Contents
tempdb database ................................................................. 29
Creating temporary tables .......................................................... 30
sybsecurity database ................................................................. 31
sybsystemdb database ................................................................. 31
sybmgmtdb database ................................................................. 32
pubs2 and pubs3 sample databases ............................................ 32
  Maintaining the sample databases .............................................. 32
  pubs2 image data ................................................................. 33
dbccdb database ........................................................................... 33
sybdiag database ........................................................................... 33
Determining the version of the installation scripts ......................... 33

CHAPTER 3
System Administration for Beginners ............................................. 35
Logical page sizes ........................................................................ 35
Using "test" servers ....................................................................... 36
  Understanding new procedures and features ............................... 36
  Planning resources ..................................................................... 36
  Achieving performance goals ...................................................... 37
Installing Sybase products ............................................................ 37
  Check product compatibility ....................................................... 38
  Install or upgrade Adaptive Server .......................................... 38
  Install additional third-party software ....................................... 38
  Configure and test client connections ....................................... 39
Allocating physical resources ........................................................ 39
  Dedicated versus shared servers .............................................. 40
  Decision support and OLTP applications ................................. 40
  Advance resource planning ..................................................... 40
  Operating system configuration .............................................. 41
Backup and recovery ................................................................. 42
  Keep up-to-date backups of master ......................................... 42
  Automate backup procedures ............................................... 43
  Verify data consistency before backing up a database ............... 44
  Monitor the log size ............................................................... 45
Ongoing maintenance and troubleshooting .................................... 45
  Starting and stopping Adaptive Server .................................... 45
  Viewing and pruning the error log ......................................... 46
Keeping records .......................................................................... 46
  Contact information ............................................................. 46
  Configuration information ..................................................... 47
  Maintenance schedules ......................................................... 47
  System information ............................................................. 48
  Disaster recovery plan ......................................................... 48
Getting more help ....................................................................... 48
CHAPTER 7  Initializing Database Devices........................................................................................................ 261
  What are database devices?......................................................................................................................... 261
  Using the disk init command ................................................................................................................... 262
  disk init syntax ........................................................................................................................................ 262
    disk init examples ................................................................................................................................. 263
  Specifying a logical device name with disk init ...................................................................................... 263
  Specifying a physical device name with disk init .................................................................................. 263
  Choosing a device number for disk init ................................................................................................. 263
  Specifying the device size with disk init ............................................................................................... 263
  Specifying the dsync setting with disk init (optional) ............................................................................. 266
  Using directio to bypass operating system buffer ................................................................................ 268
  Other optional parameters for disk init ................................................................................................. 269
  Getting information about devices ........................................................................................................ 269
  Dropping devices .................................................................................................................................... 271
  Designating default devices .................................................................................................................... 272
  Choosing default and nondefault devices ............................................................................................... 272
  Increasing the size of devices with disk resize ...................................................................................... 273
  Insufficient disk space ............................................................................................................................ 274
  disk resize syntax ................................................................................................................................... 274

CHAPTER 8  Setting Database Options........................................................................................................... 277
  What are database options? ...................................................................................................................... 277
  Using the sp_dboption procedure .......................................................................................................... 277
  Database option descriptions .................................................................................................................. 278
    abort tran on log full ............................................................................................................................. 279
    allow nulls by default ............................................................................................................................ 279
    async log service ................................................................................................................................. 279
    auto identity .......................................................................................................................................... 280
    dbo use only ......................................................................................................................................... 280
    ddl in tran ............................................................................................................................................ 280
    delayed commit ..................................................................................................................................... 282
    identity in nonunique index ................................................................................................................ 282
    no chkpt on recovery ........................................................................................................................... 282
    no free space acctg .............................................................................................................................. 283
    read only ............................................................................................................................................... 283
    select into/bulkcopy/pllssort ................................................................................................................. 283
    single user ............................................................................................................................................. 284
    trunc log on chkpt .................................................................................................................................. 284
    unique auto_identity index .................................................................................................................. 285
  Changing database options ..................................................................................................................... 286
  Viewing the options on a database .......................................................................................................... 287

CHAPTER 9  Configuring Character Sets, Sort Orders, and Languages.................................................... 289
Understanding internationalization and localization ............................... 289
Advantages of internationalized systems ............................................. 290
A sample internationalized system ..................................................... 291
Elements of an internationalized system .......................................... 293
Selecting the character set for your server ....................................... 293
    Unicode .................................................................................. 296
    Selecting the server default character set .................................. 300
Selecting the sort order ...................................................................... 303
    Using sort orders ..................................................................... 304
    Different types of sort orders ................................................. 304
    Selecting the default sort order .............................................. 305
Selecting a language for system messages ........................................ 311
Setting up your server: examples ...................................................... 313
    A Spanish-version server ..................................................... 313
    A U.S.-based company in Japan .............................................. 313
    A Japan-based company with multinational clients .................. 314
Changing the character set, sort order, or message language ............ 315
    Changing the default character set ....................................... 315
    Changing the sort order with a resources file ......................... 316
    Changing the default sort order ............................................ 317
    Reconfiguring the character set, sort order, or message language .. 317
    Unicode examples .................................................................. 318
    Preliminary steps .................................................................... 320
    Setting the user’s default language ........................................ 321
    Recovery after reconfiguration ............................................... 321
    Handling suspect partitions ................................................... 325
Installing date strings for unsupported languages ............................. 326
    Server versus client date interpretation .................................. 326
Internationalization and localization files ......................................... 327
    Types of internationalization files ....................................... 327
    Character sets directory structure ....................................... 328
    Types of localization files .................................................... 329
    Software messages directory structure .................................. 330
    Message languages and global variables ................................. 330

CHAPTER 10 Configuring Client/Server Character Set Conversions .......... 331
Character set conversion in Adaptive Server ................................... 331
Supported character set conversions .............................................. 332
    Conversion for native character sets ..................................... 332
    Conversion in a Unicode system .......................................... 333
Types of character set conversion ................................................... 334
    Adaptive Server direct conversions .................................... 334
    Unicode conversions .......................................................... 334
Which type of conversion do I use? ............................................... 335
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>Diagnosing System Problems</td>
</tr>
<tr>
<td></td>
<td>How Adaptive Server uses error messages</td>
</tr>
<tr>
<td></td>
<td>Error messages and message numbers</td>
</tr>
<tr>
<td></td>
<td>Variables in error message text</td>
</tr>
<tr>
<td></td>
<td>Adaptive Server error logging</td>
</tr>
<tr>
<td></td>
<td>Error log format</td>
</tr>
<tr>
<td></td>
<td>Severity levels</td>
</tr>
<tr>
<td></td>
<td>Severity levels 10–18</td>
</tr>
<tr>
<td></td>
<td>Severity levels 19–26</td>
</tr>
<tr>
<td></td>
<td>Reporting errors</td>
</tr>
<tr>
<td></td>
<td>Backup Server error logging</td>
</tr>
<tr>
<td></td>
<td>Killing processes</td>
</tr>
<tr>
<td></td>
<td>Using kill with statusonly</td>
</tr>
<tr>
<td></td>
<td>Using sp_lock to examine blocking processes</td>
</tr>
<tr>
<td></td>
<td>Housekeeper functionality</td>
</tr>
<tr>
<td></td>
<td>Three housekeepers</td>
</tr>
<tr>
<td></td>
<td>Housekeeper wash</td>
</tr>
<tr>
<td></td>
<td>Housekeeper chores</td>
</tr>
<tr>
<td></td>
<td>Housekeeper garbage collection</td>
</tr>
<tr>
<td></td>
<td>Configuring enable housekeeper GC</td>
</tr>
<tr>
<td></td>
<td>Configuring Adaptive Server to save SQL batch text</td>
</tr>
<tr>
<td></td>
<td>Allocating memory for batch text</td>
</tr>
<tr>
<td></td>
<td>SQL commands not represented by text</td>
</tr>
<tr>
<td></td>
<td>Viewing the query plan of a SQL statement</td>
</tr>
<tr>
<td></td>
<td>Viewing a nested procedure</td>
</tr>
<tr>
<td></td>
<td>Shutting down servers</td>
</tr>
<tr>
<td></td>
<td>Shutting down Adaptive Server</td>
</tr>
<tr>
<td></td>
<td>Shutting down a Backup Server</td>
</tr>
<tr>
<td></td>
<td>Learning about known problems</td>
</tr>
</tbody>
</table>
# PART 2 SECURITY ADMINISTRATION

## CHAPTER 12 Introduction to Security

- Introduction to security .......................................................... 375
- What is “information security?” .................................................. 376
- Information security standards ................................................... 377
- Adaptive Server version 15.0.1 available for Common Criteria configuration ................................................................. 377
- FIPS 140-2 Validated cryptographic module ................................ 378

## CHAPTER 13 Getting Started With Security Administration in Adaptive Server

- General process of security administration .................................. 379
- Recommendations for setting up security .................................... 381
- Using the “sa” login .................................................................. 381
- Changing the “sa” login password .............................................. 381
- When to enable auditing ............................................................. 381
- Assigning login names ............................................................... 382
- An example of setting up security ................................................ 382
- Introduction to Security Features in Adaptive Server .................. 383
- Identification and authentication ................................................ 384
- External authentication .............................................................. 385
- Managing remote servers .......................................................... 385
- Discretionary access controls ..................................................... 385
- Row-level access control ........................................................... 386
- Division of roles ....................................................................... 387
- Role hierarchy ......................................................................... 387
- Mutual exclusivity ................................................................. 387
- Auditing for accountability ......................................................... 388
- Confidentiality of data ............................................................. 389
- Password-Protected Database Backup ....................................... 389

## CHAPTER 14 Managing Adaptive Server Logins, Database Users, and Client Connections

- Overview .................................................................................. 391
- Choosing and creating a password .............................................. 392
- Adding logins to Adaptive Server .............................................. 393
- Login failure to Adaptive Server .............................................. 394
- Creating groups ........................................................................ 395
- Adding users to databases ........................................................... 396
- Adding a “guest” user to a database .......................................... 398
- Adding a guest user to the server ............................................... 400
- Adding remote users ................................................................. 400
Getting information about login accounts ....................................... 430
Getting information about database users ......................................... 431
Finding user names and IDs ............................................................ 431
Displaying information about roles .................................................. 433
Establishing a password and login policy .......................................... 436
Setting and changing the maximum login attempts ......................... 437
Logging in after lost password ........................................................ 439
Locking and unlocking logins and roles .......................................... 440
Displaying password information .................................................... 441
Checking passwords for at least one digit ....................................... 442
Setting and changing minimum password length .............................. 443
Password complexity checks ........................................................... 445
Enabling custom password checks .................................................. 453
Setting the login and role expiration interval for a password .......... 455
Securing login passwords stored on disk and in memory ................. 461
Using only the SHA-256 algorithm .................................................. 462
Character set considerations for passwords ..................................... 465
Upgrade and downgrade behavior ................................................... 466
Expiring passwords when allow password downgrade is set to 0 ... 472
Last login and locking inactive accounts .......................................... 474
Using passwords in a high availability environment ......................... 475
Monitoring license use ........................................................................ 477
How licenses are counted ................................................................ 478
Configuring the License Use Manager to monitor user licenses ....... 478
Monitoring license use with the housekeeper task ......................... 478
Logging the number of user licenses ............................................... 479
Getting information about usage: chargeback accounting .............. 480
Reporting current usage statistics .................................................... 480
Specifying the interval for adding accounting statistics ................. 481

CHAPTER 15  Managing Remote Servers ................................................................. 483
Overview ................................................................................................. 483
Managing remote servers ................................................................. 484
Adding a remote server ...................................................................... 485
Managing remote server names ......................................................... 486
Setting server connection options ..................................................... 487
Getting information about servers ..................................................... 489
Dropping remote servers ................................................................. 489
Adding remote logins ....................................................................... 490
Mapping users’ server IDs ................................................................. 490
Mapping remote logins to particular local names ......................... 491
Mapping all remote logins to one local name ................................... 491
Keeping remote login names for local servers ............................... 492
Example of remote user login mapping .......................................... 492
Password checking for remote users ........................................................ 494
Effects of using the untrusted mode ......................................................... 494
Getting information about remote logins ................................................. 495
Configuration parameters for remote logins ............................................ 495
  Allowing remote access.................................................................... 496
  Controlling the number of active user connections ......................... 496
  Controlling the number of remote sites .......................................... 497
  Controlling the number of active remote connections ..................... 497
  Controlling number of preread packets ............................................ 497

