ADMINISTRATION GUIDE | PUBLIC
SAP Adaptive Server Enterprise 16.0 SP03
Document Version: 1.0 – 2020-03-04

System Administration Guide: Volume 1

THE BEST RUN SAP

Content

1 Overview of System Administration

The system administrator is responsible for setting up and maintaining SAP ASE.

SAP ASE administration tasks include:

- Installing SAP ASE and Backup Server
- Creating and managing SAP ASE login accounts
- Granting roles and permissions to SAP ASE users
- Managing and monitoring the use of disk space, memory, and connections
- Backing up and restoring databases
- Diagnosing system problems
- Configuring SAP ASE to achieve the best performance

In addition, system administrators may assist with certain database design tasks that overlap with the work of application designers, such as enforcing integrity standards.

Although a system administrator generally concentrates on tasks that are independent of the applications running on SAP ASE, he or she is likely to have the best overview of all applications. For this reason, a system administrator can advise application designers about existing data, make recommendations about standardizing data definitions across applications, and so on.

However, the distinction between what is specific to an application is sometimes unclear. Owners of user databases might consult certain sections of this book. Similarly, system administrators and database owners will use the *Transact-SQL Users Guide* (especially information about data definitions, stored procedures, and triggers). Both system administrators and application designers will use the *Performance and Tuning Series*.

1.1 Roles Required for System Administration Tasks

Many SAP ASE commands and procedures require the system administrator or system security officer role. Other system administration information is relevant to database owners.

Various security-related, administrative, and operational tasks are grouped into the following user roles

- system administrator – by default, the system administrator (sa) is assigned these roles:
  - `sa_role`
  - `sso_role`
  - `oper_role`
  - `sybase_ts_role`

  The system administrator's tasks that are related to SAP ASE include:
  - Managing disk storage
  - Monitoring the automatic recovery procedure
  - Fine-tuning by changing configurable system parameters
  - Diagnosing and reporting system problems

    - Backing up and loading databases
    - Modifying and dropping server login accounts
    - Granting and revoking the system administrator role
    - Granting permissions
    - Creating user databases and granting ownership of them
    - Setting up groups, which can be used for granting and revoking permissions
- System security officer – performs security-related tasks, such as:
    - Creating server login accounts, which includes assigning initial passwords
    - Changing the password of any account
    - Granting and revoking the system security officer and operator roles
    - Creating, granting, and revoking user-defined roles
    - Granting the capability to impersonate another user throughout the server
    - Setting the password expiration interval
    - Setting up network-based security services
    - Managing the audit system
- Operator – backs up and loads databases on a server-wide basis. The operator role allows a single user to use the `dump database`, `dump transaction`, `load database`, and `load transaction` commands to back up and restore all databases on a server without having to be the owner of each one. These operations can be performed for an individual database by the database owner or by a system administrator. However, an operator can perform them for any database.

These roles provide individual accountability for users who are performing operational and administrative tasks. Their actions can be audited and attributed to them. A system administrator operates outside the discretionary access control (DAC) protection system; that is, when a system administrator accesses objects, SAP ASE does not check the DAC permissions.

In addition, two kinds of object owners have special status because of the objects they own:

- Database owner
- Database object owner

1.1.1 Database Owner

The database owner is the creator of a database or someone to whom database ownership has been transferred. A system administrator can use the `grant` command to grant users the authority to create databases.

A database owner logs in to SAP ASE using his or her assigned login name and password, and has the "dbo" account. When this user logs in to databases they did not create, this user is known by his or her regular user name.

A database owner can:

- Run the system procedure `sp_adduser` to allow other SAP ASE users access to the database
- Use the `grant` command to give other users permission to create objects and execute commands within the database

Adding users to databases is discussed in *Security Administration Guide > Managing SAP ASE Logins and Database Users*. Granting permissions to users is discussed in *Security Administration Guide > Managing User Permissions*.

The database owner does not automatically receive permissions on objects owned by other users. However, a database owner can temporarily assume the permissions of other users in the database at any time by using the `setuser` command. Using a combination of the `setuser` and `grant` commands, the database owner can acquire permissions on any object in the database.

> **i** Note
>
> Because the database owner role is so powerful, the system administrator should carefully plan is allowed to own databases in the server. The system security officer should consider auditing the database activity of all database owners.

1.1.2 Database Object Owner

A database object owner is a user who creates a database object.

Database objects include tables, indexes, views, defaults, triggers, rules, constraints, and procedures. Before a user can create a database object, the database owner must grant the user permission to create objects of a particular type. There is no special login name or password for a database object owner.

The database object owner creates an object using the appropriate `create` statement, and then grants permission to other users.

The creator of a database object is automatically granted all permissions on that object. The system administrator also has all permissions on the object. The owner of an object must explicitly grant permissions to other users before they can access the object. Even the database owner cannot use an object directly unless the object owner grants him or her the appropriate permission. However, the database owner can always use the `setuser` command to impersonate any other user in the database, including the object owner.

> **i** Note
>
> When a database object is owned by someone other than the database owner, the user (including a system administrator) must qualify the name of that object with the object owner's name—`<ownername.objectname>`—to access the object. If an object or a procedure must be accessed by a large number of users, particularly in ad hoc queries, having these objects owned by "dbo" greatly simplifies access.

1.2 Performing System Administration Tasks

You can enter SAP ASE commands using either `isql` or SAP Adaptive Server Enterprise cockpit (SAP ASE cockpit). SAP ASE cockpit is a graphical administration tool for on-board management and monitoring of SAP ASE.

For complete information about using `isql`, see the *Utility Guide*. For complete information about using SAP ASE cockpit, see the *SAP Adaptive Server Enterprise Cockpit* documentation.

.

You can use isql to enter many Transact-SQL.

To start isql on most platforms, enter this command at an operating system prompt, where `<username>` is the system administrator:

```
isql -Uusername
```

SAP ASE prompts you for your password. To prevent other users from seeing your password, do not use the `isql -P` option.

The statements that you enter in `isql` can span several lines. Statements are not processed until you enter **go** on a separate line.

For example:

```
1> select *
2> from sysobjects
3> where type = "TR"
4> go
```

The examples used throughout the SAP ASE documentation do not include the **go** command between statements. If you are typing the examples, you must enter the **go** command to see the sample output.

You can save the Transact-SQL statements you use to create or modify user databases and database objects.

The easiest way to do this is to create or copy the statements to an ASCII-formatted file. You can then use the file to supply statements to isql to re-create databases or database objects later.

The syntax for using isql with an ASCII-formatted file is the following, where filename is the full path and file name of the file that contains Transact-SQL statements:

```
isql -U<username> -i<filename>
```

On UNIX and other platforms, use the "less than" symbol (<) to redirect the file.

The Transact-SQL statements in the ASCII file must use valid syntax and the go command.

When reading commands from a file, you must:

- Supply the `-P<password>` option at the command line, or,
- Include the named user's password on the first line of the input file

.

1.3 System Tables

The `master` database contains system tables that keep track of information about SAP ASE. In addition, each database (including the `master` database) contains system tables that keep track of information that is specific to that database.

All the SAP ASE-supplied tables in the `master` database (the SAP ASE controlling database) are considered system tables. Each user database is created with a subset of these system tables. The system tables may also be called the data dictionary or the system catalogs.

A `master` database and its tables are automatically created when SAP ASE is installed. The system tables in a user database are created when the `create database` command is issued. The names of all system tables start with "sys". You cannot create tables in user databases that have the same names as system tables. See *Reference Manual: Tables* for detailed descriptions of system tables and their columns.

1.3.1 Querying the System Tables

Query system tables in the same manner as any other tables.

For example, the following statement returns the names of all triggers in the database:

```
select name
from sysobjects
where type = "TR"
```

In addition, SAP ASE supplies stored procedures (called system procedures), many of which provide shortcuts for querying the system tables.

These system procedures provide information from the system tables:

• `sp_commonkey`	• `sp_helpremotelogin`
• `sp_configure`	• `sp_help_resource_limit`
• `sp_countmedatada`	• `sp_helprotect`
• `sp_dboption`	• `sp_helpsegment`
• `sp_estspace`	• `sp_helpserver`
• `sp_help`	• `sp_helpsort`
• `sp_helpartition`	• `sp_helptext`
• `sp_helpcache`	• `sp_helpthreshold`
• `sp_helpconfig`	• `sp_helpuser`
• `sp_helpconstraint`	• `sp_lock`
• `sp_helpdb`	• `sp_monitor`
• `sp_helpdevice`	• `sp_monitorconfig`
• `sp_helpgroup`	• `sp_showcontrolinfo`
• `sp_helpindex`	• `sp_showexeclass`
• `sp_helpjava`	• `sp_showplan`

• sp_helpjoins	• sp_spaceused
• sp_helpkey	• sp_who
• sp_helplanguage	• sp_help_resource_limit
• sp_helplog	• sp_sp
• sp_p	• sp_w

For complete information about the system procedures, see the *Reference Manual: Procedures*.

1.3.2 Keys in System Tables

Primary, foreign, and common keys for system tables are defined in the `master` and `model` databases. You can generate a report on defined keys by executing `sp_helpkey`. For a report on columns in two system tables that are likely join candidates, execute `sp_helpjoins`.

1.3.3 Updating System Tables

The SAP ASE system tables contain information that is critical to the operation of your databases. Under ordinary circumstances, you need not perform direct data modifications to system tables.

Update system tables only when you are instructed to do so by SAP Technical Support.

Before you update system tables, you must issue an `sp_configure` command that enables system table updates. While this command is in effect, any user with appropriate permission can modify a system table. Other requirements for direct changes to system tables are:

- Modify system tables only inside a transaction. Issue a `begin transaction` command before you issue the data modification command.
- Verify that only the rows you wanted changed have been affected by the command, and that the data has been changed correctly.
- If the command was incorrect, issue a `rollback transaction` command. If the command was correct, issue a `commit transaction` command.

> ⚠ Caution
>
> Some system tables should not be altered by any user under any circumstances: these tables are dynamically built by system processes, contain encoded information, or display only a portion of their data when queried. Imprudent, ad hoc updates can prevent SAP ASE from running, make database objects inaccessible, scramble permissions on objects, or terminate a user session. Never attempt to alter system table definitions; for example, do not alter them to include constraints: triggers, defaults, and rules are not allowed in system tables.

1.4 System Procedures

The names of all system procedures begin with "sp_". They are located in the `sybsystemprocs` database, but you can run many of them in any database by issuing the stored procedure from the database or by qualifying the procedure name with the database name.

SAP-supplied system procedures (such as `sp_who`) are created using the `installmaster` installation script. Use `sp_version` to determine the version of `installmaster` that was most recently executed. See the *Reference Manual: System Procedures* for more information about `sp_version`.

If you execute a system procedure in a database other than `sybsystemprocs`, the procedure operates on the system tables in the database from which it was executed. For example, if the database owner of `pubs2` runs `sp_adduser` from `pubs2` or issues the command `pubs2..sp_adduser`, the new user is added to `pubs2..sysusers`. However, this does not apply to system procedures that update only tables in the master database.

Permissions on system procedures are discussed in the *Reference Manual: Procedures*.

1.4.1 Using System Procedures

A parameter is an argument to a stored or system procedure.

If a parameter value for a system procedure contains reserved words, punctuation, or embedded blanks, you must enclose it in single or double quotes. If the parameter is an object name, and the object name is qualified by a database name or owner name, enclose the entire name in single or double quotes.

System procedures can be invoked during a session using either chained or unchained transaction mode. Chained mode implicitly begins a transaction before any data retrieval or modification statement. Unchained mode requires explicit `begin transaction` statements paired with `commit transaction` or rollback transaction statements to complete the transaction. See *Transact-SQL Users Guide* > *Transactions: Maintaining Data Consistency and Recovery*.

You cannot execute the system procedures that modify data in system tables in the `master` database from within a transaction, since this may compromise recovery. You cannot run system procedures that create temporary worktables from transactions.

If no transaction is active when you execute a system procedure, SAP ASE turns off chained mode and sets `transaction isolation level 1` for the duration of the procedure. Before returning, the session's chained mode and isolation level are reset to their original settings. See *Transact-SQL Users Guide* > *Transactions: Maintaining Data Consistency and Recovery*.

All system procedures report a return status. For example, the following means that the procedure executed successfully:

```
return status = 0
```

If the system procedures do not execute successfully, the return status is a number other than 0.

1.4.2 System Procedure Tables

System procedures use several system procedure tables in the `master` and `sybsystemdb` databases to convert internal system values (for example, status bits) into human-readable format.

One of these tables, `spt_values`, is used by a variety of system procedures, including:

- `sp_configure`
- `sp_dboption`
- `sp_depends`
- `sp_help`
- `sp_helpdb`
- `sp_helpdevice`
- `sp_helpindex`
- `sp_helpkey`
- `sp_helprotect`
- `sp_lock`

The `spt_values` table can be updated only by an upgrade; you cannot modify it. To see how it is used, execute `sp_helptext` and look at the text for one of the system procedures that references it.

The other system procedure tables are `spt_monitor`, `spt_committab`, and tables needed by the catalog stored procedures. (The `spt_committab` table is located in the `sybsystemdb` database.)

In addition, several system procedures create, and then drop, temporary tables. For example, `sp_helpdb` creates `#spdbdesc`, `sp_helpdevice` creates `#spdevtab`, and `sp_helpindex` creates `#spindtab`.

1.4.3 Creating Stored Procedures

System administrators can write stored procedures that can be executed in any database.

Context

The *System Administration Guides* include discussions about some system procedures that are relevant to system administration. Many system procedures are explained in this manual, in the sections where they are relevant. For detailed reference information about all procedures, see the *Reference Manual: Procedures*.

Create a stored procedure in `sybsystemprocs` and assign it a name that begins with "`sp_`". The `uid` of the stored procedure must be 1, the `uid` of the database owner.

Most system procedures that you create query the system tables. SAP recommends that you do not create stored procedures that modify the system tables.

Procedure

1. Use `sp_configure` to set `allow updates to system tables` on.
2. Use `create procedure` to create the stored procedure.
3. Use `sp_configure` to set `allow updates to system tables` off.

> ⚠ Caution
>
> Use caution when you modify system tables. Always test the procedures that modify system tables in development or test databases, rather than in your production database.

1.5 System Extended Stored Procedures

An extended stored procedure (ESP) lets you call external language functions from within SAP ASE.

SAP ASE includes a set or predefined ESPs; users can also create their own. The names of all system extended stored procedures begin with "xp_", and are located in the `sybsystemprocs` database.

One very useful system ESP is `xp_cmdshell`, which executes an operating system command on the system that is running SAP ASE.

Invoke a system ESP just like a system procedure. The difference is that a system ESP executes procedural language code rather than Transact-SQL statements. All ESPs are implemented by an Open Server™ application called XP Server™, which runs on the same machine as SAP ASE. XP Server starts automatically on the first ESP invocation.

For information about the system ESPs provided with SAP ASE, see the *Reference Manual: Procedures*.

1.5.1 Creating System ESPs

Use `create procedure` to create a system ESP in the `sybsystemprocs` database.

System procedures are automatically included in the `sybsystemprocs` database. The name of the ESP, and its procedural language function, must begin with "xp_". The `uid` of the stored procedure must be 1, the `uid` of the database owner.

For general information about creating ESPs, see *Transact-SQL Users Guide* > *Using Extended Stored Procedures*.

1.6 Logging Error Messages

Each time it starts, SAP ASE writes start-up information to a local error log file. The installation program automatically sets the error log location when you configure a new SAP ASE.

Default location for the error log is:

- (UNIX) `$SYBASE/$SYBASE_ASE/install/server_name.log`
- (Windows) `%SYBASE%\%SYBASE_ASE%\install\errorlog`

Many error messages from SAP ASE are written only to the user's terminal. However, fatal error messages (severity levels 19 and above), kernel error messages, and informational messages from SAP ASE are recorded in the error log file.

SAP ASE keeps the error log file open until you stop the server process. Before deleting old messages to reduce the size of the error log, stop the SAP ASE process.

> **i** Note
>
> On some platforms, such as Windows, SAP ASE also records error messages in the operating system event log. See the installation guide and configuration guide for your platform.

See the *Configuration Guide* for your platform to view the default location and file name of the error log for your platform.

1.7 Connecting to SAP ASE

SAP ASE can communicate with other SAP ASEs, Open Server applications, and client software on the network.

Clients can talk to one or more servers, and servers can communicate with other servers using remote procedure calls. For products to interact with one another, each must know where the others reside on the network. This network service information is stored in the `interfaces` file.

1.7.1 The interfaces File

The `interfaces` file, which lists the name and address of every known server, is usually named `interfaces`, `interface`, or `sql.ini`, depending on the operating system. On UNIX platforms, the interfaces file is located in the `$SYBASE` installation directory. On Windows, the `sql.ini` file is located in the `%SYBASE%\ini` installation directory.

When you use a client program to connect to a server, the program looks up the server name in the `interfaces` file and then connects to the server using the address.

The name, location, and contents of the `interfaces` file differ between operating systems.

Also, the format of the SAP ASE addresses in the `interfaces` file differs between network protocols.

The SAP ASE installation program creates a simple `interfaces` file that you can use for local connections to SAP ASE over one or more network protocols. As a system administrator, modify the `interfaces` file and distribute it to users so that they can connect to SAP ASE over the network. See the *Configuration Guide* for your platform for information about the `interfaces` file.

See *Performance and Tuning Series: Basics > Networks and Performance* for more information about the interfaces file and network listeners.

1.7.2 Directory Services

A directory service manages the creation, modification, and retrieval of network service information. Directory services are provided by platform or third-party vendors and must be purchased and installed separately from SAP ASE. Two examples of directory services are Registry, and Distributed Computing Environment (DCE).

The `$SYBASE/$SYBASE_OCS/config/libtcl.cfg` file is a SAP-supplied configuration file used by servers and clients to determine:

- Which directory service to use, and
- The location of the specified directory service driver.

If no directory services are installed or listed in the `libtcl.cfg` file, SAP ASE defaults to the `interfaces` file for obtaining network service information.

The system administrator must modify the `libtcl.cfg` file as appropriate for the operating environment.

Some directory services are specific to a given platform; others can be used on several different platforms. Because of the platform-specific nature of directory services, see the configuration documentation for your platform for detailed information about configuring for directory services.

1.7.3 LDAP 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.

The LDAP server can be located on a different platform from the one on which SAP ASE or 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 metadata (data about data).

The LDAP server can store and retrieve information about:

- SAP ASE, such as IP address, port number, and network protocol
- Security mechanisms and filters
- High availability companion server name
- Authentication information for user access to SAP ASE

You can authenticate users logging in to SAP ASE through information stored in the `syslogins` directory or through a centralized LDAP server that enables a single login and password throughout the enterprise. See *Security Administration Guide > Managing SAP ASE Logins and Database Users*.

You can configure the LDAP server to use these access restrictions:

- Anonymous authentication – all data is visible to any user.
- User name and password authentication – SAP ASE uses the default user name and password from the appropriate file:
  - UNIX, 32-bit – $SYBASE/$SYBASE_OCS/config/libtcl.cfg
  - UNIX, 64-bit – $SYBASE/$SYBASE_OCS/config/libtcl64.cfg
  - Windows – `%SYBASE%\%SYBASE_OCS%\ini\libtcl.cfg`

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

> **i** Note
>
> The default user name and password stored in `libtcl.cfg` and passed to the LDAP server for authentication purposes are distinct and different from those used to access SAP ASE. The default user name and password allow access to the LDAP server for administrative tasks.

When an LDAP server is specified in the `libtcl.cfg` or `libtcl64.cfg` file (collectively called `libtcl*.cfg` file), the server information is then accessible only from the LDAP server; SAP ASE 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 use the `dataserver` command line option to specify the search order.

1.7.3.1 Multiple Directory Services

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

You can specify multiple directory services for high-availability failover protection in `libtcl*.cfg`. Not every directory service in the list must be an LDAP server.

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

```
[DIRECTORY]
   ldap=libdldap.so ldap://test:389/dc=sap,dc=com
   dce=libddce.so ditbase=/.:/subsys/sap/dataservers
   ldap=libdldap.so ldap://huey:11389/dc=sap,dc=com
```

See the *Open Client Client-Library/C Programmer's Guide* and the *Open Client Client-Library/C Reference Manual*.

1.7.3.2 LDAP Directory Services Versus the SAP Interfaces File

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

The LDAP infrastructure provides:

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

Table 1: Interfaces File Versus LDAP Directory Services

`interfaces` File	Directory Services
Platform-specific	Platform-independent
Specific to each 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

1.7.3.2.1 LDAP 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 SAP ASE is started, changes in performance are seen at login time, if at all. During normal system load, the delay should not be noticeable. During high system load with many connections, especially repeated connections with short duration, the overall performance difference of using an LDAP server versus the traditional `interfaces` file might be noticeable.

1.8 Security Features Available in SAP ASE

SAP ASE includes a series of major security features, including identification and authentication control, discretionary access control, role division, accountability, and data confidentiality.

- Identification and authentication controls – ensures that only authorized users can log in to the system. In addition to password-based login authentication, SAP ASE supports external authentication using Kerberos, LDAP, or pluggable authentication modules (PAM).
- Discretionary access controls (DACs) – provides access controls that let object owners restrict access to objects, usually with the `grant` and `revoke` commands. This type of control is dependent upon an object owner's discretion.
- Division of roles – allows an administrator to grant privileged roles to specified users so only designated users can perform certain tasks. SAP ASE has predefined roles, called "system roles," such as system

administrator and system security officer, and also SAP ASE allows system security officers to define additional roles, called "user-defined roles."

- Accountability – provides the ability to audit events such as logins, logouts, server start operations, remote procedure calls, accesses to database objects, and all actions performed by a specific user or with a particular role active. SAP ASE also provides a single option to audit a set of server-wide, security-relevant events.
- Confidentiality of data – maintains a confidentiality of data using encryption for client/server communication, available with Kerberos or Secure Sockets Layer (SSL). Inactive data is kept confidential with password-protected database backup.

See *Security Administration Guide > Getting Started with Security Administration in SAP ASE*.

2 System and Optional Databases

SAP ASE includes a number of system databases, optional SAP-supplied databases that you can install, and the `sybdiag` database, which SAP Product Support may install for diagnostic purposes.

2.1 Overview of System Databases

A default SAP ASE installation includes a number of system databases.

- The `master` database
- The `model` database
- The system procedure database, `sybsystemprocs`
- The two-phase commit transaction database, `sybsystemdb`
- The temporary database, `tempdb`

Optionally, you can install:

- The auditing database, `sybsecurity`
- The sample databases, `pubs2` and `pubs3`
- The `dbcc` database, `dbccdb`
- The Job Scheduler database, `sybmgmtdb`

For information about installing the `master`, `model`, `sybsystemprocs`, `tempdb`, and `sybmgmtdb` databases, see the installation guide for your platform. For information about installing `dbccdb`, see *System Administration Guide: Volume 2 > Checking Database Consistency*. For information about using Job Scheduler, see the *Job Scheduler Users Guide*.

The `master`, `model`, `sybsystemdb`, and temporary databases reside on the master device which is named during installation. The `master` database must be contained entirely on the master device and cannot be expanded onto any other device. Create all other databases and user objects on other devices.

> ⚠ Caution
>
> Do not store user databases on the master device; doing so makes it difficult to recover both the system databases and any user databases stored on the master device.

Install the `sybsecurity` and `sybmgmtdb` databases on their own devices and segment. See the installation documentation for your platform.

Install the `sybsystemprocs` database on a device of your choice. You may want to modify the installation scripts for `pubs2` and `pubs3` to share the device you create for `sybsystemprocs`.

Use the `installjsdb` script (located in `$SYBASE/ASE-15_0/scripts`) to install the `sybmgmtdb` database. `installjsdb` looks for a device named `sybmgmtdev` on which to create the `sybmgmtdb` database and its accompanying tables and stored procedures. If the `sybmgmtdb` database already exists, `installjsdb` creates

the Job Scheduler tables and stored procedures in the existing database. If `installjsdb` cannot find either a `sybmgmtdev` device or a `sybmgmtdb` database, it creates `sybmgmtdb` on the master device. However, SAP strongly recommends that you remove the `sybmgmtdb` database from the master device.

The `installpubs2` and the `installpubs3` scripts do not specify a device in their `create database` statement, so they are created on the default device. During installation, the master device is the default device. To change this, edit the scripts.

2.2 The master Database

The `master` database controls the operation of SAP ASE and stores information about all user databases and their associated database devices.

Table 2: Information About the master Database Stores

Information	System table
User accounts	`syslogins`
Remote user accounts	`sysremotelogins`
Remote servers that this server can interact with	`sysservers`
Ongoing processes	`sysprocesses`
Configurable environment variables	`sysconfigures`
System error messages	`sysmessages`
Databases on SAP ASE	`sysdatabases`
Storage space allocated to each database	`sysusages`
Tapes and disks mounted on the system	`sysdevices`
Active locks	`syslocks`
Character sets	`syscharsets`
Languages	`syslanguages`
Users who hold server-wide roles	`sysloginroles`
Server roles	`syssrvroles`
SAP ASE engines that are online	`sysengines`

Because the `master` database stores information about user databases and devices, you must be in the `master` database to issue the `create database`, `alter database`, `disk init`, `disk refit`, `disk reinit`, and disk mirroring commands.

The minimum size of your `master` database depends on your server's logical page size. The `master` database must contain at least 6656 logical pages, so its minimum physical size for each logical page size is:

- 2K page – 13MB
- 4K page – 26MB
- 8K page – 52MB
- 16K page – 104MB

2.2.1 Controlling Object Creation in the master Database

When you install SAP ASE, only a system administrator can create objects in the `master` database, because the system administrator implicitly becomes the database owner ("dbo") of any database he or she uses.

Any objects created on the `master` database should be used only for system administration. Set permissions in `master` so that most users cannot create objects there.

> ⚠ Caution
>
> Do not place user objects in `master`. Storing user objects in `master` causes the transaction log to fill quickly. If the transaction log runs out of space completely, you cannot use `dump transaction` commands to free space in `master`.

You may also want to use `alter login` to change the default database for users (the database to which a user is connected when he or she logs in). See *Security Administration Guide > Managing SAP ASE Logins and Database Users*.

Create any system procedures in the `sybsystemprocs` database rather than in `master`.

2.2.2 Backing Up Master and Keeping Copies of System Tables

To be prepared for hardware or software failure on SAP ASE, perform frequent backups of all databases, and keep copies of system tables.

- Perform frequent backups of the `master` database and all user databases. See *System Administration Guide: Volume 2 > Restoring the System Databases*.
- Keep a copy (preferably offline) of these system tables: `sysusages`, `sysdatabases`, `sysdevices`, `sysloginroles`, and `syslogins`. If you have copies of these scripts, and a hard-disk failure or other disaster makes your database unusable, you can use the recovery procedures described in *System Administration Guide: Volume 2 > Restoring the System Databases*. If you do not have current copies of your scripts, it is much more difficult to recover SAP ASE when the `master` database is damaged.

2.3 model Database

SAP ASE includes the `model` database, which provides a template, or prototype, for new user databases.

Each time a user enters the `create database` command, SAP ASE makes a copy of the `model` database and extends the new database to the size specified by the `create database` command.

> **i** Note
>
> New databases must be at least as large as the `model` database.

The `model` database contains the required system tables for each user database. You can modify `model` to customize the structure of newly created databases—everything you do to `model` is reflected in each new database. Some of the changes that system administrators commonly make to `model` are:

- Adding user-defined datatypes, rules, or defaults.
- Adding users who need access to all databases on SAP ASE.
- Granting default privileges, particularly for "guest" accounts.
- Setting database options such as `select into/bulkcopy/pllsort`. These settings are reflected in all new databases. The default settings for these options in `model` is `off`.

Typically, most users do not have permission to modify the `model` database. There is not much point in granting read permission either, since SAP ASE copies its entire contents into each new user database.

The `model` database cannot be larger than `tempdb`. By default, the size of `model` is six allocation units (an allocation unit is 256 logical pages.). You see error message if you increase the size of `model` without making `tempdb` at least as large.

> **i** Note
>
> Keep a backup copy of the `model` database, and back up `model` with `dump database` each time you change it. In case of media failure, restore `model` as you would a user database.

2.4 sybsystemprocs Database

SAP system procedures are stored in the database `sybsystemprocs`.

When a user in any database executes a system stored procedure (that is, a procedure that has a name beginning with `sp_`), SAP ASE first looks for that procedure in the user's current database. If there is no procedure there with that name, SAP ASE looks for it in `sybsystemprocs`. If there is no procedure in `sybsystemprocs`, SAP ASE looks for the procedure in `master`.

If the procedure modifies system tables (for example, `sp_adduser` modifies the `sysusers` table), the changes are made in the database from which the procedure was executed.

To change the default permissions on system procedures, modify those permissions in `sybsystemprocs`.

> **i** Note
>
> Any time you make changes to `sybsystemprocs`, back up the database.

2.5 Creating Temporary Tables

You can create temporary tables in the `tempdb` database, or create global temporary tables in any database.

Temporary Tables in the tempdb Database

Temporary tables created with a pound sign are accessible only by the current SAP ASE session: users on other sessions cannot access them. These nonsharable, temporary tables are destroyed at the end of each session. The first 13 bytes of the table's name, including the pound sign (#), must be unique. SAP ASE assigns the names of such tables a 17-byte number suffix. (You can see the suffix by querying `tempdb..sysobjects`.)

Temporary tables created with the "tempdb.." prefix are stored in `tempdb` and can be shared among SAP ASE sessions. SAP ASE does not change the names of temporary tables created this way. The table exists either until you restart SAP ASE or until its owner drops it using `drop table`.

System procedures work on temporary tables, but only if you use them from `tempdb`.

If a stored procedure creates temporary tables, the tables are dropped when the procedure exits. You can also explicitly drop temporary tables before a session ends.

> ⚠ Caution
>
> Do not create temporary tables with the "tempdb.." prefix from inside a stored procedure unless you intend to share those tables among other users and sessions.

Each time you restart SAP ASE, it copies `model` to `tempdb`, which clears the database. You cannot recover temporary tables.

No special permissions are required to create temporary tables or to execute commands that may require storage space in the temporary database.

You can create a temporary table either by preceding the table name in a `create table` statement with a pound sign (#), or by specifying the name prefix "tempdb..".

Global Temporary Tables

You can also create a global temporary table that contains persistent metadata across sessions, and private data per session, with data being automatically deleted at the end of the session, even if the session ends abnormally. Global temporary tables can be created in any database.

For more information about syntax, examples, and characteristics, see *Reference Manual: Commands > create global temporary table*.

2.5.1 tempdb Database

SAP ASE has a temporary database, `tempdb`, that provides a storage area for temporary tables and other temporary working storage needs. The space in `tempdb` is shared among all users of all databases on the server.

