|
| cp869 | | IBM PC Greek |
| cp1253 | | MS Windows Greek |
| greek8 | | HP GREEK8 |
| iso88597 | | ISO 8859-7 Latin/Greek |
| macgrk2 | | Macintosh Greek |

## Hebrew Character Sets

Adaptive Server supports Hebrew character sets.

- X – requires Unilib® conversion.
- No X – may use either the Unilib conversion or the built-in conversion.

| Character Set | Unilib Required | Description |
|---|---|---|
| cp1255 | X | Microsoft Windows Hebrew |
| iso88598 | X | ISO 8859-8 Hebrew |

## Japanese Character Sets

Adaptive Server supports Japanese character sets.

- X – requires Unilib® conversion.
- No X – may use either the Unilib conversion or the built-in conversion.

| Character Set | Unilib Required | Description |
|---|---|---|
| cp932 | X | IBM J-DBCS:CP897 + CP301 (Shift-JIS) |
| deckanji | | Digital UNIX JIS encoding |
| eucjis | | EUC-JIS encoding |
| sjis | | Shift-JIS (no extensions) |

## Korean Character Set

Adaptive Server supports the Korean character set.

- X – requires Unilib® conversion.
- No X – may use either the Unilib conversion or the built-in conversion.

| Character Set | Unilib Required | Description |
|---|---|---|
| eucksc | X | EUC KSC Korean encoding = CP949 |

## Thai Character Sets

Adaptive Server supports Thai character sets.

- X – requires Unilib® conversion.
- No X – may use either the Unilib conversion or the built-in conversion.

| Character Set | Unilib Required | Description |
|---|---|---|
| tis620 | X | TIS-620 Thai standard |
| cp874 | X | Microsoft Windows Thai |

## Turkish Character Sets

Adaptive Server supports Turkish character sets.

- X – requires Unilib® conversion.
- No X – may use either the Unilib conversion or the built-in conversion.

| Character Set | Unilib Required | Description |
|---|---|---|
| cp857 | | IBM PC Turkish |
| cp1254 | | Microsoft Windows Turkish |
| iso88599 | | ISO 8859-9 Latin-5 Turkish |
| macturk | | Macintosh Turkish |
| turkish8 | | HP TURKISH8 |

## Unicode Character Set

Adaptive Server supports the Unicode character set (which supports over 650 languages).

- X – requires Unilib® conversion.
- No X – may use either the Unilib conversion or the built-in conversion.

| Character Set | Unilib Required | Description |
|---|---|---|
| utf8 | X | Unicode UTF-8 encoding |

## Vietnamese Character Set

Adaptive Server supports Vietnamese character sets.

- X – requires Unilib® conversion.
- No X – may use either the Unilib conversion or the built-in conversion.

| Character Set | Unilib Required | Description |
|---|---|---|
| cp1258 | X | Microsoft Windows Vietnamese |

## Western European Character Sets

Adaptive Server supports Western European character sets.

- X – requires Unilib® conversion.
- No X – may use either the Unilib conversion or the built-in conversion.

| Character Set | Unilib Required | Description |
|---|---|---|
| ascii8 | X | US ASCII, with 8-bit data, ISO 646 |
| cp437 | | IBM CP437 – US code set |
| cp850 | | IBM CP850 – European code set |
| cp860 | X | PC Portuguese |
| cp863 | X | IBM PC Canadian French code page |
| cp1252 | X | Microsoft Windows US (ANSI) |
| iso_1 | | ISO 8859-1 Latin-1 |
| mac | | Standard Macintosh coding |
| roman8 | | HP ROMAN8 |
| iso 885915 | X | ISO 8859-15 Latin-1 with Euro support |

# Character Set Names

Supported character sets and their Sybase name.