CHAPTER 16 External Authentication .......................................................... 499
  Overview ....................................................................................... 499
  Configuring Adaptive Server for Network-Based Security ................. 500
    How applications use security services ...................................... 500
    Security services and Adaptive Server ....................................... 501
    Administering network-based security ....................................... 502
    Setting up configuration files for security .................................. 503
    Identifying users and servers to the security mechanism ............ 509
    Configuring Adaptive Server for security .................................... 510
    Rebooting the server to activate security services ....................... 514
    Adding logins to support unified login ...................................... 515
    Establishing security for remote procedures .............................. 516
    Connecting to the server and using the security services .......... 523
    Getting information about available security services ............... 526
    Using Kerberos ........................................................................ 527
    Using principal names ................................................................ 533
  Configuring Adaptive Server for LDAP User Authentication ............ 539
    Composed DN algorithm ............................................................. 540
    Searched DN algorithm ............................................................... 540
    Configuring LDAP ...................................................................... 541
    LDAP user authentication administration .................................. 542
    Adaptive Server logins and LDAP user accounts ....................... 546
    Secondary lookup server support .............................................. 546
    LDAP server state transitions .................................................... 548
    LDAP user authentication tuning .............................................. 550
    Adding tighter controls on login mapping ................................... 551
    Troubleshooting LDAP user authentication errors ..................... 554
    Configuring LDAPS ................................................................... 555
  Configuring Adaptive Server for authentication using PAM .............. 556
    Enabling PAM in Adaptive Server .............................................. 557
    Enhanced login controls ............................................................. 559
    Forcing authentication ............................................................... 560
    Mapping logins using sp_maplogin ............................................ 561
<table>
<thead>
<tr>
<th>Chapter 17</th>
<th>Managing User Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>563</td>
</tr>
<tr>
<td>Permissions for creating databases</td>
<td>565</td>
</tr>
<tr>
<td>Changing database ownership</td>
<td>565</td>
</tr>
<tr>
<td>Database owner privileges</td>
<td>566</td>
</tr>
<tr>
<td>Database object owner privileges</td>
<td>567</td>
</tr>
<tr>
<td>Other database user privileges</td>
<td>568</td>
</tr>
<tr>
<td>Permissions on system procedures</td>
<td>568</td>
</tr>
<tr>
<td>Granting and revoking permissions</td>
<td>569</td>
</tr>
<tr>
<td>Object access permissions</td>
<td>569</td>
</tr>
<tr>
<td>Granting permissions on functions</td>
<td>578</td>
</tr>
<tr>
<td>Granting and revoking permissions to execute commands</td>
<td>578</td>
</tr>
<tr>
<td>Granting permissions on dbcc commands</td>
<td>582</td>
</tr>
<tr>
<td>Permissions on system tables</td>
<td>583</td>
</tr>
<tr>
<td>Combining grant and revoke statements</td>
<td>586</td>
</tr>
<tr>
<td>Understanding permission order and hierarchy</td>
<td>587</td>
</tr>
<tr>
<td>Grant dbcc and set proxy issue warning for fipsflagger</td>
<td>588</td>
</tr>
<tr>
<td>Granting and revoking roles</td>
<td>588</td>
</tr>
<tr>
<td>Granting roles</td>
<td>588</td>
</tr>
<tr>
<td>Understanding grant and roles</td>
<td>589</td>
</tr>
<tr>
<td>Revoking roles</td>
<td>590</td>
</tr>
<tr>
<td>Acquiring the permissions of another user</td>
<td>590</td>
</tr>
<tr>
<td>Using setuser</td>
<td>590</td>
</tr>
<tr>
<td>Using proxy authorization</td>
<td>591</td>
</tr>
<tr>
<td>Reporting on permissions</td>
<td>595</td>
</tr>
<tr>
<td>Querying the sysprotects table for proxy authorization</td>
<td>596</td>
</tr>
<tr>
<td>Displaying information about users and processes</td>
<td>596</td>
</tr>
<tr>
<td>Reporting permissions on database objects or users</td>
<td>597</td>
</tr>
<tr>
<td>Reporting permissions on specific tables</td>
<td>598</td>
</tr>
<tr>
<td>Reporting permissions on specific columns</td>
<td>599</td>
</tr>
<tr>
<td>Using views and stored procedures as security mechanisms</td>
<td>600</td>
</tr>
<tr>
<td>Using views as security mechanisms</td>
<td>600</td>
</tr>
<tr>
<td>Using stored procedures as security mechanisms</td>
<td>602</td>
</tr>
<tr>
<td>Understanding ownership chains</td>
<td>603</td>
</tr>
<tr>
<td>Permissions on triggers</td>
<td>607</td>
</tr>
<tr>
<td>Using row-level access control</td>
<td>607</td>
</tr>
<tr>
<td>Access rules</td>
<td>608</td>
</tr>
<tr>
<td>Using the Application Context Facility</td>
<td>617</td>
</tr>
<tr>
<td>Creating and using application contexts</td>
<td>620</td>
</tr>
<tr>
<td>SYS_SESSION system application context</td>
<td>624</td>
</tr>
<tr>
<td>Solving a problem using an access rule and ACF</td>
<td>625</td>
</tr>
<tr>
<td>Using login triggers</td>
<td>627</td>
</tr>
<tr>
<td>Exporting set options from a login trigger</td>
<td>635</td>
</tr>
<tr>
<td>Setting global login triggers</td>
<td>637</td>
</tr>
</tbody>
</table>
CHAPTER 18  Auditing ................................................................. 639
Introduction to auditing in Adaptive Server .................. 639
Correlating Adaptive Server and operating system audit records .... 640
The audit system ......................................................... 640
Installing and setting up auditing .................................. 644
Installing the audit system ........................................... 645
Setting up audit trail management ................................ 648
Setting up transaction log management ...................... 654
Enabling and disabling auditing .................................... 656
Single-table auditing ................................................... 657
Restarting auditing ...................................................... 660
Setting global auditing options .................................... 661
Auditing options: types and requirements .................. 661
Hiding system stored procedure and command password parameters 669
Determining current auditing settings .......................... 669
Adding user-specified records to the audit trail .......... 669
Querying the audit trail .............................................. 671
Understanding the audit tables .................................... 671
Reading the extrainfo column ..................................... 672
Monitoring failed login attempts ............................... 683
Auditing login failures ............................................... 683

CHAPTER 19  Confidentiality of Data ............................................. 687
Secure Sockets Layer (SSL) in Adaptive Server ............. 687
Internet communications overview ............................ 688
SSL in Adaptive Server ............................................. 690
Enabling SSL ............................................................. 694
Performance ............................................................ 700
Cipher Suites ............................................................ 700
Setting SSL cipher suite preferences .......................... 701
Kerberos confidentiality ............................................... 707
Dumping and loading databases with password protection .... 707
Passwords and earlier versions of Adaptive Server ........ 708
Passwords and character sets ..................................... 708

Index ................................................................................. 709

xiv  Adaptive Server Enterprise
About This Book

This manual, the *System Administration Guide: Volume 1*, describes how to administer and control Sybase® Adaptive Server® Enterprise databases independent of any specific database application.

This manual is for Sybase system administrators and database owners.

This guide (System Administration Guide Volume 1) is comprised of two parts: Part One describes the concepts of system administration. Part Two discusses security administration issues. Part One includes the following chapters:

- Chapter 1, “Overview of System Administration,” describes the structure of the Sybase system.
- Chapter 2, “System and Optional Databases,” discusses the contents and function of the Adaptive Server system databases.
- Chapter 3, “System Administration for Beginners,” summarizes important tasks that new system administrators must perform.
- Chapter 4, “Introduction to the Adaptive Server Plug-in for Sybase Central,” describes how to start and use Sybase Central, a graphical user interface for managing Adaptive Server.
- Chapter 5, “Setting Configuration Parameters,” summarizes the configuration parameters that you set with `sp_configure`, which control many aspects of Adaptive Server behavior.
- Chapter 6, “Overview of Disk Resource Issues,” discusses Adaptive Server and Backup Server™ error handling and how to shut down servers and kill user processes.
- Chapter 8, “Setting Database Options,” describes how to set database options.
Chapter 9, “Configuring Character Sets, Sort Orders, and Languages,” discusses international issues, such as the files included in the Language Modules and how to configure an Adaptive Server language, sort order, and character set.

Chapter 10, “Configuring Client/Server Character Set Conversions,” discusses character set conversion between Adaptive Server and clients in a heterogeneous environment.

Chapter 11, “Diagnosing System Problems,” discusses Adaptive Server and Backup Server error handling and shows how to shut down servers and kill user processes.

Part Two includes these chapters:

- Chapter 12, “Introduction to Security,” introduces you to security concepts.
- Chapter 15, “Managing Remote Servers,” discusses the steps the system administrator and System Security Officer of each Adaptive Server must execute to enable remote procedure calls (RPCs).
- Chapter 16, “External Authentication,” describes the network-based security services that enable you to authenticate users and protect data transmitted among machines on a network.
- Chapter 17, “Managing User Permissions,” describes the use and implementation of user permissions.
- Chapter 18, “Auditing,” describes how to set up auditing for your installation.
- Chapter 19, “Confidentiality of Data,” describes how to configure Adaptive Server to ensure that all data is secure and confidential.

**Related documents**

The Adaptive Server® Enterprise documentation set consists of the following:

- The release bulletin for your platform – contains last-minute information that was too late to be included in the books.
A more recent version of the release bulletin may be available on the World Wide Web. To check for critical product or document information that was added after the release of the product CD, use the Sybase Technical Library.

- The Installation Guide for your platform – describes installation, upgrade, and configuration procedures for all Adaptive Server and related Sybase products.

- What’s New in Adaptive Server Enterprise? – describes the new features in Adaptive Server version 15.0, the system changes added to support those features, and changes that may affect your existing applications.

- ASE Replicator User’s Guide – describes how to use the Adaptive Server Replicator feature of Adaptive Server to implement basic replication from a primary server to one or more remote Adaptive Servers.

- Component Integration Services User’s Guide – explains how to use the Adaptive Server Component Integration Services feature to connect remote Sybase and non-Sybase databases.


- Glossary – defines technical terms used in the Adaptive Server documentation.


- Java in Adaptive Server Enterprise – describes how to install and use Java classes as datatypes, functions, and stored procedures in the Adaptive Server database.

- Job Scheduler User’s Guide – provides instructions on how to install and configure, and create and schedule jobs on a local or remote Adaptive Server using the command line or a graphical user interface (GUI).

- Messaging Service User’s Guide – describes how to use Real Time Messaging Services to integrate TIBCO Java Message Service and IBM WebSphere MQ messaging services with all Adaptive Server database applications.


Performance and Tuning Series – a series of books that explain how to tune Adaptive Server for maximum performance:

- Basics – the basics for understanding and investigating performance questions in Adaptive Server.
- Locking and Concurrency Control – describes how the various locking schemas can be used for improving performance in Adaptive Server, and how to select indexes to minimize concurrency.
- Query Processing and Abstract Plans – describes how the optimizer processes queries and how abstract plans can be used to change some of the optimizer plans.
- Physical Database Tuning – describes how to manage physical data placement, space allocated for data, and the temporary databases.
- Improving Performance with Statistical Analysis – describes how Adaptive Server stores and displays statistics, and how to use the set statistics command to analyze server statistics.
- Using the Monitoring Tables – describes how to query Adaptive Server’s monitoring tables for statistical and diagnostic information.

Quick Reference Guide – provides a comprehensive listing of the names and syntax for commands, functions, system procedures, extended system procedures, data types, and utilities in a pocket-sized book (regular size when viewed in PDF format).

Reference Manual – is a series of four books that contains the following detailed Transact-SQL information:

- Building Blocks – Transact-SQL datatypes, functions, global variables, expressions, identifiers and wildcards, and reserved words.
- Commands – Transact-SQL commands.
• *Procedures* – Transact-SQL system procedures, catalog stored procedures, system extended stored procedures, and dbcc stored procedures.

• *Tables* – Transact-SQL system tables and dbcc tables.

• *System Administration Guide* –
  - *Volume 1* – provides an introduction to the basics of system administration, including a description of configuration parameters, resource issues, character sets, sort orders, and diagnosing system problems. The second part of this book is an in-depth description of security administration.
  - *Volume 2* – includes instructions and guidelines for managing physical resources, mirroring devices, configuring memory and data caches, managing multiprocessor servers and user databases, mounting and unmounting databases, creating and using segments, using the `reorg` command, and checking database consistency. The second half of this book describes how to back up and restore system and user databases.

• *System Tables Diagram* – illustrates system tables and their entity relationships in a poster format. Full-size available only in print version; a compact version is available in PDF format.

• *Transact-SQL User’s Guide* – documents Transact-SQL, the Sybase enhanced version of the relational database language. This manual serves as a textbook for beginning users of the database management system. This manual also contains descriptions of the pubs2 and pubs3 sample databases.

• *Troubleshooting Series (for release 15.0)* –
  - *Troubleshooting: Error Messages Advanced Resolutions* – contains troubleshooting procedures for problems that you may encounter when using Sybase® Adaptive Server® Enterprise. The problems addressed here are those which the Sybase Technical Support staff hear about most often
  - *Troubleshooting and Error Messages Guide* – contains detailed instructions on how to resolve the most frequently occurring Adaptive Server error messages. Most of the messages presented here contain error numbers (from the `master.sysmessages` table), but some error messages do not have error numbers, and occur only in Adaptive Server’s error log.
• **User Guide for Encrypted Columns** – describes how configure and use encrypted columns with Adaptive Server.

• **Using Adaptive Server Distributed Transaction Management Features** – explains how to configure, use, and troubleshoot Adaptive Server DTM features in distributed transaction processing environments.

• **Using Sybase Failover in a High Availability System** – provides instructions for using Sybase Failover to configure an Adaptive Server as a companion server in a high availability system.

• **Unified Agent and Agent Management Console** – describes the Unified Agent, which provides runtime services to manage, monitor and control distributed Sybase resources.

• **Utility Guide** – documents the Adaptive Server utility programs, such as `isql` and `bcp`, which are executed at the operating system level.

• **Web Services User’s Guide** – explains how to configure, use, and troubleshoot Web Services for Adaptive Server.

• **XA Interface Integration Guide for CICS, Encina, and TUXEDO** – provides instructions for using the Sybase DTM XA interface with X/Open XA transaction managers.

• **XML Services in Adaptive Server Enterprise** – describes the Sybase native XML processor and the Sybase Java-based XML support, introduces XML in the database, and documents the query and mapping functions that comprise XML Services.

Other sources of information

Use the Sybase Getting Started CD, the SyBooks CD, and the Sybase Product Manuals Web site to learn more about your product:

• The Getting Started CD contains release bulletins and installation guides in PDF format, and may also contain other documents or updated information not included on the SyBooks CD. It is included with your software. To read or print documents on the Getting Started CD, you need Adobe Acrobat Reader, which you can download at no charge from the Adobe Web site using a link provided on the CD.

• The SyBooks CD contains product manuals and is included with your software. The Eclipse-based SyBooks browser allows you to access the manuals in an easy-to-use, HTML-based format.

Some documentation may be provided in PDF format, which you can access through the PDF directory on the SyBooks CD. To read or print the PDF files, you need Adobe Acrobat Reader.
Refer to the *SyBooks Installation Guide* on the Getting Started CD, or the *README.txt* file on the SyBooks CD for instructions on installing and starting SyBooks.

- The Sybase Product Manuals Web site is an online version of the SyBooks CD that you can access using a standard Web browser. In addition to product manuals, you will find links to EBFs/Maintenance, Technical Documents, Case Management, Solved Cases, newsgroups, and the Sybase Developer Network.

  To access the Sybase Product Manuals Web site, go to Product Manuals at [http://www.sybase.com/support/manuals/](http://www.sybase.com/support/manuals/).

---

**Sybase certifications on the Web**

Technical documentation at the Sybase Web site is updated frequently.

- **Finding the latest information on product certifications**
  2. Click Certification Report.
  3. In the Certification Report filter select a product, platform, and timeframe and then click Go.
  4. Click a Certification Report title to display the report.

- **Finding the latest information on component certifications**
  2. Either select the product family and product under Search by Base Product; or select the platform and product under Search by Platform.
  3. Select Search to display the availability and certification report for the selection.