The default size of `tempdb` depends on the logical page size for your server, 2, 4, 8, or 16K. Certain activities may make it necessary for you to increase the size of `tempdb`:

- Large temporary tables.
- A lot of activity on temporary tables, which fills up the `tempdb` logs.
- Large or many simultaneous sorts. Subqueries and aggregates with `group by` also cause some `tempdb` activity.

Use `alter database` to increase the size of `tempdb`. `tempdb` is initially created on the master device. You can add space to `tempdb` from the master device or from any other database device.

If you run `update index statistics` against large tables, the command fails with error number 1105 if `tempdb` is not large enough.

You can create and manage multiple temporary databases in addition to the system temporary database, `tempdb`. Multiple temporary databases reduce contention on system catalogs and logs in `tempdb`.

2.6 sybsecurity Database

The `sybsecurity` database, which contains the auditing system for SAP ASE, includes these system tables, is installed when you configure SAP ASE for auditing.

`sybsecurity` includes these system tables:

- The `sysaudits_01`, `sysaudits_02`, ... `sysaudits_08` system tables, which contain the audit trail
- The `sysauditoptions` table, which contains rows that describe the global audit options
- All other default system tables that are derived from `model`

See *Security Administration Guide > Auditing*.

2.7 sybsystemdb Database

The `sybsystemdb` database stores information about distributed transactions.

SAP ASE versions 12.0 and later can provide transaction coordination services for transactions that are propagated to remote servers using remote procedure calls (RPCs) or Component Integration System (CIS).

Information about remote servers participating in distributed transactions is stored in the `syscoordinations` table.

The `sybsystemdb` database also stores information about SYB2PC transactions that use the SAP two-phase commit protocol. The `spt_committab` table, which stores information about and tracks the completion status of each two-phase commit transaction, is stored in the `sybsystemdb` database.

See the *Configuration Guide* for your platform for information about two-phase commit transactions and how to create the `sybsystemdb` database.

2.8 sybmgmtdb Database

The `sybmgmtdb` database stores jobs, schedules, scheduled jobs information, and data needed by the internal Job Scheduler task for processing.

`sybmgmtdb` also maintains the output and results from these executed tasks. See the *Job Scheduler Users Guide*.

2.9 pubs2 and pubs3 Sample Databases

The `pubs2` and `pubs3` sample databases are provided as a learning tool for SAP ASE.

The `pubs2` sample database is used for most of the examples in the SAP ASE documentation, except for examples, where noted, that use the `pubs3` database. The sample databases are intended for training purposes only, and installing them is optional. Do not install them in an SAP ASE production environment.

For information about installing `pubs2` and `pubs3`, see the installation guide for your platform. For information about the contents of these sample databases, see the *Transact-SQL Users Guide*.

2.9.1 Maintaining the Sample Databases

The sample databases include a "guest" user login that allows access to the database by any authorized SAP ASE user.

The "guest" login has a wide range of privileges in `pubs2` and `pubs3`, including permissions to select, insert, update, and delete user tables. See *Security Administration Guide > Managing SAP ASE Logins and Database Users*.

The size of the `pubs2` and `pubs3` databases are determined by the size of the logical page size for your server; 2, 4, 8, and 16K. If possible, give each new user a clean copy of `pubs2` and `pubs3` so that she or he is not confused by other users' changes. To place `pubs2` or `pubs3` on a specific database device, edit the installation script before installing the database.

If space is a problem, instruct users to issue the `begin transaction` command before updating a sample database. After the user has finished updating one of the sample databases, he or she can issue the `rollback transaction` command to undo the changes.

2.9.2 pubs2 Image Data

SAP ASE includes a script for installing `image` data in the `pubs2` database (`pubs3` does not use the `image` data).

The `image` data consists of six pictures, two each in PICT, TIF, and Sun raster file formats. SAP does not provide any tools for displaying `image` data. You must use the appropriate screen graphics tools to display the images after you extract them from the database.

See the installation documentation for your platform for information about installing the `image` data in `pubs2`.

2.10 dbccdb Database

`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 *System Administration Guide: Volume 2 > Checking Database Consistency*.

2.11 sybdiag Database

SAP Product Support may create the `sybdiag` database on your system for debugging purposes. This database holds diagnostic configuration data, and should not be used by customers.

2.12 Determining the Version of the Installation Scripts

`sp_version` lets you determine the current version of the scripts (`installmaster`, `installdbccdb`, and so on) installed on SAP ASE, whether they ran successfully or not, and the length of time they took to complete.

The syntax for `sp_version` is:

```
sp_version [<script_file> [, "all"]]
```

where:

- `<script_file>` is the name of the installation script (the default value is NULL).
- `all` reports details about each script, such as the date executed, and the length of time for execution.

If you run `sp_version` without any parameters, it reports on all scripts.

This example describes what installation scripts were run, what time they were run, and what time they finished:

```
sp_version null, 'all'
```

```
Script           Version
Status
-----------      ------------------------------------------------------------
-----------
installmaster    15.0/EBF XXXXX/B/Sun_svr4/OS 5.8/asemain/1/32-bit/OPT/Thu Sep
23 22:12:12 2004
Complete [Started=Sep 24 2004  3:39PM]-[Completed=Sep 24 2004  3:45PM]
```

3 System Administration for Beginners

New system administrators should be aware of a number of concepts like logical page sizes, "test" servers, installation issues, physical resources, backup and recovery, maintenance, troubleshooting, and the importance of keeping accurate records.

3.1 Logical Page Sizes

Database objects are built with logical pages. A database and any of its related objects must use the same logical page size. That is, you cannot create a server that uses more than one logical page size.

SAP ASE allows you to create master devices and databases with logical page sizes of 2K, 4K, 8K, or 16K, but a given server installation can use only one of these four logical page sizes.

All databases in a server—and all objects in every database—use the same logical page size. For example, all the pages on a server with a logical page size of 4K must be 4K, even though you may not use some pages beyond the initial 2K.

Select the page size when you create the master device with `dataserver -z`.

For more information about the `dataserver` command, see the *Utility Guide*. For more information about logical page sizes, see *System Administration Guide: Volume 2 > Configuring Memory*.

3.2 Using “Test” Servers

SAP suggests that you install and use a test or development server, then remove it before you create the production server.

Using a test server makes it easier to plan and test different configurations and less stressful to recover from mistakes. It is much easier to learn how to install and administer new features when there is no risk of having to restart a production server or re-create a production database.

SAP suggests that you use a test server from installation or upgrade through configuration. It is in these initial steps that you make some of the most important decisions about your final production system.

3.2.1 Planning Resources

Using a test server helps you plan the final resource requirements for your system and helps you discover resource deficiencies that you might not have anticipated.

In particular, disk resources can have a dramatic effect on the final design of the production system. For example, you may decide that, in the event of a media failure, a particular database requires nonstop recovery.

This means you must configure one or more additional database devices to mirror the critical database. Discovering these resource requirements in a test server allows you to change the physical layout of databases and tables without affecting database users.

Use a test server to benchmark both SAP ASE and your applications using different hardware configurations. This allows you to determine the optimal setup for physical resources at both the SAP ASE level and the operating system level before bringing the entire system online for general use.

3.2.2 Achieving Performance Goals

Most performance objectives can be met only by carefully planning a database's design and configuration.

For example, you may discover that the insert and I/O performance of a particular table causes a bottleneck. In this case, the best course of action may be to re-create the table on a dedicated segment and partition the table. Changes of this nature are disruptive to a production system; even changing a configuration parameter may require you to restart SAP ASE.

3.3 Considerations When Installing SAP Products

The responsibility for installing SAP ASE and other SAP products is sometimes placed with the system administrator. If installation is one of your responsibilities, use the following pointers to help you in the process. there are several items you should consider, including product compatibility, installation and upgrade issues, third-party software installation, and configuring and testing client connections.

Check Product Compatibility

Before installing new products or upgrading existing products, always read the release bulletin included with the products to understand any compatibility issues that might affect your system.

Compatibility problems can occur between hardware and software, and between different release levels of the same software. Reading the release bulletin in advance can save the time and guesswork of troubleshooting known compatibility problems. Pay particular attention to the lists of known problems that are included in the release bulletin.

Install or Upgrade SAP ASE

Read through the installation guide for your platform before you begin a new installation or upgrade.

You may also want to consult with the operating system administrator to discuss operating system requirements for SAP ASE. These requirements can include the configuration of memory, raw devices, asynchronous I/O, and other features, depending on the platform you use. Many of these tasks must be completed before you begin the installation.

If you are upgrading a server, back up all data (including the `master` database, user databases, triggers, and system procedures) offline before you begin. After upgrading, immediately create a separate, full backup of your data, especially if there are incompatibilities between older dump files and the newer versions.

Install Additional Third-Party Software

SAP ASE generally includes support for the network protocols that are common to your hardware platform. If your network supports additional protocols, install the required protocol support.

As an alternative to the SAP `interfaces` file, you can use a directory service to obtain a server's address and other network information. Directory services are provided by platform or third-party vendors and must be purchased and installed separately from the installation of SAP ASE. See the *Configuration Guide* for your platform for a list of the directory services that SAP ASE currently supports.

Configure and Test Client Connections

A successful client connection depends on the coordination of SAP ASE, the client software, and network products.

If you are using one of the network protocols installed with SAP ASE, see the *Configuration Guide* for your platform for information about testing network connections. If you are using a different network protocol, follow the instructions that are included with the network product. You can also use "ping" utilities that are included with SAP connectivity products to test client connections with SAP ASE. For details about the name and contents of the `interfaces` file, see the *Configuration Guide* for your platform

3.4 Allocating Physical Resources

Allocating physical resources is providing SAP ASE the memory, disk space, worker processes, and CPU power required to achieve your performance and recovery goals.

When installing a new server, every system administrator must make decisions about resource utilization. If you upgrade your platform, or if the design of your database system changes, you must also reallocate SAP ASE resources by adding new memory, disk controllers, or CPUs. Early benchmarking of SAP ASE and your applications can help you identify hardware resource deficiencies that create performance bottlenecks.

See *System Administration Guide: Volume 2 > Overview of Disk Resources* to understand the kinds of disk resources required by SAP ASE. See also *System Administration Guide: Volume 2 > Configuring Memory* and *System Administration Guide: Volume 2 > Managing Multiprocessor Servers* for information about memory and CPU resources.

The following sections provide helpful pointers in determining physical resource requirements.

3.4.1 Dedicated Versus Shared Servers

Part of planning SAP ASE resources means learning about the resources that are required by other applications running on the same machine.

Generally, system administrators dedicate an entire machine for SAP ASE use, which means that only the operating system and network software consume resources that might otherwise be reserved for SAP ASE. On a shared system, other applications, such as SAP ASE client programs or print servers, run on the same machine as SAP ASE. It can be difficult to calculate the resources available to SAP ASE on a shared system, because the types of applications and their pattern of use may change over time.

It is the system administrator's responsibility to take into account the resources used by operating systems, client programs, windowing systems, and so forth when configuring resources for SAP ASE. Configure SAP ASE to use only the resources that are available to it. Otherwise, the server may perform poorly or fail to start.

3.4.2 Decision-Support and OLTP Applications

SAP ASE contains many features that optimize performance for OLTP, decision-support, and mixed-workload environments. However, to make optimal use of these features, determine in advance the requirements of your system's applications.

For mixed-workload systems, list the individual tables that you anticipate will be most heavily used for each type of application; this can help you achieve maximum performance for applications.

3.4.3 Advance Resource Planning

Plan resource usage before you make changes to SAP ASE configuration.

In the case of disk resources, for example, after you initialize and allocate a device to SAP ASE, that device cannot be used for any other purpose (even if SAP ASE never fills the device with data). Likewise, SAP ASE automatically reserves the memory for which it is configured, and this memory cannot be used by any other application.

When planning resource usage:

- For recovery purposes, always place a database's transaction log on a separate physical device from its data. See *System Administration Guide: Volume 2 > Creating and Managing User Databases*.
- Consider mirroring devices that store mission-critical data. See *System Administration Guide: Volume 2 > Mirroring Database Devices*. If your operating system supports these features, consider using disk arrays and disk mirroring for SAP ASE data.
- SAP ASE supports using both raw partitions or certified file systems for devices. However, you may find it simpler to initialize database devices as operating system files instead of using raw devices in a system designed for testing purposes, where you may not be as concerned about the buffering inherent in a system that uses file system devices.
- Changing some configuration parameters can affect the way SAP ASE consumes physical resources, especially memory (for example, `max memory`).

3.4.4 Operating System Configuration

After determining the resources that are available to SAP ASE and the resources you require, configure the physical resources at the operating system level.

- If you are using raw partitions, initialize the raw devices to the sizes required by SAP ASE. If you initialize a raw device for SAP ASE, you cannot use that device for any other purpose (for example, to store operating system files). Ask your operating system administrator for assistance in initializing and configuring raw devices to the required sizes.
- Configure the number of network connections. Make sure that the machine on which SAP ASE runs can actually support the number of connections you configure. See your operating system documentation.
- Additional configuration may be required for your operating system and the applications that you use. Read the installation guide for your platform. Also read your client software documentation or consult with your engineers to understand the operating system requirements for your applications.

3.5 Backup and Recovery

Making regular backups of your databases is crucial to the integrity of your database system. Although SAP ASE automatically recovers from system crashes (for example, power outages) or server failures, only *you* can recover from data loss caused by media failure.

See these topics in the *System Administration Guide: Volume 2* for information about how to develop and implement a backup and recovery plan:

- *Developing a Backup and Recovery Plan*
- *Backing Up and Restoring User Databases*
- *Restoring the System Databases*
- *Managing Free Space with Thresholds*

3.5.1 Keep Up-To-Date Backups of the master Database

Backing up the `master` database is the most crucial element of any backup and recovery plan.

The `master` database contains details about the structure of your entire database system. It stores information about the SAP ASE databases, devices, and device fragments that make up those databases. Because SAP ASE needs this information for recovery, it is crucial that you maintain an up-to-date backup copy of the `master` database at all times.

To ensure that your backup of `master` is always up to date, back up the database after each command or procedure that affects disks, storage, databases, or segments, including:

- Creating or deleting databases
- Initializing new database devices
- Adding new dump devices
- Using any device mirroring command

- Creating or dropping system stored procedures, if they are stored in `master`
- Creating, dropping, or modifying a segment
- Adding new SAP ASE logins

To back up `master` to a tape device, start `isql` and enter the command:

```
dump database master to "<tape_device>"
```

where `<tape_device>` is the name of the tape device (for example, `/dev/rmt0`).

3.5.1.1 Keep Offline Copies of System Tables

In addition to backing up `master` regularly, keep offline copies of the `sysdatabases`, `sysdevices`, `sysusages`, `sysloginroles`, and `syslogins` system tables

Use the `bcp` utility described in the *Utility Guide* and store a printed copy of the contents of each system table. Create a printed copy by printing the output of:

```
select * from sysusages order by vstart
select * from sysdatabases
select * from sysdevices
select * from sysloginroles
select * from syslogins
```

If you have copies of these tables, and a hard-disk failure or other disaster makes your database unusable, you can use the recovery procedures described in *System Administration Guide: Volume 2 > Restoring the System Databases*.

Also keep copies of all data definition language (DDL) scripts for user objects.

3.5.2 Automating Backup Procedures

Automate your backup procedure using an operating system script or a utility (for example, the UNIX `cron` utility) to perform the necessary backup commands.

Context

You can further automate the procedure by using thresholds, which are discussed in *System Administration Guide: Volume 2 > Managing Free Space with Thresholds*.

Although the commands required to create an automated script vary, depending on the operating system you use, all scripts should accomplish the same basic steps:

Procedure

1. Start `isql` and dump the transaction log to a holding area (for example, a temporary file).
2. Rename the dump file to a name that contains the dump date, time, and database name.
3. In a history file, record information about the new backup.
4. In a separate file, record any errors that occurred during the dump.
5. Automatically send mail to the system administrator for any error conditions.

3.5.3 Verify Data Consistency Before Backing Up a Database

Your database backups must be consistent and accurate, especially for `master`. If you back up a database that contains internal errors, the errors persist in a restored version of the database.

Use the `dbcc` commands to check a database for errors before backing it up. Always use `dbcc` commands to verify the integrity of a database before dumping it. If `dbcc` detects errors, correct them before dumping the database.

Over time, if you discover few or no errors while running `dbcc`, you may decide that the risk of database corruption is small and that you need to run `dbcc` only occasionally. If the consequences of losing data are too high, continue to run `dbcc` commands each time you back up a database.

> **i** Note
>
> For performance considerations, many sites choose to run `dbcc` checks outside of peak hours or on separate servers.

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

3.5.4 Monitor the Log Size

When the transaction log becomes nearly full, it may be impossible to use standard procedures to dump transactions and reclaim space.

The system administrator should monitor the log size and perform regular transaction log dumps (in addition to regular database dumps) to avoid this situation. Set up a threshold stored procedure that notifies you (or dumps the log) when the log reaches a certain capacity. See *System Administration Guide: Volume 2 > Managing Free Space with Thresholds*. SAP also suggests that, to shorten the time required to dump and load the database, dump the transaction log immediately prior to performing a full database dump.

You can also monitor the space used in the log segment manually using `sp_helpsegment`, as described in *System Administration Guide: Volume 2 > Creating and Using Segments*.

3.6 Ongoing Maintenance and Troubleshooting

In addition to making regularly scheduled backups, the system administrator generally also performs other maintenance activities throughout the life of a server.

3.6.1 Starting and Stopping SAP ASE

Most system administrators automate the procedure for starting SAP ASE to coincide with the start-up of the server machine. Do this by editing operating system start-up scripts, or by using other operating system procedures.

See the *Configuration Guide* for your platform to determine how to start and stop SAP ASE.

3.6.2 Viewing and Pruning the Error Log

You can use operating system scripts to scan the error log for particular messages and to automatically notify the system administrator when specific errors occur. Checking the error log regularly may help determine whether there are continuing problems of the same nature, or whether a particular database device is likely to fail.

The error log file can grow large over time, since SAP ASE appends informational and status messages to it each time it starts. You can periodically "prune" the log file by opening the file and deleting old records. Keeping the log file to a manageable size saves disk space and makes it easier to locate current errors.

3.7 Keeping Records

Keeping records about your SAP ASE system is an important part of your job as a system administrator.

Accurate records of changes and problems that you have encountered can be a valuable reference when you are contacting SAP Technical Support or recovering databases. They can also provide vital information for administrators who manage the SAP ASE system in your absence.

3.7.1 Contact Information

Maintain a list of contact information for yourself as well as the system security officer, operator, and database owners on your system.

Also, record secondary contacts for each role. Make this information available to all SAP ASE users so that the appropriate contacts receive enhancement requests and problem reports.

3.7.2 Configuration Information

If you use script files to create databases and database objects, configure SAP ASE, and store the scripts in a safe place, you can then re-create your entire system in the event of a disaster.

You can also use script files to quickly re-create database systems for evaluation purposes on new hardware platforms. If you use a third-party tool to perform system administration, remember to generate equivalent scripts after performing administration tasks.

Consider recording the following kinds of information:

- Commands used to create databases and database objects (DDL scripts)
- Commands that add new SAP ASE logins and database users
- The current SAP ASE configuration file.
- The names, locations, and sizes of all files and raw devices initialized as database devices

Maintain a dated log of all changes to the SAP ASE configuration. Mark each change with a brief description of when and why you made the change, as well a summary of the end result.

3.7.3 Maintenance Schedules

SAP recommends that you keep a calendar of regularly scheduled maintenance activities.

List any of the procedures you perform at your site:

- Using `dbcc` to check database consistency
- Backing up user and system databases
- Monitoring the space left in transaction logs (if this is not done automatically)
- Dumping the transaction log
- Examining the error log contents for SAP ASE and Backup Server
- Running the `update statistics` command (see *Performance and Tuning Series: Improving Performance with Statistical Analysis > Using the set statistics Commands*)
- Examining auditing information, if the auditing option is installed
- Recompiling stored procedures
- Monitoring resource utilization of the server machine

3.7.4 System Information

Record information about the hardware and operating system on which you run SAP ASE.

This information should include:

- Copies of operating system configuration files or start-up files
- Copies of network configuration files (for example, the `hosts` and `services` files)
- Names and permissions for the SAP ASE executable files and database devices
- Names and locations of the tape devices used for backups

- Copies of operating system scripts or programs for automated backups, starting SAP ASE, or performing other administration activities

3.7.5 Disaster Recovery Plan

Consolidate the basic backup and recovery procedures, the guidelines in *Backup and Recovery*, and your personal experiences in recovering data into a concise list of recovery steps that are tailored to your system.

This list can be useful to both yourself and to other system administrators who may need to recover a production system in the event of an emergency.

3.8 Additional Resources

The amount of information for system administrators to learn may seem overwhelming. There are several software tools that can help you learn and facilitate basic administration tasks.

The SAP Adaptive Server Enterprise cockpit (SAP ASE cockpit) simplifies many administration tasks and provides monitoring information. There are also many third-party software packages available designed to help system administrators manage daily maintenance activities.

4 Managing and Monitoring SAP ASE

SAP ASE includes managing and monitoring tools.

4.1 Managing SAP ASE with SAP ASE Cockpit

SAP ASE cockpit provides a single, comprehensive, Web-administration console for real-time performance, status, and availability monitoring of large-scale SAP enterprise servers.

SAP ASE cockpit includes historical monitoring, threshold-based alerts and notifications, alert-based script execution, and intelligent tools for identifying performance and usage trends. SAP ASE cockpit replaces the SAP Control Center (SCC). Unlike SAP Control Center, SAP ASE cockpit is designed as an onboard management solution, where you install the cockpit on each SAP ASE host to manage and monitor that system.

It is possible to continue to run SCC 3.x on your system (allowing you access to previously collected and archived statistics), but SCC does not recognize any features or functionality introduced in 16.0 SP02.

SAP ASE cockpit is installed as part of an SAP ASE server installation. The SAP ASE cockpit agent continually broadcasts information about the server to subscribing clients via the SAP ASE cockpit framework. Messages from the Web client pass through the SAP ASE cockpit framework to the agent, which in turn, processes the request on the SAP ASE server.

SAP ASE cockpit is free of charge to customers who have a paid license for SAP ASE.

For detailed SAP ASE cockpit information, see the*SAP Adaptive Server Enterprise Cockpit* documentation, which includes SAP ASE configuration procedures, user assistance, and troubleshooting information.

4.2 Configuration History Tracking

SAP ASE allows you to track the history of configuration changes made to your server. The `sp_confighistory` system procedure manages the history of configuration changes, and stores data about the changes in the `sybsecurity` database.

Configuration properties that are tracked include:

- Server-wide configuration parameters
- Database options
- Data cache and data cache pool properties
- Engine threads
- Changes to the server configuration file.

You must install the `sybsecurity` auditing database to track these properties.

`sp_confighistory` displays SAP ASE configuration changes, including which configuration option has been changed, the old and new values, the user who made the change, and when the change was made. SAP ASE stores records of configuration changes in the `sybsecurity` database. Query the `ch_events` view or run `sp_confighistory` to access these records.

For example, the output below includes changes that include enabling auditing and changing the value of `max online engines` from 5 to 7:

```
area type target element oldvalue newvalue mode timestamp username instanceid
---- ---- ------ ------- -------- -------- ---- --------- -------- ----------
AUDIT global auditing  NULL NULL off on NULL Jul 15 2013 2:22PM sa NULL
SERVER sp_configure NULL max online engines 5 7 static Jul 15 2013 2:23PM sa NULL
```

4.2.1 Configuring SAP ASE to Track Configuration Changes

To install `sp_confighistory`, run the `installsecurity` script.

Context

Install and enable the audit system. See the *Security Administration Guide*.

Procedure

1. Enable configuration history tracking (requires the sa_role, sso_role, or `manage auditing` if granular permission is enabled):

   ```
   sp_audit "config_history", "all", "all", "on"
   ```

   > **i** Note
   >
   > Issuing `sp_audit` is recorded in the configuration history.

2. Enable auditing:

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

3. Move to the `sybsecurity` database:

   ```
   use sybsecurity
   ```

4. Create the `ch_events` view:

   ```
   sp_confighistory create_view
   ```

   `ch_events` collects information from all the audit tables, and becomes out of sync if you add or remove audit tables. If this occurs, `ch_events` may not include some configuration history record changes, or you

may see error messages 208 (`table not found`) and 4413 (`view unusable`) when you query `ch_events`.

Use `sp_confighistory create_view` to update `ch_events` when you add or remove audit tables. `sp_confighistory create_view` drops the view if it exists, and creates a new view that corresponds to the current audit tables.

4.2.2 Changes Captured

When configuration history auditing is enabled, SAP ASE captures a number of events.

The `ch_events` view does not record changes if the new value is the same as the old value.

Startup Configuration Changes

If you modify the SAP ASE configuration file while the SAP ASE is shut down, SAP ASE records any changes to the configuration in the `ch_events` table it starts (recording a value of NULL for the `mode` and `username` values for these changes).

Reading the Configuration File

`ch_events` records the event when you read, write, verify, and restore the configuration file, but does not record the configuration value changes. For example, if you change the value for `number of user connections` and then issue:

```
sp_configure "configuration file", 0, "read", "srv.config"
```

`ch_events` records that you read the configuration file, but does not record the configuration value changes.

sp_configure Changes

SAP ASE records all changes made by `sp_configure`, including:

- Name of the configuration parameter
- Old configuration value
- New configuration value
- Whether the parameter is dynamic or static
- Timestamp of the date and time the change was made
- Login of the user making the change

Configuration changes caused by reading from the configuration file are not recorded. That is, SAP ASE records the reading, writing, verifying, and restoring operations, but does not record the configuration changes

caused by a reading operation. You can also change configuration values by reading a different or manually modified configuration file. Although SAP ASE records that it read the file, it does not record the individual parameter changes.

Changes to Server Options

`ch_events` records all changes made by `sp_serveroption`, including:

- Name of affected server
- Name of option that was changed
- Old option value
- New option value
- Timestamp of the date and time the change was made
- Login of the user making the change

Changes to Database Options

`ch_events` records these changes made by `sp_dboption`:

- Name of affected database
- Name of option that was changed
- Old option value
- New option value
- Timestamp of the date and time the change was made
- Login of the user making the change

Changes to Cache Configuration

`ch_events` records all changes made by `sp_cacheconfig` and `sp_poolconfig` to cache configurations.

Recorded changes from `sp_cacheconfig` include:

- Name of affected cache
- Old cache size
- New cache size
- Attribute (cache type, cache replacement policy, partition number), if applicable
- Timestamp of the date and time the change was made
- Login of the user making the change
- (Cluster Edition only) Instance to which this change applies

Recorded changes from `sp_poolconfig` include:

- Name of affected cache

- Configuration pool
- Old cache pool size
- New cache pool size
- Name of changed attribute (affected pool, wash size, asynchronous prefetch (APF) percentage)
- Timestamp of the date and time the change was made
- Login of the user making the change
- (Cluster Edition only) Instance to which this change applies

Changes to Trace flags and Switches

`ch_events` records changes to `dbcc traceon | off` and `set switch on | off`.

Recorded changes from `dbcc traceon | off` include:

- Trace flag affected
- Session ID
- Old trace flag state (`on` or `off`)
- New trace flag state (`on` or `off`)
- Timestamp of the date and time the change was made
- Login of the user making the change
- (Cluster Edition only) Instance to which this change applies

Recorded changes from `set switch on | off` include:

- Switch number affected
- Old switch state (`on` or `off`)
- New switch state (`on` or `off`)
- Name of changed attribute (server-wide or session-specific, with override, with no_info)
- Timestamp of the date and time the change was made
- Login of the user making the change

Changes to Number of Engines

The changes tracked by `create thread pool`, `alter thread pool`, and `drop thread pool` include:

- Name of pool name affected
- Old pool size
- New pool size
- Name of changed attribute (new pool name, idle timeout, and so on)
- Timestamp of the date and time the change was made
- Login of the user making the change
- (Cluster Edition only) Instance to which this change applies

SAP ASE Startup and Shutdown Events

`ch_events` records this information for startup, `shutdown`, `shutdown with nowait`, and abrupt (unscheduled) shutdown events for SAP ASE and instances from the Cluster Edition:

- Name of the action (startup, `shutdown`, `shutdown with nowait`, abrupt shutdown).
- Time spent waiting for a shutdown. Not applicable for `shutdown with no_wait`.
- Name of the host on which the server starts.
- Timestamp of the date and time the change was made.
- Login of the user making the change.
- (Cluster Edition only) Instance to which this change applies.

> **i** Note
>
> Because `ch_events` records the shutdown when you issue the `shutdown` command, `ch_events` may record multiple shutdowns during a polite shutdown if you issue the command more than once.

Enabling or Disabling Auditing

`ch_events` records this information about tracking, global auditing, and configuration history auditing:

- Name of the action (`enable` or `disable`)
- Timestamp of the date and time the change was made
- Login of the user making the change

4.2.3 Querying ch_events

SAP ASE includes the `ch_events` view as part of the `sybsecurity` database.

`ch_events` presents configuration change history data in an easy to read format. You can query `ch_events` directly, or use the `sp_confighistory` system procedure to generate reports on configuration changes history. Either method provides the same information.

Using the `select` command provides the flexibility of the Transact-SQL™ language to qualify your result set (you must first move to the `sybsecurity` database before selecting from the `ch_events` view). `sp_confighistory` provides a more streamlined result set.