| Character Sets | Sybase Name |
|---|---|
| ASCII 8 | acsii_8 |
| Big 5 | big5 |
| Big 5HK | big5hk |
| CP 437 | cp437 |
| CP 850 | cp850 |
| CP 852 | cp852 |
| CP 855 | cp855 |

| Character Sets | Sybase Name |
|---|---|
| CP 857 | cp857 |
| CP 858 | cp858 |
| CP 860 | cp860 |
| CP 864 | cp864 |
| CP 866 | cp866 |
| CP 869 | cp869 |
| CP 874 | cp874 |
| CP 932 | cp932 |
| CP 936 | cp936 |
| CP 950 | cp950 |
| CP 1250 | cp1250 |
| CP 1251 | cp1251 |
| CP 1252 | cp1252 |
| CP 1253 | cp1253 |
| CP 1254 | cp1254 |
| CP 1255 | cp1255 |
| CP 1256 | cp1256 |
| CP 1257 | cp1257 |
| CP 1258 | cp1258 |
| DEC Kanji | deckanji |
| EUC-CNS | euccns |
| EUC-GB | eucgb |
| EUC-JIS | eucjis |
| EUC-KSC | eucksc |
| GB 18030 | gb18030 |
| GREEK8 | greek8 |
| ISO 8859-1 | iso_1 |
| ISO 8859-2 | iso88592 |

| Character Sets | Sybase Name |
|---|---|
| ISO 8859-5 | iso88595 |
| ISO 8859-6 | iso88596 |
| ISO 8859-7 | iso88597 |
| ISO 8859-8 | iso88598 |
| ISO 8859-9 | iso88599 |
| ISO 8859-15 | iso15 |
| Kazakhstan Cyrillic | kz1048 |
| Koi8 | koi8 |
| ROMAN8 | roman8 |
| ROMAN9 | roman9 |
| Shift-JIS | sjis |
| TIS 620 | tis620 |
| TURKISH8 | turkish8 |
| UTF-8 | utf8 |

## Load Character Sets

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 during 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. Keywords and Options for Charsets**

| Keywords and Options | Description |
|---|---|
| **-U** | If you are not already logged in to your operating system as “sa”, you must specify “-Usa” or “/username = sa” in the command line. |
| **-P** | Specifies the “sa” password on the command line. If not specified, the user is prompted for the “sa” password. |
| **-S** | 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 “SYBASE.” |
| **-I** | Specifies the interfaces file to use. If not specified, **charset** uses the interfaces file in the SYBASE directory. |
| **-v** | Prints the Sybase version, then exits. Use with no other options specified. |
| *sort_order* | 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. |
| *charset* | Specifies the directory of the character set to be used by Adaptive Server. |

## Language Definition Files Sort Order

Supported sort orders.

Languages not list listed do not have a language-specific sort order; use a binary sort order.

**Table 8. Available Sort Orders**

| Language or Script | Sort Orders | File Name | ID |
|---|---|---|---|
| All languages | Binary order | `binary.srt` | 50 |
| Cyrillic | Dictionary order, case sensitive, accent sensitive | `cyrdict.srt` | 63 |

| Language or Script | Sort Orders | File Name | ID |
|---|---|---|---|
| | Dictionary order, case sensitive, accent sensitive | `cyrnocs.srt` | 64 |
| English<br>French<br>German<br>These sort orders work with all Western European character sets. | Dictionary order, case sensitive, accent sensitive | dictiona.srt | 51 |
| | Dictionary order, case insensitive, accent sensitive | `nocase.srt` | 52 |
| | Dictionary order, case insensitive, accent sensitive, with preference | `nocasepr.srt` | 53 |
| | Dictionary order, case insensitive, accent insensitive | `noaccent.srt` | 54 |
| English<br>French<br>German<br>These sort orders work only with CP 850. | Alternate dictionary order, case sensitive | `altdict.srt` | 45 |
| | Alternate dictionary order, case sensitive, accent insensitive | `altnoacc.srt` | 39 |
| | Alternate dictionary order, case sensitive, with preference | `altnocsp.srt` | 46 |
| Greek<br>This sort order works only with ISO 8859-7. | Dictionary order, case sensitive, accent sensitive | `elldict.srt` | 65 |
| Hungarian<br>These sort orders work only with ISO 8859-2. | Dictionary order, case sensitive, accent sensitive | `hundict.srt` | 69 |
| | Dictionary order, case insensitive, accent sensitive | `hunnoac.srt` | 70 |
| | Dictionary order, case insensitive, accent insensitive | `hunnocs.srt` | 71 |