- **Creating a personalized view of the Sybase Web site (including support pages)**
  Set up a MySybase profile. MySybase is a free service that allows you to create a personalized view of Sybase Web pages.
  2. Click MySybase and create a MySybase profile.
Finding the latest information on EBFs and software maintenance

2. Select EBFs/Maintenance. If prompted, enter your MySybase user name and password.
3. Select a product.
4. Specify a time frame and click Go. A list of EBF/Maintenance releases is displayed.
   Padlock icons indicate that you do not have download authorization for certain EBF/Maintenance releases because you are not registered as a Technical Support Contact. If you have not registered, but have valid information provided by your Sybase representative or through your support contract, click Edit Roles to add the “Technical Support Contact” role to your MySybase profile.
5. Click the Info icon to display the EBF/Maintenance report, or click the product description to download the software.

Conventions

The following sections describe conventions used in this manual.

SQL is a free-form language. There are no rules about the number of words you can put on a line or where you must break a line. However, for readability, all examples and most syntax statements in this manual are formatted so that each clause of a statement begins on a new line.Clauses that have more than one part extend to additional lines, which are indented. Complex commands are formatted using modified Backus Naur Form (BNF) notation.

Table 1 shows the conventions for syntax statements that appear in this manual:

<table>
<thead>
<tr>
<th>Element</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command names, procedure names, utility names, and other keywords display in sans serif font</td>
<td>select, sp_configure</td>
</tr>
<tr>
<td>Database names and datatypes are in sans serif font.</td>
<td>master, database</td>
</tr>
<tr>
<td>Book names, file names, variables, and path names are in italics.</td>
<td>System Administration Guide, sql.ini file, column_name, $SYBASE/ASE directory</td>
</tr>
</tbody>
</table>
Variables—or words that stand for values that you fill in—when they are part of a query or statement, are in italics in Courier font.

```
select column_name
from table_name
where search_conditions
```

Type parentheses as part of the command.

```
compute row_aggregate (column_name)
```

Double colon, equals sign indicates that the syntax is written in BNF notation. Do not type this symbol. Indicates “is defined as”.

```
::=
```

Curly braces mean that you must choose at least one of the enclosed options. Do not type the braces.

```
{cash, check, credit}
```

Brackets mean that to choose one or more of the enclosed options is optional. Do not type the brackets.

```
[cash | check | credit]
```

The comma means you may choose as many of the options shown as you want. Separate your choices with commas as part of the command.

```
cash, check, credit
```

The pipe or vertical bar (|) means you may select only one of the options shown.

```
cash | check | credit
```

An ellipsis (...) means that you can repeat the last unit as many times as you like.

```
buy thing = price [cash | check | credit] [, thing = price [cash | check | credit]]...
```

You must buy at least one thing and give its price. You may choose a method of payment: one of the items enclosed in square brackets. You may also choose to buy additional things: as many of them as you like. For each thing you buy, give its name, its price, and (optionally) a method of payment.

• Syntax statements (displaying the syntax and all options for a command) appear as follows:

```
sp_dropdevice [device_name]
```

For a command with more options:

```
select column_name
from table_name
where search_conditions
```

In syntax statements, keywords (commands) are in normal font and identifiers are in lowercase. Italic font shows user-supplied words.

• Examples showing the use of Transact-SQL commands are printed like this:

```
select * from publishers
```

• Examples of output from the computer appear as follows:
In this manual, most of the examples are in lowercase. However, you can disregard case when typing Transact-SQL keywords. For example, SELECT, Select, and select are the same.

Adaptive Server's sensitivity to the case of database objects, such as table names, depends on the sort order installed on Adaptive Server. You can change case sensitivity for single-byte character sets by reconfiguring the Adaptive Server sort order. For more information, see the System Administration Guide.

This document is available in an HTML version that is specialized for accessibility. You can navigate the HTML with an adaptive technology such as a screen reader, or view it with a screen enlarger.

Adaptive Server HTML documentation has been tested for compliance with U.S. government Section 508 Accessibility requirements. Documents that comply with Section 508 generally also meet non-U.S. accessibility guidelines, such as the World Wide Web Consortium (W3C) guidelines for Web sites.

Note You might need to configure your accessibility tool for optimal use. Some screen readers pronounce text based on its case; for example, they pronounce ALL UPPERCASE TEXT as initials, and MixedCase Text as words. You might find it helpful to configure your tool to announce syntax conventions. Consult the documentation for your tool.

For information about how Sybase supports accessibility, see Sybase Accessibility at http://www.sybase.com/accessibility. The Sybase Accessibility site includes links to information on Section 508 and W3C standards.

If you need help Each Sybase installation that has purchased a support contract has one or more designated people who are authorized to contact Sybase Technical Support. If you cannot resolve a problem using the manuals or online help, please have the designated person contact Sybase Technical Support or the Sybase subsidiary in your area.
The following chapters introduce the concepts of system administration in Adaptive Server:

- Chapter 1, “Overview of System Administration,” describes the structure of the Sybase system.
- Chapter 2, “System and Optional Databases,” discusses the contents and function of the Adaptive Server system databases.
- Chapter 3, “System Administration for Beginners,” summarizes important tasks that new system administrators must perform.
- Chapter 4, “Introduction to the Adaptive Server Plug-in for Sybase Central,” describes how to start and use Sybase Central, a graphical user interface for managing Adaptive Server.
- Chapter 5, “Setting Configuration Parameters,” summarizes the configuration parameters that you set with sp_configure, which control many aspects of Adaptive Server behavior.
- Chapter 8, “Setting Database Options,” describes how to set database options.
Chapter 9, “Configuring Character Sets, Sort Orders, and Languages,” discusses international issues, such as the files included in the Language Modules and how to configure an Adaptive Server language, sort order, and character set.

Chapter 10, “Configuring Client/Server Character Set Conversions,” discusses character set conversion between Adaptive Server and clients in a heterogeneous environment.

Chapter 11, “Diagnosing System Problems,” discusses Adaptive Server and Backup Server™ error handling and how to shut down servers and kill user processes.
CHAPTER 1
Overview of System Administration

This chapter introduces the basic topics of Adaptive Server system administration.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaptive Server administration tasks</td>
<td>3</td>
</tr>
<tr>
<td>System tables</td>
<td>9</td>
</tr>
<tr>
<td>System procedures</td>
<td>12</td>
</tr>
<tr>
<td>System extended stored procedures</td>
<td>15</td>
</tr>
<tr>
<td>Logging error messages</td>
<td>15</td>
</tr>
<tr>
<td>Connecting to Adaptive Server</td>
<td>16</td>
</tr>
<tr>
<td>Security features available in Adaptive Server</td>
<td>21</td>
</tr>
</tbody>
</table>

Adaptive Server administration tasks

Administering Adaptive Server includes tasks such as:

- Installing Adaptive Server and Backup Server
- Creating and managing Adaptive Server login accounts
- Granting roles and permissions to Adaptive Server users
- Managing and monitoring the use of disk space, memory, and connections
- Backing up and restoring databases
- Diagnosing system problems
- Configuring Adaptive Server 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.
Adaptive Server administration tasks

Although a system administrator concentrates on tasks that are independent of the applications running on Adaptive Server, he or she is likely to be the person with the best overview of all the applications. For this reason, a system administrator can advise application designers about the data that already exists on Adaptive Server, 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 may consult certain sections of this book. Similarly, system administrators and database owners will use the *Transact-SQL User’s Guide* (especially the chapters on data definition, stored procedures, and triggers). Both system administrators and application designers will use the *Performance and Tuning Series*.

Roles required for system administration tasks

Many of the commands and procedures discussed in this manual require the system administrator or System Security Officer role. Other sections in this manual are relevant to database owners.

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

- **system administrator** – by default the system administrator (sa) has the following roles:
  - sa_role
  - sso_role
  - oper_role
  - sybase_ts_role

The system administrator’s tasks include:

- Managing disk storage
- Monitoring Adaptive Server’s automatic recovery procedure
- Fine-tuning Adaptive Server 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 to Adaptive Server users
• 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 Adaptive Server to use network-based security services
  • Managing the audit system

• **Operator** – a user who can back up and load 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 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, Adaptive Server does not check the DAC permissions.

In addition, two kinds of object owners have special status because of the objects they own. These ownership types are:

• Database owner
• Database object owner
Database owner

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

A database owner logs in to Adaptive Server using his or her assigned login name and password. In other databases, that owner is known by his or her regular user name. In the database, Adaptive Server recognizes the user as having the “dbo” account.

A database owner can:

- Run the system procedure `sp_adduser` to allow other Adaptive Server 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 Chapter 14, “Managing Adaptive Server Logins, Database Users, and Client Connections.” Granting permissions to users is discussed in Chapter 17, “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.

*Note* Because the database owner role is so powerful, the system administrator should plan carefully who should own databases in the server. The System Security Officer should consider auditing the database activity of all database owners.

Database object owner

A **Database object owner** is a user who creates a database object. **Database objects** are 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.

**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.

### Using *isql* to perform system administration tasks

This book assumes that the system administration tasks described in this guide are performed using the command line utility *isql*. This section provides some basic information about using *isql*. For complete information, see the *Utility Guide*.

You can also use the graphic tool Sybase Central™ to perform many of the tasks described in this book, as described in “Using Sybase Central for system administration tasks” on page 8.

### Starting *isql*

To start *isql* on most platforms, type this command at an operating system prompt, where `username` is the user name of the system administrator:

```
isql -U username
```

Adaptive Server prompts you for your password.

**Note** Do not use the `-P` option of *isql* to specify your password; another user might then see your password.

You can use *isql* in command line mode to enter many of the Transact-SQL examples in this manual.
Adaptive Server administration tasks

Entering statements

The statements that you enter in isql can span several lines. isql does not process statements until you type “go” on a separate line. For example:

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

The examples in this manual 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.

Saving and reusing statements

This manual frequently suggests that you 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 -ifilename
```

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 -Ppassword option at the command line, or,
- Include the named user’s password on the first line of the input file.

Using Sybase Central for system administration tasks

You can accomplish many of the system administration tasks detailed in this book with Sybase Central, a graphic tool that comes with Adaptive Server.

Here are some of the tasks you can use Sybase Central for:
- Initializing database devices
- Setting configuration parameters
- Viewing the amount of free log space in a database
- Generating data definition language (DDL)
- Creating logins
- Adding remote servers
- Creating databases
- Creating stored procedures
- Defining roles
- Adding data caches
- Setting database options
- Backing up and restoring databases

You can also use the Monitor Viewer feature of Sybase Central to access Adaptive Server Monitor™. Sybase Central also comes with extensive online help.

You can use the Sybase Central DDL-generation feature to record your work to Transact-SQL scripts. The DDL-generation feature lets you save to a script the actions you performed in an entire server or within a specific database.

System tables

The master database contains system tables that keep track of information about Adaptive Server. In addition, each database (including the master database) contains system tables that keep track of information specific to that database.

All the Adaptive Server-supplied tables in the master database (Adaptive Server’s controlling database) are considered system tables. Each user database is created with a subset of these system tables. The system tables may also be referred to as the data dictionary or the system catalogs.
A master database and its tables are created automatically when Adaptive Server 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. An explanation of the system tables and their columns is in the Reference Manual.

### Querying the system tables

You can query system tables in the same manner as any other tables. For example, the following statement returns the names of all the triggers in the database:

```sql
select name
from sysobjects
where type = 'TR'
```

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

Here are the system procedures that provide information from the system tables:

- sp_commonkey
- sp_configure
- sp_countmediatada
- sp_dboption
- sp_estspace
- sp_help
- sp_helpartition
- sp_helpcache
- sp_helconfig
- sp_helpconstraint
- sp_helddb
- sp_helpdevice
- sp_helgroup
- sp_helindex
- sp_helpjava
- sp_helpjoins
- sp_helpkey
- sp_helplanguage
- sp_helpremotelogin
- sp_help_resource_limit
- sp_helpprotect
- sp_helpsegment
- sp_helserver
- sp_helpsort
- sp_helptext
- sp_helphelp
- sp_helpthreshold
- sp_heluser
- sp_lock
- sp_monitor
- sp_monitorconfig
- sp_showcontrolinfo
- sp_showexeclass
- sp_showplan
- sp_spaceused
- sp_who
- sp_help_resource_limit
• sp_helplog

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

**Keys in system tables**

Primary, foreign, and common keys for the 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.

The Adaptive Server System Tables Diagram included with Adaptive Server shows the relationships between columns in the system tables.

**Updating system tables**

The Adaptive Server 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 Sybase Technical Support or by an instruction in the Error Messaging and Troubleshooting Guide or in this manual.

When 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 were affected by the command and that the data was changed correctly.
If the command was incorrect, issue a rollback transaction command. If the command was correct, issue a commit transaction command.

Warning! Some system tables should not be altered by any user under any circumstances. Some system tables are built dynamically by system processes, contain encoded information, or display only a portion of their data when queried. Imprudent, ad hoc updates to certain system tables can make Adaptive Server unable to run, make database objects inaccessible, scramble permissions on objects, or terminate a user session. Moreover, you should never attempt to alter the definition of the system tables in any way. For example, do not alter system tables to include constraints. Triggers, defaults, and rules are not allowed in system tables. If you try to create a trigger or bind a rule or default to a system table, you will get an error message.

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.

Sybase-supplied system procedures (such as sp_who) are created using the installmaster installation script. You can use sp_version to determine which version of installmaster was run last. See the Reference Manual: System Procedures for more information about sp_version.

If you execute a system procedure in a database other than sybsystemprocs, it 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.
CHAPTER 1  Overview of System Administration

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, it must be enclosed 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, the entire name must be enclosed in single or double quotes.

System procedures can be invoked by sessions using either chained or unchained transaction mode. However, 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 the system procedures that create temporary worktables from transactions.

If no transaction is active when you execute a system procedure, Adaptive Server 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. For more information about transaction modes and isolation levels, see the Reference Manual.

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

    return status = 0

System procedure tables

The 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_helptext
- sp_helpindex
- sp_helpkey
- sp_helpoptions
- sp_helpprotect
- sp_lock

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

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 of the system procedures create and then drop temporary tables. For example, sp_helpdb creates #spdbdesc, sp_helpdevice creates #spdevtab, and sp_helpindex creates #spindtab.

Creating system procedures

Many of the system procedures are explained in this manual, in the sections where they are relevant. For complete information about system procedures, see the Reference Manual: System Procedures.

System administrators can write system procedures that can be executed in any database. Simply create a stored procedure in sybsystemprocs and give it a name that begins with “sp_”. The uid of the stored procedure must be 1, the uid of the database owner.

Most of the system procedures that you create query the system tables. You can also create stored procedures that modify the system tables, although this is not recommended.

To create a stored procedure that modifies system tables, a System Security Officer must first turn on the allow updates to system tables configuration parameter. Any stored procedure created while this parameter is set to “on” will always be able to update system tables, even when allow updates to system tables is set to “off.” To create a stored procedure that updates the system tables:

1 Use sp_configure to set allow updates to system tables to “on.”
2 Create the stored procedure with the create procedure command.
3 Use sp_configure to set allow updates to system tables to “off.”

Warning! Use extreme caution when you modify system tables. Always test the procedures that modify system tables in development or test databases, not in your production database.
**System extended stored procedures**

An extended stored procedure (ESP) provides a way to call external language functions from within Adaptive Server. Adaptive Server provides a set of 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 Adaptive Server.

You can 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 Adaptive Server. XP Server starts automatically on the first ESP invocation.

For information about the system ESPs provided with Adaptive Server, see the *Reference Manual*.

**Creating system ESPs**

Create a system ESP in the `sybsystemprocs` database using the *create procedure* command. System procedures are automatically included in the `sybsystemprocs` database. The name of the ESP, and its procedural language function, should 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 Chapter 17, “Using Extended Stored Procedures,” in the *Transact-SQL User’s Guide*.

**Logging error messages**

Adaptive Server writes start-up information to a local error log file each time it starts. The installation program automatically sets the error log location when you configure a new Adaptive Server. See the *Configuration Guide* for your platform to learn the default location and file name of the error log.

Many error messages from Adaptive Server go to the user’s terminal only. However, fatal error messages (severity levels 19 and above), kernel error messages, and informational messages from Adaptive Server are recorded in the error log file.
Connecting to Adaptive Server

Adaptive Server keeps the error log file open until you stop the server process. To reduce the size of the error log by deleting old messages, stop the Adaptive Server process before you do so.

Note On some platforms, such as Windows, Adaptive Server also records error messages in the operating system event log. See the installation and configuration guide for your platform for additional information about error logs.

Connecting to Adaptive Server

Adaptive Server can communicate with other Adaptive Servers, Open Server applications, and client software on the network. Clients can talk to one or more servers, and servers can communicate with other servers via remote procedure calls. For products to interact with one another, each needs to know where the others reside on the network. This network service information is stored in the interfaces file.

The interfaces file

The interfaces file is usually named interfaces, interface, or sql.ini, depending on the operating system.

The interfaces file is like an address book. It lists the name and address of every known server. 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, as shown in Figure 1-1.
The name, location, and contents of the `interfaces` file differ between operating systems. Also, the format of the Adaptive Server addresses in the `interfaces` file differs between network protocols.

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

See Chapter 2, “Networks and Performance” in the `Performance and Tuning Series: Basics` for more information about the `interfaces` file and network listeners.

**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 Adaptive Server. Two examples of directory services are Registry and Distributed Computing Environment (DCE).

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

- 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, Adaptive Server 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, refer to the configuration documentation for your platform for detailed information on configuring for directory services.

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 Adaptive Server or the clients are running. LDAP defines the communication protocol and the contents of messages exchanged between clients and servers. Messages are operators, such as client requests for read, write and query, and server responses, including data-format information.

The LDAP server can store and retrieve information about:

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

You can authenticate users logging in to Adaptive Server through information stored in the syslogins directory or through a centralized LDAP server that enables a single login and password throughout the enterprise. See Chapter 14, “Managing Adaptive Server Logins, Database Users, and Client Connections,” for more information.

The LDAP server can be configured with these access restrictions:

- Anonymous authentication – all data is visible to any user.
• User name and password authentication – Adaptive Server uses the default user name and password from the file:

  UNIX, 32-bit – $SYBASE/$SYBASE_OCS/config/libtcl.cfg
  UNIX, 64-bit – $SYBASE/$SYBASE_OCS/config/libtcl64.cfg
  NT – %SYBASE%\%SYBASE_OCS%\ini\libtcl.cfg

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

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 Adaptive Server. 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 accessible only from the LDAP server. Adaptive Server ignores the interfaces file.

If multiple directory services are supported in a server, then the order in which they are searched is specified in libtcl*.cfg. You cannot specify the search order with the dataserver command line option.

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 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=sybase,dc=com
dce=libddce.so ditbase=/./subsys/sybase/dataservers
Connecting to Adaptive Server

ldap=libdldap.so ldap://huey:11389/dc=sybase,dc=com


LDAP directory services versus the Sybase interfaces file

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

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

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

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

Table 1-1: interfaces file versus LDAP directory services

Performance

Performance when using an LDAP server may be slower than when using an interfaces file because the LDAP server requires time to make a network connection and retrieve data. Since this connection is made when Adaptive Server is started, changes in performance 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.
**Security features available in Adaptive Server**

Table 1-2 summarizes the major security features available for Adaptive Server. For information about configuring Adaptive Server for security, see Part Two of this manual.

*Table 1-2: Major security features*

<table>
<thead>
<tr>
<th>Security feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification and authentication controls</td>
<td>Ensures that only authorized users can log into the system. In addition to password based login authentication, Adaptive Server supports external authentication using Kerberos, LDAP, or PAM.</td>
</tr>
<tr>
<td>Discretionary Access Controls (DAC)</td>
<td>Provides access controls that give object owners the ability to restrict access to objects, usually with the grant and revoke commands. This type of control is dependent upon an object owner’s discretion.</td>
</tr>
<tr>
<td>Division of roles</td>
<td>Allows an administrator to grant privileged roles to specified users so only designated users can perform certain tasks. Adaptive Server has predefined roles, called “system roles,” such as system administrator and System Security Officer. In addition, Adaptive Server allows System Security Officers to define additional roles, called “user-defined roles.”</td>
</tr>
<tr>
<td>Accountability</td>
<td>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. Adaptive Server also provides a single option to audit a set of server-wide security-relevant events.</td>
</tr>
<tr>
<td>Confidentiality of data</td>
<td>Maintains a confidentiality of data using encryption for Client-Server communications, available with Kerberos or SSL. Data that is not active is kept confidential with password-protected database backup.</td>
</tr>
</tbody>
</table>
Security features available in Adaptive Server
CHAPTER 2

System and Optional Databases

This chapter describes the system databases that reside on all Adaptive Server systems. It also describes optional Sybase-supplied databases that you can install, and a database that Sybase Technical Support may install for diagnostic purposes.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview of system databases</td>
<td>23</td>
</tr>
<tr>
<td>master database</td>
<td>25</td>
</tr>
<tr>
<td>model database</td>
<td>28</td>
</tr>
<tr>
<td>sybsystemprocs database</td>
<td>29</td>
</tr>
<tr>
<td>tempdb database</td>
<td>29</td>
</tr>
<tr>
<td>sybsecurity database</td>
<td>31</td>
</tr>
<tr>
<td>sybsystemdb database</td>
<td>31</td>
</tr>
<tr>
<td>Chapter, “sybmgmtdb database,”</td>
<td>32</td>
</tr>
<tr>
<td>pubs2 and pubs3 sample databases</td>
<td>32</td>
</tr>
<tr>
<td>dbccdb database</td>
<td>33</td>
</tr>
<tr>
<td>sybdiag database</td>
<td>33</td>
</tr>
</tbody>
</table>

Overview of system databases

When you install Adaptive Server, it includes these 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
Overview of system databases

- 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 documentation for your platform. For information on installing dbccdb, see Chapter 10, “Checking Database Consistency,” in System Administration Guide: Volume 2. For information about using Job Scheduler, see the Job Scheduler User’s Guide.

The master, model, sybsystemdb, and temporary databases reside on the device named during installation, which is known as the master device. The master database is contained entirely on the master device and cannot be expanded onto any other device. All other databases and user objects should be created on other devices.

Warning! Do not store user databases on the master device. Storing user databases on the master device makes it difficult to recover the system databases if they become damaged. Also, you cannot recover user databases stored on the master device.

You should install the sybsecurity and sybmgmtdb databases on their own devices and segment. For more information, see the installation documentation for your platform.

You can 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.

You install the sybmgmtdb database with the installjsdb script (located in $SYBASE/ASE-15_0/scripts). 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, Sybase 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. At installation time, the master device is the default device. To change this, you can either edit the scripts or follow the instructions in Chapter 7, “Initializing Database Devices,” for information about adding more database devices and designating default devices.

**master database**

The master database controls the operation of Adaptive Server and stores information about all user databases and their associated database devices. Table 2-1 describes the information that the master database tracks.
**Table 2-1: Information the master database tracks**

<table>
<thead>
<tr>
<th>Information</th>
<th>System table</th>
</tr>
</thead>
<tbody>
<tr>
<td>User accounts</td>
<td>syslogins</td>
</tr>
<tr>
<td>Remote user accounts</td>
<td>syremotelogins</td>
</tr>
<tr>
<td>Remote servers that this server can interact with</td>
<td>syservers</td>
</tr>
<tr>
<td>Ongoing processes</td>
<td>sysprocesses</td>
</tr>
<tr>
<td>Configurable environment variables</td>
<td>sysconfigures</td>
</tr>
<tr>
<td>System error messages</td>
<td>sysmessages</td>
</tr>
<tr>
<td>Databases on Adaptive Server</td>
<td>sysdatabases</td>
</tr>
<tr>
<td>Storage space allocated to each database</td>
<td>sysusages</td>
</tr>
<tr>
<td>Tapes and disks mounted on the system</td>
<td>sysdevices</td>
</tr>
<tr>
<td>Active locks</td>
<td>syslocks</td>
</tr>
<tr>
<td>Character sets</td>
<td>syscharsets</td>
</tr>
<tr>
<td>Languages</td>
<td>syslanguages</td>
</tr>
<tr>
<td>Users who hold server-wide roles</td>
<td>sysloginroles</td>
</tr>
<tr>
<td>Server roles</td>
<td>syssrvroles</td>
</tr>
<tr>
<td>Adaptive Server engines that are online</td>
<td>sysengines</td>
</tr>
</tbody>
</table>

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.

**Note** 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
CHAPTER 2  System and Optional Databases

Controlling object creation in *master*

When you initially install Adaptive Server, only a system administrator can create objects in the *master* database, because the system administrator implicitly becomes “dbo” of any database he or she uses. Any objects created on the *master* database should be used for the administration of the system as a whole. Permissions in *master* should remain set so that most users cannot create objects there.

**Warning!** Never place user objects in *master*. Storing user objects in *master* can cause 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*.

Another way to discourage users from creating objects in *master* is to change the default database for users (the database to which a user is connected when he or she logs in) with *sp_modifylogin*. See “Adding users to databases” on page 396 for more information.

If you create your own system procedures, create them in the *sybsystemprocs* database rather than in *master*.

**Backing up *master* and keeping copies of system tables**

To be prepared for hardware or software failure on Adaptive Server:

- Perform frequent backups of the *master* database and all user databases. See “Keep up-to-date backups of *master*” on page 42 for more information. See also Chapter 13, “Restoring the System Databases,” in *System Administration Guide: Volume 2* for an overview of the process for recovering the *master* database.

- Keep a copy (preferably offline) of these system tables: *sysusages*, *sysdatabases*, *sysdevices*, *sysloginroles*, and *syslogins*. See “Keep offline copies of system tables” on page 43 for more information. If you have copies of these scripts, and a hard disk crash or other disaster makes your database unusable, you can use the recovery procedures described in Chapter 13, “Restoring the System Databases,” in *System Administration Guide: Volume 2*. If you do not have current copies of your scripts, it is much more difficult to recover Adaptive Server when the *master* database is damaged.
Adaptive Server includes the `model database`, which provides a template, or prototype, for new user databases. Each time a user enters the `create database` command, Adaptive Server makes a copy of the `model database` and extends the new database to the size specified by the `create database` command.

**Note** A new database cannot be smaller than 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 should have access to all databases on Adaptive Server.
- Granting default privileges, particularly for “guest” accounts.
- Setting database options such as `select into/bulkcopy/pllsort`. The settings are reflected in all new databases. Their original value in `model` is `off`. For more information about the database options, see Chapter 8, “Setting Database Options.”

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

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

**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.
**sybsystemprocs database**

Sybase system procedures are stored in the database sybsystemprocs. When a user in any database executes a system stored procedure (that is, a procedure whose name begins with \texttt{sp_}), Adaptive Server first looks for that procedure in the user’s current database. If there is no procedure there with that name, Adaptive Server looks for it in sybsystemprocs. If there is no procedure in sybsystemprocs by that name, Adaptive Server looks for the procedure in master.

If the procedure modifies system tables (for example, \texttt{sp\_adduser} modifies the \texttt{sysusers} table), the changes are made in the database from which the procedure was executed.

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

\textbf{Note} Any time you make changes to sybsystemprocs, you should back up the database.

**tempdb database**

Adaptive Server has a \textbf{temporary database}, tempdb, 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. The most common of these are:

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

You can increase the size of tempdb with \texttt{alter database}. tempdb is initially created on the master device. Space can be added from the master device or from any other database device.
tempdb database

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

Adaptive Server allows you to 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.

Creating temporary tables

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

Create temporary tables either by preceding the table name in a create table statement with a pound sign (#) or by specifying the name prefix “tempdb..”.

Temporary tables created with a pound sign are accessible only by the current Adaptive Server 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. Adaptive Server assigns the names of such tables a 17-byte number suffix. (You can see the suffix when you query tempdb..sysobjects.)

Temporary tables created with the “tempdb..” prefix are stored in tempdb and can be shared among Adaptive Server sessions. Adaptive Server does not change the names of temporary tables created this way. The table exists either until you restart Adaptive Server 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. Temporary tables can also be dropped explicitly before a session ends.

Warning! 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 Adaptive Server, it copies model to tempdb, which clears the database. Temporary tables are not recoverable.
**sybsecurity database**

The sybsecurity database contains the audit system for Adaptive Server. It consists of:

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

The audit system is discussed in more detail in Chapter 18, “Auditing.”

**sybsystemdb database**

The sybsystemdb database stores information about distributed transactions. Adaptive Server 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.

---

**Note** Distributed transaction management (DTM) services are available in Adaptive Server version 12.0 and later as a separately-licensed feature. You must purchase and install a valid license for Distributed Transaction Management before you can use it. See *Using Adaptive Server Distributed Transaction Management Features* and the installation guide for more information.

The sybsystemdb database also stores information about SYB2PC transactions that use the Sybase 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.

Two-phase commit transactions and how to create the sybsystemdb database is discussed in detail in the configuration documentation for your platform.
The sybmgmtdb database stores jobs, schedules, scheduled jobs information, and data the internal Job Scheduler task needs for processing. sybmgmtdb also maintains the output and results from these executed tasks. For more information on the Job Scheduler and sybmgmtdb, refer to the Job Scheduler User’s Guide.

Installing the pubs2 and pubs3 sample databases is optional. These databases are provided as a learning tool for Adaptive Server. The pubs2 sample database is used for most of the examples in the Adaptive Server documentation, except for examples, where noted, that use the pubs3 database. For information about installing pubs2 and pubs3, see the installation documentation for your platform. For information about the contents of these sample databases, see the Transact-SQL User’s Guide.

The sample databases contain a “guest” user login that allows access to the database by any authorized Adaptive Server user. The “guest” login has been given a wide range of privileges in pubs2 and pubs3, including permissions to select, insert, update, and delete user tables. For more information about the “guest” login and a list of the guest permissions in pubs2 and pubs3, see Chapter 14, “Managing Adaptive Server Logins, Database Users, and Client Connections.”

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.
**pubs2 image data**

Adaptive Server 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. Sybase 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 the installation documentation for your platform for information about installing the image data in pubs2.

**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. For more information, see Chapter 10, “Checking Database Consistency.”

**sybdiag database**

Sybase Technical 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.

**Determining the version of the installation scripts**

sp_version allows you to determine the current version of the scripts (installmaster, installdbccdb, and so on) installed on Adaptive Server, whether they ran successfully or not, and the time they took to complete.

The syntax for sp_version is:

```
sp_version [script_file [, "all"]]
```
Determining the version of the installation scripts

where:

- \textit{script_file} is the name of the installation script (the default value is NULL).
- \textit{all} reports details about the installation scripts, such as the date it was run and the time it took to run.

For example, the following reports the latest version of \textit{installmaster} that was run:

```
1> sp_version installmaster
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
```

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
--------- ------------------------------------------------------------
---------
```
CHAPTER 3

System Administration for Beginners

This chapter:

- Introduces new system administrators to important topics
- Helps system administrators find information in the Sybase documentation

Experienced administrators may also find this chapter useful for organizing their ongoing maintenance activities.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical page sizes</td>
<td>35</td>
</tr>
<tr>
<td>Using “test” servers</td>
<td>36</td>
</tr>
<tr>
<td>Installing Sybase products</td>
<td>37</td>
</tr>
<tr>
<td>Allocating physical resources</td>
<td>39</td>
</tr>
<tr>
<td>Backup and recovery</td>
<td>42</td>
</tr>
<tr>
<td>Ongoing maintenance and troubleshooting</td>
<td>45</td>
</tr>
<tr>
<td>Keeping records</td>
<td>46</td>
</tr>
<tr>
<td>Getting more help</td>
<td>48</td>
</tr>
</tbody>
</table>

Logical page sizes

The logical page size is a server-wide setting. You cannot have databases with different-sized logical pages within the same server. Adaptive Server 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.

You select the page size when you create the master device with `datasever -z`. All the logical pages of a server must be the same size. For instance, 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.
Using “test” servers

For more information about the dataserver, command see the Utility Guide. For more information about logical page sizes, see Chapter 3, “Configuring Memory,” in System Administration Guide: Volume 2.

Using “test” servers

Sybase suggests that you install and use a “test” and/or “development” Adaptive 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.

If you use a test server, Sybase suggests that you do so from the point of installing or upgrading Adaptive Server through the process of configuring the server. It is in these steps that you make some of the most important decisions about your final production system. The following sections describe how using a test server can help system administrators.

Understanding new procedures and features

Using a test server allows you to practice basic administration procedures before performing them in a production environment. If you are a new Adaptive Server administrator, many of the procedures discussed in this book may be unfamiliar to you, and it may take several attempts to complete a task successfully. However, even experienced administrators may benefit from practicing techniques that are introduced by new features in Adaptive Server.

Planning resources

Working with 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 a particular database requires nonstop recovery in the event of a media failure. This necessitates configuring 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.

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

**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 Adaptive Server.

**Installing Sybase products**

The responsibility for installing Adaptive Server and other Sybase 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.
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.

Also, refer to the lists of known problems that are installed with Adaptive Server. See the release bulletin for more information.

Install or upgrade Adaptive Server

Read through the installation documentation for your platform before you begin a new installation or upgrade. You must plan parts of the installation and configure the operating system before installing Adaptive Server. You may also want to consult with the operating system administrator to discuss operating system requirements for Adaptive Server. 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 have begun 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

Network protocols

Adaptive Server 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.

Directory services

As an alternative to the Sybase 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 Adaptive Server. For more information on directory services currently supported by Adaptive Server, see the configuration documentation for your platform. See also “Directory services” on page 17.
Configure and test client connections

A successful client connection depends on the coordination of Adaptive Server, the client software, and network products. If you are using one of the network protocols installed with Adaptive Server, see the configuration documentation 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 Sybase connectivity products to test client connections with Adaptive Server. For a general description of how clients connect to Adaptive Server, see “Connecting to Adaptive Server” on page 16. See also the configuration documentation for your platform for details about the name and contents of the interfaces file.

Allocating physical resources

Allocating physical resources is the process of providing Adaptive Server 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. You must also reallocate Adaptive Server’s resources if you upgrade your platform by adding new memory, disk controllers, or CPUs, or if the design of your database system changes. Early benchmarking of Adaptive Server and your applications can help you spot deficiencies in hardware resources that create performance bottlenecks.


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

Dedicated versus shared servers

The first step in planning Adaptive Server resources is understanding the resources required by other applications running on the same machine. In most cases, system administrators dedicate an entire machine for Adaptive Server use. This means that only the operating system and network software consume resources that otherwise might be reserved for Adaptive Server. On a shared system, other applications, such as Adaptive Server client programs or print servers, run on the same machine as Adaptive Server. It can be difficult to calculate the resources available to Adaptive Server on a shared system, because the types of programs and their pattern of use may change over time.

In either case, 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 Adaptive Server. Configure Adaptive Server to use only the resources that are available to it. Otherwise, the server may perform poorly or fail to start.

Decision support and OLTP applications

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

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.

Advance resource planning

It is extremely important that you understand and plan resource usage in advance. In the case of disk resources, for example, after you initialize and allocate a device to Adaptive Server, that device cannot be used for any other purpose (even if Adaptive Server never fills the device with data). Likewise, Adaptive Server automatically reserves the memory for which it is configured, and this memory cannot be used by any other application.

The following suggestions can help you plan resource usage:
For recovery purposes, it is always best to place a database’s transaction log on a separate physical device from its data. See Chapter 6, “Creating and Managing User Databases,” in System Administration Guide: Volume 2.

Consider mirroring devices that store mission-critical data. See Chapter 2, “Mirroring Database Devices,” in System Administration Guide: Volume 2. You may also consider using disk arrays and disk mirroring for Adaptive Server data if your operating system supports these features.

If you are working with a test Adaptive Server, it is sometimes easier to initialize database devices as operating system files, rather than raw devices, for convenience. Adaptive Server supports either raw partitions or certified file systems for its devices.

Keep in mind that changing configuration options can affect the way Adaptive Server consumes physical resources. This is especially true of memory resources. See Chapter 5, “Setting Configuration Parameters,” for details about the amount of memory used by individual parameters.

Operating system configuration

Once you have determined the resources that are available to Adaptive Server and the resources you require, configure these physical resources at the operating system level:

- If you are using raw partitions, initialize the raw devices to the sizes required by Adaptive Server. If you initialize a raw device for Adaptive Server, that device cannot be used 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 Adaptive Server 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 documentation for your platform. Also read your client software documentation or consult with your engineers to understand the operating system requirements for your applications.
Backup and recovery

Making regular backups of your databases is crucial to the integrity of your database system. Although Adaptive Server automatically recovers from system crashes (for example, power outages) or server crashes, only you can recover from data loss caused by media failure. Follow the basic guidelines below for backing up your system.

The following chapters, from the *System Administration Guide: Volume 2*, describe how to develop and implement a backup and recovery plan:

- Chapter 11, “Developing a Backup and Recovery Plan”
- Chapter 12, “Backing Up and Restoring User Databases”
- Chapter 13, “Restoring the System Databases”
- Chapter 16, “Managing Free Space with Thresholds”

Keep up-to-date backups of master

Backing up the *master* database is the cornerstone of any backup and recovery plan. The *master* database contains details about the structure of your entire database system. It keeps track of the Adaptive Server databases, devices, and device fragments that make up those databases. Because Adaptive Server needs this information during 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 that affects disks, storage, databases, or segments. This means you should back up *master* after performing any of the following procedures:

- 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 Adaptive Server logins
To back up master to a tape device, start isql and enter the command, where tape_device is the name of the tape device (for example, /dev/rmt0):

dump database master to "tape_device"

Keep offline copies of system tables

In addition to backing up master regularly, keep offline copies of the contents of the following system tables: sysdatabases, sysdevices, sysusages, sysloginroles, and syslogins. Do this by using the bcp utility described in the Utility Guide, and by storing a printed copy of the contents of each system table. You can create a printed copy by printing the output of the following queries:

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 crash or some other disaster makes your database unusable, you can use the recovery procedures described in Chapter 13, “Restoring the System Databases,” in System Administration Guide: Volume 2.

You should also keep copies of all data definition language (DDL) scripts for user objects, as described under “Keeping records” on page 46.

Automate backup procedures

Creating an automated backup procedure takes the guesswork out of performing backups and makes the procedure easier and quicker to perform. Automating backups can be as simple as using an operating system script or a utility (for example, the UNIX cron utility) to perform the necessary backup commands. Or you can automate the procedure further by using thresholds, which are discussed in Chapter 16, “Managing Free Space with Thresholds,” in System Administration Guide: Volume 2.

Creating an automated backup procedure

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:
Backup and recovery

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  Make a note about the new backup in a history file.
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.

Verify data consistency before backing up a database

Having backups of a database sometimes is not enough—you must have consistent, accurate backups (especially for master). If you back up a database that contains internal errors, the database has the same errors when you restore it.

Using the dbcc commands, you can 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, you can begin to think of running dbcc as insurance for your databases. If you discovered few or no errors while running dbcc in the past, you may decide that the risk of database corruption is small and that you need to run dbcc occasionally. If the consequences of losing data are too high, continue to run dbcc commands each time you back up a database.

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

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 make sure this situation never occurs. Use the preferred method of setting up a threshold stored procedure that notifies you (or dumps the log) when the log reaches a certain capacity. See Chapter 16, “Managing Free Space with Thresholds,” for information about using threshold procedures. Sybase also suggests that you dump the transaction log just prior to doing a full database dump to shorten the time required to dump and load the database.

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

Ongoing maintenance and troubleshooting

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

Starting and stopping Adaptive Server

Most system administrators automate the procedure for starting Adaptive Server to coincide with the start-up of the server machine. This can be accomplished by editing operating system start-up scripts or through other operating system procedures. See the configuration documentation for your platform to determine how to start and stop Adaptive Server.
Keeping records

Viewing and pruning the error log

Examine the contents of the error log on a regular basis to determine if any serious errors have occurred. You can also use operating system scripts to scan the error log for particular messages and to notify the system administrator when specific errors occur. Checking the error log regularly helps you determine whether there are continuing problems of the same nature or whether a particular database device is going bad. See Chapter 11, “Diagnosing System Problems,” for more information about error messages and their severity.

The error log file can grow large over time, since Adaptive Server appends informational and status messages to it each time it starts up. 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.

Keeping records

Keeping records about your Adaptive Server 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 Sybase Technical Support or recovering databases. They can also provide vital information for administrators who manage the Adaptive Server system in your absence. The following sections describe the kinds of records that are most valuable to maintain.

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 Adaptive Server users so that the appropriate contacts receive enhancement requests and problem reports.
CHAPTER 3    System Administration for Beginners

Configuration information

Ideally, create databases and database objects, and configure Adaptive Server using script files that you later store in a safe place. Storing the script files use makes it possible to re-create your entire system in the event of a disaster. It also allows you to re-create database systems quickly on new hardware platforms for evaluation purposes. 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 Adaptive Server logins and database users
- The current Adaptive Server configuration file, as described in “Using sp_configure with a configuration file” on page 67
- The names, locations, and sizes of all files and raw devices initialized as database devices

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

Maintenance schedules

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 Adaptive Server, Backup Server, and Adaptive Server Monitor
- Running the update statistics command (see Chapter 4, “Using the set statistics Commands,” in Performance and Tuning: Monitoring and Analyzing)
- Examining auditing information, if the auditing option is installed
Getting more help

- Recompiling stored procedures
- Monitoring the resource utilization of the server machine

System information

Record information about the hardware and operating system on which you run Adaptive Server. This can 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 Adaptive Server 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 Adaptive Server, or performing other administration activities

Disaster recovery plan

Consolidate the basic backup and recovery procedures, the hints provided in “Backup and recovery” on page 42, and your personal experiences in recovering data into a concise list of recovery steps tailored to your system. This 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.