For example, if you make these configuration changes in SAP ASE:

```
sp_dboption sybsystemprocs, "delayed commit", false
sp_cacheconfig pub_cache, '10M'
sp_cacheconfig pub_log_cache, '2000K', logonly
```

Then shut down and restart the server, `sp_confighistory` returns:

```
sp_confighistory
 area     type            target         element             oldvalue
```

```
         newvalue    mode timestamp           username instanceid
 -------- --------------- -------------- ------------------- --------
         ----------- ---- ------------------ -------- ----------
 AUDIT    global auditing NULL           NULL                off
         on          NULL Aug 22 2013 11:56AM sa             NULL
 DATABASE sp_dboption     sybsystemprocs delayed commit      true
         false       NULL Aug 22 2013  3:16PM sa             NULL
 CACHE    sp_cacheconfig  pub_cache      NULL                10240
         not changed NULL Aug 22 2013  3:18PM sa             NULL
 CACHE    sp_cacheconfig  pub_log_cache  cache type: logonly 2000
         not changed NULL Aug 22 2013  3:19PM sa             NULL
 SUSD     shutdown        NULL           NULL                NULL
         NULL        NULL Aug 22 2013  3:49PM sa             NULL
 SUSD     startup         NULL           tigger              NULL
         NULL        NULL Aug 22 2013  3:50PM NULL           NULL
```

Include the date with `sp_confighistory` to select the changes over a period of time. This example shows all changes made after August 23, 2013:

```
sp_confighistory "Aug 23 2013"
 area type     target element oldvalue newvalue mode timestamp
         username instanceid
 ---- -------- ------ ------- -------- -------- ---- --------------------
         -------- ----------
 SUSD shutdown NULL   NULL    NULL     NULL     NULL Aug 23 2013  9:00AM
         sa             NULL
 SUSD startup  NULL   tigger  NULL     NULL     NULL Aug 23 2013 10:38AM
         NULL           NULL
```

Issuing `select` provides this result set:

```
use sybsecurity
go
select * from ch_events
go
area        type                 target            element
oldvalue    newvalue     mode
                  timestamp                username        instanceid
----------  -------------------  ----------------  -----------------
----------  ---------    --------------
                  -----------------------   -----------     ---------------
 AUDIT      global auditing      NULL              NULL
off         on           NULL
                  Aug 22 2013 11:56AM       sa              NULL
 DATABASE   sp_dboption          sybsystemprocs    delayed commit
true        false        NULL
                  Aug 22 2013  3:16PM       sa              NULL
 CACHE      sp_cacheconfig       pub_cache         NULL
10240       not changed  NULL
                  Aug 22 2013  3:18PM       sa              NULL
 CACHE      sp_cacheconfig       pub_log_cache     cache type: logonly
2000        not changed  NULL
                  Aug 22 2013  3:19PM       sa              NULL
 SUSD       shutdown             NULL              NULL
NULL        NULL         NULL
                  Aug 22 2013  3:49PM       sa              NULL
 SUSD       startup              NULL              tiger
NULL        NULL         NULL
                  Aug 22 2013  3:50PM       NULL            NULL
```

Include the `last` parameter to see the last items changed:

```
sp_confighistory last
 area type    target element oldvalue newvalue mode timestamp
         username instanceid
```

```
---- ------- ------ ------- -------- -------- ---- -------------------
         -------- ----------
SUSD startup NULL   tigger  NULL     NULL     NULL Aug 22 2013  3:50PM
        NULL           NULL
```

5 Setting Configuration Parameters

Configuration parameters are set using `sp_configure` and are used to tailor SAP ASE for an installation's particular needs. Determine which configuration parameters you should reset to optimize server performance.

For a complete list of configuration parameters, see *Reference Manual: Configuration Parameters*.

> ⚠ Caution
>
> Change configuration parameters with caution. Arbitrary changes in parameter values can adversely affect SAP ASE performance and other aspects of server operation.

In many cases, the maximum allowable values for configuration parameters are usually limited by available memory, rather than by `sp_configure` limitations.

SAP ASE stores the values of configuration parameters in a configuration file, which is an ASCII text file. When you install a new SAP ASE, your parameters are set to the default configuration; the default name of the file is `server_name.cfg`, and the default location of the file is the SAP ASE home directory ($SYBASE_ASE). Each time you modify a configuration parameter, SAP ASE creates a copy of the outdated configuration file, using the naming convention `server_name.001`, `server_name.002`, `server_name.003`...`server_name.999`. SAP ASE writes the new values to the file `server_name.cfg` or to a file name you specify at start-up.

5.1 Modifying Configuration Parameters

You can modify configuration parameters using a variety of methods.

These methods include:

- Executing `sp_configure` with the appropriate parameters and values,
- Editing your configuration file and then invoking `sp_configure` with the `configuration file` option, or
- Specifying the name of a configuration file at start-up.

Configuration parameters are either dynamic or static. Dynamic parameters take effect as soon as you execute `sp_configure`. Static parameters require memory to be reallocated, so they take effect only after you have restarted SAP ASE. The description of each parameter in this chapter indicates whether it is static or dynamic.

SAP ASE writes the new value to the system table `sysconfigures` and to the configuration file when you change the value. The current configuration file and `sysconfigures` reflect configured values, not run values. The system table `syscurconfigs` reflects current run values of configuration parameters.

5.2 Required Roles for Modifying Configuration Parameters

The SAP ASE role you have determines how you execute `sp_configure`.

- Any user can execute `sp_configure` to display information about parameters and their current values.
- Only a system administrator or a system security officer can execute `sp_configure` to modify configuration parameters.
- Only a system security officer can execute `sp_configure` to modify values for:
  - `allow procedure grouping`
  - `allow remote access`
  - `allow sendmsg`
  - `allow updates to system tables`
  - `auditing`
  - `audit queue size`
  - `check password for digit`
  - `current audit table`
  - `enable ldap user auth`
  - `enable pam user auth`
  - `enable ssl`
  - `log audit logon failure`
  - `log audit logon success`
  - `maximum failed logins`
  - `minimum password length`
  - `msg confidentiality reqd`
  - `msg integrity reqd`
  - `secure default login`
  - `select on syscomments.text`
  - `SQL Perfmon Integration`
  - `syb_sendmsg port number`
  - `suspended audit when device full`
  - `systemwide password expiration`
  - `unified login required`
  - `use security services`

5.3 Unit Specification Using sp_configure

`sp_configure` allows you to specify the value for configuration parameters in unit specifiers.

The unit specifiers are `p` or `P` for pages, `m` or `M` for megabytes, and `g` or `G` for gigabytes. If you do not specify a unit, and you are configuring a parameter that controls memory, SAP ASE uses the logical page size for the basic unit.

> **Note**
>
> When you are configuring memory-related parameters, use only the `P` (page size) parameter for your unit specification. If you use any other parameter to configure memory related parameters, SAP ASE may issue an arithmetic overflow error message.

The syntax to indicate a particular unit specification is:

```
sp_configure "parameter name", 0, "p|P|k|K|m|M|g|G"
```

You must include the "0" as a placeholder.

You can use this unit specification to configure any parameter. For example, when setting `number of locks` to 1024 you can enter:

```
sp_configure "number of locks", 1024
```

or:

```
sp_configure "number of locks", 0, "1K"
```

This functionality does not change the way in which SAP ASE reports `sp_configure` output.

5.4 Global Versus Session Settings

Some SAP ASE global settings are similar to session-level settings. For example, both `sp_configure` and `sp_passwordpolicy` include a `minimum password length` parameter.

If a global and session-level parameters configure the same setting, only the session level setting applies to the login (login level settings override global settings).

For example, if you set this global configuration parameter:

```
sp_configure 'minimum password length' 12
```

And you attempt to add user joe to the server with a insufficiently long password:

```
sp_adduser joe, joejoe
```

The command fails:

```
Msg 10317, Level 14, State 1:
Procedure 'sp_password', Line 118:
The specified password is too short. Passwords must be at least 12 character(s)
long.
Msg 17720, Level 16, State 1:
Procedure 'sp_password', Line 128:
Error:  Unable to set the Password.
```

However, if you set `sp_passwordpolicy` to a shorter length, SAP ASE requires a shorter password length for this session:

```
sp_passwordpolicy 'set', 'minimum password length', 6
```

The command succeeds:

```
sp_addlogin joe, joejoe
```

```
Password correctly set.
Account unlocked.
New login created.
(return status = 0)
```

SAP ASE tests all configuration limits independently, and the login attempt must pass all the applicable limits to succeed.

5.5 Getting Help Information on Configuration Parameters

Use either `sp_helpconfig` or `sp_configure` to display information about a particular configuration parameter.

For example:

```
sp_helpconfig "number of open"
```

```
Configuration option is not unique.
option_name                    config_value  run_value
------------------------------ ------------ -----------
number of open databases                 12          12
number of open indexes                  500         500
number of open objects                  500         500
```

```
sp_helpconfig "number of open indexes"
```

```
number of open indexes sets the maximum number of indexes that can be open at
one time on SQL Server. The default value is 500.
Minimum Value Maximum Value Default Value Current Value Memory Used
------------- ------------- ------------- ------------- -----------
          100    2147483647           500           500         208
```

```
sp_configure "number of open indexes"
```

```
Parameter Name           Default  Memory Used  Config Value  Run Value
-----------------------  -------  -----------  ------------  ---------
number of open indexes       500          208           500        500
```

See *System Administration Guide: Volume 2 > Configuring Memory*.

5.6 Using sp_configure

`sp_configure` displays and resets configuration parameters.

You can restrict the number of parameters that `sp_configure` shows by using `sp_displaylevel` to set your display level to one of:

- Basic
- Intermediate
- Comprehensive

For information about `sp_displaylevel`, see the *Reference Manual: Procedures*.

In this table, the "Effect" column assumes that your display level is set to "comprehensive."

Command	Effect
`sp_configure`	Displays all configuration parameters by group, their current values, their default values, the value to which they have most recently been set, and the amount of memory used by this particular setting.
`sp_configure "<parameter>"`	Displays current value, default value, most recently changed value, and amount of memory used by the specified parameter.
`sp_configure "<parameter>", <value>`	Resets `<parameter>` to `<value>`.
`sp_configure "<parameter>", 0, "default"`	Resets the specified parameter to its default value.
`sp_configure "<group_name>"`	Displays all configuration parameters in `<group_name>`, their current values, their default values, the values to which they were recently set, and the amount of memory used by each.
`sp_configure "configuration file", 0, "<sub_command>", "<file_name>"`	Sets configuration parameters from the configuration file.

5.6.1 sp_configure Syntax Elements

`sp_configure` includes a variety of variables to help you configure SAP ASE.

- `<parameter>` – is any valid SAP ASE configuration parameter or parameter substring.
- `<value>` – is any integer within the valid range for that parameter. (See the descriptions of the individual parameters for valid range information.) Parameters that work as toggles have only two valid values: 1 (on) and 0 (off).
- `<group_name>` – is the name of any group in the parameter hierarchy.

`sp_configure` parses each parameter (and parameter name fragment) as "`<%parameter%>`". A string that does not uniquely identify a particular parameter returns values for all parameters matching the string.

The following example returns values for all configuration parameters that include "lock," such as `lock shared memory`, `number of locks`, `lock promotion HWM`, `server clock tick length`, `print deadlock information`, and `deadlock retries`:

```
sp_configure "lock"
```

> **i** Note
>
> If you attempt to set a parameter value with a nonunique parameter name fragment, `sp_configure` returns the current values for all parameters matching the fragment and asks you to specify a unique parameter name.

5.6.2 Issue sp_configure with the Configuration File

Configure SAP ASE either interactively, by using `sp_configure`, or noninteractively, by instructing SAP ASE to read values from an edited or restored version of the configuration file.

By making your changes from the configuration file, you can:

- Replicate a specific configuration across multiple servers by using the same configuration file.
- Use a configuration file as a baseline for testing configuration values on your server.
- Use a configuration file to perform validation checking on parameter values before actually setting the values.
- Create multiple configuration files and switch between them as your resource needs change.

5.6.2.1 Configuration File Naming Recommendations

To work with a configuration file that has a name other than the default name, keep the `<server_name>` part of the file name, and include at least one alphabetic character in the extension (for example `my_server.A001`).

Alternatively, you can change the `<server_name>` part of the file name (for example, `A_my_server.001`). Doing this avoids confusion with the backup configuration files generated by SAP ASE when you modify a parameter.

5.6.2.2 Read or Write the Configuration File with sp_configure

`sp_configure` includes syntax options for using configuration files.

The syntax is:

```
sp_configure "configuration file", 0, "<subcommand>", "<file_name>"
```

where:

- `"configuration file"` – including quotes, specifies that this command uses the configuration file.
- `0` – required—for backward compatibility—after the `configuration file` parameter.
- `"<subcommand>"` – is one of:
  - `write` – creates a file named `<file_name>` with the current configuration. If `<file_name>` already exists, a message is written to the error log; the existing file is renamed using the convention `<server_name>.001`, `<server_name>.<002>`, and so on. If you have changed a static parameter, but you have not restarted your server, `write` displays the currently running value for that parameter. If you do not specify a directory with `<file_name>`, the file is written to the directory from which SAP ASE was started.
  - `read` – performs validation checking on values contained in `<file_name>` and reads those values that pass validation into the server. If any parameters are missing from `<file_name>`, the current values for those parameters are used.
    If the value of a static parameter in `<file_name>` is different from its current running value, `read` fails and a message is printed. However, validation is still performed on the values in `<file_name>`.
  - `verify` – performs validation checking on the values in `<file_name>`. This is useful if you have edited the configuration file, as it prevents you from attempting to configure your server with invalid configuration values.
  - `restore` – creates `<file_name>` with the most recently configured values. If you have configured static parameters to new values, this subcommand writes the configured, not the currently running, values to the file. This is useful if all copies of the configuration file have been lost and you must generate a new copy. If you do not specify a directory with `<file_name>`, the file is written to the directory from which SAP ASE was started.
- `<file_name>` – specifies the configuration file to use in conjunction with any `<subcommand>`. If you do not specify a directory as part of the file name, the directory where SAP ASE was started is used.

Examples

- Example 1 – performs validation checking on the values in the file `srv.config` and reads the parameters that pass validation into the server. Current run values are substituted for values that do not pass validation checking:

  ```
  sp_configure "configuration file", 0, "read", "srv.config"
  ```

- Example 2 – creates the file `my_server.config` and writes the current configuration values the server is using to that file:

  ```
  sp_configure "configuration file", 0, "write", "my_server.config"
  ```

5.6.2.3 Edit the Configuration File

The configuration file is an ASCII file that you can edit with any text editor that can save files in ASCII format.

The syntax for each parameter is:

```
<parameter_name>={<value> | DEFAULT}
```

where:

- `<parameter_name>` – is the name of the parameter you want to specify.
- `<value>` – is the numeric value for set `<parameter_name>`.
- "DEFAULT" – specifies that you want to use the default value for `<parameter_name>`.

Examples

- Example 1 – specifies that the transaction can retry its attempt to acquire a lock one time when deadlocking occurs during an index `page split` or shrink:

  ```
  deadlock retries = 1
  ```

- Example 2 – specifies that the default value for the parameter `cpu accounting flush interval` should be used:

  ```
  cpu accounting flush interval=DEFAULT
  ```

When you edit a configuration file, your edits are not validated until you check the file using the `verify` option, read the file with the `read` option, or restart SAP ASE with that configuration file.

If all your configuration files are lost or corrupted, you can re-create one from a running server by using the `restore` subcommand and specifying a name for the new file. The parameters in the new file are set to the values with which your server is currently running.

5.6.2.3.1 Permissions for Configuration Files

Configuration files are nonencrypted ASCII text files.

By default, configuration files are created with read and write permissions set for the file owner. If you created the configuration file at the operating system level, you are the file owner; if you created the configuration file from SAP ASE, using the `write` or `restore` parameter, the file owner is the user who started SAP ASE. Usually, this is the user "sybase." To restrict access to configuration files, use your operating system's file permission command to set read, write, and execute permissions as appropriate.

> **i Note**
>
> You must set permissions accordingly on *each* configuration file created.

5.6.2.3.2 Backing Up Configuration Files

Configuration files are not automatically backed up when you back up the `master` database. They are operating system files—back them up in the same way you back up other operating system files.

5.6.2.3.3 Verify the Name of the Configuration File Currently in Use

Due to space limitations, `sp_configure` output truncates the name of the configuration file.

To see the full name of the configuration file, use:

```
select s1.value2
from syscurconfigs s1, sysconfigures s2
where s1.config = s2.config
and s2.name = "configuration file"
```

5.6.2.4 Start SAP ASE Using a Configuration File

By default, SAP ASE reads the configuration file `server_name.cfg` in the start-up directory when it starts. If this file does not exist, it creates a new file and uses SAP ASE defaults for all values.

You can start SAP ASE with a specified configuration file. For more information, see the *Utility Guide*.

If the configuration file you specify does not exist, SAP ASE prints an error message and does not start.

If the command is successful, the file `server_name.bak` is created. This file contains the configuration values stored in `sysconfigures` prior to the time `sysconfigures` was updated with the values read in from the configuration file you specified. This file is overwritten with each subsequent start-up.

5.6.2.4.1 Configuration File Errors

When there are errors in the configuration file, SAP ASE may not start, or may use default values.

SAP ASE uses default values if:

- There are illegal values. For example, if a parameter requires a numeric value, and the configuration file contains a character string, SAP ASE uses the default value.
- Values are below the minimum allowable value.

5.6.3 The Parameter Hierarchy

Configuration parameters are grouped according to the area of SAP ASE behavior they affect. This makes it easier to identify all parameters that you might need to tune to improve a particular area of SAP ASE performance.

Although each parameter has a primary group to which it belongs, many have secondary groups to which they also belong. For example, `number of remote connections` belongs primarily to the network communication group, but it also belongs secondarily to the memory use group. This reflects the fact that some parameters have implications for a number of areas of SAP ASE behavior. `sp_configure` displays parameters in all groups to which they belong.

Table 3: Configuration Groups

Parameter Group	Configures SAP ASE for
Application Functionality	Controls the default setting for a subset of configuration options
Backup/Recovery	Backing up and recovering data
Cache Manager	Data and procedure caches
Component Integration Services	Component Integration Services administration
DTM Administration	Distributed transaction management (DTM) facilities
Diagnostics	Diagnostic principles
Disk I/O	Disk I/O
Error log	Error log, and the logging of SAP ASE events to the Windows event log
Extended Stored Procedures	The behavior of extended stored procedures (ESPs).
General Information	Basic system administration
Java Services	Memory for Java in SAP ASE See *Java in SAP Adaptive Server Enterprise* for complete information about Java in the database. If you use method calls to JDBC, you may need to increase the size of the execution stack available to the user.
Languages	Languages, sort orders, and character sets
Lock Manager	Locking
Memory Use	Memory consumption

Parameter Group	Configures SAP ASE for
Meta-Data Caches	Setting the metadata cache size for frequently used system catalog information. The metadata cache is a reserved area of memory used for tracking information on databases, indexes, or objects. The greater the number of open databases, indexes, or objects, the larger the metadata cache size. For a discussion of metadata caches in a memory-usage context, see *System Administration Guide: Volume 2 > Configuring Memory*.
Monitoring	Collecting monitoring information. By default, SAP ASE does not collect monitoring information. See *Performance and Tuning Guide: Monitoring and Analyzing > Monitoring Tables*.
Network Communication	Communication between SAP ASE and remote servers, and between SAP ASE and client programs
O/S Resources	Use of operating system resources
Physical Memory	Your machine's physical memory resources
Physical Resources	Your machine's physical resources
Processors	Processors in an SMP environment
Query Tuning	Query optimization
RepAgent Thread Administration	Replication via Replication Server
Security Related	Security-related features
Shared Disk Cluster	Cluster Edition
SQL Server Administration	General SAP ASE administration
SSD Cache Manager	SSD Cache Manager
Unicode	Unicode-related features
User Environment	User environments

The syntax for displaying all groups and their associated parameters, and the current values for the parameters, is:

```
sp_configure
```

> **i Note**
>
> The number of parameters returned by `sp_configure` depends on the value to which you have your display level set.

The following is the syntax for displaying a particular group and its associated parameter:

```
sp_configure "<group_name>"
```

For example, to display the disk I/O group, enter:

```
sp_configure "Disk I/O"
```

```
Group: Disk I/O
Parameter Name           Default Memory Used Config Value Run Value
unit         type
--------------           ------- ----------- ------------ ---------
------       -------------
allow sql server async i/o     1           0            1         1
switch       static
diable disk mirroring          1           0            1         1
switch       static
disk i/o structures          256           0          256       256
number       dynamic
number of devices             10           0           10        10
number       dynamic
number of large I/O buffers    6       12352            6         6
number       dynamic
page utilization percent      95           0           95        95
percent      dynamic
```

> **i** Note
>
> If the server uses a case-insensitive sort order, `sp_configure` with no parameters returns a list of all configuration parameters and groups in alphabetical order with no grouping displayed.

5.6.4 User-Defined Subsets of the Parameter Hierarchy: Display Levels

Depending on how you use SAP ASE, you may need to adjust some parameters more frequently than others. It may be easier to work with a subset of parameters.

The default display level is comprehensive. When you set your display level, the setting persists across multiple sessions. However, you can reset it at any time.

- Basic – shows only the most basic parameters, and is appropriate for general server tuning.
- Intermediate – includes parameters that are somewhat more complex, in addition to the Basic parameters.
- Comprehensive – includes all the parameters, including the most complex ones. This level is appropriate for users doing highly detailed server tuning.

The syntax for showing your current display level is:

```
sp_displaylevel
```

To set the display level, use the following, where `<user_name>` is your SAP ASE login name:

```
sp_displaylevel <user_name>[, basic | intermediate | comprehensive]
```

5.6.4.1 The Effect of the Display Level on sp_configure Output

If your display level is set to either `basic` or `intermediate`, `sp_configure` returns only a subset of the parameters that are returned when your display level is set to `comprehensive`.

For example, if your display level is set to `intermediate`, and you want to see the parameters in the languages group, enter:

```
sp_configure "Languages"
```

The output looks like this:

```
sp_configure
Group: Languages
Parameter Name         Default Memory Used Config Value Run Value Unit Type
---------------        ------- ----------- ------------ --------- ---- ----
default character set    1          0            1          1     id   static
default language id      0          0            0         0      id   dyna
. . .
```

This represents only a subset of the parameters in the languages group; some language parameters appear only when your display level is `comprehensive`.

5.6.5 Performance Tuning with sp_configure and sp_sysmon

`sp_sysmon` monitors SAP ASE performance and generates statistical information that describes the behavior of your SAP ASE system.

You can run `sp_sysmon` before and after using `sp_configure` to adjust configuration parameters. The output gives you a basis for performance tuning and allows you to observe the results of configuration changes.

See the *Performance and Tuning Series: Monitoring SAP Adaptive Server with sp_sysmon*.

5.6.6 Preventing Unwanted Upgrades

The configuration parameter `prevent automatic upgrades` prevents users from accidentally booting to a newer version of SAP ASE that would prevent them from restarting their older server version.

Valid values are:

- 0 (default) - allows a newer version of SAP ASE to boot without requiring the explicit permission that is given by the flag `--upgrade-ok`.
- 1 - any attempt to boot to a later SAP ASE version will fail.

When set to 1, SAP ASE will shut down after printing message 110325: "The current database version '%d' is lower than the server's version '%d', and the server is configured to require upgrade permission. Boot the server with flag '--upgrade-ok' to permit this boot."

If you want to perform an upgrade installation, perform one of the following:

- In the older SAP ASE version, execute:

```
sp_configure 'prevent automatic upgrade', 0
```

- Or, boot the newer SAP ASE version with the command-line flag `--upgrade-ok`.

5.6.7 Using Configuration Parameters in a Clustered Environment

In the Cluster Edition, SAP supports both cluster-wide and instance-specific configuration.

Cluster-wide configuration parameters are applied to all instances in the cluster. Local configuration parameters are applied only to a specified instance.

- Local configuration overrides cluster-wide configuration.
- If an instance-specific configuration has not been applied, the cluster-wide configuration applies.
- Some parameters, such as `default character set id`, cannot be applied to a specific instance. These parameters can only be used over an entire cluster.

The cluster configuration file includes an instance-specific configuration block. Parameter settings in the instance-specific block override cluster-wide settings. For example:

```
max online engines = DEFAULT
[Instance:ase1]
max online engines = 5
[Instance:ase2]
max online engines = 3
```

See the *Clusters Users Guide*.

5.7 sp_configure Output

The values sp_configure prints vary, depending on your platform and on what values you have already changed.

This output shows the type of information `sp_configure` prints if your display level is `comprehensive`, and you execute `sp_configure` with no parameters:

```
sp_configure
Group: Configuration Options
Group: Backup/Recovery
Parameter Name        Default Memory Used Config Value Run Value Unit Type
--------------        ------- ----------- ------------ --------- ---- ----
allow remote access         1           0            1         1    switch dyn
print recovery info         0           0            0         0    switch dyn
recovery interval in m      5           0            5         5    minutes dyn
...
```

> **i Note**
>
> All configuration groups and parameters appears in output if your display level is set to “comprehensive.”

where:

- The “Default” column displays the default value. If you do not explicitly reconfigure a parameter, it retains its default value.
- “Memory Used” shows the amount of memory, in kilobytes, used by the parameter at its current value. Some related parameters draw from the same memory pool. For instance, the memory used for `stack size` and `stack guard size` is already accounted for in the memory used for `number of user connections`. If you added the memory used by each of these parameters separately, the sum is more than the amount actually used. Parameters that “share” memory with other parameters are marked with a hash mark (“#”).
- “Config Value” displays the most recent value to which the configuration parameter has been set. When you execute `sp_configure` to modify a dynamic parameter:
  - The configuration and run values are updated.
  - The configuration file is updated.
  - The change takes effect immediately.

  When you modify a static parameter:
  - The configuration value is updated.
  - The configuration file is updated.
  - The change takes effect only when you restart SAP ASE.
- “Run Value” displays the value SAP ASE is currently using. It changes when you modify a dynamic parameter’s value and, for static parameters, after you restart SAP ASE.
- “Unit” displays the unit value of the configuration parameter. SAP ASE displays information in the following units:

Name of Unit	Unit Description
number	Number of items.
clock ticks	Number of clock ticks.
microseconds	Number of microseconds.
milliseconds	Number of milliseconds.
seconds	Number of seconds.
minutes	Number of minutes.
hours	Number of hours.
bytes	Number of bytes.
days	Number of days.
kilobytes	Number of kilobytes.
megabytes	Number of megabytes.
memory pages (2K)	Number of 2K memory pages.

Name of Unit	Unit Description
virtual pages (2K)	Number of 2K virtual pages.
logical pages	Number of logical pages. This value depends on the logical page size your server is using: 2, 4, 8, or 16K.
percent	Value of the configured parameter as a percentage.
ratio	Value of the configured parameter as a ratio.
switch	Value of the parameter is either TRUE (the parameter is turned on), or FALSE.
id	ID of the configured parameter you are investigating.
name	Character string name assigned to the run or configure value of the parameter. For example, "binary" appears under the "Run Value or "Config Value" column for the output of `sp_configure "lock scheme"`.
row	Number of rows

- "Type" displays whether the configuration option is static or dynamic. Changes to static parameters require that you restart SAP ASE for the changes to take effect. Changes to dynamic parameters take effect immediately without having to restart SAP ASE.

5.8 sysconfigures and syscurconfigs Tables

The report shown by `sp_configure` is constructed mainly from the `master..sysconfigures` and `master..syscurconfigs` system tables, with additional information provided from `sysattributes`, `sysdevices`, and other system tables.

The `value` column in the `sysconfigures` table records the last value set from `sp_configure` or the configuration file; the `value` column in `syscurconfigs` stores the value currently in use. For dynamic parameters, the two values match; for static parameters, which require a restart of the server to take effect, the two values are different if the values have been changed since SAP ASE was last started. The values may also be different when the default values are used. In this case, `sysconfigures` stores 0, and `syscurconfigs` stores the value that SAP ASE computes and uses.

`sp_configure` performs a join on `sysconfigures` and `syscurconfigs` to display the values reported by `sp_configure`.

5.8.1 Example syscurconfigs and sysconfigures Query

You can query `sysconfigures` and `syscurconfigs` to organize information the way you want.

For example, `sp_configure` without any arguments lists the memory used for configuration parameters, but does not list minimum and maximum values. Use this query to get a complete list of current memory usage, as well as minimum, maximum, and default values:

```
select b.name, memory_used, minimum_value, maximum_value, defvalue
from master.dbo.sysconfigures b,
master.dbo.syscurconfigs c
where b.config *= c.config and parent != 19
and b.config > 100
```

5.9 Named Cache Configuration Parameter Group

The Named Cache configuration parameter group provides details for named caches.

- `cache size` – size of the cache. By default SAP ASE creates 8MB caches. Change this parameter dynamically with `sp_cacheconfig`, or change the value in the server configuration file to have the change take place after the next server restart.
- `cache status` – status of the cache. One of `default data cache`, `log on ly`, `mixed`, or `in-memory storage`. The default data cache must have a cache status of `default data cache`, and cannot be changed. `cache status` for named caches can be `log only`, `mixed`, or, for in-memory databases, `in-memory storage` (you cannot change the `cache status` for in-memory databases).
  You cannot dynamically change the `cache status` in a clustered environment from `log only` on a local cache while other instance use a different `cache status`.
- `cache replacement` – describes the cache replacement policy. For named caches and default data caches, the replacement policy is `strict LRU` or `relaxed LRU`. Change this parameter dynamically with `sp_cacheconfig`, or change the value in the server configuration file to have the change take place after the next server restart. The cache replacement policy must be `none` for in-memory databases because they do not use buffer or page replacement.
- `local cache partition number` – number of cache partitions. You may partition a named cache into multiple cache partitions. The acceptable values are 0, 2, 4, 8, 16, 32, 64 or 128. You cannot change the number of cache partitions dynamically; you must restart SAP ASE for the change to take effect.

5.10 Dump Configuration Parameter Group

The `dump configuration` configuration parameter group represents these user-created dump configurations.

- `stripe directory` – is the directory in which files are archived during the dump operation. Archived files are typically named using this convention:

  ```
  database_name.nump_type.date-timestamp.stripeID
  ```

- `external api name` – is the name of the external API (byte stream device) to be used for the dump operation, and must conform to this format:

  ```
  External API Name::Options
  ```

- `number of stripes` – is the number of stripe devices to use during the dump operation. By default, a single stripe device is used.
- `number of retries` – is the number of times the server tries the dump operation for nonfatal errors up to a maximum of 5 times. The default is 0.
- `block size` – is the block size for the dump device and overrides the default block size for the device. `blocksize` must be at least 1 database page, and an exact multiple of the database page size.
- `compression level` – is the compression level for compressed dumps. By default, compression is disabled.
- `retain days` – is the number of days during which the dump cannot be overwritten. Backup Server requires confirmation to overwrite an unexpired volume. By default, `retaindays` is 0, and dumps can be overwritten.
- `init` – specifies whether the volume must be reinitialized. The default is "`noinit`".
- `verify` – specifies if Backup Server must perform a minimal page-header or full structural row check on the data pages as they are copied to archives. There is no structural check made to global allocation map (GAM), object allocation map (OAM), allocation pages, indexes, text, or log pages. By default, there is no verification of data pages during archiving.
- `notify` – the default message destination to Backup Server. One of:
  - `client` - route messages to the terminal that initiated the `dump` command.
  - `operator_console` - route messages to the terminal on which Backup Server is running
- `remote backup server name` – specifies the remote Backup Server used for a dump operation. The default is SYB_BACKUP.