| Language or Script | Sort Orders | File Name | ID |
|---|---|---|---|
| Russian<br>This sort order works with all Cyrillic character sets except CP 855. | Dictionary order, case sensitive, accent sensitive | `rusdict.srt` | 58 |
| | Dictionary order, case insensitive, accent sensitive | `rusnocs.srt` | 59 |
| Scandinavian<br>These sort orders work only with CP 850. | Dictionary order, case sensitive, accent sensitive | `scandict.srt` | 47 |
| | Dictionary order, case insensitive, with preference | `scannocp.srt` | 48 |
| Spanish | Dictionary order, case sensitive, accent sensitive | `espdict.srt` | 55 |
| | Dictionary order, case insensitive, accent sensitive | `espnocs.srt` | 56 |
| | Dictionary order, case insensitive, accent insensitive | `espnoac.srt` | 57 |
| Thai | Dictionary order | `dictionary.srt` | 51 |
| Turkish<br>These sort orders work only with ISO 8859-9. | Dictionary order, case sensitive, accent sensitive | `turdict.srt` | 72 |
| | Dictionary order, case insensitive, accent insensitive | `turnoac.srt` | 73 |
| | Dictionary order, case insensitive, accent sensitive | `turnocs.srt` | 74 |

# Character Set Conversion

Backup Server passes messages to Adaptive Server in the client's language and in the Adaptive Server character set. Adaptive Server converts the messages and issues them in the client's language and character set.

Keep these 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, 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. See the *System Administration Guide: Volume 1*.

If either Adaptive Server or Backup Server does not support a client's language or character set, the server issues a warning message. Errors also occur when the Backup Server character set is incompatible 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 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 US 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. The server can support only one sort order at a time, so select one that works 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 for Character Sets

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`

You can specify sort orders during installation, or later, using the **sqlloc** utility.

| Sort Order Name | Description |
|---|---|
| Binary order | 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.<br>All character sets have binary order as the default. If binary order does not meet your needs, specify one of the other sort orders during installation, or by, using the **sqlloc** utility. |

| Sort Order Name | Description |
|---|---|
| Dictionary order, case sensitive, accent 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. |
| Dictionary order, case insensitive, accent sensitive | Uppercase letters are equivalent to their lowercase counterparts and are intermingled in sorting results. |
| Dictionary order, case insensitive, accent insensitive | Diacritical marks are ignored. |
| Dictionary order, case insensitive with preference | A word written with uppercase letters is equivalent to the same word written with lowercase letters.<br>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.<br>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. |
| Alternate dictionary order, case sensitive | Use with several of the Western European languages. |
| Alternate dictionary order, case insensitive, accent insensitive | Use with several of the Western European languages. |
| Alternate dictionary order, case insensitive, uppercase preference | Use with several of the Western European languages. |
| Spanish dictionary order, case sensitive | Use with Spanish and for most Latin American locales. |
| Spanish dictionary order, case insensitive | Use with Spanish and for most Latin American locales. |
| Spanish dictionary order case insensitive, accent insensitive | Use with Spanish and for most Latin American locales. |
| Scandinavian dictionary order, case sensitive | Case-sensitive dictionary sort order.<br>Use with Scandinavian languages. |

| Sort Order Name | Description |
|---|---|
| Scandinavian dictionary order, case insensitive, uppercase preference | Use with Scandinavian languages. |

Use **sqlloc** to display the sort orders for the character sets you plan to use. See *Configuring Character Sets, Sort Orders*, and *Languages* in the *System Administration Guide: Volume 1*.

# 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.

By default, Adaptive Server and Backup Server installs English system messages in English are installed by defaut.