Getting more help

The amount of new information that system administrators must learn may seem overwhelming. There are several software tools that can help you learn and facilitate basic administration tasks. These include Adaptive Server Monitor, used for monitoring server performance and other activities, and Sybase Central, which simplifies many administration tasks. There are also many third-party software packages available designed to help system administrators manage daily maintenance activities.
CHAPTER 4
Introduction to the Adaptive Server Plug-in for Sybase Central

This chapter describes how to use Sybase Central to manage Adaptive Server. This chapter is meant as an overview to introduce you to Sybase Central. For a complete description of the Adaptive Server plug-in features, see the Sybase Central online help.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview for Adaptive Server Sybase Central Plug-in</td>
<td>49</td>
</tr>
<tr>
<td>Using the Adaptive Server Plug-in</td>
<td>50</td>
</tr>
<tr>
<td>Starting and stopping Sybase Central</td>
<td>51</td>
</tr>
<tr>
<td>Registering Adaptive Server Plug-in</td>
<td>52</td>
</tr>
<tr>
<td>Performing common tasks</td>
<td>52</td>
</tr>
<tr>
<td>Using Interactive SQL</td>
<td>59</td>
</tr>
</tbody>
</table>

Overview for Adaptive Server Sybase Central Plug-in

Sybase Central is a graphical user interface (GUI) management tool. Sybase Central accepts a variety of “plug-ins” that allow you to manage specific Sybase products. The Adaptive Server plug-in allows you to manage Adaptive Server and helps you perform complex administration tasks without the need to remember the syntax of Transact-SQL commands or system stored procedures. You can use the Adaptive Server plug-in to:

- **Manage multiple servers from one console** – You can manage all the Adaptive Server installations from the Sybase Central main window.
- **Generate database definition language (DDL)** – You can generate DDL for the objects in Adaptive Server.
Using the Adaptive Server Plug-in

- **Visually represent objects** – You can see the databases and logins in each Adaptive Server and the objects in each database, and windows expand and contract to display information about databases and logins. The Adaptive Server plug-in expands to display information about many items, including:
  - Databases and tables
  - Disk devices
  - Active processes and locks
  - Logins and users
  - Data caches
  - ASE Replicator, Job Scheduler, and Messaging Services
  - Access to other utilities such as Interactive SQL (for sending queries and displaying query results).
- **Navigate between related objects** – To get more information about a database object related to the one whose property sheet you are displaying, navigate directly through the displayed object’s dialog box to the related object.

**Using the Adaptive Server Plug-in**

The Adaptive Server plug-in for Sybase Central provides you with an intuitive and easy way to administer Adaptive Server Enterprise. Sybase Central displays the Adaptive Server plug-in in its left-hand pane. Included in this pane is a hierarchical list of folders that represent different objects the plug-in can manage, including:

- Viewing and changing the characteristics of the object
- Creating another object:
- Generating the SQL text for creating an object (which allows you to reverse engineer Adaptive Server objects)
- Deleting an object
- Configuring Adaptive Server
- Managing:
Starting and stopping Sybase Central

To start Sybase Central:

- On UNIX, move to the $SYBASE/shared/sycentral43 directory and run the scjview.sh script.
- On Windows, choose Programs | Sybase | Sybase Central v4.3 from the Start menu, or
  On Windows, move to the %SYBASE%\Shared\Sybase Central 4.3\ directory and run the scjview.bat script.

To stop Sybase Central, select File | Exit
Registering Adaptive Server Plug-in

The Adaptive Server plug-in is registered in Sybase Central as part of the server installation. However, if Adaptive Server plug-in is not correctly registered, you can manually register the Adaptive Server plug-in:

- On Unix, run `$SYBASE/ASEP/bin/registerASEP`.
- On Windows, run `%SYBASE%\ASEP\bin\registerASEP.bat`.
- You can register the Adaptive Server plug-in manually by:
  a. Select Register from Tools | Plug-ins. A registration wizard appears.
  b. Select Register.
  c. Select “Register a plug-in by specifying a plug-in registration file.”
  d. Click Browse.
  e. Navigate to `$SYBASE/ASEP/bin` (%SYBASE%\ASEP\bin on Windows) and select `ASEPlugin.jpr`. Follow the wizard to register the Adaptive Server plug-in.

Performing common tasks

The following are some common tasks users perform with the Adaptive Server plug-in.

For more information about all the following tasks, see the Adaptive Server plug-in online help.

Starting and stopping Adaptive Server

If the Unified Agent is monitoring Adaptive Server, you can start, stop, and restart the server by right-clicking on the server and selecting Shutdown, Start, or Restart.

If the Unified Agent is not monitoring Adaptive Server, you can shutdown the server by selecting Shutdown.

Connecting to Adaptive Server

You can connect to an Adaptive Server by any of these methods:

- Select the Connect icon from the tool bar.
- Right click on Adaptive Server Enterprise and select Connect from the menu.
- Right click on any server group and select Connect from the menu.
The connected server is displayed in the Default server group if the connection is initiated from the Adaptive Server Enterprise folder or the connect icon. The plug-in displays “Connected to server” in the corresponding server group if the connection is initiated from the server group.

You can also specify a server to which you want to connect by any of the following:

- Specifying the server’s hostname and port number in the Connect dialog box.
- Selecting a pre-defined Adaptive Server from the server name dropdown list. This drop down list is derived from the servers listed in the interfaces file (UNIX) and sql.ini files (Windows) and LDAP servers.
- Discover which Adaptive Servers are available by clicking on Find in the Connect dialog. Before you can use this method, you must first define the discovery servers in Server Discovery tab located in the Adaptive Server Enterprise property page.

Creating a database

Before creating a database, make sure enough space is available on the database devices you plan to use.

To create a database:

- Right-click on the Add Database icon in the right-hand panel, or,
  1 Select the Databases folder.
  2 Choose File | New | Database or click on the Add Database option in the Databases folder. The Create a New Database wizard opens. The Create a New Database wizard asks for the following information:
Performing common tasks

Table 4-1: Inputs to create a new database wizard

<table>
<thead>
<tr>
<th>Input</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database name</td>
<td>Enter a name for the database</td>
</tr>
<tr>
<td>Database device</td>
<td>Specify the database device or devices on which to allocate the new database</td>
</tr>
<tr>
<td>Database device size</td>
<td>Specify a size for each database device</td>
</tr>
<tr>
<td>Data or log</td>
<td>Specify whether the database device will store data or the transaction log.</td>
</tr>
<tr>
<td>With override</td>
<td>Specify with override if you want to store data and log on the same device.</td>
</tr>
<tr>
<td>For load</td>
<td>If you are creating the database so you can restore it from a backup, check the For Load check box. This is the case only if you are recovering from media failure or if you are moving a database from one location to another.</td>
</tr>
<tr>
<td>Guest account</td>
<td>Specify whether to create a guest user in the database.</td>
</tr>
</tbody>
</table>

If you do not enter a size, Adaptive Server allocates either the value of the database size configuration variable or the size of the model database, whichever is larger.

If you have limited storage and must put the transaction log and the data on the same logical device, specifying With Override allows Adaptive Server to maintain the log on separate device fragments from the data.

You cannot remove or change a database device after creating the database unless you first delete the database.

Warning! Deleting a database also deletes all its objects.

Deleting a database

Only the owner of a database can delete it.

To delete a database:

1. Select the database icon.
2. Choose Edit | Delete.
3. Confirm the deletion in the confirmation dialog box.

Note Sybase recommends that you back up the master database after you delete a user database.

Adding a user

Database owners can add and delete users in the databases they own.
To create a user:

1. Expand the databases folder (select the “+” icon) and select the Users folder.
2. Choose File | New | User.

The Add a New User wizard opens and asks for this information:

<table>
<thead>
<tr>
<th>Table 4-2: Inputs to Add a New User wizard</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input</strong></td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Login name</td>
</tr>
<tr>
<td>Group</td>
</tr>
</tbody>
</table>

Note: A user can be a member of one assigned group or the default “public” group.

You can also select the Users folder. In the right pane, double-click the Add User icon.

Deleting a user

You cannot delete a user who owns objects. Since there is no command to transfer ownership of objects, you must delete objects owned by a user before you can delete the user. Also, you cannot delete a user who has granted permissions to other users without first revoking the permissions with cascade. If appropriate, re-grant the permissions to the other users.

Locking a login is a simple alternative to deleting a user.

To delete a user:

1. Select the user icon.
2. Choose Edit | Delete.
3. Confirm the deletion in the confirmation dialog box.

You can also select the user folder by right-clicking on the user icon and select Delete.

Before you delete a user:

1. Revoke the user’s command and object permissions with cascade.
2. Re-grant the permissions to the other users, if appropriate.
3. Delete the user’s objects.
### Creating a table

Only a database owner or a user with `create table` permission can create a table.

To create a table:

1. In a database you are working in, select the User Tables folder.
2. Choose File | New | Table or click on the Add Table icon in the User Tables folder.
   
   The Table Editor opens.
3. In the Name box, enter a name.
4. From the Owner list, choose an owner. The default is “dbo”.

You can also select the User Tables folder. In the right pane, double-click the Add Table icon.

### Deleting a table

Before you delete a table, be sure that no other objects reference it. If any objects reference it, you must edit those objects to avoid errors. To find out if other objects reference a table, check its dependencies.

**Note** When you delete a table, Adaptive Server deletes the indexes and triggers associated with the table and unbounds the rules or defaults that are bound to its columns.

Only table owners can delete tables.

To delete a table:

- Follow these steps:
  - Select the table icon.
  - Choose Edit | Delete.
  - Confirm the deletion in the confirmation dialog box, or,

- You can also select the table by right-clicking on the table icon and selecting Delete.

### Creating a server group

To create a server group:

1. Select Adaptive Server Enterprise
2. Choose File | New | Server Group
3. Follow the steps provided by the Create New Server Group wizard.

You can also add a server group by double-clicking on the Add Server Group from the right-hand pane.
CHAPTER 4  Introduction to the Adaptive Server Plug-in for Sybase Central

Getting server status

If the Unified Agent is monitoring Adaptive Server, check the server status by any of the following:

- Click on the server group to which the server belongs. Check the Status column in the Details pane of the server group.
- Click on the Adaptive Server Enterprise listed under Sybase Central, and then click on Servers tab on the right hand side panel. The server status is printed in the Status column.
- A green triangle on the lower right-hand side of the server icon indicates that Adaptive Server is running. A red square indicates that Adaptive Server is stopped.

Note: By default, the Adaptive Server plug-in does not have Check Server Status enabled. To enable Unified Agent to monitor Adaptive Server:

- Right click on Adaptive Server Enterprise and select Properties.
- Select Preferences and check “Enable Unified Agent (UA) related features.”

Getting the server log

If the Unified Agent is monitoring Adaptive Server, retrieve the server log by selecting the server and clicking on the Server Log tab in the right-hand pane.

The server log is retrieved based on how you have configured the filter for the server log. To configure the server log filtering, right-click on the server and select Server Log Filter. By default, the Adaptive Server plug-in retrieves the last 1000 lines from the server log. You can configure the server filter to retrieve:

- The entire log file.
- The last n number of lines.
- The log from the last n number of days.
- The lines that match the regular expression

Logging SQL statements

To log all SQL statements executed through the Adaptive Server plug-in:

- Right click on a server and select “Log SQL Statement.”
- Select whether you want SQL statements logged directly to a window or to a file.

Executing SQL statements

You can execute SQL statements from within the Adaptive Server plug-in by using the Interactive SQL query tool. To start the Interactive SQL tool, you can either:
Performing common tasks

- Right-click the server on which you want to execute the SQL statements and select Open Interactive SQL from the menu, or

1. Click on Adaptive Server Enterprise.
2. Click the Utilities tab on the right-hand pane and select Interactive SQL

You can execute SQL statements simultaneously on a set of servers belonging to a server group:
1. Right-click the server group and choose Execute SQL.
2. Select the servers on which you want to execute the SQL statements
3. Click Execute.

The result set for each server is listed in the Result Set pane of the SQL Execution dialog.

You can use the Adaptive Server plug-in to view a GUI version of the SQL execution plan for individual queries (much like a GUI version of `showplan`) and execution plans for all queries in a stored procedure. This GUI display includes nodes for each of the operators of the execution plan.

To get the GUI plan:
1. Start Interactive SQL.
2. Execute the query or stored procedure
3. Click on the plan tab in the Results pane of Interactive SQL
4. Select a query from the queries drop-down list.
5. Click the Details tab to see the GUI plan of the selected query. Click on an operator node to see the detailed statistics for that node.
6. Click on the XML tab to see an XML representation of the execution plan for the selected query
7. Click on the Text tab to see the execution plan in a text format for the submitted queries

For more information about Interactive SQL, see “Starting Interactive SQL” on page 60.

Viewing and updating object properties

You can view and modify the configuration of any object represented in the Adaptive Server plug-in using the Property dialog.

To bring up the Property dialog:
1. Click on the object you want to view or modify.
CHAPTER 4  Introduction to the Adaptive Server Plug-in for Sybase Central

2 Right-click on the object and select Properties.
3 Select the appropriate tab to perform your task.
4 Make any modification in the Property dialog.
5 Click on Apply, OK, or Cancel.

Generate the SQL text for creating an object
You can generate the SQL text required for creating an object, which allows you to reverse engineer the object. To generate SQL text, right-click on the object and select “Generate DDL.”

Viewing and updating Adaptive Server configuration parameters
You can view and update the Adaptive Server configuration parameters using the Server Properties dialog.
To view and update configuration parameters:
1 Right click on the server and select Configuration in the menu
2 Select the functional group from the drop down list in the Show Configuration Parameters
3 Find and select the parameter you want to view or update
4 Enter new valuing the value column if update is necessary
5 Click on Apply/OK/Cancel accordingly

Using Interactive SQL
Interactive SQL allows you to execute SQL statements, build scripts, and display database data to the server. You can use it to:
• Browse the information in a database.
• Test SQL statements that you plan to include in an application.
• Save query results to a file.
• Edit data in result sets.
• Load data into a database and carry out administrative tasks.

In addition, Interactive SQL can run command files or script files. For example, you can build repeatable scripts to run against a database and then use Interactive SQL to execute these scripts as batches.
Starting Interactive SQL

To start Interactive SQL, either:

- Select a database in Sybase Central and select File | Open Interactive SQL. Interactive SQL connects to the database. You can also right-click on the database and select Open Interactive SQL. The menu item Open Interactive SQL opens a connection to a server. However, when you select the menu item for a server, Interactive SQL opens a connection to the default database for that server. When you select a specific database from the Open Interactive SQL menu, Interactive SQL opens to the selected database.

- To start Interactive SQL without a connection to a server, select Tools | Adaptive Server Enterprise | Open Interactive SQL. The Connect dialog appears. How you start Interactive SQL from the command line depends on your operating system.

If you start Interactive SQL independently, the Connect dialog appears, which lets you connect to a database just as you would in Sybase Central.

- For UNIX, change to the $SYBROOT/DBISQL/bin directory and enter:
  
  `dbisql`

      On Windows, change to the %SYBROOT%\DBISQL\bin directory and enter:

  `dbisql.bat`

- In the Connection dialog, enter the information to connect to a database in the Connect dialog box and click OK.