This example contains multiple dump configurations as created in the configuration file:

```
[dump configuration : dmp_cfg1]
   stripe_dir = /work/dmp_cfg1_dir
   ext_api = DEFAULT
   num_stripes = 5
   retry = 0
   blocksize = DEFAULT
   compression = 9
   retaindays = DEFAULT
   init = DEFAULT
   verify = DEFAULT
   backup_srv_name = DEFAULT
[dump configuration : dmp_cfg2]
   stripe_dir = /work/dmp_cfg2_dir
   ext_api = syb_tsm
```

```
   num_stripes = DEFAULT
   retry = 3
   blocksize = DEFAULT
   compression = DEFAULT
   retaindays = DEFAULT
   init = DEFAULT
   verify = DEFAULT
   backup_srv_name = SYB_REMOTE
```

5.11 Bucket Pool Configuration Parameter Group

The bucket pool configuration parameter group represents bucket fragment manager configuration.

The bucket fragment manager allows you to tune the memory manager according to the varying needs of different server components. Since the tuning process requires the knowledge of internal functionality, you can modify bucket pool configuration with the guidance of SAP support.

Each bucket fragment pool in the system has an entry in the configuration file similar to:

```
[Bucket Pool:Network Buffers]
        enabled = 1
        bucket size = 32
        seed = 0
        maxfill = 0
        bucket size = 64
        seed = 0
        maxfill = 0
        bucket size = 96
        seed = 0
        maxfill = 0
        bucket size = 128
        seed = 0
        maxfill = 0
        bucket size = 160
        seed = 0
        maxfill = 0
        bucket size = 192
        seed = 0
        maxfill = 0
        bucket size = 224
        seed = 0
        maxfill = 0
        bucket size = 256
        seed = 0
        maxfill = 0
        bucket size = 288
        seed = 0
        maxfill = 0
        bucket size = 320
        seed = 0
        maxfill = 0
        bucket size = 352
        seed = 0
        maxfill = 0
        instances = 1
        instance seed = 0
        defragment = 0
        autotune = 0
        overhead = 0
```

Exact entries vary according to the bucket pool options.

A system administrator can safely modify `instances` parameter. You can improve efficiency by allowing groups of engines to have individual areas of the memory pool. When `instances` is set to:

- `1` - all engines allocate and free memory to the entire pool.
- a number representing the number of engines that are currently online - each engine allocates and free memory from its own section of the pool.

For optimal performance, set `instances` to a number between these two extremes.

5.12 Limiting the Number of Elements in the in Clause

You can set the `max number of IN elements` configuration parameter to limit the number of elements in the largest `in` clause in a query.

Syntax

```
sp_configure 'max number of IN elements', <limit_value>
```

The default value is 0, which means the configuration parameter is turned off. Valid values range from 256 to 128000. See *max number of IN elements* in the *Reference Manual: Configuration Parameters* for more details about the configuration parameter.

Execute the following command to check the value of the configuration parameter:

```
sp_configure 'max number of IN elements'
```

Usage

- When the limit value is not valid or crossed, SAP ASE reports an error.
- The limit setting applies to the largest `in` clause only and takes runtime check into account, which means the limit check is not done at object creation time, but is done at runtime.
- If a stored procedure or statement has been created or cached before the `max number of IN elements` configuration parameter is turned on, the configuration parameter does not take effect to it and the runtime check is not performed. If you want the configuration parameter to take effect, execute the stored procedure with `recompile` option, or run `dbcc purgesqlcache` to purge the statement cache and issue the statement again.

Example

In a new installation, the default value of the `max number of IN elements` is 0. Execute the following command to check the value of the configuration parameter:

```
sp_configure 'max number of IN elements'
```

The returned result is:

```
 Parameter Name            Default     Memory Used   Config Value   Run Value
Unit   Type
 ------------------------- ----------- ------------- -------------- ------------
------ -------
 max number of IN elements           0           0              0              0
number dynamic
(1 row affected)
(return status = 0)
```

Enable the configuration parameter by changing the value from 0 to a value between 256 and 128000:

```
sp_configure 'max number of IN elements', 256
```

The returned result is:

```
 Parameter Name            Default     Memory Used   Config Value   Run Value
Unit   Type
 ------------------------- ----------- ------------- -------------- ------------
------ -------
 max number of IN elements           0           0            256            256
number dynamic
```

6 Disk Resource Issues

Many SAP ASE defaults are set to reasonable values for aspects of storage management, such as database, table, and index location, and how much space is allocated for each one. Responsibility for storage allocation and management is often centralized, and usually, the system administrator has ultimate control over the allocation of disk resources to SAP ASE and the physical placement of databases, tables, and indexes on those resources.

6.1 Device Allocation and Object Placement

When configuring a new system, the system administrator must consider several issues that have a direct impact on the number and size of disk resources required. These device allocation issues refer to commands and procedures that add disk resources to SAP ASE.

Table 4: Device allocation documentation

Task	See
Initialize and allocate a default pool of database devices	*System Administration Guide: Volume 1 > Initializing Database Devices*
Mirror database devices for recovery	*System Administration Guide: Volume 2 > Mirroring Database Devices*

After the initial disk resources have been allocated to SAP ASE, the system administrator, database owner, and object owners should consider how to place databases and database objects on specific database devices. These object placement issues determine where database objects reside on your system and whether or not the objects share devices.

Table 5: Object placement

Task	See
Place databases on specific database devices	*System Administration Guide: Volume 2 > Creating and Managing User Databases*
Place tables and indexes on specific database devices	*System Administration Guide: Volume 2 > Creating and Managing User Databases*

Do not consider allocating devices separately from object placement. For example, if you decide that a particular table must reside on a dedicated pair of devices, first allocate those devices to SAP ASE.

6.2 Commands for Managing Disk Resources

SAP ASE offers many commands and strategies for managing disk resources.

Command	Task	See
`disk init name = "<dev_name>" physname = "<phys_name>"...`	Makes a physical device available to a particular SAP ASE. Assigns a database device name (`<dev_name>`) that is used to identify the device in other SAP ASE commands.	*System Administration Guide: Volume 1 > Initializing Database Devices*
`sp_deviceattr <logicalname>, <optname>, <optvalue>`	Changes the `<dsync>` setting of an existing database device file.	*System Administration Guide: Volume 1 > Initializing Database Devices*
`sp_diskdefault "<dev_name>"...`	Adds `<dev_name>` to the general pool of default database space.	*System Administration Guide: Volume 1 > Initializing Database Devices*
`disk resize name = "<device_name>", size = <additional_space>`	Dynamically increases the size of database devices.	*System Administration Guide: Volume 1 > Initializing Database Devices*
`disk mirror name = "<dev_name>" mirror = "<phys_name>"...`	Mirrors a database device on a specific physical device.	*System Administration Guide: Volume 2 > Mirroring Database Devices*

The table below lists the commands used in object placement. For information about how object placement affects performance, see *Performance and Tuning Series: Physical Database Tuning > Controlling Physical Data Placement*.

Table 6: Commands for placing objects on disk resources

Command	Task	See
`create database...on <dev_name>` or `alter database...on <dev_name>`	Makes database devices available to a particular SAP ASE database. The `log on` clause to `create database` places the database's logs on a particular database device.	*System Administration Guide: Volume 2 > Creating and Managing User Databases*

Command	Task	See

`create database...`
or
`alter database...`
	When used without the `on <dev_name>` clause, these commands allocate space on the default database devices.	*System Administration Guide: Volume 2 > Creating and Managing User Databases*

`sp_addsegment <seg_name>, <dbname>, <devname>`
and
`sp_extendsegment <seg_name>, <dbname>, <devname>`
	Creates a segment—a named collection of space—from the devices available to a particular database.	*System Administration Guide: Volume 2 > Creating and Using Segments*

`create table...on <seg_name>`
or
`create index...on <seg_name>`
	Creates database objects, placing them on a specific segment of the database's assigned disk space.	*System Administration Guide: Volume 2 > Creating and Using Segments*

`create table...`
or
`create index...`
	When used without `on <seg_name>`, tables and indexes occupy the general pool of space allocated to the database (the default devices).	*System Administration Guide: Volume 2 > Creating and Using Segments in System*

6.3 Considerations in Storage Management Decisions

The system administrator must make many decisions regarding the physical allocation of space to SAP ASE databases.

The major considerations are:

- Recovery – disk mirroring and maintaining logs on a separate physical device provide two mechanisms for full recovery in the event of physical disk failures.
- Performance – for tables or databases where speed of disk reads and writes is crucial, properly placing database objects on physical devices yields performance improvements. Disk mirroring slows the speed of disk writes.

6.3.1 Recovery

Recovery is the key motivation for using several disk devices. You can ensure full recovery by storing a database's log on a separate physical device. You can also mirror database devices to achieve nonstop recovery.

- Storing logs on a separate device – unless a database device is mirrored, full recovery requires that a database's transaction log be stored on a different device from the actual data (including indexes) of a database.
  In the event of a hard disk failure, you can create an up-to-date database by loading a dump of the database and then applying the log records that were safely stored on another device. See *System Administration Guide: Volume 2 > Creating and Managing User Databases* for information about the `log on` clause of `create database`.
- Disk mirroring – Nonstop recovery in the event of a hard disk failure is guaranteed by mirroring all SAP ASE devices to a separate physical disk. See *System Administration Guide: Volume 2 > Mirroring Database Devices*.

6.3.2 Performance

You can improve system performance by placing logs and database objects on separate devices.

For example:

- Place a table on one hard disk and nonclustered indexes on another to ensure that physical reads and writes are faster, since the work is split between two disk drives.
- Split large tables across two disks to improve performance, particularly for multiuser applications.
- When log and data share devices, disable user log cache buffering of transaction log records.
- Use partitioning to provide multiple insertion points for a heap table, add a degree of parallelism to systems configured to perform parallel query processing, and make it possible to distribute a table's I/O across multiple database devices.

See *Performance and Tuning Series: Physical Database Tuning > Controlling Physical Data Placement* for a detailed discussion of how object placement affects performance.

6.4 Status and Defaults at Installation

The installation program and scripts initialize the master device and set up the `master`, `model`, `sybsystemprocs`, `sybsecurity`, and temporary databases for you.

When you install SAP ASE, the system databases, system-defined segments, and database devices are organized as follows:

- The `master`, `model`, and `tempdb` databases are installed on the master device.
- The `sybsystemprocs` database is installed on a device that you specified.
- Three segments are created in each database: `system`, `default`, and `logsegment`.

- The master device is the default storage device for all user-created databases.

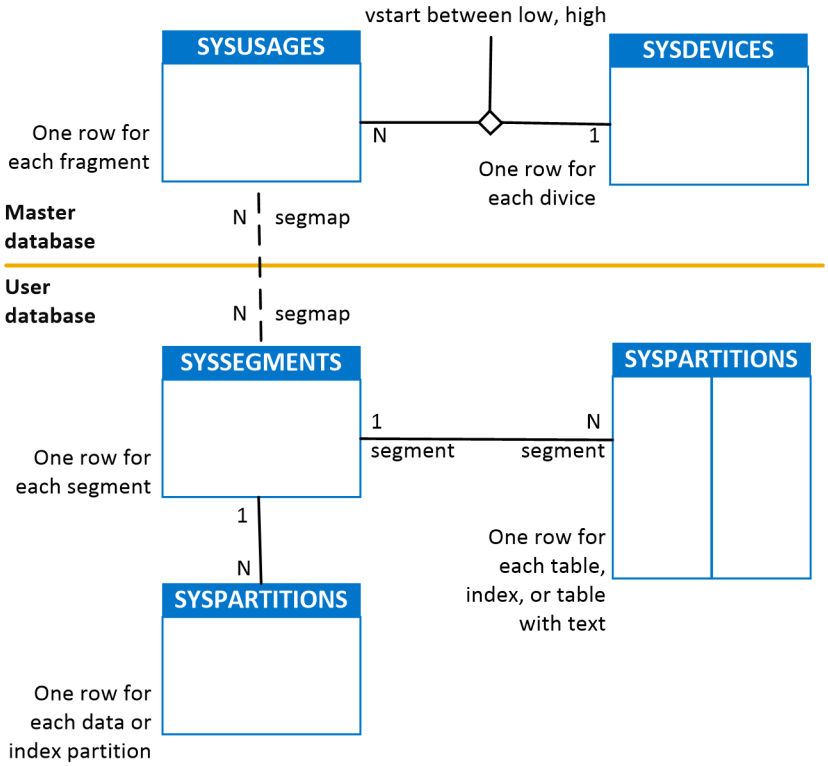
  > **i** Note
  >
  > After initializing new devices for default storage, remove the master device from the default storage area with `sp_diskdefault`. Do not store user databases and objects on the master device.

- If you install the audit database, `sybsecurity`, it is located on its own device.

6.5 System Tables That Manage Storage

Two system tables in the `master` database, `sysusages` and `sysdevices`, and three more in each user database (`syssegments`, `sysindexes`, and `syspartitions`) track the placement of databases, tables (including the transaction log table, `syslogs`), and indexes.

The relationship between the tables is illustrated below.



6.5.1 The sysdevices Table

The `sysdevices` table in the `master` database contains one row for each database device and may contain a row for each dump device (tape, disk, or operating system file) available to SAP ASE.

The `disk init` command adds entries for database devices to `master..sysdevices`. Dump devices, added using `sp_addumpdevice`, are discussed in *System Administration Guide: Volume 2 > Developing a Backup and Recovery Plan*.

`sysdevices` stores two names for each device:

- A logical name or device name, used in all subsequent storage-management commands, is stored in the `name` column of `sysdevices`. This is usually a user-friendly name, perhaps indicating the planned use for the device, for example, "logdev" or "userdbdev."
- The physical name is the actual operating system name of the device. Use this name only in the `disk init` command; after that, all SAP ASE data storage commands use the logical name.

Place a database or transaction log on one or more devices by specifying the logical name of the device in the `create database` or `alter database` statement. The `log on` clause to `create database` places a database's transaction log on a separate device to ensure full recoverability. The log device must also have an entry in `sysdevices` before you can use `log on`.

A database can reside on one or more devices, and a device can store one or more databases. See *System Administration Guide: Volume 2 > Creating and Managing User Databases* for information about creating databases on specific database devices.

6.5.2 The sysusages Table

The `sysusages` table in the `master` database keeps track of the space you assign to all SAP ASE databases.

`create database` and `alter database` allocate new space to the database by adding a row to `sysusages` for each database device or device fragment. When you allocate only a portion of the space on a device with `create` or `alter database`, that portion is called a fragment.

`sp_addsegment`, `sp_dropsegment`, and `sp_extendsegment` change the `segmap` column in `sysusages` for the device that is mapped or unmapped to a segment. See *System Administration Guide: Volume 2 > Creating and Using Segments*.

6.5.3 The syssegments Table

The `syssegments` table, one in each database, lists the segments in a database.

A segment is a collection of the database devices and fragments available to a particular database. Tables and indexes can be assigned to a particular segment—and therefore to a particular physical device—or can span a set of physical devices.

`create database` makes default entries in `syssegments`. `sp_addsegment` and `sp_dropsegment` to add and remove entries from `syssegments`.

6.5.4 The sysindexes Table

The `sysindexes` table lists each table and index and the segment where each table, clustered index, nonclustered index, and chain of text pages is stored. It also lists other information, such as the `max_rows_per_page` setting for the table or index.

The `create table`, `create index`, and `alter table` commands create new rows in `sysindexes`. Partitioning a table changes the function of `sysindexes` entries for the table.

6.5.5 The syspartitions Table

The `syspartitions` table lists each table and index partition and the segment where the partition is stored.

`syspartitions` maintains key storage management information such as the first page of a data or index page chain, the last page of a heap, the root page of an index partition, and so on.

Use `create table`, `create index` and `alter table` to create new rows in `syspartitions`.

6.6 Setting Errorlog Thresholds

Use the `errorlog size` configuration parameter to set the threshold for the size of the error log. Once this threshold is reached, SAP ASE closes the current error log and creates a new one.

The `errorlog size` configuration parameter allows you to manage the disk space. You can discard the older log files or archive them. By default `errorlog size` is disabled, and is enabled by setting the threshold to a size between 1MB and 200GB. For example, this sets the threshold to 100MB:

```
sp_configure 'errorlog size', 100MB
```

Once you enable the error log threshold, SAP ASE checks the size of its error log every minute, and if it exceeds the threshold, it creates a new errorlog with the current timestamp as its extension and renaming the original error log by appending a slightly older timestamp as its extension. If the previous error log already has a timestamp as an extension (meaning it is not the one the server starts with), its name remains unchanged. The timestamp is in this format:

```
YearMonthDateHourMinuteSecond
```

To disable the error log threshold, issue:

```
sp_configure ‘errorlog size’, 0
```

7 Managing Remote Servers

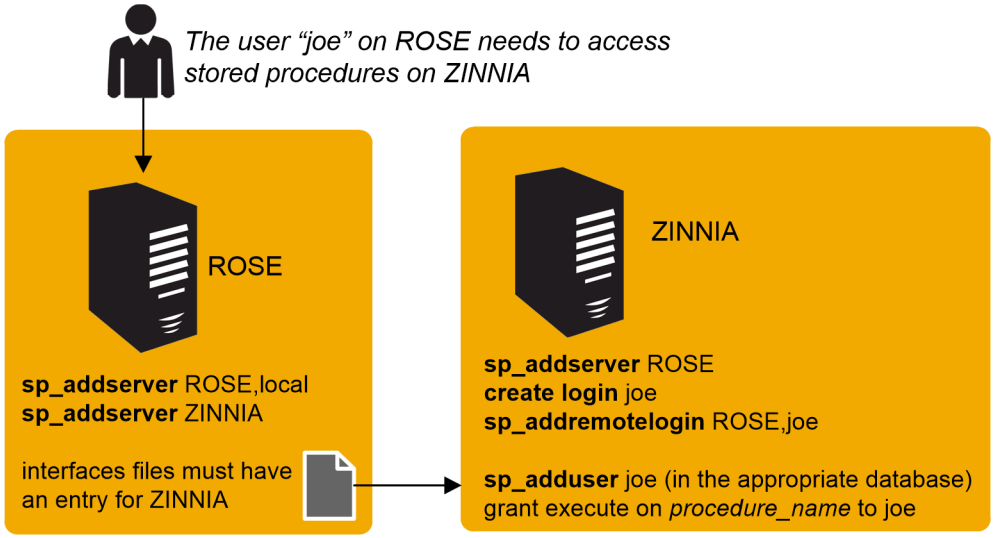
Users on a local SAP ASE can execute stored procedures on a remote SAP ASE. Executing an remote procedure calls (RPC) sends the results of the remote process to the calling process, which usually appears on the user's screen.

Users on a local SAP ASE can execute stored procedures on a remote SAP ASE. Executing an RPC sends the results of the remote process to the calling process, which usually appears on the user's screen.

To enable RPCs, the system administrator and system security officer of each SAP ASE must execute the following steps:

- On the local server:
  - System security officer – use `sp_addserver` to list the local server and remote server in the system table `master..sysservers`.
  - List the remote server in the `interfaces` file or directory service for the local server.
  - Restart the local server so the global variable `<@@servername >`is set to the name of the local server. If this variable is not set properly, users cannot execute RPCs from the local server on any remote server.
- On the remote server:
  - System security officer – use `sp_addserver` to list the server originating the RPC in the system table `master..sysservers`.
  - To allow the user who is originating the remote procedure access to the server, a system security officer uses `create login`, and a system administrator uses `sp_addremotelogin`.
  - Add the remote login name as a user of the appropriate database and grant that login permission to execute the procedure. (If `execute` permission is granted to "public," the user does not need to be granted specific permission.)

The figure below shows how to set up servers for remote access.



For operating-system-specific information about handling remote servers, see the installation documentation for your platform.

7.1 Adding Remote Logins

The system security officer and system administrator of any SAP ASE share control over which remote users can access the server, and the identity that remote users assume.

The system administrator uses `sp_addremotelogin` to add remote logins and `sp_dropremotelogin` to drop remote logins. The system security officer uses `sp_remoteoption` to control whether password checking is required.

7.1.1 Map Users' Server IDs

You can map logins from a remote server to a local server.

You can map:

- A particular remote login to a particular local login name. For example, user "joe" on the remote server might be mapped to "joesmith".
- All logins from one remote server to one local name. For example, all users sending remote procedure calls from the MAIN server might be mapped to "remusers".
- All logins from one remote server to use their remote names.

The first option can be combined with the other two options, and its specific mapping takes precedence over the other two more general mappings. The second and third options are mutually exclusive; you can use either of them, but not both.

Changing the mapping option

Use `sp_dropremotelogin` to remove the old mapping.

Use `sp_addremotelogin` to add remote logins:

```
sp_addremotelogin <remoteserver> [, <loginame>
     [, <remotename>]]
```

If the local names are not listed in `master..syslogins`, use `create login` to add them as SAP ASE logins before you add the remote logins.

Only a system administrator can execute `sp_addremotelogin`. See the *Reference Manual: Procedures*.

7.1.2 Map Remote Logins to Particular Local Names

You can map remote logins to specific local names.

This example maps the login "pogo" from a remote system to the local login "bob".

The user logs in to the remote system as "pogo". When "pogo" executes remote procedure calls from GATEWAY, the local system maps the remote login name to "bob".

```
create login bob with password itsA8secret
sp_addremotelogin GATEWAY, bob, pogo
```

7.1.3 Map All Remote Logins to One Local Name

You can create entries that map all remote logins to a single name.

This example creates an entry that maps all remote login names to the local name "albert". All names are mapped to "albert", except those with specific mappings. For example, if you mapped "pogo" to "bob", and then the rest of the logins to "albert", "pogo" still maps to "bob".

```
create login albert with password itsA8secret
sp_addremotelogin GATEWAY, albert
```

If you use `sp_addremotelogin` to map all users from a remote server to the same local name, use `sp_remoteoption` to specify the "trusted" option for those users. For example, if all users from server GATEWAY that are mapped to "albert" are to be trusted, specify:

```
sp_remoteoption GATEWAY, albert, NULL, trusted, true
```

If you do not specify logins as trusted, they cannot execute RPCs on the local server unless they specify passwords for the local server when they log in to the remote server. Users can run `ct_remote_pwd` to specify a password for server-to-server connections when they use Open Client Client-Library. `isql` and `bcp` do not permit users to specify a password for RPC connections.

> ⚠ Caution
>
> Do not map more than one remote login to a single local login, as it reduces individual accountability on the server. Audited actions can be traced only to the local server login, not to the individual logins on the remote server.

If you are using network-based security

If users are logged in to the remote server using unified login, the logins must be designated as trusted on the local server, or they must specify passwords for the server when they log in to the remote server.

> ⚠ Caution
>
> Using the `trusted` mode of `sp_remoteoption` reduces the security of your server, as passwords from such "trusted" users are not verified.

7.1.4 Keeping Remote Login Names for Local Servers

You can retain remote login names for your local server.

Procedure

1. Use `create login` to create a login for each login from the remote server.
2. Use `sp_addremotelogin` for the server to create an entry in `master..sysremotelogins` with a null value for the remote login name and a value of -1 for the `suid`. For example:

   ```
   sp_addremotelogin GATEWAY
   ```

7.1.5 Example of Remote User Login Mapping

The remote user mapping procedures and the ability to set permissions for individual stored procedures give you control over which remote users can access local procedures.

For example, you can allow the "vp" login from the CORPORATE server to execute certain local procedures and all other logins from CORPORATE to execute the procedures for which the "admin" login has permission.

> ℹ Note
>
> Typically, the passwords for users on the remote server must match passwords on the local server.

This statement displays the local and remote server information recorded in `master..sysservers`:

```
select srvid, srvname from sysservers
```

```
srvid  srvname
-----  ----------
    0  SALES
    1  CORPORATE
    2  MARKETING
    3  PUBLICATIONS
    4  ENGINEERING
```

The SALES server is local. The other servers are remote.

This statement displays information about the remote servers and users stored in `master..sysremotelogins`:

```
select remoteserverid, remoteusername, suid
  from sysremotelogins
```

```
remoteserverid   remoteusername   suid
--------------   --------------   ------
1                joe              1
1                nancy            2
1                NULL             3
3                NULL             4
4                NULL             -1
```

By matching the value of `remoteserverid` in this result and the value of `srvid` in the previous result, you can find the name of the server for which the `remoteusername` is valid. For example, in the first result, `srvid` 1 indicates the CORPORATE server; in the second result, `remoteserverid` 1 indicates that same server. Therefore, the remote user login names "joe" and "nancy" are valid on the CORPORATE server.

The following statement shows the entries in `master..syslogins`:

```
select suid, name from syslogins
```

```
suid    name
------   ------------
    1   sa
    2   vp
    3   admin
    4   writer
```

The results of all three queries together show:

- The remote user name "joe" (`suid` 1) on the remote CORPORATE server (`srvid` and `remoteserverid` 1) is mapped to the "sa" login (`suid` 1).
- The remote user name "nancy" (`suid` 2) on the remote CORPORATE server (`srvid` and `remoteserverid` 1) is mapped to the "vp" login (`suid` 2).
- The other logins from the CORPORATE server (`remoteusername` "NULL") are mapped to the "admin" login (`suid` 3).
- All logins from the PUBLICATIONS server (`srvid` and `remoteserverid` 3) are mapped to the "writer" login (`suid` 4).
- All logins from the ENGINEERING server (`srvid` and `remoteserverid` 4) are looked up in `master..syslogins` by their remote user names (`suid` -1).
- There is no `remoteserverid` entry for the MARKETING server in `sysremotelogins`. Therefore, users who log in to the MARKETING server cannot run remote procedure calls from that server.

7.2 Password Checking for Remote Users

A system security officer can use `sp_remoteoption` to determine whether passwords are checked when remote users log in to the local server.

By default, passwords are verified (this is the "untrusted" mode). In `trusted` mode, the local server accepts remote logins from other servers and front-end applications without user-access verification for the particular login.

When `sp_remoteoption` is used with arguments, it changes the mode for the named user:

```
sp_remoteoption [<remoteserver>, <loginame>, <remotename>,
     <optname>, {true | false}]
```

For example, to set `trusted` mode for the user "bob", enter

```
sp_remoteoption GATEWAY, pogo, bob, trusted,
    true
```

7.2.1 Effects of Using the Untrusted Mode

The effects of the untrusted mode depend on the user's client program.

`isql` and some user applications require that logins have the same password on the remote server and the local server. You can write Open Client applications to allow local logins to have different passwords on different servers.

To change your password in "untrusted" mode, you must first change it on all the remote systems you access before you can change it on your local server. If you change your password on the local server first, when you issue the remote procedure call to execute `alter login` on the remote server, your passwords no longer match.

The syntax for changing your password on the remote server is:

```
<remote_server>...alter login <login_name >
    with password, <caller_passwd>
    modify password [immediately] <new_password> ]
```

See *Security Administration Guide > Managing SAP ASE Logins and Database Users*.

7.3 Getting Information About Remote Logins

`sp_helpremotelogin` displays information about the remote logins on a server.

This example shows the remote login "pogo" mapped locally to login name "bob", with all other remote logins keeping their remote names:

```
sp_helpremotelogin
```

```
server     remote_user_name     local_user_name      options
---------    ----------------     -----------------    --------
GATEWAY      **mapped locally**   **use local name**    untrusted
GATEWAY      pogo                 bob                   untrusted
```

7.4 Configuration Parameters for Remote Logins

Some configuration parameters affect RPCs.

This table lists the configuration parameters that affect RPCs. Configuration parameters are set using `sp_configure`, and most on this list do not take effect until you restart SAP ASE.

Configuration parameter	Default
`allow remote access`	1
`number of remote logins`	20
`number of remote sites`	10
`number of remote connections`	20
`remote server pre-read packets`	3

8 Initializing Database Devices

A database device stores the objects that make up databases.

The term device does not necessarily refer to a distinct physical device: it can refer to any piece of a disk (such as a disk partition) or a file in the file system that is used to store databases and their objects.

Each database device or file must be prepared and made known to SAP ASE before it can be used for database storage. This process is called initialization.

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

- Allocated to the default pool of devices for the `create` and `alter database` commands
- Assigned to the pool of space available to a user database
- Assigned to a user database and used to store one or more database objects
- Assigned to store a database's transaction logs

8.1 Using the disk init command

System administrators use the disk init command to initialize new database devices with the `disk init` command.

The `disk init` command:

- Maps the specified physical disk device or operating system file to a database device name
- Lists the new device in `master..sysdevices`
- Prepares the device for database storage

> **i** Note
>
> Before you run `disk init`, see the installation documentation for your platform for information about choosing a database device and preparing it for use with SAP ASE. You may want to repartition the disks on your computer to provide maximum performance for your SAP databases.

`disk init` divides the database devices into allocation units, groups of 256 logical pages. The size of the allocation unit depends on which logical page size your server is configured for (2, 4, 8, or 16K). In each allocation unit, the `disk init` command initializes the first page as the allocation page, which contains information about the database (if any) that resides on the allocation unit.

> **⚠** Caution
>
> After you run the `disk init` command, dump the `master` database. This makes recovery easier and safer in case `master` is damaged. See *System Administration Guide: Volume 2 > Restoring the System Databases*.

The `disk init` command initializes a physical device or file and makes it usable by SAP ASE.

See the *Reference Manual: Commands* for the `disk init` syntax.

8.2 Getting Information about Devices

`sp_helpdevice` provides information about the devices in the `sysdevices` table.

When used without a device name, `sp_helpdevice` lists all the devices available on SAP ASE. When used with a device name, it lists information about that device. Here, `sp_helpdevice` is used to report information about the master device:

```
sp_helpdevice master
device_name  physical_name  description
-----------  -------------- ------------------------------------------
master        d_master         special, default disk, physical disk, 30 MB
status      cntrltype    vdevno    vpn_low   vpn_high
------      ----------   -------    -------   ---------
3           0            0          0         10239
```

Each row in `master..sysdevices` describes:

- A dump device (tape, disk, or file) to be used for backing up databases, or
- A database device to be used for database storage.

The initial contents of `sysdevices` are operating-system-dependent. `sysdevices` entries usually include:

- One for the master device
- One for the `sybsystemprocs` database, which you can use to store additional databases such as `pubs2` and `sybsyntax`, or for user databases and logs
- Two for tape dump devices

If you installed auditing, there is a separate device for `sybsecurity`.

The `vpn_low` and `vpn_high` columns represent the page numbers that have been assigned to the device. For dump devices, these columns represent the media capacity of the device.

The `status` field indicates the type of device, whether a disk device is used as a default storage device when users issue a `create` or `alter database` command without specifying a database device, disk mirroring information, and `dsync` settings.