## Installing a New Language Module

A full installation of Adaptive Server installs all the language components automatically. If you did not select a full installation, manually install additional language modules as required.

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.

## Message Languages

By default, US English is installed as the language for messages in Adaptive Server.

These rules apply to language modules:

- During Adaptive Server installation or reconfiguration, you can specify a default language other than US English. Make sure you have also installed the language module for the language you specify.
- If your clients require Adaptive Server messages in a language other than US English, you must load the language module for those languages. You can then 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.

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

# Localization Directories

Sybase localization configuration involves the `locales` and `charsets` directories.

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

| | | | |
|---|---|---|---|
| `%SYBASE%/` or `$SYBASE/` | `charsets` | `charset_name` | `*.srt` files |
| | | `charset_name...` | `charset.loc` |
| | | `unicode` | `*.uct` files |
| | `locales` | `language_name` | `charset_name` |
| | | `language_name...` | `charset_name...` |
| | | `locales.dat` | |
| | | `message` | `language_name` |
| | | | `language_name...` |

## About the locales Directory

The `$SYBASE/locales` directory contains a subdirectory for each available language. Each language subdirectory contains a subdirectory for each available character set for 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 any available sort order.

## Format of locales.dat File Entries

You can edit the `locales.dat` file to change the default language or character set for a platform, or to add new associations between platform locale names and Sybase language and character set names.

Each entry in the `locales.dat` file links a platform-specific locale definition to a Sybase language and character set combination. Each entry has this 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 `$SYBASE/locales/language_name`.
- *syb_charset* is the character set that determines the character set conversion method and identifies the directory location of the message files for clients from within `$SYBASE/locales/language_name/charset_name`.

For example, this 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.

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

   ```
   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.

# Changing Adaptive Server Localization Configuration

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

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. See the *System Administration Guide*.

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. 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.

### See also

- *Chapter 4, About Changing Adaptive Server Configuration* on page 13

## Configuring Adaptive Server for Other Character Sets

Configure Adaptive Server with the character set and sort order for your language.

### Prerequisites

The server must be running and you must have system administrator privileges.

### Task

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

   ```
   $SYBASE/$SYBASE_ASE/bin/charset -Usa -Ppassword
   -Sserver_name sort_order_file character_set
   ```

   Replace `sort_order_file` with the name of the sort order file. Replace `character_set` with the Sybase name for your character set.
2. Use **charset** utility to load any additional character sets.

   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. Configure your server for the new character set and sort order:

   ```
   1> sp_configure "default sortorder_id",
   2> sort_order_id, "character_set"
   3> go
   ```

   Replace *sort_order_id* with the ID for your sort order. Replace *character_set* with the Sybase name for your character set.
5. Shut down the server to start the reconfiguration process.
6. Use your normal process on your UNIX system to restarts 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. Your system messages appear in the default language, English.

## Editing the locales.dat File

Use a text editor to make edits to the `locales.dat` file.

### Prerequisites

Make a copy of the original file, in case you have problems with the resulting edited version.

### Task

1. Using a text editor such as Notepad, open the `locales.dat` file copy.
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 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 do not run properly.

   For example, for Open Client messages to appear in French, when Adaptive Server is using the Roman8 character set, check the `locales.dat` entries for your platform and look for:

   ```
   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 Backup Server Localization Configuration

Localize Backup Server by modifying the `RUN_<backup_server_name>` file.

See the *Utility Guide* for more information on the **backupserver** command arguments.

# CHAPTER 16 Adaptive Server Error Logging

Upon start-up, Adaptive Server, it writes information to a local error log file, called the Adaptive Server error log.

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.

The error log file,`$SYBASE/$SYBASE_ASE/install/server_name.log`:

- 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 start-up messages from Adaptive Server

If the error log become too large, you can:

- Use **sp_errorlog** to dynamically change its path. Once the older error log is not being used by Adaptive Server, you can move it, and make space available.
- Stop the Adaptive Server and delete logged messages.