To open a new Interactive SQL window:

1. Choose Window | New Window. The Connect dialog appears.
2. In the Connect dialog, enter connection options, and click OK to connect. The connection information (including the database name, your user ID, and the database server) appears on the title bar above the SQL Statements pane.

You can also connect to or disconnect from a database with the Connect and Disconnect commands in the SQL menu, or by executing a `connect` or `disconnect` statement in the SQL Statements pane.
CHAPTER 5

Setting Configuration Parameters

This chapter describes the Adaptive Server configuration parameters. The parameters are listed alphabetically.

A configuration parameter is a user-definable setting that you set, using the system procedure `sp_configure`. Configuration parameters are used for a wide range of services, from basic to specific server operations, and for performance tuning.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>What are configuration parameters?</td>
<td>61</td>
</tr>
<tr>
<td>Using sp_configure</td>
<td>65</td>
</tr>
<tr>
<td>Output from sp_configure</td>
<td>76</td>
</tr>
<tr>
<td>The sysconfigure and syscurconfigs tables</td>
<td>78</td>
</tr>
<tr>
<td>Configuration parameters</td>
<td>79</td>
</tr>
</tbody>
</table>

What are configuration parameters?

Configuration parameters are user-definable settings that control various aspects of Adaptive Server’s behavior. Adaptive Server supplies default values for all configuration parameters. You can use configuration parameters to tailor Adaptive Server for an installation’s particular needs.

Read this chapter carefully to determine which configuration parameters you should reset to optimize server performance.

Warning! Change configuration parameters with caution. Arbitrary changes in parameter values can adversely affect Adaptive Server performance and other aspects of server operation.
What are configuration parameters?

The Adaptive Server configuration file

Adaptive Server stores the values of configuration parameters in a configuration file, which is an ASCII text file. When you install a new Adaptive Server, 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 Sybase installation directory ($SYBASE). When you change a configuration parameter, Adaptive Server saves a copy of the old configuration file as `server_name.001`, `server_name.002`, and so on. Adaptive Server writes the new values to the file `server_name.cfg` or to a file name you specify at start-up.

How to modify configuration parameters

Set or change configuration parameters in one of the following ways:

- By executing `sp_configure` with the appropriate parameters and values,
- By editing your configuration file and then invoking `sp_configure` with the configuration file option, or
- By 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 Adaptive Server to reallocate memory, so they take effect only after you have restarted. The description of each parameter in this chapter indicates whether it is static or dynamic. Adaptive Server 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.

Who can modify configuration parameters?

The roles required for using `sp_configure` are as follows:

- 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
What are configuration parameters?

Unit specification using \textit{sp\_configure}

\texttt{sp\_configure} allows you to specify the value for configuration parameters in unit specifiers. The unit specifiers are \texttt{p} or \texttt{P} for pages, \texttt{m} or \texttt{M} for megabytes, and \texttt{g} or \texttt{G} for gigabytes. If you do not specify a unit, and you are configuring a parameter that controls memory, Adaptive Server uses the logical page size for the basic unit.

The syntax to indicate a particular unit specification is:

\[ \text{sp\_configure "parameter name", 0, "p|P|k|K|m|M|g|G"} \]

You must include the “\texttt{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:

\[ \text{sp\_configure "number of locks", 1024} \]

or:

\[ \text{sp\_configure "number of locks", 0, 1K} \]

This functionality does not change the way in which Adaptive Server reports \texttt{sp\_configure} output.

\begin{itemize}
  \item \textbf{Note} When you are configuring memory-related parameters, use only the \texttt{P} (pagesize) parameter for your unit specification. If you use any other parameter to configure memory related parameters, Adaptive Server may issue an arithmetic overflow error message.
\end{itemize}

Getting help information on configuration parameters

Use either \texttt{sp\_helpconfig} or \texttt{sp\_configure} to display information on a particular configuration parameter. For example:

\[ \text{sp\_helpconfig "number of open"} \]

\begin{tabular}{lll}
\hline
option_name & config_value & run_value \\
\hline
number of open databases & 12 & 12 \\
number of open indexes & 500 & 500 \\
number of open objects & 500 & 500 \\
\hline
\end{tabular}

\[ \text{sp\_helpconfig "number of open indexes"} \]
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.

<table>
<thead>
<tr>
<th>Minimum Value</th>
<th>Maximum Value</th>
<th>Default Value</th>
<th>Current Value</th>
<th>Memory Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>2147483647</td>
<td>500</td>
<td>500</td>
<td>208</td>
</tr>
</tbody>
</table>

For more information, see “Using sp_helpconfig” on page 60 in System

Using **sp_configure**

**sp_configure** displays and resets configuration parameters. You can restrict the
number of parameters displayed by **sp_configure** using **sp_displaylevel** to set
your display level to one of three values:

- Basic
- Intermediate
- Comprehensive

For information about display levels, see “User-defined subsets of the
parameter hierarchy: display levels” on page 74. For information about

Table 5-1 describes the syntax for **sp_configure**. The information in the
“Effect” column assumes that your display level is set to “comprehensive.”

<table>
<thead>
<tr>
<th>Command</th>
<th>Effect</th>
</tr>
</thead>
</table>
| **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 setting for all parameters matching parameter.     |
| **sp_configure “parameter”, value** | Resets **parameter** to **value**.                                       |
Using *sp_configure*

<table>
<thead>
<tr>
<th>Command</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>sp_configure &quot;parameter&quot;, 0, &quot;default&quot;</code></td>
<td>Resets parameter to its default value.</td>
</tr>
<tr>
<td><code>sp_config &quot;group_name&quot;</code></td>
<td>Displays all configuration parameters in <code>group_name</code>, their current values, their default values, the values to which they were recently set, and the amount of memory used by each setting.</td>
</tr>
<tr>
<td><code>sp_configure &quot;configuration file&quot;, 0, &quot;sub_command&quot;, &quot;file_name&quot;</code></td>
<td>Sets configuration parameters from the configuration file. See “Using <em>sp_configure</em> with a configuration file” on page 67 for descriptions of the parameters.</td>
</tr>
</tbody>
</table>

**Syntax elements**

The commands in Table 5-1 use the following variables:

- **parameter** – is any valid Adaptive Server 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.

**Parameter parsing**

`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"
```

**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 for a unique parameter name.
Using *sp_configure* with a configuration file

You can configure Adaptive Server either interactively, by using *sp_configure* as described above, or noninteractively, by instructing Adaptive Server to read values from an edited or restored version of the configuration file.

The benefits of using configuration files include:

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

You can make a copy of the configuration file using *sp_configure* with the parameter “configuration file” and then edit the file at the operating system level. Then, you can use *sp_configure* with the parameter “configuration file” to instruct Adaptive Server to read values from the edited file. Or you can specify the name of the configuration file at start-up.

For information on editing the file, see “Editing the configuration file” on page 69. For information on specifying the name of the configuration file at start-up, see “Starting Adaptive Server with a configuration file” on page 71.

Naming tips for the configuration file

Each time you modify a configuration parameter with *sp_configure*, Adaptive Server creates a copy of the outdated configuration file, using the naming convention `server_name.001`, `server_name.002`, `server_name.003...server_name.999`.

To work with a configuration file with a name other than the default name, keeping the `server_name` part of the file name, include at least one alphabetic character in the extension. Alternatively, you can change the `server_name` to part of the file name. Doing this avoids confusion with the backup configuration files generated by Adaptive Server when you modify a parameter.
Using sp_configure

Using sp_configure to read or write the configuration file

The syntax for using the configuration file option with sp_configure is:

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

where:

- "configuration file" – including quotes, specifies the configuration file parameter.
- 0 – must be included as the second parameter to sp_configure for backward compatibility.
- "subcommand" – is one of the commands described below.
- "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 Adaptive Server was started is used.

Parameters for using configuration files

You can use the four parameters described below with configuration files.

- **write** – creates file_name from the current configuration. If file_name already exists, a message is written to the error log; the existing file is renamed using the convention file_name.001, file_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 Adaptive Server 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 Adaptive Server was started.

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"

Example 3 Runs validation checking on the values in the file generic.config:

sp_configure "configuration file", 0, "verify", "generic.config"

Example 4 Writes configured values to the file restore.config:

sp_configure "configuration file", 0, "restore", "restore.config"

Editing the configuration file

The configuration file is an operating system 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 The following example specifies that the transaction can retry to acquire a lock one time when deadlocking occurs during an index page split or shrink:

`deadlock retries = 1`

Example 2 The following example specifies that the default value for the parameter `cpu accounting flush interval` should be used:
Using sp_configure

    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 Adaptive Server 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.

Permissions for configuration files

    Configuration files are nonencrypted ASCII text files. By default, they are created with read and write permissions set for the file owner and read permission set for all other users. If you created the configuration file at the operating system level, you are the file owner; if you created the configuration file from Adaptive Server, using the write or restore parameter, the file owner is the user who started Adaptive Server. 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.

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

Backing up configuration files

    Configuration files are not automatically backed up when you back up the master database. They are operating system files, and you should back them up in the same way you back up your other operating system files.

Checking the name of the configuration file currently in use

    The output from sp_configure truncates the name of the configuration file due to space limitations. 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"
Starting Adaptive Server with a configuration file

By default, Adaptive Server 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 Adaptive Server defaults for all values.

You can start Adaptive Server with a specified configuration file. For more information, see the Utility Guide.

If the configuration file you specify does not exist, Adaptive Server 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.

Configuration file errors

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

Adaptive Server 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, Adaptive Server uses the default value.
- Values are below the minimum allowable value.

The parameter hierarchy

Configuration parameters are grouped according to the area of Adaptive Server behavior they affect. This makes it easier to identify all parameters that you might need to tune to improve a particular area of Adaptive Server 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 Adaptive Server behavior. sp_configure displays parameters in all groups to which they belong.

Table 5-2 lists the configuration parameter groups.
Using sp_configure

Table 5-2: Configuration groups

<table>
<thead>
<tr>
<th>Parameter group</th>
<th>Configures Adaptive Server for:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backup/Recovery</td>
<td>Backing up and recovering data</td>
</tr>
<tr>
<td>Cache manager</td>
<td>The data and procedure caches</td>
</tr>
<tr>
<td>Component Integration Services</td>
<td>Component Integration Services</td>
</tr>
<tr>
<td>administration</td>
<td></td>
</tr>
<tr>
<td>DTM administration</td>
<td>Distributed transaction management (DTM) facilities</td>
</tr>
<tr>
<td>Diagnostics</td>
<td>Diagnostic principles</td>
</tr>
<tr>
<td>Disk I/O</td>
<td>Disk I/O</td>
</tr>
<tr>
<td>Error log</td>
<td>Error log and the logging of Adaptive Server events to the Windows Event Log</td>
</tr>
<tr>
<td>Extended stored procedures</td>
<td>Affecting the behavior of extended stored procedures (ESPs).</td>
</tr>
<tr>
<td>General information</td>
<td>Basic system administration</td>
</tr>
<tr>
<td>Java services</td>
<td>Memory for Java in Adaptive Server</td>
</tr>
<tr>
<td></td>
<td>See the Java in Adaptive Server Enterprise manual for complete information about Java in the database.</td>
</tr>
<tr>
<td></td>
<td>If you use method calls to JDBC, you may need to increase the size of the execution stack available to the user. See “stack size” on page 230 for information about setting the stack size parameter.</td>
</tr>
<tr>
<td>Languages</td>
<td>Languages, sort orders, and character sets</td>
</tr>
<tr>
<td>Lock manager</td>
<td>Locks</td>
</tr>
<tr>
<td>Memory use</td>
<td>Memory consumption</td>
</tr>
<tr>
<td>Meta-data caches</td>
<td>Setting the metadata cache size for frequently used system catalog information.</td>
</tr>
<tr>
<td></td>
<td>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 “Configuring Memory” on page 41 in System Administration Guide: Volume 2.</td>
</tr>
<tr>
<td>Network communication</td>
<td>Communication between Adaptive Server and remote servers, and between Adaptive Server and client programs</td>
</tr>
<tr>
<td>O/S resources</td>
<td>Use of operating system resources</td>
</tr>
<tr>
<td>Physical memory</td>
<td>Your machine’s physical memory resources</td>
</tr>
<tr>
<td>Processors</td>
<td>Processors in an SMP environment</td>
</tr>
<tr>
<td>Query Tuning</td>
<td>Query optimization</td>
</tr>
<tr>
<td>RepAgent thread administration</td>
<td>Replication via Replication Server</td>
</tr>
<tr>
<td>SQL Server administration</td>
<td>General Adaptive Server administration.</td>
</tr>
</tbody>
</table>
The syntax for displaying all groups and their associated parameters, and the current values for the parameters, is:

```
sp_configure
```

Note The number of parameters `sp_configure` returns depends on the value to which you have your display level set. See “User-defined subsets of the parameter hierarchy: display levels” on page 74 for further information about display levels.

The following is the syntax for displaying a particular group and its associated parameter, where `group_name` is the name of the group you are interested in:

```
sp_configure "group_name"
```

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

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

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Default</th>
<th>Memory Used</th>
<th>Config Value</th>
<th>Run Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>allow sql server async i/o</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>switch</td>
<td>static</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>disable disk mirroring</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>switch</td>
<td>static</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>disk i/o structures</td>
<td>256</td>
<td>0</td>
<td>256</td>
<td>256</td>
</tr>
<tr>
<td>number</td>
<td>dynamic</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>number of devices</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>number</td>
<td>dynamic</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>number of large I/O buffers</td>
<td>6</td>
<td>12352</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>number</td>
<td>dynamic</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>page utilization percent</td>
<td>95</td>
<td>0</td>
<td>95</td>
<td>95</td>
</tr>
<tr>
<td>percent</td>
<td>dynamic</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Parameter group | Configures Adaptive Server for:
-----------------|----------------------------------|
Security related | Security-related features        |
Unicode          | Unicode-related features         |
User environment | User environments                |

Security related Security-related features
Unicode Unicode-related features
User environment User environments
Using sp_configure

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.

User-defined subsets of the parameter hierarchy: display levels

Depending on your use of Adaptive Server, you may need to adjust some parameters more frequently than others. You may find it is easier to work with a subset of parameters than having to see the entire group when you are working with only a few. You can set your display level to one of three values to give you the subset of parameters that best suits your working style.

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 to see more or fewer configuration parameters.

- “Basic” shows only the most basic parameters, and is appropriate for very general server tuning.
- “Intermediate” includes parameters that are somewhat more complex, in addition to the “basic” parameters. This level is appropriate for a moderately complex level of server tuning.
- “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

The following is the syntax for setting your display level, where user_name is your Adaptive Server login name:

    sp_displaylevel user_name[, basic | intermediate | comprehensive]
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 instance, 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
... 
```

However, this is only a subset of the parameters in the languages group, because some parameters in that group are displayed only at the “comprehensive” level.