Table 7: Status bits in sysdevices

Bit	Meaning
1	Default disk (may be used by any `create` or `alter database` command that does not specify a location)
2	Physical disk
4	Logical disk (not used)
8	Skip header (used with tape dump devices)
16	Dump device
32	Serial writes
64	Device mirrored

Bit	Meaning
128	Reads mirrored
256	Secondary mirror side only
512	Mirror enabled
2048	Used internally; set after `disk unmirror, side = retain`
4096	Primary device needs to be unmirrored (used internally)
8192	Secondary device needs to be unmirrored (used internally)
16384	UNIX file device uses `dsync` setting (writes occur directly to physical media)

For more information about dump devices and `sp_addumpdevice`, See *System Administration Guide: Volume 2 Developing a Backup and Recovery Plan*.

8.3 Dropping devices

Use `sp_dropdevice` to drop database and dump devices.

The syntax is:

```
sp_dropdevice< logicalname>
```

You cannot drop a device that is in use by a database. You must drop the database first.

`sp_dropdevice` removes the device name from `sysdevices`. `sp_dropdevice` does not remove an operating system file; it only makes the file inaccessible to SAP ASE. Use operating system commands to delete a file after using `sp_dropdevice`.

8.4 Designating Default Devices

To create a pool of default database devices to be used by all SAP ASE users for creating databases, use `sp_diskdefault` after the devices are initialized.

`sp_diskdefault` marks these devices in `sysdevices` as default devices. Whenever users create (or alter) databases without specifying a database device, new disk space is allocated from the pool of default disk space.

The syntax for `sp_diskdefault` is:

```
sp_diskdefault <logicalname>, {defaulton | defaultoff}
```

After adding user devices, use the `defaultoff` option to remove the master device from the pool of default space:

```
sp_diskdefault master, defaultoff
```

The following designates `sprocdev`, the device that holds the `sybsystemprocs` database, a default device:

```
sp_diskdefault sprocdev, defaulton
```

SAP ASE can have multiple default devices. They are used in the order in which they appear in the `sysdevices` table (that is, alphabetical order). When the first default device is filled, the second default device is used, and so on.

> **i** Note
>
> After initializing a set of database devices, you may want to assign them to specific databases or database objects rather than adding them to the default pool of devices. For example, you may want to make sure a table never grows beyond the size of a particular device.

8.4.1 Choosing Default and Nondefault Devices

`sp_diskdefault` lets you plan space usage for performance and recovery, while allowing users to create or alter databases.

Do not use these devices as default devices:

- The master device
- The device used for `sybsecurity`
- Any device intended solely for logs
- Devices where high-performance databases reside

You can use the device that holds `sybsystemprocs` for other user databases.

> **i** Note
>
> If you are using disk mirroring or segments, exercise caution in deciding which devices you add to the default list. In most cases, devices that are to be mirrored, or databases that contain objects placed on segments should specifically allocate devices, rather than being made part of default storage.

8.5 Increasing the Size of Devices with disk resize

The `disk resize` command allows you to increase the size of your database devices dynamically, rather than initializing a new device.

For example, if `/SAP_ASE/testdev.dat` requires an additional 10MB of space, you can run `disk resize` and allocate this amount of space to the device. The `create` and `alter database` commands can use this added space.

Use `disk resize` to increase the size for both devices on raw partitions and for file systems. The minimum amount of space by which you can increase a device is 1MB or an allocation unit, whichever is greater.

Page Size	Allocation Unit Size	Minimum Incremental Size
2K	0.5MB	1MB
4K	1MB	1MB
8K	2MB	2MB
16K	4MB	4MB

You cannot use `disk resize` on dump or load devices.

Any properties that are set on the device continue to be set after you increase its size. That is, if a device has `dsync` set before you increase its size, it has `dsync` set afterwards. Also, any access rights that were set before you increased the size of the device remain set.

A user with the `sa_role` can execute the `disk resize` command, which:

- Updates the high value in `master....sysdevices`, and
- Prepares the additional space for database storage.

Use audit trails on `disk resize` to track the number of times a device is resized. The device being resized is always online and available for users during the resize operation.

Resizing a disk requires that:

- You have already initialized the device with `disk init`.
- `<device_name>` must refer to a valid logical device name.
- You disable mirroring while the resize operation is in progress. You can reestablish mirroring when the resize operation is complete.

In this example, the configuration of the device `testdev` is:

```
sp_helpdevice testdev
device_name  physical_name          description
  status  cntrltype  vdevno          vpn_low          vpn_high
-----------  -------------------    -------------
  ------- ---------  -------------   -------------     --------------
testdev      /SAP_ASE/dev/testdev.dat   special, dsync on, directio off,
physical disk, 10.00MB
  16386    0          1                    0             5119
```

To increase the size of `testdev` by 4MB using `disk resize`, enter:

```
disk resize
name = "test_dev",
size = "4M"
```

`testdev.dat` is now 14MB:

```
sp_helpdevice testdev
device_name  physical_name          description
  status  cntrltype  vdevno          vpn_low          vpn_high
-----------  -------------------    -------------
  ------- ---------  -------------   -------------     --------------
```

```
testdev      /SAP_ASE/dev/testdev.dat   special, dsync on, directio off,
physical disk, 14.00MB
  16386           0         1          0                 7167
```

See the *Reference Manual: Commands* for `disk resize` syntax.

8.5.1 Insufficient Disk Space

During the physical initialization of the disk, if an error occurs due to insufficient disk space, `disk resize` extends the database device to the largest size possible.

For example, on a server that uses 4K logical pages, if you try to increase the size of the device by 40MB, but only 39.5MB is available, the device is extended only by 39.5MB.

You cannot use `disk resize` to decrease the size of a device.

8.6 Restoring Database Devices with disk reinit

`disk reinit` allows you to re-create a database device that has been removed (planned or unplanned) from the operating system when you include only the `name` parameter.

Prerequisites

The device you are re-creating cannot be active when you issue `disk reinit`. The server issues this error message if you attempt to rebuild an active device:

```
Unable to execute 'DISK REINIT' on '<device_name>' because it's still active.
Please,
drop the device and recreate it using DISK INIT.
```

You must deactivate the device by stopping and restarting SAP ASE before you re-create the device. The device must already exist in `sysdevices`.

Procedure

1. Verify the device is listed in `sysdevices` (this example is verifying that `user_disk.dat` is listed):

```
select * from sysdevices
        low  high
        status
        cntrltype
         name
         phyname
```

```
          mirrorname
          vdevno      crdate
          resizedate  tatus2     class
         ----------- ----------- ------ ---------
          ------------------------------------------------------------
          ------------------------------------------------------------------
          ----------- -------------------------------
          ------------------------------ ----------- -----
. . .
            0        5119      2         0
          user_disk
          /sybase/Devices/user_disk.dat
. . .
```

2. If necessary, stop and restart SAP ASE to make sure the device is not active.
3. Issue `disk reinit` to rebuild the device (this example rebuilds the `user_disk` device):

```
disk reinit name="user_disk"
```

8.7 Shrinking a Device

SAP ASE allows you to shrink a device, freeing up requested space to reuse or delete.

Use `sp_shrink` to shrink a device to:

- Clear the specified space and truncate the device.
- Move the data from the specified device to other eligible devices, then drop the device once the data is completely moved off this device.
- Truncate the free space at the end of a device.

The syntax is:

```
sp_shrink 'device', <device_name> [, {'size' | 'drop'}]
```

This example shrinks a device named `dev1` by 5 GB:

```
sp_shrink 'device', 'dev1', '5G'
```

Alternatively, this example moves databases off a device named `mydev` and drop the device:

```
sp_shrink 'device', 'mydev', 'drop'
```

This example truncates free space in a device named `datadev1`:

```
sp_shrink 'device', 'datadev1'
```

See *Reference Manual: Procedures* for more information about `sp_shrink`.

9 Setting Database Options

Use `sp_dboption` to change settings, which remain in effect until they are changed, for an entire database.

`sp_dboption`:

- Displays a complete list of the database options when it is used without a parameter
- Changes a database option when used with parameters

You can change options only for user databases. You cannot change options for the `master` database. To change a database option in a user database (or to display a list of the database options), execute `sp_dboption` while using the `master` database.

The syntax is:

```
sp_dboption [<dbname>, <optname>, {true | false}]
```

> **i** Note
>
> Changes to `model`'s database options do not affect `tempdb` or current user-defined multiple temporary databases when you restart SAP ASE. These changes appear only in databases that you create after you change the `model` database. Restarting SAP ASE clears objects and data contained in the temporary databases, but does not reset database options.

9.1 Database Option Descriptions

All users with access to the `master` database can execute `sp_dboption` with no parameters to display a list of the database options.

The report from `sp_dboption` looks like this:

```
sp_dboption
Settable database options.
--------------------
abort tran on log full
allow nulls by default
allow wide dol rows
async log service
auto identity
dbo use only
ddl in tran
delayed commit
enforce dump tran sequence
full logging for all
full logging for alter table
full logging for reorg rebuild
full logging for select into
identity in nonunique index
no chkpt on recovery
no free space acctg
```

```
read only
select into/bulkcopy/pllsort
single user
trunc log on chkpt
trunc. log on chkpt.
unique auto_identity index
```

For a report on which options have been set in a particular database, execute `sp_helpdb` in that database.

See the *Reference Manual: Procedures* for information about each database option in detail.

9.2 Viewing the Options on a Database

Use `sp_helpdb` to determine the options that are set for a particular database. `sp_helpdb` lists each active option in the "status" column of its output.

The following example shows that the `read only` option is turned on in `mydb`:

```
sp_helpdb mydb
```

```
name           db_size  owner  dbid   created          status
-----          -------  -----  ----   ------------     -------------------
mydb           20.0 MB    sa     5    Mar 05, 2005     read only
device_fragments      size    usage             created        free kbytes
-----------------     ------  ------------      --------       -----------
master                10.0 MB  data and log      Mar 05 2005          1792
device                         segment
------------------------------ ------------------------------
master                         default
master                         logsegment
master                         system
```

To display a summary of the options for all databases, use `sp_helpdb` without specifying a database:

```
sp_helpdb
```

```
name            db_size  owner dbid   created           status
-------------   -------- ----- ----   ------------      -------------------
master          48.0 MB   sa      1   Apr 12, 2005      mixed log and data
model            8.0 MB   sa      3   Apr 12, 2005      mixed log and data
pubs2           20.0 MB   sa      6   Apr 12, 2005      select into/
     bulkcopy/pllsort, trunc log on chkpt, mixed log and data
sybsystemdb      8.0 MB   sa      5   Apr 12, 2005      mixed log and data
sybsystemprocs 112.0 MB   sa      4   Apr 12, 2005      trunc log on chkpt,
     mixed log and data
tempdb           8.0 MB   sa      2   Apr 12, 2005      select into/
     bulkcopy/pllsort, trunc log on chkpt, mixed log and data
```

9.3 Displaying Currently Set Switches with sysoptions

The `number` column of `sysoptions` contains the switch ID for currently set switches.

`sysoptions` shows these switches:

- Trace flag set in the runserver file with the `-T` flag
- Trace flag set with `dbcc traceon(flag_number)` or `set switch serverwide on`
- Trace flags and switches set for a specific system process ID (spid) with `set switch on`

`sysoptions` shows only the switches that are visible to the user. Users cannot see switches set privately by other spids. The value for `number` is `Null` for all option categories other than switches.

`sysoptions` displays this information for switches:

- `spid` – spid for the current session.
- `name` – name of the switch. If an unnamed switch is set, `name` contains the number of the switch converted to a string.
- `category` – specifies the string `Switch`.
- `currentsetting` – set to 1 if the switch is set, 0 if the switch is not set.
- `defaultsetting` – contains 0.
- `scope` – indicates whether the switch is set server-wide or for the session. Values are:
  - 0 – switch is not set.
  - 8 – switch is set server-wide.
  - 16 – switch is set privately to the current spid.
  - 24 – switch is set server-wide and privately.
- `number` – contains the switch ID as an integer.

This query displays all `sysoptions` entries for switches:

```
select * from sysoptions
where category = 'Switch'
```

This query displays the switches set for the current session:

```
select * from sysoptions
where category = 'Switch'
and currentsetting = '1'
```

This shows the `sysoptions` output after setting trace flag 3604:

```
spid       name                            category          currentsetting
  defaultsetting         scope               number
------     ----------------------------     ---------------  ---------------
      ----------------       ------------         --------------
    11           print_output_to_client             Switch                1
                     0                   8                               3604
```

10 Configuring Character Sets, Sort Orders, and Languages

SAP provides both internationalization and localization support.

Internationalization is enabling an application to support multiple languages and cultural conventions.

An internationalized application uses external files to provide language-specific information at execution time. Because it contains no language-specific code, an internationalized application can be deployed in any native language environment without code changes. A single version of a software product can be adapted to different languages or regions, conforming to local requirements and customs without engineering changes. This approach to software development saves significant time and money over the lifetime of an application.

Localization is adapting an internationalized product to meet the requirements of one particular language or region, for example Spanish, including providing translated system messages; translations for the user interface; and the correct formats for date, time, and currency. One version of a software product may have many localized versions.

SAP ASE includes the character set definition files and sort order definition files required for data processing support for the major business languages in Western Europe, Eastern Europe, the Middle East, Latin America, and Asia.

SAP Language Modules provide translated system messages and formats for Chinese (Simplified), French, German, Japanese, Korean, Brazilian Portuguese, and Spanish. By default, SAP ASE comes with U.S. English message files.

This chapter describes the available character sets and language modules and summarizes the steps necessary to change the default character set, sort order, or message language for SAP ASE.

10.1 Advantages of Internationalized Systems

Designing an application to work outside its country of origin can seem daunting. Often, programmers think that internationalizing means hard-coding dependencies based on cultural and linguistic conventions for just one country.

A better approach is to write an internationalized application: that is, one that examines the local computing environment to determine what language to use and loads files containing language-specific information at runtime.

When you use an internationalized application, a single application can be deployed in all countries. This has several advantages:

- You write and maintain one application.
- The application can be deployed, without change, in new countries as needed. You need only supply the correct localization files.
- All sites can expect standard features and behavior.

10.1.1 A Sample Internationalized System

An internationalized system may include internationalized client applications, gateways, and servers running on different platforms in different native language environments.

For example, an international system might include the following components:

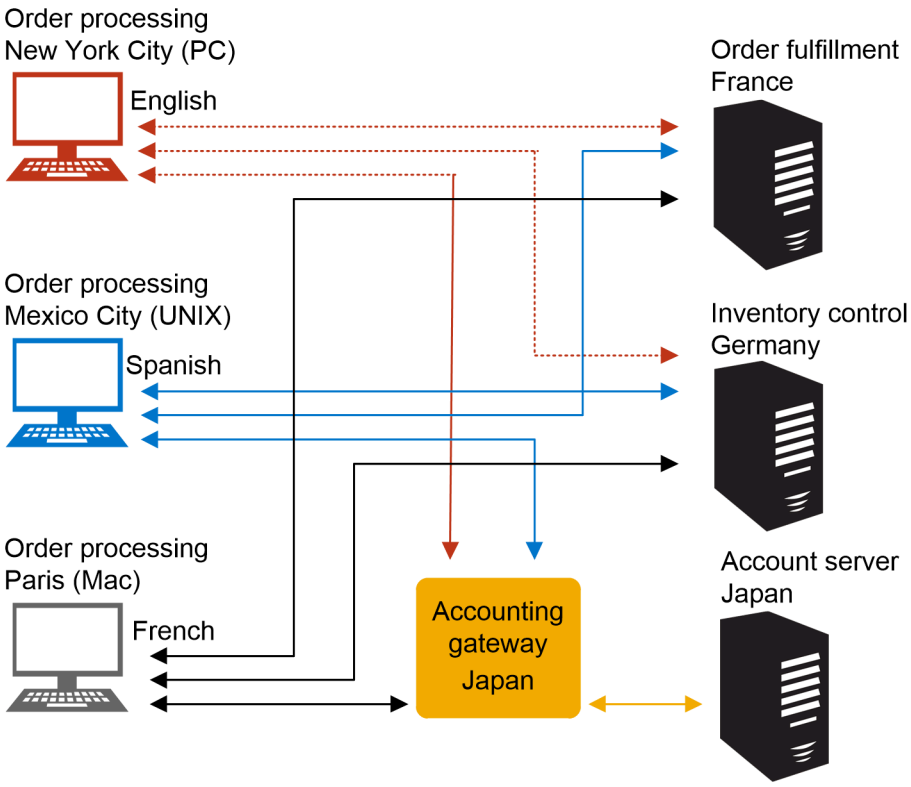
- Order processing applications in New York City, Mexico City, and Paris (Client-Library applications)
- An inventory control server in Germany (SAP ASE)
- An order fulfillment server in France (SAP ASE)
- A central accounting application in Japan (an Open Server application working with an SAP ASE)

In this system, the order processing applications:

- Query the inventory control server to determine if requested items are in stock
- Place orders with the order fulfillment server
- Send financial information to the accounting application

The inventory control server and the order fulfillment server respond to queries, and the accounting application collects financial data and generates reports.

The system looks like this:



In this example, all applications and servers use local languages and character sets to accept input and output messages.

10.1.2 Elements of an Internationalized System

In an internationalized environment, you can manipulate the character set, the sort order, and system messages to configure your server language. SAP suggests that you review each of these elements and carefully plan the client/server network you want to create.

- Character set – the language in which the server sends and receives data to and from the client servers. Select the character set after carefully planning and analyzing the language needs of all client servers.
- Sort order – sort order options are dependent on the language and character set you select.
- System messages – messages display in one of several languages provided by SAP. If your server language is not one of the languages provided, your system messages display in English, the default.

10.2 Selecting the Character Set for Your Server

In your server, all data is encoded in a special code. For example, the letter "a" is encoded as "97" in decimal. A character set is a specific collection of characters (including alphabetic and numeric characters, symbols, and nonprinting control characters) and their assigned numerical values, or codes.

A character set generally contains the characters for an alphabet, for example, the Latin alphabet used in the English language, or a script such as Cyrillic used with languages such as Russian, Serbian, and Bulgarian. Character sets that are platform-specific and support a subset of languages, for example, the Western European languages, are called native or national character sets. All character sets that come with SAP ASE, except for Unicode UTF-8, are native character sets.

A script is a writing system, a collection of all the elements that characterize the written form of a human language—for example, Latin, Japanese, or Arabic. Depending on the languages supported by an alphabet or script, a character set can support one or more languages. For example, the Latin alphabet supports the languages of Western Europe (see table below). On the other hand, the Japanese script supports only one language, Japanese. Therefore, the Group 1 character sets support multiple languages, while many character sets, such as those in Group 101, support only one language.

The language or languages that are covered by a character set is called a language group. A language group can contain many languages or only one language; a native character set is the platform-specific encoding of the characters for the language or languages of a particular language group.

Within a client/server network, you can support data processing in multiple languages if all the languages belong to the same language group (see the table below). For example, if data in the server is encoded in a Group 1 character set, you could have French, German, and Italian data and any of the other Group 1 languages in the same database. However, you cannot store data from another language group in the same database. For example, you cannot store Japanese data with French or German data.

Unlike the native character sets just described, Unicode is an international character set that supports over 650 of the world's languages, such as Japanese, Chinese, Russian, French, and German. Unicode allows you to mix different languages from different language groups in the same server, no matter what the platform.

Since all character sets support the Latin script, and therefore English, a character set always supports at least two languages—English and one other language.

Many languages are supported by more than one character set. The character set you install for a language depends on the client's platform and operating system.

Table 8: Supported languages and character sets

Language group	Languages	Character sets
Group 1	*Western European*: Albanian, Catalan, Danish, Dutch, English, Faeroese, Finnish, French, Galician, German, Icelandic, Irish, Italian, Norwegian, Portuguese, Spanish, Swedish	ASCII 8, CP 437, CP 850, CP 860, CP 863, CP 1252 , ISO 8859-1, ISO 8859-15, Macintosh Roman, ROMAN8, ROMAN9, ISO-15, CP 858 CP 1252 is identical to ISO 8859-1 except for the 0x80–0x9F code points which are mapped to characters in CP 1252.
Group 2	*Eastern European*: Croatian, Czech, Estonian, Hungarian, Latvian, Lithuanian, Polish, Romanian, Slovak, Slovene (and English)	CP 852, CP 1250, ISO 8859-2, Macintosh Central European
Group 4	Baltic (and English)	CP 1257
Group 5	*Cyrillic*: Bulgarian, Byelorussian, Macedonian, Russian, Serbian, Ukrainian (and English)	CP 855, CP 866, CP 1251, ISO 8859-5, Koi8, Macintosh Cyrillic
Group 6	Arabic (and English)	CP 864, CP 1256, ISO 8859-6
Group 7	Greek (and English)	CP 869, CP 1253, GREEK8, ISO 8859-7, Macintosh Greek
Group 8	Hebrew (and English)	CP 1255, ISO 8859-8
Group 9	Turkish (and English)	CP 857, CP 1254, ISO 8859-9, Macintosh Turkish, TURKISH8
Group 101	Japanese (and English)	CP 932 DEC Kanji, EUC-JIS, Shift-JIS
Group 102	Simplified Chinese (PRC) (and English)	CP 936, EUC-GB, GB18030
Group 103	Traditional Chinese (ROC) (and English)	Big 5, CP 950, EUC-CNS, Big 5 HKSCS CP 950 is identical to Big 5.
Group 104	Korean (and English)	EUC-KSC, cp949
Group 105	Thai (and English)	CP 874, TIS 620
Group 106	Vietnamese (and English)	CP 1258
Unicode	Over 650 languages	UTF-8

> **i Note**
>
> The English language is supported by all character sets because the first 128 (decimal) characters of any character set include the Latin alphabet (defined as “ASCII-7”). The characters beyond the first 128 differ between character sets and are used to support the characters in different native languages. For example, code points 0-127 of CP 932 and CP 874 both support English and the Latin alphabet. However, code points

128-255 support Japanese characters in CP 932 and code points 128-255 support Thai characters in CP 874.

The following character sets support the European currency symbol, the "euro": CP 1252 (Western Europe); CP 1250 (Eastern Europe); CP 1251 (Cyrillic); CP 1256 (Arabic); CP 1253 (Greek); CP 1255 (Hebrew); CP 1254 (Turkish); CP 874 (Thai); iso15, roman9 and CP858. Unicode UTF-8 also supports:

- Traditional Chinese on the Windows and Solaris platforms
- Arabic, Hebrew, Thai, and Russian on the Linux platform

> **i** Note
>
> iso_1 and ISO 8859-1 are different names for the same character set.

To mix languages from different language groups you *must* use Unicode. If your server character set is Unicode, you can support more than 650 languages in a single server and mix languages from any language group.

10.2.1 Unicode

Unicode enables all the world's languages to be encoded in the same data set.

Prior to the introduction of Unicode, if you wanted to store data in, for example, Chinese, you had to choose a character set appropriate for that language—to the exclusion of most other languages. It was either impossible or impractical to mix character sets, and thus diverse languages, in the same data set.

SAP supported Unicode in the form of three datatypes: `unichar`, `univarchar`, and `unitext`. These datatypes store data in the UTF-16 encoding of Unicode.

UTF-16 is an encoding wherein Unicode scalar values are represented by a single 16-bit value (or, in rare cases, as a pair of 16-bit values). The three encodings are equivalent insofar as either encoding can be used to represent any Unicode character. The choice of UTF-16 datatypes, rather than a UTF-16 server default character set, promotes easy, step-wise migration for existing database applications.

SAP ASE supports Unicode literals in SQL queries and a wide range of sort orders for UTF-8.

The character set model used by SAP ASE is based on a single, configurable, server-wide character set. All data stored in SAP ASE, using any of the "character" datatypes (`char`, `varchar`, `nchar`, `nvarchar`, and `text`), is interpreted as being in this character set. Sort orders are defined using this character set, as are language modules—collections of server messages translated into local languages.

During the connection dialog, a client application declares its native character set and language. If properly configured, the server thereafter attempts to convert any character data between its own character set and that of the client (character data includes any data stored in the database, as well as server messages in the client's native language).This works well as long as the server's and client's character sets are compatible. It does not work well when characters are not defined in the other character set, as is the case for the character sets SJIS, used for Japanese, and KOI8, used for Russian and other Cyrillic languages. Such incompatibilities are the reason for Unicode, which can be thought of as a character superset, including definitions for characters in all other character sets.

The Unicode datatypes `unichar`, `univarchar`, and `unitext` are completely independent of the traditional character set model. Clients send and receive Unicode data independently of whatever other character data they send and receive.

10.2.1.1 Configuration Parameters

The UTF-16 encoding of Unicode includes "surrogate pairs," which are pairs of 16-bit values that represent infrequently used characters.

Additional checking is built in to SAP ASE to ensure the integrity of surrogate pairs. You can switch this checking off by setting the `enable surrogate processing` configuration parameter to 0. This yields slightly higher performance, although the integrity of surrogate pairs is no longer guaranteed.

Unicode also defines "normalization," which is the process by which all possible representations of a single character are transformed into a single representation. Many base characters followed by combining diacritical marks are equivalent to precomposed characters, although their bit patterns are different. For example, the following two sequences are equivalent:

```
0x00E9  -- é (LATIN SMALL LETTER E WITH ACUTE)
```

```
0x00650301  -- e (LATIN SMALL LETTER E), ´ (COMBINING ACUTE ACCENT)
```

The `enable unicode normalization` configuration parameter controls whether or not SAP ASE normalizes incoming Unicode data.

Significant performance increases are possible when the `default Unicode sortorder` is set to "binary" and the `enable Unicode normalization` configuration parameter is set to 1. This combination allows SAP ASE to make several assumptions about the nature of the Unicode data, and code has been implemented to take advantage of these assumptions.

10.2.1.2 Functions

All functions that take `char` parameters accept `unichar` as well. Functions with more than one parameter, when called with at least one `unichar` parameter, results in implicit conversion of any non-`unichar` parameters to `unichar`.

To guarantee the integrity of surrogate pairs when `enable surrogate processing` is set to 1 (the default), the string functions do not allow surrogate pairs to be split. Positions fall at the beginning of a surrogate pair.

Several functions round out the `unichar` support. Included are the functions `to_unichar()` and `uscalar()`, which are analogous to `char()` and `ascii()`. The functions `uhighsurr()` and `ulowsurr()` allow the explicit handling of surrogate pairs in user code.

There are restrictions when using unitext with functions. For information, see the restriction description under the "Usage" section for each function in the *Reference Manual: Blocks*.

10.2.1.3 Using unichar Columns

When using the `isql` or `bcp` utilities, Unicode values display in hexadecimal form unless the `–Jutf8` flag is used, indicating the client's character set is UTF-8. In this case, the utility converts any Unicode data it receives from the server into UTF-8. For example:

```
% isql -Usa -P -Jiso_1
1> select unicode_name from people where unicode_name = 'Jones'
2> go
```

```
unicode_name
------------------------------------------------------------------|
0x004a006f006e00650073
(1 row affected)
```

whereas:

```
% isql -Usa -P -Jutf8
1> select unicode_name from people where unicode_name = 'Jones'
2> go
```

```
unicode_name
------------------------------------------------------------------
Jones
(1 row affected)
```

This facilitates ad hoc queries. Not all terminal windows are capable of displaying the full repertoire of Unicode characters, but simple tests involving ASCII characters are greatly simplified.

10.2.1.4 Using unitext

The variable-length `unitext` datatype can hold up to 1,073,741,823 Unicode characters (2,147,483,646 bytes). You can use `unitext` anywhere you use the `text` datatype, with the same semantics. `unitext` columns are stored in UTF-16 encoding, regardless of the SAP ASE default character set.

10.2.1.5 Open Client Interoperability

The Open Client libraries support the datatype `cs_unichar`, which can be bound to user variables declared as an array of short integers. This Open Client datatype interfaces directly with the server's `unichar`, `unitext`, and `univarchar`.

10.2.1.6 Java Interoperability

The internal JDBC driver efficiently transfers `unichar` data between SQL and Java contexts.

Going from SQL to Java, the class `java.sql.ResultSet` provides a number of "get" methods to retrieve data from the columns of a result set. Any of these get methods work with columns defined as `unichar`, `unitext`, or `univarchar`. The method `getString()` is particularly efficient since no conversion needs to be performed.

Use the `setString()` method of the class `java.sql.PreparedStatement` to go from Java to SQL. The internal JDBC driver copies Java string data directly into the SQL parameter defined as `unichar`, `unitext`, or `univarchar`.

The external JDBC driver (jConnect) has been modified to support the same seamless interface as the internal driver.

10.2.2 Selecting the Server Default Character Set

You must specify a default character set, which is the one in which the server stores and manipulates data, when you configure the server. Each server can have only one default character set.

By default, the installation tool assumes that the native character set of the platform operating system is the server's default character set. However, you can select any character set supported by SAP ASE as the default on your server (see the table below).

For example, if you are installing the server on IBM RS/6000 running AIX, and you select one of the Western European languages to install, the installation tool assumes the default character set to be ISO 8859-1.

If you are installing a Unicode server, select UTF–8 as your default character set.

For non-Unicode servers, determine what platform most of your client systems use and use the character set for this platform as the default character set on the server.

This has two advantages:

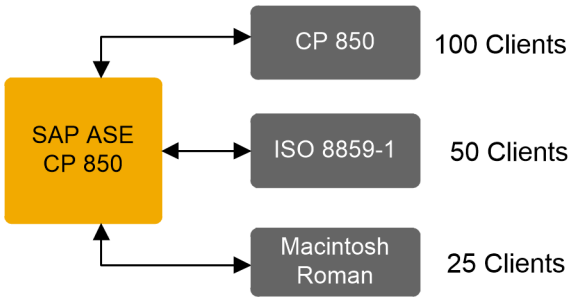
- The number of unmappable characters between character sets is minimized.
  Since there is usually not a complete one-to-one mapping between the characters in two character sets, there is a potential for some data loss. This is usually minor because most unconverted characters are special symbols that are not commonly used or are specific to a platform.
- This minimizes the character set conversion that is required.
  When the character set on the client system differs from the default character set on the server, data must be converted in order to ensure data integrity. Although the measured performance decrease that results

from character set conversion is insignificant, it is good practice to select the default character set that results in the fewest conversions.

For example, if most of your clients use CP 850, specify CP 850 on your server. You can do this even if your server is on an HP-UX system (where its native character set for the Group 1 languages is ROMAN8).

> **i Note**
>
> SAP strongly recommends that you decide which character set to use as your default before you create any databases or make any changes to the SAP-supplied databases.

In the example below, 175 clients all access the same SAP ASE. The clients are on different platforms and use different character sets. The critical factor that allows these clients to function together is that *all* of the character sets in the client/server system belong to the same language group. The default language for the SAP ASE is CP 850, which is the character set used by the largest number of clients. This allows the server to operate most efficiently, with the least amount of character set conversion.



To help you choose the default character set for your server, the following tables list the most commonly used character sets by platform and language.