See *Diagnosing System Problems* in the *System Administraton Guide* for a description of the error log format.

## Set 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.

## Set the Adaptive Server Error Log Path

Change the error log path by editing the RUN_*server_name* file.

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

```
$SYBASE/ASE-15_0/bin/dataserver -d/Devices/ASE_2K.dat -sASE_2K -i/
ASE_150 -e$SYBASE/ASE_2K.log -M/ASE_150
```

For information about using the `RUN_server_name` file, see the *Utility Guide*.

### See also

- *Chapter 7, Start and Stop Servers* on page 23

# Manage Messages

Manage event logging by controlling whether a message is logged, or by adding user messages.

When event logging is enabled, you can:

- 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.

  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.

## Log User-Defined Messages

You can write a user-defined messages to the Adaptive Server error log.

Adaptive Server lets you manage:

- 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.

See **sp_addmessage** and **sp_altermessage** in the *Reference Manual: Procedures*.

## Log Auditing Events

Use **sp_configure** parameters to specify whether Adaptive Server writes auditing events, such as logins, to the Adaptive Server error log.

By default, Adaptive Server does not log auditing events. To log events, you can use **sp_configure** with these possible parameters and values:

- **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
  ```

See the *System Administration Guide: Volume 1*.

# CHAPTER 17 Manage Database Devices

*database device* is a disk or a portion of a disk that stores Adaptive Server databases and database objects.

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.

## Device Requirements

Adaptive Server devices and databases are subject to maximum and minimum values.

- The maximum device size is 4TB.
- The minimum usable device size depends on the server's 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.
  - 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 8TB.
  - The maximum database size on a 4K page server is 16TB.
  - The maximum database size on an 8K page server is 32TB.
  - The maximum database size on a 16K page server is 64TB.
- 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.

## Create Files for Database Devices

You can create new database devices using **disk init**.

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, 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**.

An example of creating a database device using **disk init**:

```
disk init name = "user_device1",
physname = "/work/data/device1.dat",
size = 2048
```

`"size = 2048"` allocates 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.You cannot specify a device number that is 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.

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 18 Database Management System Auditing

Security-related system activity is recorded in an audit trail, which you can use to detect system penetration and resource abuse.

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.

**See also**

- *sybsecurity Device and Database* on page 7
- *Install Auditing* on page 108
- *Determine the Size of a Database Device* on page 9

## Audit System Devices and Databases

The main components of the audit system are the `sybsecurity` device and the `sybsecurity` database.

- The `sybsecurity` device and the `sybsecurity` database store audit information.
- The audit trail, consists of several audit devices and tables that you specify during configuration.
- The syslogs transaction log device stores transaction logs.

### *The sybsecurity device and database*

The sybsecurity device stores the `sybsecurity` database, which 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 tracking 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 audit table 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 the current audit table.

When you configure Adaptive Server for auditing, determine the number of audit tables for your installation. 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.

# Install Auditing

Use the **auditinit** utility or **installsecurity** script to install auditing.

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

- Use the **installsecurity** script. See the *Security Administration Guide*.
- Use the **auditinit** utility.

### See also

- *sybsecurity Device and Database* on page 7
- *Chapter 18, Database Management System Auditing* on page 107
- *Determine the Size of a Database Device* on page 9

## Preinstallation Recommendations for Auditing Devices

Complete the preinstallation recommendations before installing auditing.

- Determine the location of the devices for the `sybsecurity`, `syslogs`, and `sysaudits` table devices. You will need to provide this information later.
- Configure your system with the minimum number of auditing devices you require – you must configure at least three devices. You can use **sp_addaudittable** to add more auditing devices later. See the *Reference Manual: Procedures*.
- Install auditing tables and devices in a one-to-one ratio. Tables that share the same device also 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. 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. You can later expand the table size toward 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.

### See also

- *The directio Parameter* on page 8
- *The dsync Parameter* on page 8

## Configuring Adaptive Server for Auditing

Use the **auditinit** command to configure Adaptive Server for auditing.

1. Source `SYBASE.csh` or `SYBASE.sh` file if you have not set up the Sybase environment variables.
2. At the UNIX prompt, enter:

```
$SYBASE/$SYBASE_ASE/install/auditinit
```

**auditinit** displays:

```
AUDITINIT
1. Release directory:  /usr/u/sybase
2. Configure a Server product
```

3. Select **Configure a Server Product**.
4. Select **Adaptive Server**.
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 datbase transaction log:
```

```
Logical name    Physical name    Segment name    Table
name    Size
```

9. From the Configure Auditing screen, select **Configure Auditing**.
10. Restart Adaptive Server for the changes to take effect.

## Creating a Device for the Audit Database Transaction Log

Add a device for the audit 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**.

   **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:
   - 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. Use only a subset of the device, edit the Size of the Device value.
6. Press Ctrl+A to accept the settings that appear 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/
aud1.dat’ sysaudits_01   5
7. Audit_02'   /secret1/sybase_  dr/install/
aud2.dat' sysaudits_02   5
8. auditlog    /secret1/.../
auditlog.dat logsegment   syslogs        2
```

7. When you are ready to execute the audit configuration, press Ctrl+A.
8. On the Sybase Server Configuration screen, press Ctrl+A again. You see:

   ```
   Execute the Sybase Server Configuration now?
   ```

9. Enter “y” (yes).

   **auditinit** executes the tasks to install auditing. When installation completes successfully, you see:

   ```
   Running task: install auditing capabilities.
   ....................Done
   Auditing capability installed.
   Task succeeded: install auditing capabilities.
   Configuration completed successfully.
   Press <return> to continue.
   ```

## Enabling Auditing

Use **sp_configure** to enable auditing.

### Prerequisites

You must have SSO permission to enable auditing.

### Task

To enable auditing, enter:

```
sp_configure 'auditing', 1
```

See the *Security Administration Guide*.

## Deleting a Device Entry

Delete an auditing device.

1. From the Configure Auditing menu, select **Delete a Device Entry**.
2. Enter the number of the device to delete.
3. Press **Return**.

## Changing a Device Entry

Change a device name or size for auditing.

1. From the Configure Auditing menu, select **Change a Device Entry** .
2. Enter the number of the device to change.

   The Add/Change a New Device for Auditing menu with information about 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.

# CHAPTER 19 Install Online Help for Transact-SQL Syntax

The `$SYBASE/$SYBASE_ASE/scripts` directory contains scripts for installing the syntax help database, `sybsyntax`.

You can retrieve this data using **sp_syntax**. See the *Reference Manual: Procedures*.

The `scripts` directory contains one or more of the **sp_syntax** scripts shown in, depending on which Sybase products are included with your server:

**Table 9. sp_syntax Installation Scripts**

| Script | Product |
|---|---|
| **ins_syn_cl** | Open Client Client-Library™ |
| **ins_syn_esql** | Embedded SQL™ |
| **ins_syn_os** | Open Server |
| **ins_syn_sql** | Transact-SQL |

All Adaptive Server installations include the **ins_syn_sql** script, which includes syntax information for Transact-SQL, the system procedures, and the Sybase utilities. Execute this script to 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 necessary tables and indexes. Any scripts that you execute after the first one add to the existing information in the database. If you reexecute a script, 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

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 recommend that you do not use 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, either:

- Use **sp_diskdefault** to specify a default device other than the master device. For information about **sp_diskdefault**.
- Modify each `sybsyntax` installation script that you plan to execute to specify a different device.

See the *Reference Manual: Procedures*.

## Installing sybsyntax

The `sybsyntax` installation script installs the database and the necessary tables and indexes.

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.
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.

   Comment out this section, which specifies the default device:

```
/* 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
 end
```

4. 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.

5. Execute the script:

```
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 *servername*, you can replace the server name with $DSQUERY.

6. 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**:

```
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."

# Index