Table 9: Most Widely Used Western European Client Platforms

Platform	Language	Character set
Win 95, 98	U.S. English, Western Europe	CP 1252
Windows	U.S. English, Western Europe	CP 1252
Win 2000	U.S. English, Western Europe	CP 1252
Sun Solaris	U.S. English, Western Europe	ISO 8859-1
HP-UX 10,11	U.S. English, Western Europe	ROMAN8
IBM AIX 4.x	U.S. English, Western Europe	ISO 8859-1

Table 10: Most Widely Used Japanese Client Platforms

Platform	Language	Character set
Win 95, 98	Japanese	CP 932 for Windows
Win NT 4.0	Japanese	CP 932 for Windows

Platform	Language	Character set
Win 2000	Japanese	CP 932 for Windows
Sun Solaris	Japanese	EUC-JIS
HP-UX 10,11	Japanese	EUC-JIS
IBM AIX 4.x	Japanese	EUC-JIS

Table 11: Most Widely Used Chinese Client Platforms

Platform	Language	Character set
Win 95, 98	Chinese (simplified)	CP 936 for Windows
Win NT 4.0	Chinese (simplified)	CP 936 for Windows
Win 2000	Chinese (simplified)	CP 936 for Windows
Sun Solaris	Chinese (simplified)	EUC-GB
HP-UX 10,11	Chinese (simplified)	EUC-GBS
IBM AIX 4.x	Chinese (simplified)	EUC-GB

10.3 Selecting the Sort Order

Different languages sort the same characters differently. SAP ASE uses sort orders to create indexes, store date into indexed tables, and specify an `order by` clause.

For example, in English, *Cho* is sorted before *Co*, whereas in Spanish, the opposite is true. In German, β is a single character, however in dictionaries it is treated as the double character *ss* and sorted accordingly. Accented characters are sorted in a particular order so that *aménité* comes before *amène*, whereas if you ignored the accents, the reverse would be true. Therefore, language-specific sort orders are required so that characters are sorted correctly.

Each character set comes with one or more sort orders that SAP ASE uses to collate data. A sort order is tied to a particular language or set of languages and to a specific character set. The same sort orders can be used for English, French, and German because they sort the same characters identically, for example, *A*, *a*, *B*, *b*, and so on. Or the characters are specific to one of the languages—for example, the accented characters, *é* , *à*, and *á*, are used in French but not in English or German—and therefore, there is no conflict in how those characters are sorted. The same is not true for Spanish however, where the double letters *ch* and *ll* are sorted differently. Therefore, although the same character sets support all four languages, there is one set of sort orders for English, French and German, and a different set of sort orders for Spanish.

In addition, a sort order is tied to a particular character set. Therefore, there is one set of sort orders for English, French, and German in the ISO 8859-1 character set, another set in the CP 850 character set, and so on. The sort orders available for a particular character set are located in sort order definition files (`*.srt` files) in the character set directory.

10.3.1 Different Types of Sort Orders

All character sets are offered with a binary sort order at a minimum, which blindly sorts all data based only on the arithmetic value of the code assigned to represent each letter (the "binary" code) in the character set.

Binary sort order works well for the first 128 characters of each character set (ASCII English) and for Asian languages.When a character set supports more than one language (for example, Group 1 or Unicode) the binary sort order most likely give incorrect results, and you should select another sort order.

Character sets may also have one or more of the following dictionary sort orders:

- *Dictionary order, case-sensitive, accent-sensitive* – sorts uppercase and lowercase letters separately. Dictionary order recognizes the various accented forms of a letter and sorts them after the associated unaccented letter.
- *Dictionary order, case-insensitive, accent-sensitive* – sorts data in dictionary order but does not recognize case differences. Uppercase letters are equivalent to their lowercase counterparts and are intermingled in sorting results. Useful for avoiding duplicate entries in tables of names.
- *Dictionary order, case-insensitive, accent-sensitive, order with preference* – does not recognize case difference in determining equivalency of items. A word in uppercase is equivalent to the same word in lowercase. Preference is given to uppercase letters (they appear first) if all other conditions are equal. Using case-insensitive with preference may cause poor performance in large tables when the columns specified in an `order by` clause match the key of the table's clustered index. Do not select case-insensitive order with preference unless your installation requires that uppercase letters be sorted before lowercase letters in otherwise equivalent strings for `order by` clauses.
- *Dictionary order, case-insensitive, accent-insensitive* – treats accented forms of a letter as equivalent to the associated unaccented letter. It intermingles accented letters in sorting results.

10.3.2 Selecting the Default Sort Order

SAP servers can support only one default sort order at a time. If your users all use the same language, or their languages all use the same sort order, selecting the default sort order is straightforward. .

For example, if your users are using French data and expect French sorting, then you can pick one of the French dictionary sort orders. Or if your users are using data in multiple languages and the languages use the same sort order, for example English, French, and German, you can pick one sort order and it works for all your users in all languages.

However, if you have users using different languages that require different sort orders, for example French and Spanish, then you must select one of the sort orders as the default. If you pick, for example, a French sort order, your Spanish users will not see the *ch* and *ll* double characters sorted as they would expect. The installation procedure, by default, configures the server with the binary sort order.

You can use the `sortkey` function to set up customized alternative sort orders for your data—one for each language.These sort orders can be selected dynamically to meet the needs of different users. The `sortkey` function is separate from the default sort order, but can coexist in the same server. The range and depth of sort orders provided by the `sortkey` function is better than those provided by the default sort order mechanism. For more information, see `sortkey` and `compare` in the *Reference Manual: Building Blocks*.

Table 12: Available sort orders

Language or script	Character sets	Sort orders
All languages	UTF-8	Multiple sort orders
Cyrillic: Bulgarian, Byelorussian, Macedonian, Russian, Serbian, Ukrainian	CP 855, CP 866, CP 1251, ISO 8859-5, Koi8, Macintosh Cyrillic	Dictionary order, case sensitive, accent sensitive
Eastern European: Czech, Slovak	CP 852, ISO 8859-2, CP 1250	Dictionary order, case sensitive, accent sensitive Dictionary order, case insensitive, accent sensitive Dictionary order, case sensitive, accent sensitive, with preference Dictionary order, case insensitive, accent insensitive
English, French, German	ASCII 8, CP 437, CP850, CP 860, CP 863, CP 1252a, ISO 8859-1, ISO 8859-15, Macintosh Roman, ROMAN8, ROMAN9, ISO 15	Dictionary order, case sensitive, accent sensitive Dictionary order, case insensitive, accent sensitive Dictionary order, case sensitive, accent sensitive, with preference Dictionary order, case insensitive, accent insensitive
English, French, German	CP 850, CP 858	Alternate dictionary order, case sensitive Alternate dictionary order, case sensitive, accent insensitive Alternate dictionary order, case sensitive, with preference
Greek	ISO 8859-7	Dictionary order, case sensitive, accent sensitive
Hungarian	ISO 8859-2	Dictionary order, case sensitive, accent sensitive Dictionary order, case insensitive, accent sensitive Dictionary order, case insensitive, accent insensitive
Japanese	EUCJIS, SJIS, DECKANJI	General purpose case-insensitive dictionary ordering
Kazakh	87	50
Russian	CP 866, CP 1251, ISO 8859-5, Koi8, Macintosh Cyrillic	Dictionary order, case sensitive, accent sensitive Dictionary order, case insensitive, accent sensitive
Scandinavian	CP 850	Dictionary order, case sensitive, accent sensitive Dictionary order, case insensitive, with preference
Simplified Chinese	EUC-GB, GB-18030, CP936	General purpose case-insensitive dictionary ordering

Language or script	Character sets	Sort orders
Spanish	ASCII 8, CP 437, CP850, CP 860, CP 863, CP 1252, ISO 8859-1, ISO 8859-15, Macintosh Roman, ROMAN8	Dictionary order, case sensitive, accent sensitive Dictionary order, case insensitive, accent sensitive Dictionary order, case insensitive, accent insensitive
Thai	CP 874, TIS 620	Dictionary order
Turkish	ISO 8859-9	Dictionary order, case sensitive, accent sensitive Dictionary order, case insensitive, accent insensitive Dictionary order, case insensitive, accent sensitive
Western European	CP 1252	Dictionary order, case insensitive, case sensitive, with preference, accent insensitive, Spanish dictionary, Spanish case insensitive, Spanish accent insensitive

If your language does not appear here, there is no language-specific sort order for your language. Select a binary sort order and then investigate whether the `sortkey` function meets your needs. As this table illustrates, many languages have more than one sort order.

10.3.2.1 Chinese Pinyin Sort Order

Pinyin, more formally known as "Hanyu Pinyin," uses the Roman alphabet to represent the standard Chinese pronunciation system.

Pinyin consists of a system of transliteration to Roman alphabets for reading and writing Mandarin without Chinese characters. Pinyin uses accents to represent the four tones of Mandarin.

Earlier versions of SAP ASE used the Simplified Chinese (GB) sort orders, `gbpinyin` and `gbpinyinocs`, using the Unilib character set, significantly impacting the performance of databases using the GB character sets.

SAP ASE version 15.0.3 automatically uses the `gbpinyin` and `gbpinyinocs` sort orders, eliminating a processing step and significantly improving performance.

In earlier versions, the default `size of unilib cache` configuration parameter was 268 KB. In version 15.0.3, the default has been increased to 302 KB.

Improved performance occurs in queries that access ASCII and `gbpinyin` data. However, if the data set has a mixture of other characters, you may not see any performance improvement.

10.3.2.2 Selecting Case-Insensitive Sort Orders for Chinese and Japanese Character Sets

Use the `sp_helpsort` and `sp_configure` system procedures to select case-insensitive sort orders.

`sp_helpsort` lists the available case-insensitive sort orders.

```
sp_helpsort
-----------
Name                     ID
-------------------------
nocase_eucgb             52
nocase_cp936             52
nocase_gb18030           52
nocase_eucjis            52
nocase_sjis              52
nocase_deckanji          52
```

Use `sp_configure` to switch to a case-insensitive sort order:

```
sp_configure 'default sortorder id', 52
```

10.3.2.3 Selecting the Default Unicode Sort Order

The value for `default unicode sort order` database option is different than the sort order for the server's default character set.

This separate configuration parameter is a static parameter that requires that you restart your server and reindex the `unichar` data if it is changed. This sort order is identified using a string parameter, rather than a numeric parameter, to guarantee that the sort order is unique.

This table lists the available default Unicode sort orders:

Name	ID	Description
defaultml	20	Default Unicode multilingual ordering
thaidict	21	Thai dictionary ordering
iso14651	22	Ordering as per ISO14651 standard
utf8bin	24	Ordering for UTF-16 that matches the UTF-8 binary
binary	25	Binary sort
altnoacc	39	Alternate accent-insensitive
altdict	45	Alternate dictionary ordering
altnocsp	46	Alternate case-insensitive with preference

Name	ID	Description
scandict	47	Scandinavian dictionary ordering
scannocp	48	Scandinavian case-insensitive with preference
bin_utf8	50	UTF-8 binary sort order
dict	51	General-purpose dictionary ordering
nocase	52	General-purpose case-insensitive dictionary ordering
nocasep	53	General-purpose case-insensitive with preference
noaccent	54	Dictionary order, case-insensitive, accent-insensitive
espdict	55	Spanish dictionary ordering
espnocs	56	Spanish case-insensitive dictionary ordering
espnoac	57	Spanish accent-insensitive dictionary ordering
rusnocs	59	Russian case-insensitive dictionary ordering
cyrnocs	64	Cyrillic case-insensitive dictionary ordering
elldict	65	Greek dictionary ordering
hundict	69	Hungarian dictionary ordering
hunnoac	70	Hungarian accent-insensitive dictionary ordering
hunnocs	71	Hungarian case-insensitive dictionary ordering
turknoac	73	Turkish accent-insensitive dictionary ordering
binaryalt	99	Binary sort order that matches the Business Suite (and ABAP) binary sort order

This table lists the loadable sort orders:

Name	ID	Description
cp932bin	129	Ordering that matches the binary ordering of CP932
dynix	130	Chinese phonetic ordering
gb3213bn	137	Ordering that matches the binary ordering of GB2312
cyrdict	140	Common cyrillic dictionary ordering
turdict	155	Turkish Dictionary ordering
euckscbn	161	Ordering that matches the binary ordering of EUCKSC

Name	ID	Description
gbpinyin	163	Chinese phonetic ordering
rusdict	165	Russian dictionary ordering
sjisbin	179	Ordering that matches the binary ordering of SJIS
eucjisbn	192	Ordering that matches the binary ordering of EUCJIS
big5bin	194	Ordering that matches the binary ordering of BIG5

To view this sort order list in SAP ASE, use `sp_helpsort`. See the *Reference Manual: Procedures*.

You can add sort orders using external files in the `$/collate/Unicode` directory. The names and collation IDs are stored in `syscharsets`. The names of external Unicode sort orders do not have to be in `syscharsets` before you can set the default Unicode sort order.

> **i** Note
>
> External Unicode sort orders are provided by SAP. Do not attempt to create external Unicode sort orders.

Sort order associated with Unicode data is completely independent of the sort order associated with traditional character data. All relational expressions involving the Unicode datatypes are performed using the Unicode sort order. This includes mixed-mode expressions involving Unicode and non-Unicode data. For example, in the following query the `varchar` character constant 'Mü' is implicitly cast to `unichar` and the comparison is performed according to the Unicode sort order:

```
select * from authors where unicode_name > 'Mü'
```

The same holds true for all other comparison operators, as well as the concatenation operator "+", the operator "in", and the operator "between." Once again, the goal is to retain compatibility with existing database applications.

Tables joins based on equality (equijoins) deserve special mention. These are generally optimized by the server to take advantage of indexes that defined on the participating columns. When a `unichar` column is joined with a `char` column, the latter requires a conversion, and since the character sort order and the Unicode sort order are distinct, the optimizer will ignore the index on the `char` column.

In SAP ASE version 12.5.1 and later, when the server's default character set is configured to UTF-8, you can configure the server's default sort order (for `char` data) to be any of the above sort orders. Prior to this version, the binary sort order "bin_utf8" (ID=50) was the only well-behaved sort order for UTF-8. Although not required, the sort order for `char` data in UTF-8 can be selected so that it corresponds with the sort order for `unichar`.

There is a potential confusion regarding choice of binary sort orders for Unicode. The sort order named "binary" is the most efficient one for `unichar` data (UTF-16), and is thus the default. This order is based on the Unicode scalar value, meaning that all 32-bit surrogate pairs are placed after all 16-bit Unicode values. The sort order named "utf8bin" is designed to match the order of the default (most efficient) binary order for UTF-8 `char` data, namely "bin_utf8". The recommended matching combinations are thus "binary" for `unichar` and "binary" for UTF-8 `char`, or "utf8bin" for `unichar` and "bin_utf8" for UTF-8 `char`. The former favors `unichar` efficiency, while the latter favors `char` efficiency. Avoid using "utf8bin" for UTF-8 `char`, since it is equivalent to "bin_utf8" but less efficient.

10.4 Select a Language for System Messages

Any installation of SAP ASE can use Language Modules, which contains files of messages in different languages.

SAP ASE provides Language Modules for messages in the following languages: English, Chinese (Simplified), French, German, Japanese, Korean, Brazilian Portuguese, and Spanish. If your client language is *not* one of these languages, you see system messages in English, the default language.

Each client can choose to view messages in their own language at the same time, from the same server; for example, one client views system messages in French, another in Spanish, and another in German. To do this, however, all selected languages must be part of the same language group. For example, French, Spanish and German are all part of language group 1. Japanese, on the other hand, is part of language group 101, which contains no other languages. Therefore, if Japanese is your server language, you can display system messages only in Japanese or English. Remember that all language groups can display messages in English. There is also a server-wide default language, used if the user has not selected a specific language. If you use Unicode, you can view system messages in any of the supported languages.

You can select the language for your system messages in one of two ways:

- Select a language as part of your user profile
- Enter a language in the `locales.dat` file

This table displays the supported system message languages and their language groups. Each user can select only one language per session for system messages:

Language group	System message languages	Character sets
Group 1	French, German, Spanish, Brazilian Portuguese	ASCII 8, CP 437, CP 850, CP 860, CP 863, CP 1252, ISO 8859-1, ISO 8859-15, Macintosh Roman, ROMAN8
Group 2	Polish	Cp 1250, CP 852, ISO 8859-2
Group 101	Japanese	CP 932, DEC Kanji, EUC-JIS, Shift-JIS
Group 102	Simplified Chinese (PRC)	CP 936, EUC-GB, GB18030
Group 104	Korean	EUC-KSC, CP 949
Group 105	Thai	CP 874, TIS 620
Unicode	French, German, Spanish, Brazilian Portuguese, Japanese, Simplified Chinese, Korean	UTF-8
All Other Language Groups	English	

Install Language Modules for all languages in which clients will receive messages. These Language Modules, located in the `locales` subdirectory of the SAP ASE installation directory, are part of a group of files called localization files.

10.5 A Spanish-Version Server

This server requires only a Spanish language group.

Procedure

1. Select the server language, in this case, Spanish, which is part of language group 1. Based on your platform, select a character set from language group 1. SAP recommends that you select the character set used by the greatest number of clients. Or, if you think your company might someday expand into other countries and languages, you might consider installing Unicode.
2. Install the Spanish Language Module in the server. This allows clients to view system messages in Spanish.
3. Select the default sort order. Spanish has three possible sort orders, in addition to binary sort order. Select a sort order.
4. Restart the server.

10.6 A Company Needs to Support Both English and Japanese Languages

This company needs to support both English and Japanese.

Procedure

1. Select the default character set for your server. If you install a character set from language group 101 (Japanese), you can support both Japanese and English data in the same server.
2. Install the Japanese Language Module so that system messages are available in Japanese.
3. Select the sort order. Because a binary sort order is the only sort order available for Japanese, both the English and Japanese clients have a default binary sort order. Consider using the `sortkey` function to provide solutions for both audiences.
4. Make sure that each Japanese user requests Japanese messages by default. Since you are using a character set from language group 101, and you have already installed the Japanese Language Module, your client in Japan sees messages in Japanese, while clients in the U.S. can choose to see messages in either English or Japanese.

10.7 A Japan-Based Company with Multinational Clients

This company is located in Japan, and has clients in France, Germany, and Spain, so you need to mix European and Asian languages in the same server.

Procedure

1. Select the default server language and character set. Since your company is based in Japan and most of your clients are located in Japan, the default server language should be Japanese. But you also want your clients in France, Germany, and Spain to be able to send and receive data in their native languages. Japanese is part of language group 101, while French, German, and Spanish are part of language group 1. Since the languages you need are not part of the same language group, the only way you can have all of these languages on the same server is to select Unicode as your default character set.
2. Install the Language Modules for Japanese, French, German, and Spanish.
3. Select the binary sort order, since this is the only sort order available for the Unicode character set. (You can, however, consider using the `sortkey` function inside your application code to supply data sorted according to each user's preference.)
4. Select Japanese as the default language for system messages. Clients in other countries can select their own native language for messages.

10.8 Changing the character set, sort order, or message language

Even after you have configured your server, a system administrator can change the default character set, sort order, or message language used by SAP ASE.

Context

Because a sort order is built on a specific character set, changing character sets always involves a change in sort order. However, you can change the sort order without changing character sets, because more than one sort order may be available for a character set.

> **i** Note
>
> You cannot change the SAP ASE default character set and sort order if it includes an archived database.

Procedure

To display SAP ASE's default sort order, character set, and a table of its primary sort orders, enter:

```
sp_helpsort
```

10.8.1 Changing the default character set

SAP ASE can have only one default character set, in which data is stored in its databases. When you install SAP ASE, you specify a default character set.

Context

> ⚠ Caution
>
> Read the following carefully, and exercise caution when changing the default character set in SAP ASE. SAP strongly recommends that you perform backups before you change a default character set.

When you change the default character set in SAP ASE, you must convert any existing data to the new default character set. Conversion is unnecessary only if:

- There is no user data in the server.
- It is acceptable to destroy user data in the server.
- You are absolutely certain that data in the server uses only ASCII-7. In this case, you can change the default without first copying your data out of the server.

To change the default character set:

Procedure

1. Copy the data out using `bcp`.
2. Change the default character set.
3. Use `bcp` with the appropriate flags for data conversion to copy the data back into the server.

Next Steps

See the *Utility Guide* for more information about using `bcp` to copy data.

> ⚠ Caution
>
> After converting data to a different character set (particularly to UTF-8), the data may be too large for the allocated column size. Re-create the columns affected with a larger size.

Code conversion between the character set of the existing data and the new default character set must be supported. If it is not, conversion errors will occur and the data is not converted correctly.

Even if conversions are supported between the character sets, some errors may occur due to minor differences between the character sets, or because some characters do not have equivalents in other character sets. Rows containing problematic data may not get copied back into the database, or data may contain partial or invalid characters.

10.8.2 Changing the sort order with a resource file

You can install and modify languages, character sets, and sort order defaults using the `sqllocres` utility.

Use the (UNIX only) `sqllocres` utility to manually modify languages, character sets, and sort order defaults. A sample `sqlloc.rs` resource file is located in the `$SYBASE/ASE-16_0/init/sample_resource_files/` directory. The resource file is similar to:

```
sybinit.release_directory: USE_DEFAULT
sqlsrv.server_name: PUT_YOUR_SERVER_NAME_HERE
sqlsrv.sa_login: sa
sqlsrv.sa_password:
sqlsrv.default_language: USE_DEFAULT
sqlsrv.language_install_list: USE_DEFAULT
sqlsrv.language_remove_list: USE_DEFAULT
sqlsrv.default_characterset: USE_DEFAULT
sqlsrv.characterset_install_list: USE_DEFAULT
sqlsrv.characterset_remove_list: USE_DEFAULT
sqlsrv.sort_order: USE_DEFAULT
# An example sqlloc resource file...
# sybinit.release_directory: USE_DEFAULT
# sqlsrv.server_name: PUT_YOUR_SERVER_NAME_HERE
# sqlsrv.sa_login: sa
# sqlsrv.sa_password:
# sqlsrv.default_language: french
# sqlsrv.language_install_list: spanish,german
# sqlsrv.language_remove_list: USE_DEFAULT
# sqlsrv.default_characterset: cp437
# sqlsrv.characterset_install_list: mac,cp850
# sqlsrv.characterset_remove_list: USE_DEFAULT
# "sqlsrv.sort_order" can be sort order name (dictionary_cp437) or file name
(dictionary.srt)
# sqlsrv.sort_order: dictionary_cp437
```

For information about the `sqllocres` utility, see *Utility Guide* > *sqllocres*.

10.8.3 Change the Default Sort Order

SAP ASE can have only one default sort order, which is the collating sequence it uses to order data.

When you consider changing the sort order for character data on a particular SAP ASE, keep in mind that all of your organization's SAP ASEs should have the same sort order. A single sort order enforces consistency and makes distributed processing easier to administer.

You may have to rebuild your indexes after changing the default sort order.

10.8.4 Reconfiguring the Character Set, Sort Order, or Message Language

Changing SAP ASE's default character set, sort order, or message language involves extracting the data, installing the new character set, sort order, or message language, shutting down and restarting the server, and reloading the data. For procedures on how to configure the character set, sort order, or message language for a new server, see the configuration documentation for your platform.

Back up all databases in SAP ASE before and after you change character sets or sort orders. After you back up your databases, use `bcp` to copy the data in and out of your databases if:

- A database contains character data and you want to convert the data to a new character set. Do not load a database dump of the data into a server that uses the new default character set. SAP ASE assumes the loaded data is in the new character set, and corrupts the data.
- You are changing the default sort order only and not the default character set. You cannot load a database from a dump performed prior to changing the sort order—if you attempt to, an error message appears, and SAP ASE aborts the load.
- You change the default character set, and either the old or the new sort order is not binary. You cannot load a database dump that was made before you changed the character set.

You cannot reload your data from a database dump once you have reconfigured the default character set and sort order (unless both old and new character sets use a binary sort order and no conversion is required between the old and new character sets).

10.8.5 Example: Converting a Unicode Database to UTF-8

A fictitious database named `xpubs` is modified to use `univarchar` columns.

Prerequisites

Assume a database was created using the following script on a server that has all the installation defaults, namely character set "iso_1" and default sort order ID 50, "binary_iso_1":

```
create database xpubs
```

```
go
use xpubs
go
create table authors (au_id int, au_lname varchar(255), au_fname varchar(255))
go
create index au_idx on authors(au_lname, au_fname)
go
```

Then the data was loaded into the server using a series of inserts and updates.

To convert the data to UTF-8:

Procedure

1. Extract the data and convert it to UTF-8 form (the conversion occurs with the `-J` parameter):

   ```
   % bcp xpubs..authors out authors.utf8.bcp -c -Jutf8 -Usa -P
   ```

2. Install UTF-8 as the default character set:

   ```
   charset -Usa -P binary.srt utf8
   isql -Usa -P
   sp_configure 'default sortorder id', 50, 'utf8'
   ```

3. Shutdown the server.
4. Restart the server and modify the default character set and re-create indexes on the system tables. :

   ```
   isql -Usa -P
   sp_dboption xpubs, 'select into', true
   go
   use xpubs
   go
   checkpoint
   go
   delete from authors
   go
   quit
   ```

5. Restart the server.
6. Reload the data:

   ```
   bcp xpubs..authors in authors.utf8.bcp -c -Jutf8 -Usa -P
   ```

10.8.5.1 Migrating Selected Columns to unichar

With a working database running with UTF-8 as the default character set, you can convert select columns to `univarchar`.

The columns are modified to the new datatypes, the data is converted in place, and the index is re-created.

For example, to migrate a select of columns from the xpubs database to unichar:

```
% isql -Usa -P
> use xpubs
```

```
> go
> alter table authors modify au_lname univarchar(255), au_fname univarchar(255)
> go
```

> **i Note**
>
> Currently, the `alter table modify` command does not support `text`, `image`, or unitext columns. To migrate from a `text` to a unitext column, you must first use `bcp`, create a table with unitext columns, and then use `bcp` again to place data into the new table. This migration path only works when you invoke `bcp` with `-Jutf8` option.

10.8.5.2 Migrating to or from unitext

The `alter table modify` command does not support `text`, `image`, or unitext columns. To migrate from a `text` to a unitext column, you must first use `bcp`, create a table with unitext columns, and then use `bcp` again to place data into the new table. This migration path only works when you invoke `bcp` with `-Jutf8` option.

10.8.6 Before Changing the Character Set or Sort Order

You must perform some preliminary steps before you change the character set or sort order.

Procedure

1. Dump all user databases and the `master` database. If you have made changes to `model` or `sybsystemprocs`, dump them also.
2. Load the Language Module if it is not already loaded (see the configuration documentation for your platform for complete instructions).
3. If you are changing the SAP ASE default character set, and your current databases contain non ASCII-7 data, use `bcp` with the necessary flags to copy the existing data out of your databases.

Results

Once you have loaded the Language Module, you can run the SAP ASE installation program, which allows you to:

- Install or remove message languages and character sets included with SAP ASE
- Change the default message language or character set
- Select a different sort order

See the configuration documentation for your platform for instructions on using the installation program

> **i Note**
>
> Before you change the character set or sort order, SAP ASE must have as many open databases as there are databases managed by the server. If SAP ASE does not have a sufficient number of open databases when it is re-started after a change in sort order, SAP ASE prints this message to the error log and the server will revert to the former sort order:
>
> ```
> The configuration parameter 'number of open databases' must be at least as
> large as the number of databases,
> in order to change the character set or sort order." Re-start Adaptive
> Server, use sp_configure to increase
> 'number of open databases' to at least %d, then re-configure the character
> set or sort order
> ```

To reconfigure the language, character set, or sort order, use the `sqlloc` utility, described in *Utility Guide*. If you are using Windows, use the `Server Config` utility, described in *Configuration Guide* > *Default SAP ASE Configuration*.

If you installed additional languages but did not change the SAP ASE character set or sort order, you have completed the reconfiguration process.

10.8.7 Set the User's Default Language

If you install an additional language, users running client programs can run `create login` to set that language as their default language, or set the LANG variable on the client machine, with the appropriate entries in `locales.dat`.

10.8.8 Manage Suspect Partitions

Partitions are marked suspect because of a sort order or character set change on a range-partitioned table, or because of a cross-platform dump and load with a hash-partitioned table.

If the table is marked with suspect partitions:

- All updates and cursor activities are suspended on this table.
- No `alter table` commands, except `partition by`, are allowed. `create index` and `drop index` are not allowed on a table with suspect partitions.
- The `select` command is allowed on tables containing suspect partitions. However, the optimizer treats such tables as round-robin partitioned tables, to avoid using the possibly corrupt `partition` condition.

To fix table with suspect partitions:

- If the `partition` condition needs fixing after a sort-order change, you can use `alter table` with the `partition by` option to repartition a table that has suspect partitions.
- If the partition condition does not need fixing, you can use the `reorg rebuild table` command to rebuild the table, redistributing only the data rows among the partitions.
- If the indexes as well as the partitions on a table are marked suspect, use `partition by` or `reorg rebuild` to fix both the suspect index and suspect partitions.

To manage suspect partitions in cross-platform dump and load operations:

- During the first `online database` command, after you execute `load database` across two platforms with different endian types, the hash partition is marked suspect.
- Any global clustered index on a round-robin partition, which has an internally generated partition condition with a `unichar` or `varchar` partition key, is marked suspect.
- After the database is online, use `sp_post_xpload` to fix the suspect partitions and indexes.

10.9 Installing Date Strings for Unsupported Languages

Use `sp_addlanguage` to install names for the days of the week and months of the year for languages that do not have language modules.

Context

`sp_addlanguage` lets you define:

- A language name and (optionally) an alias for the name
- A list of the full names of months and a list of abbreviations for the month names
- A list of the full names of the days of the week
- The date format for entering dates (such as month/day/year)
- The number of the first day of the week

Procedure

To add the information for Italian:

```
sp_addlanguage italian, italiano,
"gennaio,febbraio,marzo,aprile,maggio,giugno,luglio,agosto,settembre,ottobre,
novembre,dicembre",
"genn,feb,mar,apr,mag,giu,lug,ago,sett,ott,nov,dic",
"lunedi,martedi,mercoledi,giovedi,venerdi,sabato,domenica", dmy, 1
```

Next Steps

`sp_addlanguage` enforces strict data entry rules. The lists of month names, month abbreviations, and days of the week must be comma-separated lists with no spaces or line feeds (returns). Also, they must contain the correct number of elements (12 for month strings, 7 for day-of-the-week strings.)

Valid values for the date formats are: `mdy`, `dmy`, `ymd`, `ydm`, `myd`, and `dym`. The `dmy` value indicates that the dates are in day/month/year order. This format affects only data entry; to change output format, you must use the `convert` function.

10.9.1 Server Versus Client Date Interpretation

Generally, date values are resolved on the client. When a user selects date values, SAP ASE sends them to the client in an internal format.

The client uses the `common.loc` file and other localization files in the default language subdirectory of the `locales` directory on the client to convert the internal format to character data. For example, if the user's default language is Spanish, SAP ASE looks for the `common.loc` file in `$SYBASE/$SYBASE_ASE/locales/spanish/char_set`. It uses the information in the file to display, for example, `12 febrero 2007`.

Assume that the user's default language is set to Italian, a language for which SAP ASE does not provide a language module, and that the date values in Italian have been added. When the client connects to the server and looks for the `common.loc` file for Italian, it does not find the file. The client prints an error message and connects to the server. If the user then selects date values, the dates are displayed in U.S. English format.To display the date values added with `sp_addlanguage`, use the `convert` function to force the dates to be converted to character data at the server.

The following query generates a result set with the dates in U.S. English format:

```
select pubdate from titles
```

The query below, however, returns the date with the month names in Italian:

```
select convert(char(19),pubdate) from titles
```

10.10 Internationalization and localization files

The files that support data processing in a particular language are called internationalization files. Several types of internationalization files come with SAP ASE.

Table 13: Internationalization files

File	Location	Purpose and contents
`charset.loc`	In each character set subdirectory of the `charsets` directory	Character set definition files that define the lexical properties of each character, such as alphanumeric, punctuation, operand, and uppercase or lowercase. Used by SAP ASE to correctly process data.
`*.srt`	In each character set subdirectory of the `charsets` directory	Defines the sort order for alphanumeric and special characters, including ligatures, diacritics, and other language-specific considerations.

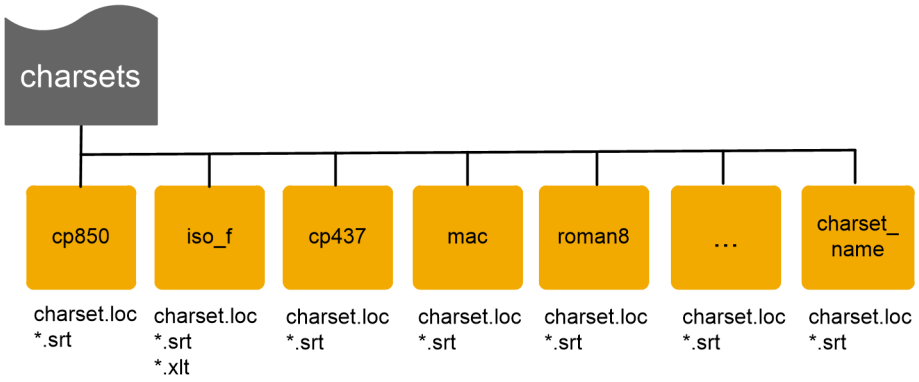
File	Location	Purpose and contents
`*.xlt`	In each character set subdirectory of the `charsets` directory	Terminal-specific character translation files for use with utilities such as `bcp` and `isql`. For more information about how the `.xlt` files are used, see the *Utility Guide*.

> ⚠ Caution
>
> Do not alter any of the internationalization files. If you need to install a new terminal definition or sort order, contact your local SAP office or distributor.

10.10.1 Character sets directory structure

The `charsets` directory is located in `$SYBASE/`. There is a separate subdirectory for each character set in the `charsets` directory.

Within the subdirectory for each character set (for example, `cp850`) are the character set and sort order definition files and terminal-specific files.



If you load additional character sets, they also appear in the `charsets` directory.

10.10.2 Types of localization files

SAP ASE includes several localization files for each Language Module.

> ℹ Note
>
> All SAP ASE-related locales files (used by `dataserver`, `sqlloc`, `syconfig`, and so on) are in `$SYBASE/$SYBASE_ASE/locales`, except `locales.dat`, which is in `$SYBASE/locales`. All Open Client/Server-related locales files (`ctlib`, `ctisql`, `ctbcp`, `optdiag`, `installjava`, and so on) are located in `$SYBASE/locales`.

SAP ASE localization files include:

- `locales.dat` – in the `locales` directory. Used by client applications to identify the default message language and character set.
- `server.loc` – in the character set subdirectories under each language subdirectory in the `$SYBASE/$SYBASE_ASE/locales` directory. Software messages translated into the local language. SAP products have product-specific `*.loc` files. If an entry is not translated, the software message or string appears in U.S. English instead of the local language.
- `common.loc` – in each language and character set directory of the `locales` directory. `common.loc` contains the local names of the months of the year and their abbreviations, and information about the local date, time, and money formats.

> ⚠ Caution
>
> Do not alter any of the localization files. If you need to alter any information in those files, contact your local SAP office or distributor.

10.10.3 Software Messages Directory Structure

Within the `$SYBASE/$SYBASE_ASE/locales` directory is a subdirectory for each language installed.

There is always a `us_english` subdirectory (`english` on Windows).)

During installation, when you are prompted to select the languages you want installed on SAP ASE, the installation program lists the supported software message languages. If you install language modules for additional languages, you see subdirectories for those languages. Within each language subdirectory are subdirectories for the supported character sets; for example, `cp850` is a supported character set for `us_english`. Software message files for each SAP product reside in the character set subdirectories.

10.10.4 Global variables for Languages and Character Sets

SAP ASE includes global variables that display information about languages and character sets.

The following global variables contain information about languages:

- `<@@langid>` – contains the local language ID of the language currently in use (specified in `syslanguages.langid`)
- `<@@language>` – contains the name of the language currently in use (specified in `syslanguages.name`)

The following global variables contain information about character sets:

- `<@@char_convert>` – contains 0 if character set conversion is not in effect. Contains 1 if character set conversion is in effect.
- `<@@client_csname>` – the client's character set name. Set to NULL if client character set has never been initialized; otherwise, contains the name of the character set for the connection.
- `<@@client_csid>` – the client's character set ID. Set to -1 if client character set has never been initialized; otherwise, contains the client character set ID from `syscharsets` for the connection.
- `<@@client_csexpansion>` – returns the expansion factor used when converting from server's character set to client's character set.

- `<@@maxcharlen>` – the maximum length, in bytes, of a character in the SAP ASE default character set.
- `<@@ncharsize>` – the maximum length, in bytes, of a character set in the current server default character set.
- `<@@unicharsize>` – equals 2.

See *Reference Manual: Building Blocks > Global Variables* for a list of all global variables.

11 Configuring Client/Server Character Set Conversions

In a heterogeneous environment, SAP ASE may need to communicate with clients running on different platforms using different character sets.

Although different character sets may support the same language group (for example, ISO 8858-1 and CP 850 support the group 1 languages), they may encode the same characters differently. For example, in ISO 8859-1, the character *à* is encoded as *0xE0* in hexadecimal. However, in CP 850 the same character is encoded as *0x85* in hexadecimal.

To maintain data integrity between your clients and servers, data must be converted between the character sets. The goal is to ensure that an "a" remains an "a" even when crossing between machine and character set boundaries. This process is known as character set conversion.

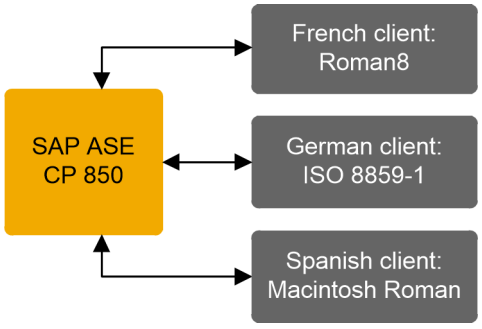
11.1 Supported Character Set Conversions

Character set conversion occurs between a pair of character sets. The supported conversions in any particular client/server system depend on the character sets used by the server and its clients. One type of character set conversion occurs if the server uses a native character set as the default; a different type of conversion is used if the server default is Unicode UTF-8.

11.1.1 Conversion for Native Character Sets

SAP ASE supports character set conversion between native character sets that belong to the same language group.

If the server has a native character set as its default, the clients' character sets must belong to the same language group. In figure below, the clients' character sets and the SAP ASE default character set all belong to language group 1. Data is correctly converted between the client character sets and the server default character set. Since they all belong to the same language group, the clients can view all data on the server, no matter which client submitted the data.

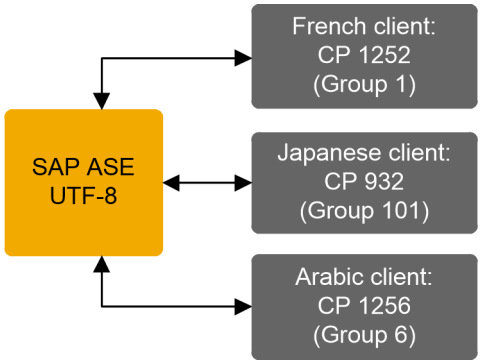


11.1.2 Conversion in a Unicode System

SAP ASE supports character set conversion between UTF-8 and any native character set that SAP supports.

In a Unicode system, since the server default character set is UTF-8, the client character set may be a native character set from any language group. Therefore, a Japanese client (group 101), a French client (group 1), and an Arabic client (group 6) can all send and receive data from the same server. Data from each client is correctly converted as it passes between each client and the server.

The following figure illustrates character set conversion in a Unicode system:



Each client can view data only in the language supported by its character set. Therefore, the Japanese client can view any Japanese data on the server, but it cannot view Arabic or French data. Likewise, the French client can view French or any other Western European language supported by its character set, but not Japanese or Arabic.

The following figure illustrates the process of viewing Unicode data.

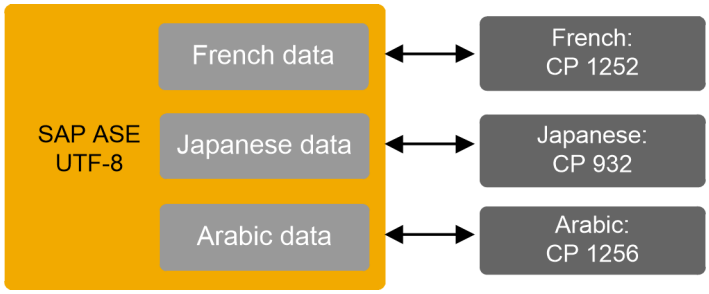


Figure 1: Viewing Unicode data

An additional character set, ASCII-7, is a subset of *every* character set, including Unicode, and is therefore compatible with `all` character sets in `all` language groups. If either the SAP ASE or the client's character set is ASCII-7, any 7-bit ASCII character can pass between the client and server unaltered and without conversion.

SAP recommends that you do not configure a server for ASCII-7. You can achieve the same benefits of compatibility by restricting each client to use only the first 128 characters of each native character set.

11.1.3 SAP ASE Direct Conversions

SAP ASE direct conversions occur between two native character sets of the same language group.

For example, SAP ASE supports conversion between CP 437 and CP 850, because both belong to language group 1. SAP ASE direct conversions exist between many, but not all, native character sets of a language group.

11.1.4 Unicode Conversions

Unicode conversions exists for all native character sets. When converting between two native character sets, Unicode conversion uses Unicode as an intermediate character set.

For example, to convert between the server default character set (CP 437), and the client character set (CP 860), CP 437 is first converted to Unicode; Unicode is then converted to CP 860.

Unicode conversions may be used either when the default character set of the server is UTF-8, or a native character set. You must specifically configure your server to use Unicode conversions (unless the server's default character set is UTF-8).

Earlier versions of SAP ASE used direct conversions, and it is the default method for character set conversions. However, Unicode conversions allow easier and less complex character set conversion. While SAP ASE direct conversions are still supported, SAP now also uses Unicode conversions to provide complete conversion support for all character sets and has no plans to add new direct conversions.

11.1.4.1 Allowing Unicode noncharacters

In versions of SAP ASE earlier than 15.7, the `unichar`, `univarchar`, `unitext`, `char`, `varchar`, and `text` datatypes under the `utf-8` default character set did not accept Unicode noncharacters (code points are permanently reserved for internal use).

SAP ASE versions 15.7 and later allows you to ignore Unicode noncharacters by enabling the `enable functionality group` or the `enable permissive unicode` configuration parameters.

If you do not enable this feature, SAP ASE rejects these noncharacters as in earlier versions.

For more information about the Unicode standard, see the Unicode Consortium Web site .

When you enable this feature, Unicode noncharacters are not detected in:

- Parameters:
  - Presented as `univarchar` and `unitext` (UTF-16) datatypes
  - Presented as `varchar` and `text` (UTF-8) datatypes
  - As parameters to dynamic SQL statements
  - As input to parameterized language statements
- String literals when the server's character set is UTF-8
- Escaped string literals (those prefixed with U&), regardless of the server's character set
- Conversion processes between `unichar` (UTF-16) and `varchar` (UTF-8) in either direction

In addition, Unicode noncharacters are acceptable in simple expressions such as comparisons, where they sort higher than legal Unicode characters.

You can use Unicode noncharacters as parameters to these functions:

`ascii()`	`lower()`	`sortkey()`
`char_length()`	`ltrim()`	`soundex()`
`charindex()`	`patindex()`	`str_replace()`
`compare()`	`replicate()`	`stuff()`
`datalength()`	`reverse()`	`substring()`
`difference()`	`right()`	`upper()`
`left()`	`rtrim()`	`uscalar()`
`len()`		

> **i** Note
>
> This feature does not affect UTF-16 surrogate handling enabled with the `enable surrogate handling` configuration parameter. See *Reference Manual: Configuration Parameters*.

11.2 Choosing a Conversion Type

By default, SAP ASE uses direct conversions to convert data between different character sets.

Set the `enable unicode conversions` option to either 1 or 2 to use the Unicode conversions:

- Set to 1 – uses SAP ASE direct conversions or Unicode conversions. SAP ASE first checks to see if an SAP ASE direct conversion exists for the server and client character set. If a direct conversion is used; if no direct conversion exists, the Unicode conversion is used.
  Use this setting if the character sets in your client/server system fall into both columns 1 and 2 in the table below.
- Set to 2 – uses Unicode conversions only. SAP ASE uses Unicode conversions, without attempting to find an SAP ASE direct conversion. Use this setting if the client/server conversions result in a change in the data length.

If all character sets fall into column 2 in the table below, set `enable unicode conversions` to 2 to always use Unicode conversions.

For SAP ASE versions 15.0 and later, the default value for `enable unicode conversions` is 1.

If the server default is UTF-8, the server automatically uses Unicode conversions only.

Non-Unicode Client/Server Systems

In a non-Unicode system, the character sets of the server and clients are native character sets; therefore, you can use the SAP ASE direct conversions.

However, there are some character sets for which there is no SAP ASE direct conversion; in this situation, you must use Unicode conversions.

- If all character sets in your client/server system are column 1 in the table below, use the SAP ASE direct conversions. The character sets must all belong to the same language group.
- If the character sets in your client/server system are in column 2 in the table below, or some combination of columns 1 and 2, configure your server to use Unicode conversions. Again, the character sets must all belong to the same language group.

For example, assume the server default character set is CP 850 and the clients' character sets are either ISO 8859-1 or ROMAN 8. The table below shows that direct conversions exist between CP 850 and the client character sets. Now, suppose you add a client using CP 1252. Since there is no direct conversion between CP 1252 and CP 850, (the default server character set), you must use Unicode conversions to convert between CP 1252 and CP 850. When you have a mixture of character sets—some where you can use SAP ASE direct conversions and others where you must use Unicode conversions—you can specify that a combination of SAP ASE direct conversion and Unicode conversion be used.

Unicode Client/Server Systems

If your server default is Unicode UTF-8, then all conversions are between UTF-8 and the native character set being used on the client systems. In a Unicode system, Unicode conversions are used exclusively.

Table 14: Conversion methods for character sets

Language group	Column 1 – SAP ASE direct conversions and Unicode conversions	Column 2 – Unicode conversions only
Group 1	CP 437, CP 850, ISO 8859-1, Macintosh Roman	CP 860, CP 1252, ISO 8859-15, CP 863
Group 2	CP 852, CP 1250, CP 8859-1, Macintosh Central European	ISO 8859-2
Group 4	No conversions needed (only one character set supported)	
Group 5	CP 855, CP 866, CP 1251, ISO 8859-5, Koi8, Macintosh Cyrillic	
Group 6		CP 864, CP 1256, ISO 8859-6
Group 7	CP 869, CP 1253, GREEK8, ISO 8859-7, Macintosh Greek	
Group 8		CP 1255, ISO 8859-8
Group 9	CP 857, CP 1254, ISO 8859-9, Macintosh Turkish, TURKISH8	
Group 101	DEC Kanjii, EUC-JIS, Shift-JIS	CP 932

Language group	Column 1 – SAP ASE direct conversions and Unicode conversions	Column 2 – Unicode conversions only
Group 102		CP 936, EUG-GB, GB18303
Group 103		Big 5, CP 950, EUC-CNS
Group 104		EUCKSC, CP 949
Group 105		CP 874, TIS 620
Group 106	No conversions needed (only one character set supported)	
Unicode	No conversions needed (only one character set supported)	

11.3 Enabling and Disabling Character Set Conversion

A client that requests a connection identifies its character set to SAP ASE. SAP ASE compares the client character set with its default character set, and if the two names are identical, no conversion is required.

If the names differ, SAP ASE determines whether it supports conversion between its default and the client's character set. If it does not, it sends an error message to the client and continues with the login process. If it does, character set conversion is automatically enabled. If the default character set of the server is UTF-8, Unicode conversions are automatically used. If the default is a native character set, the server uses SAP ASE direct conversions, unless the user requests Unicode conversions.

You can disable character set conversion at the server level. You may want to do this if:

- All of your clients are using the same character set as the server default, and therefore, no conversion is required.
- Conversion between the client character set and the server default is not supported.
- You want to store data in the server without changing the encoding.

To disable character set conversion at the server level, set the `disable character set conversion` parameter to 1.

You can control character set conversion at the connection level using the `set char_convert` command from within a client session. `set char_convert off` turns conversion off between a particular client and the server. You may want to `set char_convert off` if the client and the server use the same character set, which makes conversion unnecessary. `set char_convert on` turns conversion back on.

11.3.1 Characters That Cannot Be Converted

You cannot convert all character sets.

Characters may not be converted if:

- The character exists (is encoded) in the source character set, but does not exist in the target character set. For example, the OE ligature is part of the Macintosh character set (code point 0xCE). This character does not exist in the ISO 8859-1 character set. If the OE ligature exists in data that is being converted from the Macintosh to the ISO 8859-1 character set, it causes a conversion error.
- The character exists in both the source and the target character set, but in the target character set, the character is represented by a different number of bytes than in the source character set.
  For example, 1-byte accented characters (such as á, è) are 2-byte characters in UTF-8; 2-byte Thai characters are 3-byte characters in UTF-8. Avoid this limitation by configuring the `enable unicode conversion` option to 1 or 2.

11.4 Error handling in character set conversion

The SAP ASE character set conversion reports errors when a character exists in the client's character set but not in the server's character set, or vice versa.

SAP ASE must guarantee that data successfully converted on input to the server can be successfully converted back to the client's character set when the client retrieves that data. To do this effectively, SAP ASE must avoid putting suspect data into the database.

When SAP ASE encounters a conversion error in the data being entered, it generates this message:

```
Msg 2402, Severity 16 (EX_USER):
Error converting client characters into server's
character set. Some character(s) could not be converted.
```

A conversion error prevents query execution on insert and update statements. If this occurs, review your data for problem characters and replace them.

When SAP ASE encounters a conversion error while sending data to the client, it replaces the bytes of the suspect characters with ASCII question marks (?). The query batch continues to completion. When the statement is complete, SAP ASE sends the following message:

```
Msg 2403, Severity 16 (EX_INFO):
WARNING! Some character(s) could not be converted into
client's character set. Unconverted bytes were changed
to question marks ('?').
```

11.5 Conversions and Changes to Data Lengths

In some cases, converting data between the server's character set and the client's character set results in a change to the length of the data. For example, when the character set on one system uses one byte to represent each character and the character set on the other system requires two bytes per character.

When character set conversion results in a change in data length, there are two possibilities:

- The data length decreases, as in the following examples:
  - Greek or Russian in multibyte UTF-8 to a single-byte Greek or Russian character set

    - Japanese two-byte Hankaku Katakana characters in EUC-JIS to single-byte characters in Shift-JIS
- The data length increases, as in the following examples:
    - Single-byte Thai to multibyte Thai in UTF-8
    - Single-byte Japanese characters in Shift-JIS to two-byte Hankaku Katakana in EUC-JIS

To configure your system or application:

1. Configure the server to use Unicode conversions. If the data length increases between the server and the client, you must also complete steps 2 and 3.
2. The client must be using Open Client 11.1 or later. It must inform the server that it can handle CS_LONGCHAR data at connection time, using the Open Client `ct_capability` function.
   The `<capability>` parameter must be set to CS_DATA_LCHAR and the `<value>` parameter must be set to CS_TRUE, where `<connection>` is a pointer to a CS_CONNECTION structure:

   ```
   CS_INT capval = CS_TRUE
   ct_capability(<connection>,CS_SET,CS_CAP_RESPONS,
        CS_DATA_LCHAR,&capval)
   ```

3. When conversions result in an increase in data length, `char` and `varchar` data are converted to the client's character set and are sent to the client as CS_LONGCHAR data. The client application must be coded to extract the data received as CS_LONGCHAR.

11.6 Specify the Character Set for Utility Programs

The SAP utility programs assume that the default character set of the client platform is the same character set the client is using.

However, occasionally the client character set differs from the character set for the platform. For this reason, you may need to specify the client character set at the command line. A command line option for the `isql`, `bcp`, and `defncopy` utilities specifies the client's character set, and temporarily overrides settings of the LANG variable or settings in `locales.dat`.

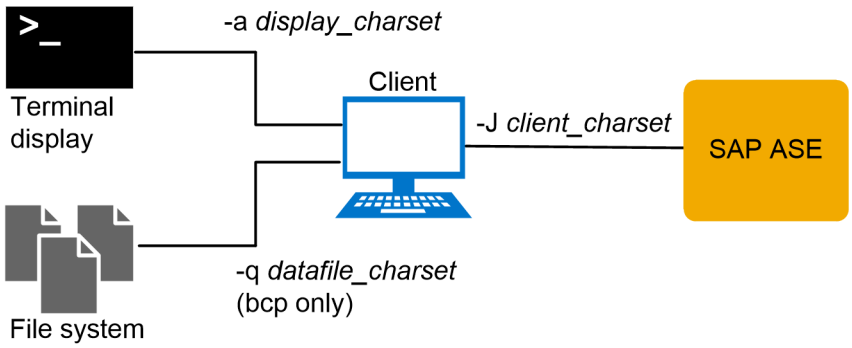
`-J< ><charset_name>` (UNIX and PC) sets the client's character set to the `<charset_name>`.

If you omit the client character set's command line flag, the platform's default character set is used. See the *Utility Guide*.

11.6.1 Display and file character set command line options

You might require character set conversion between the client and a terminal, and between the client and a file system.

This figure illustrates the paths and command line options that are available in the standalone utilities `isql`, `bcp`, and `defncopy`:



Use:

- `-J` or `/clientcharset` command line option to specify the character set used by the client when it sends and receives character data to and from SAP ASE.
- `-a` command line option if you are running the client from a terminal with a character set that differs from the client character set. In the figure above, the `-a` option and the `-J` option are used together to identify the character set translation file (`.xlt` file) needed for the conversion.
- `-a` without the `-J` parameter only if the client character set is the same as the default character set.
- `-q` command line option if you are running `bcp` to copy character data to or from a file system that uses a character set that differs from the client character set. In the figure above, use the `-q` or `/filecharset` option and the `-J` or `/clientcharset` option together to identify the character set translation file (`.xlt` file) needed for the conversion.

12 Diagnosing System Problems

Diagnosing system problems involves, among other duties, reviewing error messages, managing processes, performing dumps, starting and stopping servers.

12.1 How SAP ASE Uses Error Messages

SAP ASE displays an error message when it encounters a problem.

The error message includes:

- A message number, which uniquely identifies the error message
- A severity level number between 10 and 24, which indicates the type and severity of the problem
- An error state number, which allows unique identification of the line of SAP ASE code at which the error was raised
- An error message, which tells you what the problem is, and may suggest how to fix it

See the *Configuration Guide* for your platform for a description of the error log format.

For example, if you try to access a table that does not exist, you see:

```
select * from publisher
```

```
Msg 208, Level 16, State 1:
publisher not found. Specify owner.objectname or use sp_help to check whether
the object exists (sp_help may produce lots of output).
```

There may be more than one error message for a single query. If there is more than one error in a batch or query, SAP ASE usually reports only the first one. Subsequent errors are reported the next time you execute the batch or query.

Error messages are stored in `master..sysmessages`, which is updated with each new version of SAP ASE (and has thousands of rows). Here are the first few rows (from an SAP ASE that uses us_english as the default language):

```
select error, severity, description
from sysmessages
where error >=101 and error <=106
and langid is null
```

```
error severity description
----- -------- ------------------------------------------------
  101       15 Line %d: SQL syntax error.
  102       15 Incorrect syntax near '%.*s'.
  103       15 The %S_MSG that starts with '%.*s' is too long.
               Maximum length is %d.
  104       15 Order-by items must appear in the select-list if
               the statement contains set operators.
  105       15 Unclosed quote before the character string '%.*s'.
```

```
106       16 Too many table names in the query. The maximum
             allowable is %d.
```

You can query `sysmessages` to generate a custom list of error messages:

- If your server supports more than one language, `sysmessages` stores each message in each language. The column `langid` is NULL for us_english and matches the `syslanguages.langid` for other languages installed on the server.
- The `sqlstate` column stores the SQLSTATE value for error conditions and exceptions defined in ANSI SQL92.
- Message numbers 17000 and higher are system procedure error messages and message strings.

12.1.1 Error Log Format

All SAP ASE error messages use the same format.

```
<instance_ID>:<thread_ID>:<family_ID>:<spid> <date> <time> {<server> | <kernel>}
<message>
```

where:

- `<instance_ID>` – the instance running in a clustered environment. This value is 00 if SAP ASE is running in a nonclustered environment.
- `<thread_ID>` – the thread on which the error occurred. Always appears as a four-digit number (for example, 10 is shown as 0010).

  > **i Note**
  >
  > In process mode, SAP ASE displays the engine involved for each log entry. The engine number is indicated by a four-digit number. If only one engine is online, it is indicated by 00.

- `<family_ID>` – family ID of the originating thread:
  - In serial processing, you see 00000.
  - In parallel processing, you see the server process ID number of the parent of the originating thread.
- `<spid>` – the server process ID of the originating thread:
  - In serial processing, this is the server process ID number of the thread that generated the message. If the thread is a system task, the spid value is 00000.
  - In parallel processing, you see the server process ID number of the originating thread.
- `<date>` – appeared in the format `yyyy/mm/dd`, which allows you to sort error messages by date.
- `<time>` – shown in 24-hour format, which includes seconds and hundredths of a second.
- `<server>` | `<kernel>` – this entry is for use only by SAP Technical Support.
- `<message>` – the error message

For example:

```
00:0024:00000:00000:2010/04/27 10:28:07.82 kernel Thread 24 (LWP 24390) of
Threadpool syb_default_pool online as engine 0
```

- `<instance_ID>` = 0 – the server is not configured for a clustered environment
- `<thread_ID>` = 0024 – SAP ASE brought engine 0 online as thread number 24.

- <family_ID> = 00000 – the process is running in serial mode.
- <spid> = 00000 – this is a system task.
- <date> and <time> = 2010/04/27 10:28:07.82.
- <server> | <kernel> = kernel.

12.1.2 Error Messages and Message Numbers

A combination of message number (error) and language ID (langid) uniquely identifies each error message. Messages that share the same message number but have different language IDs.

```
select error, description, langid
from sysmessages
where error = 101
```

```
error description                          langid
----- ------------------------------------ ------
  101 Line %d: SQL syntax error.             NULL
  101 Ligne %1!: erreur de syntaxe SQL.         1
  101 Zeile %1!: SQL Syntaxfehler.              2
(3 rows affected)
```

The error message text describes the problem. The descriptions often include a line number, a reference to a type of database object (a table, column, stored procedure, and so forth), or the name of a particular database object.

In the description field of sysmessages, a percent sign (%) followed by a character or character string serves as a placeholder for these pieces of data, which SAP ASE supplies when it encounters the problem and generates the error message. "%d" is a placeholder for a number; "%S_MSG" is a placeholder for a type of database object; "%.*s"—all within quotes—is a placeholder for the name of a particular database object.

For example, the description field for message number 103 is:

```
The %S_MSG that starts with '%.*s' is too long. Maximum length is %d.
```

The actual error message that appears to a user might be:

```
The column that starts with 'title' is too long. Maximum length is 80.
```

For errors that you report to Technical Support, include the numbers, object types, and object names.

12.1.3 Variables in Error Message Text

Error messages include variables to stand for characters, numbers, database structures, and so on.

Symbol	Stands for
%d, %D	Decimal number

Symbol	Stands for
%x,%X,%.*x,%lx, %04x, %08lx	Hexadecimal number
%s	Null-terminated string
%.*s, %*s, %*.s	String, usually the name of a particular database object
%S_<type>	SAP ASE-defined structure
%c	Single character
%f	Floating-point number
%ld	Long decimal
%lf	Double floating-point number

12.2 SAP ASE error logging

Error messages from SAP ASE are sent to the user's screen and to the error log file.

The stack trace from fatal error messages (severity levels 19 and higher) and error messages from the kernel are sent to an error log file. The name of this file varies; see the configuration documentation for your platform or the *Utility Guide*.

> **i** Note
>
> The error log file is owned by the user who installed SAP ASE (or the person who started SAP ASE after an error log was removed). Permissions or ownership problems with the error log at the operating system level can block successful start-up of SAP ASE.

SAP ASE creates an error log for you if one does not already exist. Specify the location of the error log at start-up with the <errorlogfile> parameter in the runserver file or at the command line. The SAP installer utility configures the runserver file with `$SYBASE/$SYBASE_ASE/install` as the location of the error log if you do not choose an alternate location. If you do not specify the location in the runserver file or at the command line, the location of the error log is the directory from which you start SAP ASE. For more information about specifying the location of the error log, see `dataserver` in the *Utility Guide*.

> **i** Note
>
> Always start SAP ASE from the same directory, or with the runserver file or the error log flag, so that you can locate your error log.

Each time you start a server, messages in the error log provide information on the success (or failure) of the start and the recovery of each database on the server. Subsequent fatal error messages and all kernel error messages are appended to the error log file. To reduce the size of the error log by deleting old or unneeded messages, "prune" the log while SAP ASE is shut down.

Reporting Errors

When you report an error to SAP Technical Support, include:

- The message number, level number, and state number.
- Any numbers, database object types, or database object names that are included in the error message.
- The context in which the message was generated, that is, the command that was running at the time. You can help by providing a hard copy of the error log.

12.2.1 Severity Levels

The severity level of a message indicates the type and severity of the problem that SAP ASE has encountered.

For maximum integrity, when SAP ASE responds to error conditions, it displays messages from `sysmessages`, but takes action according to an internal table. A few corresponding messages differ in severity levels, so you may occasionally notice a difference in expected behavior if you are developing applications or procedures that refer to SAP ASE messages and severity levels.

> ⚠ Caution
>
> You can create your own error numbers and messages based on SAP ASE error numbers (for example, by adding 20,000 to the SAP ASE value). However, you cannot alter the SAP ASE-supplied system messages in the `sysmessages` system table.

You can add user-defined error messages to `sysusermessages` with `sp_addmessage`. See the *Reference Manual: Procedures*.

Users should inform the system administrator whenever problems that generate severity levels of 17 and higher occur. The system administrator is responsible for resolving them and tracking their frequency.

If the problem has affected an entire database, the system administrator may have to use the database consistency checker (`dbcc`) to determine the extent of the damage. The `dbcc` may identify some objects that have to be removed. It can repair some damage, but you may have to reload the database.

For more information, see the following chapters in the *System Administration Guide: Volume 2*:

- *Checking Database Consistency* – `dbcc`
- *Backing Upa and Restoring User Databases* – loading a user database
- *Restoring the System Databases* – loading system databases

12.2.2 Severity Levels 10 – 18

Error messages with severity levels 10–16 are generated by problems that are caused by user errors. These problems can be corrected by the user. Severity levels 17 and 18 do not terminate the user's session.

Error messages with severity levels 17 and higher should be reported to the system administrator or database owner.

12.2.2.1 Level 10: Status Information

Messages with severity level 10 provide additional information after certain commands have been executed and, typically, do not display the message number or severity level.

For example, after a `create database` command, SAP ASE displays a message telling the user how much of the requested space has been allocated for the new database.

12.2.2.2 Level 11: Specified Database Object Not Found

Messages with severity level 11 indicate that SAP ASE cannot find an object that is referenced in a command.

This is often because the user has made a mistake in typing the name of a database object, because the user did not specify the object owner's name, or because of confusion about which database is current. Check the spelling of object names, use the owner names if the object is not owned by the user or "dbo," and make sure you are in the correct database.

12.2.2.3 Level 12: Wrong Datatype Encountered

Messages with severity level 12 indicate a problem with datatypes. For example, the user may have tried to enter a value of the wrong datatype in a column or to compare columns of different and incompatible datatypes.

To correct comparison problems, use the `convert` function with `select`. See the *Reference Manual: Building Blocks* or the *Transact-SQL Users Guide*.

12.2.2.4 Level 13: User Transaction Syntax Error

Messages with severity level 13 indicate that something is wrong with the current user-defined transaction.

For example, the user may have issued a `commit transaction` command without having issued a `begin transaction`, or they may have tried to roll back a transaction to a savepoint that has not been defined (sometimes there may be a typing or spelling mistake in the name of the savepoint).

Severity level 13 can also indicate a deadlock, in which case the deadlock victim's process is rolled back. The user must restart his or her command.

12.2.2.5 Level 14: Insufficient Permission to Execute Command

Messages with severity level 14 mean that the user does not have the necessary permission to execute the command or access the database object. Users can ask the owner of the database object, the owner of the database, or the system administrator to grant them permission to use the command or object in question.

12.2.2.6 Level 15: Syntax Error in SQL Statement

Messages with severity level 15 indicate that the user has made a mistake in the syntax of the command. The text of these error messages includes the line numbers on which the mistake occurs and the specific word near which it occurs.

12.2.2.7 Level 16: Miscellaneous User Error

Most error messages with severity level 16 reflect that the user has made a nonfatal mistake that does not fall into any of the other categories. Severity level 16 and higher might also indicate software or hardware errors.

For example, the user may have tried to update a view in a way that violates the restrictions. Another error that falls into this category is unqualified column names in a command that includes more than one table with that column name. SAP ASE has no way to determine which one the user intends. Check the command syntax and working database context.

Messages that ordinarily have severities greater than 16 show severity 16 when they are raised by `dbcc checktable` or `dbcc checkalloc` so that checks can continue to the next object.

> **i** Note
>
> Levels 17 and 18 are usually not reported in the error log. Users should be instructed to notify the system administrator when level 17 and 18 errors occur.

12.2.2.8 Level 17: Insufficient Resources

Error messages with severity level 17 mean that the command has caused SAP ASE to run out of resources or to exceed some limit set by the system administrator. The user can continue, although he or she might not be able to execute a particular command

These system limits include the number of databases that can be open at the same time and the number of connections allowed to SAP ASE. They are stored in system tables and can be checked with `sp_configure`.

The database owner can correct the level 17 error messages indicating that the user has run out of space. Other level 17 error messages should be corrected by the system administrator.

12.2.2.9 Level 18: Nonfatal Internal Error Detected

Error messages with severity level 18 indicate an internal software bug. However, the command runs to completion, and the connection to SAP ASE is maintained.

The user can continue with the work they are doing, although they may not be able to execute a particular command. An example of a situation that generates severity level 18 is SAP ASE detecting that a decision about the access path for a particular query has been made without a valid reason.

Since problems that generate such messages do not keep users from their work, users tend not to report them. However, users should be instructed to inform the system administrator every time an error message with this severity level (or higher) occurs so that the system administrator can report them.

12.2.3 Severity Levels 19 – 26

Fatal problems generate error messages with severity levels 19 and higher. They break the user's connection to SAP ASE (some of the higher severity levels shut down SAP ASE). To continue working, the user must restart the client program.

When a fatal error occurs, the process freezes its state before it stops, recording information about what has happened. The process is then killed and disappears.

When the user's connection is broken, he or she may or may not be able to reconnect and resume working. Some problems with severity levels in this range affect only one user and one process. Others affect all the processes in the database. In some cases, the system administrator must restart SAP ASE. These problems do not necessarily damage a database or its objects, but they can. They may also result from earlier damage to a database or its objects. Other problems are caused by hardware malfunctions.

Error messages from the kernel are directed to the error log file.

12.2.3.1 Level 19: SAP ASE Fatal Error in Resource

Error messages with severity level 19 indicate that some nonconfigurable internal limit has been exceeded and that SAP ASE cannot recover gracefully. You must reconnect to SAP ASE.

12.2.3.2 Level 20: SAP ASE Fatal Error in Current Process

Error messages with severity level 20 indicate that SAP ASE has encountered a bug in a command. The problem has affected only the current process, and the database is unlikely to have been damaged. Run `dbcc` diagnostics. The user must reconnect to SAP ASE.

12.2.3.3 Level 21: SAP ASE Fatal Error in Database Processes

Error messages with severity level 21 indicate that SAP ASE has encountered a bug that affects all the processes in the current database. However, it is unlikely that the database itself has been damaged. Restart SAP ASE and run `dbcc` diagnostics. The user must reconnect to SAP ASE.

12.2.3.4 Level 22: SAP ASE Fatal Error: Table Integrity Suspect

Error messages with severity level 22 indicate that the table or index specified in the message has been previously damaged by a software or hardware problem.

The first step is to restart SAP ASE and run `dbcc` to determine whether other objects in the database are also damaged. Whatever the report from `dbcc` may be, The problem may be only in the cache, and not on the disk itself. If so, restarting SAP ASE fixes the problem.

If restarting does not help, then the problem is on the disk as well. Sometimes, the problem can be solved by dropping the object specified in the error message. For example, if the message tells you that SAP ASE has found a row with length 0 in a nonclustered index, the table owner can drop the index and re-create it.

SAP ASE takes any pages or indexes offline that it finds to be suspect during recovery. Use `sp_setsuspect_granularity` to determine whether recovery marks an entire database or only individual pages as suspect. See `sp_setsuspect_granularity` in the *Reference Manual: Procedures*.

The user must reconnect to SAP ASE.

12.2.3.5 Level 23: Fatal Error: Database Integrity Suspect

Error messages with severity level 23 indicate that the integrity of the entire database is suspect due to previous damage caused by a software or hardware problem. Restart SAP ASE and run `dbcc` diagnostics.

Even when a level 23 error indicates that the entire database is suspect, the damage may be confined to the cache, and the disk itself may be fine. If so, restarting SAP ASE with `startserver` fixes the problem.

12.2.3.6 Level 24: Hardware Error or System Table Corruption

Error messages with severity level 24 reflect a media failure or (in rare cases) the corruption of `sysusages`. The system administrator may have to reload the database. You may need to call your hardware vendor.

12.2.3.7 Level 25: SAP ASE Internal Error

Users do not see level 25 errors, as this level is used only for SAP ASE internal errors.

12.2.3.8 Level 26: Rule Error

Error messages with severity level 26 reflect that an internal locking or synchronization rule has been broken. You must shut down and restart SAP ASE.

12.3 Backup Server Error Logging

Like SAP ASE, Backup Server creates an error log if one does not already exist.

Specify the location of the error log at start-up with the `<error_log_file>` parameter in the runserver file or at the command line. The SAP installer configures the runserver file with `$SYBASE/install` as the location of the error log if you do not choose an alternate location during installation. If you do not specify the location in the runserver file or at the command line, the location of the error log is the directory from which you start Backup Server. Use the `backupserver -V` option (`bcksvr -V` on Windows) to limit the messages printed to the error log. See the sections describing Backup Server in the *Utility Guide*.

Backup Server error messages are in this form:

```
MMM DD YYY: Backup Server:N.N.N.N: Message Text
```

Backup Server message numbers consist of four integers separated by periods, in the form N.N.N.N. Messages in the form N.N.N are sent by Open Server.

The four components of a Backup Server error message are `<major.minor.severity.state>`:

- The <major> component generally indicates the functional area of the Backup Server code where the error occurred:
  - 1 – system errors.
  - 2 – Open Server event errors.
  - 3 – Backup Server remote procedure call errors.
  - 4 – I/O service layer errors.
  - 5 – network data transfer errors.
  - 6 – volume-handling errors.
  - 7 – option-parsing errors.

  Major error categories 1– 6 may result from Backup Server internal errors or a variety of system problems. Major errors in category 7 are almost always due to problems in the options you specified in your dump or load command.
- <minor> numbers are assigned in order within a major category.
- <severity> is:
  - 1 – informational, no user action necessary.
  - 2, 3 – an unexpected condition, possibly fatal to the session, has occurred. The error may have occurred with usage, environment, or internal logic, or any combination of these factors.
  - 4 – an unexpected condition, fatal to the execution of the Backup Server, has occurred. The Backup Server must exit immediately.
- <state> codes have a one-to-one mapping to instances of the error report within the code. If you need to contact Technical Support about Backup Server errors, the state code helps determine the exact cause of the error.

12.4 Killing Processes

A process is a unit of execution carried out by SAP ASE. The `kill` command removes ongoing processes.

Each process is assigned a unique process identification number when it starts. This number is called a `spid`. These numbers are stored, along with other information about each process, in `master..sysprocesses`. Processes running in a parallel-processes environment create child processes, each of which has its own `spid`s. Several processes create and assign `spids`: starting SAP ASE, login tasks, checkpoints, the housekeeper tasks, and so on. You can see most of the information by running `sp_who`.

Running `sp_who` on a single-engine server shows the `sp_who` process running and all other processes that are "runnable" or in one of the sleep states. In multi-engine servers, there can be a process running for each engine.

The most frequent reason for killing a process is that it interferes with other users, and the person responsible for running it is not available. The process may hold locks that block access to database objects, or there may be many sleeping processes occupying the available user connections. A system administrator can kill most running or "runnable" processes, including those that are waiting for:

- An alarm, such as a `waitfor` command
- Network sends or receives
- A lock
- Synchronization messages from another process in a family

SAP ASE allows you to kill processes only if it can cleanly roll back any uncompleted transactions and release all system resources that are used by the process. For processes that are part of a family, killing any of the child processes also kills all other processes in the family. However, it is easiest to kill the parent process. For a family of processes, the `kill` command is detected more quickly if the status of the child processes is `sync sleep`.

This table shows the status values that `sp_who` reports and when the `kill` command takes effect:

Status	Indicates	Effects of `kill` command
`recv sleep`	Waiting on a network read.	Immediate.
`send sleep`	Waiting on a network send.	Immediate.
`alarm sleep`	Waiting on an alarm such as: `waitfor delay "10:00"`	Immediate.
`lock sleep`	Waiting on a lock acquisition.	Immediate.
`sync sleep`	Waiting on a synchronization message from another process in the family.	Immediate. Other processes in the family must also be brought to state in which they can be killed.
`sleeping`	Waiting on a disk I/O, or some other resource. Probably indicates a process that is running, but doing extensive disk I/O	Killed when it "wakes up," usually immediate; a few sleeping processes do not wake up and require a server restart to clear.
`runnable`	In the queue of runnable processes.	Immediate.
`running`	Actively running on one of the server engines.	Immediate.
`infected`	Server has detected serious error condition; extremely rare.	`kill` command not recommended. Server restart probably required to clear process.
`background`	A process, such as a threshold procedure, run by SAP ASE rather than by a user process.	Immediate; use `kill` with extreme care. Recommend a careful check of `sysprocesses` before killing a background process.
`log suspend`	Processes suspended by reaching the last-chance threshold on the log.	Immediate.

Only system administrators can issue the `kill` command; permission to use it cannot be transferred.

The syntax is:

```
kill <spid>
```

You can kill only one process at a time, but you can perform a series of kill commands in a batch. For example:

```
1> kill 7
2> kill 8
3> kill 9
4> go
```

A `kill` command is irreversible and cannot be included in a user-defined transaction. `spid` must be a numeric constant; you cannot use a variable. Here is some sample output from `sp_who`:

```
sp_who
```

```
fid  spid  status      loginame  origname  hostname        blk_spid  dbname
        tempdbname    cmd               block_xloid   threadpool
---  ----  ----------  --------  --------  ----------------  --------  ------
        ----------    ----------------  -----------   ------------------
0       1  recv sleep    howard    howard          svr30eng         0  master
            tempdb    AWAITING COMMAND            0      syb_default_pool
0       2  sleeping        NULL      NULL                           0  master
            tempdb    NETWORK HANDLER             0      syb_default_pool
0       3  sleeping        NULL      NULL                           0  master
            tempdb    DEADLOCK TUNE               0      syb_default_pool
0       4  sleeping        NULL      NULL                           0  master
            tempdb    MIRROR HANDLER              0      syb_default_pool
0       5  sleeping        NULL      NULL                           0  master
            tempdb    CHECKPOINT SLEEP            0      syb_default_pool
0       6  sleeping        NULL      NULL                           0  master
            tempdb    HOUSEKEEPER                 0      syb_default_pool
0       7  recv sleep      bill      bill           bigblue         0  master
            tempdb    AWAITING COMMAND            0      syb_default_pool
0       8  recv sleep    wilbur    wilbur             hazel         0  master
            tempdb    AWAITING COMMAND            0      syb_default_pool
0       9  recv sleep      joan      joan          luv2work         0  master
            tempdb    AWAITING COMMAND            0      syb_default_pool
0      10  running        foote     foote          svr47hum         0  master
            tempdb    SELECT                      0      syb_default_master
(10 rows affected, return status = 0)
```

In the example above, processes 2–6 cannot be killed: they are system processes. The login name NULL and the lack of a host name identify processes them as system processes. NETWORK HANDLER, MIRROR HANDLER, HOUSEKEEPER, and CHECKPOINT SLEEP (or, rarely, CHECKPOINT) always appear in `sp_who` output. AUDIT PROCESS appears if auditing is available.

Processes 1, 8, 9, and 10 can be killed, since they have the status values "recv sleep," "send sleep," "alarm sleep," and "lock sleep."

In `sp_who` output, you cannot tell whether a is "recv sleep" belongs to a user who is using SAP ASE and may be pausing to examine the results of a command, or whether a user has restarted a PC or other terminal, and left a stranded process. Query the `sysprocesses` table to learn more about questionable processes. For example, this query shows the host process ID and client software used by process 8:

```
select hostprocess, program_name
    from sysprocesses
where spid = 8
```

```
hostprocess program_name
----------- ----------------
3993        isql
```

This query, plus the information about the user and host from the `sp_who` results, provides additional information for tracking down the process from the operating system level.

12.4.1 Using kill with statusonly

The `kill ...statusonly` command reports on the progress of a server process ID (spid) in rollback status, but does not terminate the spid.

The `statusonly` report displays:

- The percent of rollback completed
- The estimated length of time, in seconds, before the rollback completes
- Information about the amount of log space scanned, and to be scanned, for a full transaction rollback
- The different types of rollbacks, and reports a subset of information in cases where the rollback is not a full transaction rollback

To track the progress of a rollback, you must run `kill...with statusonly` multiple times:

```
kill <spid> with statusonly
```

Where `<spid>` is the number of the process you are terminating.

For example, to report on the process of the rollback of spid number 13:

```
kill 13 with statusonly
```

```
spid: 13 Transaction rollback in progress. Estimated rollback completion: 17%
Estimated time left: 13 seconds
```

If the rollback of the spid has completed when you issue `kill...statusonly` or if SAP ASE cannot roll back the specified spid, `kill...statusonly` returns the following message:

```
Status report cannot be obtained. KILL spid:<nn> is not in progress.
```

12.4.2 Using sp_lock to Examine Blocking Processes

In addition to `sp_who`, `sp_lock` can help identify processes that are blocking other processes.

If the `blk_spid` column in the `sp_who` report indicates that another process has been blocked while waiting to acquire locks, `sp_lock` can display information about the blocking process.

For example, say process 10 in the `sp_who` output above is blocked by process 7. To see information about process 7, execute:

```
sp_lock 7
```

For more information about locking in SAP ASE, see the *Performance and Tuning Series: Locking and Concurrency Control*.

12.5 Using Shared Memory Dumps

You can configure SAP ASE to dump shared memory to files when specific conditions occur (for example, segmentation faults).

The shared memory dump file provides a snapshot of the server at the time the specified condition occurred, eliminating the difficulty of using a debugger or specialized instrumented code to trap an error at the moment it occurs. Technical support engineers can use this dump file to diagnose the cause of the shared memory dump, and determine how to configure Adaptive Server to avoid the problem in the future.

12.5.1 Configuring Shared Memory Dump Conditions

Use `sp_shmdumpconfig` to define or modify the shared memory dump conditions and system-wide attributes of those conditions.

Context

The syntax for `sp_shmdumpconfig` is:

```
sp_shmdumpconfig "<action>", <type>, <value>, <max_dumps>, <dump_dir>,
<dump_file>, <option1>, <option2>, <option3>, <option4>, <option5>
```

> **i** Note
>
> `sp_shmdumpconfig` uses positional parameters. When setting a parameter that falls to the right of parameters you do not want to set, specify null values for the unset parameters.
>
> `sp_shmdumpconfig` allows you to:
>
> - Create new dump conditions
> - Modify existing dump conditions
> - Drop existing dump conditions
> - Reset the dump count for a dump condition
> - Display the current shared memory dump configuration
> - Set and modify system wide attributes for a dump condition.
>
> Set these parameters to use `sp_shmdumpconfig`:
>
> - `dump on conditions` – determines whether SAP ASE generates a dump of data in shared memory
> - `maximum dump conditions` – sets the maximum number of dump conditions that can be configured
>
> See *Reference Manual: Configuration Parameters*.

This example describes a typical shared memory dump:

Procedure

1. Enable SAP ASE to generate a shared memory dump:

   ```
   sp_configure "dump on conditions", 1
   ```

2. Specify the dump condition using `sp_shmdumpconfig`. This example request a one-time memory dump on a signal 11:

   ```
   sp_shmdumpconfig "add", signal, 11, 1 ,"/sybase"
   ```

3. You can turn off collection by deleting the dump condition after collecting the desired data. For example, to drop the condition for error 631 and disable shared memory dumps:

   ```
   sp_shmdumpconfig "drop", "error", 631
   ```

4. Disable shared memory dumps on SAP ASE:

   ```
   sp_configure "dump on conditions", 0
   ```

12.5.2 System-Wide Default Settings

Each dump condition can inherit attributes from the system-wide default settings. Specifying an attribute value for a condition overrides the system-wide defaults. To update the system-wide default settings, use the `sp_shmdumpconfig` stored procedure with a `type` value of `<defaults>`.

It is typically more convenient to specify any attribute values in the system-wide default settings, and to allow all dump conditions to inherit these values.

SAP recommends that all dump conditions have an explicit dump directory (the `<dump_dir>` parameter). To ensure this, specify a dump directory in the system-wide defaults and allow all dump conditions to inherit this value.

The default attribute values for the system-wide settings are:

```
sp_shmdumpconfig
```

```
Configured Shared Memory Dump Conditions
----------------------------------------
Defaults   ---
Maximum Dumps:          1
Halt Engines:           Halt
Cluster:                Local
Page Cache:             Omit
Procedure Cache:        Include
Unused Space:           Omit
Dump Directory:         $SYBASE
Dump File Name:         Generated File Name
Estimated File Size:    101 MB
Current number of conditions: 0
```

```
Maximum number of conditions: 10
Configurable Shared Memory Dump Configuration Settings
-----------------------------------------------------
Dump on conditions: 0
You must run 'sp_configure 'dump on conditions',1' to
enable the shared memory dump facility.
Number of dump threads: 1
Include errorlog in dump file: 1
Merge parallel files after dump: 1
Server Memory Allocation
Procedure Cache Data Caches Server Memory Total Memory
-------------  ----------- ------------- -----------
         16 MB         9 MB         86 MB      109 MB
```

12.6 Housekeeper functionality

The housekeeper task provides important background tasks.

These tasks include:

- The housekeeper wash, housekeeper garbage collection, and housekeeper chores tasks. `sp_who` displays all three tasks in its output as `HK WASH`, `HK GC`, and `HK CHORES`:

  ```
  sp_who
  ```

  ```
  fid    spid   status       loginame     origname     hostname
  blk_spid  dbname  tempdbname  cmd         block_xloid  threadpool
  ----  -----  ----------    ----------    ---------   ----------
  --------  -------  ---------  -------------  --------- -------------
     0      8    sleeping          NULL         NULL     luv2work
         0   master     tempdb        HK WASH         0  syb_default_pool
     0      9    sleeping          NULL         NULL         NULL
         0   master     tempdb          HK GC         0  syb_default_pool
     0     10    sleeping          NULL         NULL         NULL
         0   master     tempdb      HK CHORES         0  syb_default_pool
  ```

- The general automatic restart of housekeeper-related system tasks: you need not restart the server if these system tasks quit unexpectedly.

A system administrator can change all housekeeper task priorities.

`sp_showpsexe`, as well as `sp_who`, recognizes all three housekeeper names.

For more information about `sp_who` and `sp_showpsexe`, see the *Reference Manual: Procedures*.

12.6.1 Housekeeper wash

Washing buffers is an optional task that, if enabled, runs only during idle times.

You can turn off this task using the configuration parameter `housekeeper free write percent`. The housekeeper wash task is the only housekeeper task for which you use this configuration parameter.

12.6.2 Housekeeper chores

The housekeeper chores task runs only at idle times, and does not use a common configuration parameter.

It manages miscellaneous chores, such as:

- Flushing table statistics.
- Flushing account statistics.
- Handling timeout of detached transactions. You can turn off this task using the configuration parameter `dtm detach timeout period`.
- Checking license usage. You can turn off this task using the configuration parameter `license information`.

12.6.3 Housekeeper Garbage Collection

There are two forms of garbage collection, lazy and aggressive.

These terms describe two distinct tests for finding empty pages:

- Lazy garbage collection refers to an inexpensive test to find empty pages. This test may not be effective during long-running transactions, and empty pages may accumulate. Lazy garbage collection is inexpensive to use, but can lower performance, which is affected by the fragmentation of allocated table space, and by the accumulation of empty pages that must be evaluated during queries.
- Aggressive garbage collection refers to a sophisticated test for empty pages. This test is more expensive than the lazy garbage collection test, because it checks each deleted row in a page to determine whether the deleted transactions are committed.

Use the `enable housekeeper GC` configuration parameter to configure the `delete` command and the housekeeper garbage collection task for aggressive or lazy garbage collection.

The aggressive housekeeper garbage collection self-tunes the frequency with which the housekeeper garbage collection task examines the housekeeper list, so that the frequency of examination matches the rate at which the application generates empty pages.

12.6.3.1 Configure SAP ASE Priority Level

The housekeeper garbage collection task typically operates at the priority level of an ordinary user, competing for CPU time with ordinary user tasks, preventing the list of empty pages from growing faster than the housekeeper can delete them.

However, if SAP ASE is configured for threaded mode, use the `sp_bindexeclass 'sv'` object type to change the server-wide priority setting for the housekeeper wash task. Set the priority to the `EC1`, `EC2`, `EC3` level or define a new user-created execution class. This example sets the housekeeper wash task to the highest, EC1, priority:

```
sp_bindexeclass 'HK WASH', 'sv', NULL, 'EC1'
```

Use `sp_setpsexe` to set the priority level for the housekeeper task for the session if SAP ASE is configured for threaded or process mode.

This example sets the priority level for the housekeeper wash task (with a spid of 8) to `HIGH` for the current session:

```
sp_setpsexe 8, 'priority', 'HIGH'
```

See the *Reference Manual: Procedures*.

12.6.4 Configuring enable housekeeper GC

You can configure the level for SAP ASE garbage collection.

Context

Configure SAP ASE for garbage collection task using:

```
sp_configure "enable housekeeper GC", <value>
```

The valid values for the `enable housekeeper GC` configuration parameter are:

- 0 – disables the housekeeper garbage collection task, but enables lazy garbage collection by the `delete` command. You must use `reorg reclaim_space` to deallocate empty pages. This is the cheapest option with the lowest performance impact, but it may cause performance problems if many empty pages accumulate. SAP recommends that you do not use this value.
- 1 – enables lazy garbage collection, by both the housekeeper garbage collection task and the `delete` command. This is the default value. If more empty pages accumulate than your application allows, consider options 4 or 5. You can use the `optdiag` utility to obtain statistics of empty pages.
- 2 – reserved for future use.
- 3 – reserved for future use.
- 4 – enables aggressive garbage collection for both the housekeeper garbage collection task and the `delete` command. This option is the most effective, but the `delete` command is the most expensive. This option is ideal if the deletes on your dataonly locked tables are in a batch.
- 5 – enables aggressive garbage collection for the housekeeper, and lazy garbage collection by `delete`. This option is less expensive for deletes than option 4. This option is suitable when deletes are caused by concurrent transactions.

Procedure

For example, enter:

```
sp_configure "enable housekeeper GC", 4
```

12.6.4.1 Using the reorg Command

Garbage collection is most effective when you set `enable housekeeper GC` to 4 or 5.

SAP recommends that you set the parameter value to 5. However, if performance considerations prevent setting this parameter to 4 or 5, and you have an accumulation of empty pages, run `reorg` on the affected tables. You can obtain statistics on empty pages through the `optdiag` utility.

When the server is shut down or crashes, requests to deallocate pages that the housekeeper garbage collection task has not yet serviced are lost. These pages, empty but not deallocated by the housekeeper garbage collection task, remain allocated until you remove them by running `reorg`.

See *System Administration Guide: Volume 2 > Using the reorg Command*.

12.7 Shutting Down Servers

A system administrator can shut down SAP ASE or Backup Server using the `shutdown` command.

The syntax is:

```
shutdown [<backup_server_name>] [with {wait|nowait}]
```

The default for the `shutdown` command is `with wait`. That is, `shutdown` and `shutdown with wait` do exactly the same thing.

12.7.1 Shutting Down SAP ASE

`shutdown` minimizes the amount of work that automatic recovery must do when you restart SAP ASE.

If you do not provide a server name, `shutdown` shuts down the SAP ASE you are using. When you issue a `shutdown` command, SAP ASE:

1. Disables logins, except for system administrators
2. Performs a checkpoint in each database, flushing pages that have changed from memory to disk
3. Waits for currently executing SQL statements or procedures to finish

The `with nowait` option shuts down SAP ASE immediately. User processes are aborted, and recovery may take longer after a `shutdown with nowait`. You can help minimize recovery time by issuing a `checkpoint` command before you issue a `shutdown with nowait` command.

12.7.2 Shutting down a Backup Server

Include the Backup Server name when you using the `shutdown` command.

Context

The default is `with wait`, so any dumps or loads in progress complete before the Backup Server process halts. After you issue a `shutdown` command, no new dump or load sessions can be started on the Backup Server.

To see the names of the Backup Servers that are accessible from your SAP ASE, execute `sp_helpserver`. Use the value in the `name` column in the `shutdown` command. You can shut down a Backup Server only if it is:

- Listed in `sysservers` on your SAP ASE, and
- Listed in your local `interfaces` file.

Use `sp_addserver` to add a Backup Server to `sysservers`.

Procedure

1. Run `sp_who` to check for active dumps and load currently running on your Backup Server:

```
SYB_BACKUP...sp_who
spid    status     loginame   hostname     blk      cmd
-----   --------   --------   ----------   ---      --------------
    1   sleeping       NULL         NULL     0      CONNECT HANDLER
    2   sleeping       NULL         NULL     0      DEFERRED HANDLER
    3   runnable       NULL         NULL     0      SCHEDULER
    4   runnable       NULL         NULL     0      SITE HANDLER
    5   running          sa   heliotrope     0      NULL
```

2. Shut down a Backup Server:

```
shutdown SYB_BACKUP
```

12.7.2.1 Using nowait on a Backup Server

The `shutdown <backup_server> with nowait` command shuts down the Backup Server, regardless of current activity. Use it only in severe circumstances. It can leave your dumps or loads in incomplete or inconsistent states.

If you use `shutdown with nowait` during a log or database dump, check for the message indicating that the dump completed. If you did not receive this message, or if you are not sure whether the dump completed, your next dump should be a `dump database`, not a transaction dump. This guarantees that you are not relying on possibly inconsistent dumps.

If you use `shutdown with nowait` during a load of any kind, and you did not receive the message indicating that the load completed, you may not be able to issue further `load transaction` commands on the database. Run a full database consistency check (`dbcc`) on the database before you use it. You may have to reissue the full set of load commands, starting with `load database`.

12.8 Learning about known problems

The release bulletin is a valuable resource for learning about known problems or incompatibilities with SAP ASE and Backup Server. Reading the release bulletin in advance can save you the time and guesswork of troubleshooting known problems.

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.
About the icons:

- Links with the icon  : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
  - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
  - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon  : You are leaving the documentation for that particular SAP product or service and are entering a SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.
The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Gender-Related Language

We try not to use gender-specific word forms and formulations. As appropriate for context and readability, SAP may use masculine word forms to refer to all genders.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

www.sap.com/contactsap

THE BEST RUN SAP