The *reconfigure* command

Pre-11.0 SQL Server versions required you to execute *reconfigure* after executing *sp_configure*. Beginning with SQL Server version 11.0, this was no longer required. The *reconfigure* command still exists, but it does not have any effect. It is included in this version of Adaptive Server so you can run pre-11.0 SQL scripts without modification.

Scripts using *reconfigure* still run in the current version, but change them at your earliest convenience because *reconfigure* will not be supported in future versions of Adaptive Server.

Performance tuning with *sp_configure* and *sp_sysmon*

*sp_sysmon* monitors Adaptive Server performance and generates statistical information that describes the behavior of your Adaptive Server system. See the *Performance and Tuning Series: Monitoring Adaptive Server with sp_sysmon* for more information.
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.

### Output from `sp_configure`

The sample output below shows the type of information `sp_configure` prints if you have your display level set to “comprehensive” and you execute it with no parameters. The values it prints vary, depending on your platform and on what values you have already changed.

```
sp_configure
Group: Configuration Options

Group: Backup/Recovery

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Default</th>
<th>Memory Used</th>
<th>Config Value</th>
<th>Run Value</th>
<th>Unit</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>allow remote access</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>switch</td>
<td>dyn</td>
</tr>
<tr>
<td>print recovery info</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>switch</td>
<td>dyn</td>
</tr>
<tr>
<td>recovery interval in m</td>
<td>5</td>
<td>0</td>
<td>5</td>
<td>5</td>
<td>minutes</td>
<td>dyn</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

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

Where:

- The “Default” column displays the value Adaptive Server is shipped with. If you do not explicitly reconfigure a parameter, it retains its default value.
- The “Memory Used” column displays the amount of memory used (in kilobytes) 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, it would total more than the amount actually used. In the “Memory Used” column, parameters that “share” memory with other parameters are marked with a hash mark (“#”).
• The “Config Value” column 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 Adaptive Server.

• The “Run Value” column displays the value Adaptive Server is currently using. It changes when you modify a dynamic parameter’s value and, for static parameters, after you restart Adaptive Server.

• The “Unit” column displays the unit value in which the configuration parameter is displayed. Adaptive Server displays information in the following units:

<table>
<thead>
<tr>
<th>Name of unit</th>
<th>Unit description</th>
</tr>
</thead>
<tbody>
<tr>
<td>number</td>
<td>Displays the number of items for which a parameter is configured.</td>
</tr>
<tr>
<td>clock ticks</td>
<td>Number of clock ticks for which a parameter is set.</td>
</tr>
<tr>
<td>microseconds</td>
<td>Number of microseconds for which a parameter is set.</td>
</tr>
<tr>
<td>milliseconds</td>
<td>Number of milliseconds for which a parameter is set.</td>
</tr>
<tr>
<td>seconds</td>
<td>Number of seconds for which a parameter is set.</td>
</tr>
<tr>
<td>minutes</td>
<td>Number of minutes for which a parameter is set.</td>
</tr>
<tr>
<td>hours</td>
<td>Number of hours for which a parameter is set.</td>
</tr>
<tr>
<td>bytes</td>
<td>Number of bytes for which a parameter is set.</td>
</tr>
<tr>
<td>days</td>
<td>Number of days for which a parameter is set.</td>
</tr>
<tr>
<td>kilobytes</td>
<td>Number of kilobytes for which a parameter is set.</td>
</tr>
<tr>
<td>megabytes</td>
<td>Number of megabytes for which a parameter is set.</td>
</tr>
<tr>
<td>memory pages (2K)</td>
<td>Number of 2K memory pages for which the parameter is set.</td>
</tr>
<tr>
<td>virtual pages (2K)</td>
<td>Number of 2K virtual pages for which the parameter is set.</td>
</tr>
<tr>
<td>logical pages</td>
<td>Number of logical pages for which the parameter is configured. This value depends on which logical page size your server is using: 2, 4, 8, or 16K.</td>
</tr>
<tr>
<td>percent</td>
<td>Displays the value of the configured parameter as a percentage.</td>
</tr>
<tr>
<td>ratio</td>
<td>Displays the value of the configured parameter as a ratio.</td>
</tr>
<tr>
<td>switch</td>
<td>Value of the parameter is either TRUE (the parameter is turned on, or FALSE</td>
</tr>
</tbody>
</table>
The sysconfigures and syscurconfigs tables

<table>
<thead>
<tr>
<th>Name of unit</th>
<th>Unit description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>ID of the configured parameter you are investigating.</td>
</tr>
<tr>
<td>name</td>
<td>Character string name assigned to the run or configure value of the parameter. For example, the string “binary” appears under the the run or configure value column for the output of sp_configure &quot;lock scheme&quot;.</td>
</tr>
<tr>
<td>row</td>
<td>Number of rows for which the specified parameter is configured.</td>
</tr>
</tbody>
</table>

- The “Type” column displays whether the configuration option is static or dynamic. Changes to static parameters require that you restart Adaptive Server for the changes to take effect. Changes to dynamic parameters take effect immediately without having to restart Adaptive Server.

The sysconfigures and syscurconfigs tables

The report displayed 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 Adaptive Server 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 Adaptive Server computes and uses.

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

Querying syscurconfigs and sysconfigures: an example

You might want to query sysconfigures and syscurconfigs to get information organized 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. You can query these system tables to get a complete list of current memory usage, as well as minimum, maximum, and default values, with the following query:
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

Configuration parameters

In many cases, the maximum allowable values for configuration parameters are extremely high. The maximum value for your server is usually limited by available memory, rather than by \textit{sp\_configure} limitations.

\textbf{Note} To find the maximum supported values for your platform and version of Adaptive Server, see the table “Adaptive Server Specifications” in the \textit{Installation Guide} for your platform.

Alphabetical listing of configuration parameters

The following sections include both summary and detailed information about each configuration parameter.

\textit{abstract plan cache}

\begin{tabular}{|l|l|}
\hline
\textbf{Summary information} & \\
\hline
Default value & 0 (off) \\
Range of values & 0 (off), 1 (on) \\
Status & Dynamic \\
Display level & Comprehensive \\
Required role & System administrator \\
Configuration group & Query Tuning \\
\hline
\end{tabular}
**Configuration parameters**

**abstract plan cache** enables caching of abstract plan hash keys. By default, caching is not enabled. For more information, see Chapter 12, “Creating and Using Abstract Plans” in the *Performance and Tuning Series: Query Processing and Abstract Plans*. **abstract plan load** must be enabled in order for plan caching to take effect.

**abstract plan dump**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

**abstract plan dump** enables the saving of abstract plans to the `ap_stdout` abstract plans group. For more information, see Chapter 12, “Creating and Using Abstract Plans” in the *Performance and Tuning Series: Query Processing and Abstract Plans*.

**abstract plan load**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

**abstract plan load** enables association of queries with abstract plans in the `ap_stdin` abstract plans group. For more information, see Chapter 12, “Creating and Using Abstract Plans” in the *Performance and Tuning Series: Query Processing and Abstract Plans*.
abstract plan replace

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>0 (off)</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>0 (off), 1 (on)</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Dynamic</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Comprehensive</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
<tr>
<td>Query Tuning</td>
</tr>
</tbody>
</table>

abstract plan replace enables plan replacement for abstract plans in the ap_stdout abstract plans group. For more information, see Chapter 12, “Creating and Using Abstract Plans” in the Performance and Tuning Series: Query Processing and Abstract Plans. abstract plan load must be enabled in order for replace mode to take effect.

additional network memory

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>0–2147483647</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Dynamic</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Intermediate</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
<tr>
<td>Memory Use, Network Communication, Physical Memory</td>
</tr>
</tbody>
</table>

additional network memory sets the maximum size of additional memory that can be used for network packets that are larger than the default packet size. Adaptive Server rounds down the value you enter to the nearest 2K value. The default value indicates that no extra space is allocated for large packets.
When a login requests a large packet size, Adaptive Server verifies it has sufficient memory available to satisfy the request. If it does, the login continues. If it does not, Adaptive Server finds the largest available block of memory and tries the appropriate size (which is a multiple of default network packet size) less than the largest memory block. If that fails, Adaptive Server decreases the value of the request by the number of bytes equal to default network packet size, if this is available. Adaptive Server continues for 10 iterations or until the size equals the value of default network packet size, whichever comes first. On the tenth iteration, Adaptive Server uses the value of the default network packet size for the packet size.

If you increase `max network packet size` but do not increase additional network memory, clients cannot use packet sizes that are larger than the default size, because all allocated network memory is reserved for users at the default size. Adaptive Server guarantees that every user connection can log in at the default packet size. In this situation, users who request a large packet size when they log in receive a warning message telling them that their application will use the default size.

If you request a large packet size, Adaptive Server checks if the memory is available to satisfy the request.

- If the memory is available, the login continues
- If the memory is not available, Adaptive Server finds the largest available block of memory and tries again with a packet size equal to the largest block less the default network packet size. If that fails, Adaptive Server retries, but reduces the size of the request by the value of default network packet size. It repeats this process for 10 attempts until the packet size is equal to the configured default network packet size. At the tenth attempt, Adaptive Server drops the packet size to the configured default network packet size, since this size is always available. The "additional network memory" parameter is used to guarantee memory is available for these larger packet size allocations.

Increasing additional network memory may improve performance for applications that transfer large amounts of data. To determine the value for additional network memory when your applications use larger packet sizes:

- Estimate the number of simultaneous users who will request the large packet sizes, and the sizes their applications will request,
- Multiply this sum by three, since each connection needs three buffers,
- Add two percent for overhead for 32-bit servers or four percent for 64-bit servers,
• Round the value to the next highest multiple of 2048.

For example, if you estimate these simultaneous needs for larger packet sizes:

<table>
<thead>
<tr>
<th>Application</th>
<th>Packet size</th>
<th>Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td>bcp</td>
<td>8192</td>
<td></td>
</tr>
<tr>
<td>Client-Library</td>
<td>8192</td>
<td></td>
</tr>
<tr>
<td>Client-Library</td>
<td>4096</td>
<td></td>
</tr>
<tr>
<td>Client-Library</td>
<td>4096</td>
<td></td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>24576</strong></td>
<td></td>
</tr>
</tbody>
</table>

Multiply by 3 buffers/user: 24576 * 3 = 73728

Compute 2% overhead: 73728 * .02 = 1474.56

Add overhead: 73728 + 1474 = 75202

Additional network memory: 75202

Round up to multiple of 2048: 75776

You should set additional network memory to 75,776 bytes.

**allocate max shared memory**

<table>
<thead>
<tr>
<th>Summary information</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
<td>0</td>
</tr>
<tr>
<td>Range of values</td>
<td>0,1</td>
</tr>
<tr>
<td>Status</td>
<td>Dynamic</td>
</tr>
<tr>
<td>Display level</td>
<td>Basic</td>
</tr>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration groups</td>
<td>Memory Use, Physical Memory</td>
</tr>
</tbody>
</table>

allocate max shared memory determines whether Adaptive Server allocates all the memory specified by max memory at start-up or only the amount of memory the configuration parameter requires.

By setting allocate max shared memory to 0, you ensure that Adaptive Server uses only the amount of shared memory required by the current configuration, and allocates only the amount of memory required by the configuration parameters at start-up, which is a smaller value than max memory.
Configuration parameters

If you set `allocate max shared memory` to 1, Adaptive Server allocates all the memory specified by `max memory` at start-up. If you set `allocate max shared memory` to 1, and if you increase `max memory`, Adaptive Server attempts to allocate the memory immediately. If the memory allocation fails, Adaptive Server writes messages to the error log. Make sure you check the errorlog to verify that no errors have occurred.

A successful memory allocation means that Adaptive Server always has the memory required for any memory configuration changes you make and there is no performance degradation while the server readjusts for additional memory. However, if you do not predict memory growth accurately, and `max memory` is set to a large value, you may waste total physical memory.

**allow backward scans**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Valid values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

`allow backward scans` controls how the optimizer performs `select` queries that contain the `order by...desc` command:

- When the value is set to 1, the optimizer can access the index or table rows by following the page chain in descending index order.
- When the value is set to 0, the optimizer selects the rows into a worktable by following the index page pointers in ascending order and then sorts the worktable in descending order.

The first method—performing backward scans—can speed access to tables that need results ordered by descending column values. Some applications, however, may experience deadlocks due to backward scans. In particular, look for increased deadlocking if you have `delete` or `update` queries that scan forward using the same index. There may also be deadlocks due to page splits in the index.
You can use `print deadlock information` to send messages about deadlocks to the error log. See “print deadlock information” on page 209. Alternatively, you can use `sp_sysmon` to check for deadlocking. See the *Performance and Tuning Series: Locking and Concurrency Control* for more information on deadlocks.

**allow nested triggers**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Valid values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

allow nested triggers controls the use of nested triggers. When the value is set to 1, data modifications made by triggers can fire other triggers. Set allow nested triggers to 0 to disable nested triggers. A set option, `self_recursion`, controls whether the modifications made by a trigger can cause that trigger to fire again.

**allow procedure grouping**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

allow procedure grouping controls the ability to group stored procedures of the same name so that they can be dropped with a single `drop procedure` statement.

**allow remote access**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
</tbody>
</table>
### allow remote access

Controls logins from remote Adaptive Servers. The default value of 1 allows Adaptive Server to communicate with Backup Server. Only a system security officer can set `allow remote access`.

Setting the value to 0 disables server-to-server RPCs. Since Adaptive Server communicates with Backup Server via RPCs, setting this parameter to 0 makes it impossible to back up a database.

Since other system administration actions are required to enable remote servers other than Backup Server to execute RPCs, leaving this option set to 1 does not constitute a security risk.

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valid values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>

### allow resource limits

Controls the use of resource limits. When the value is set to 1, the server allocates internal memory for time ranges, resource limits, and internal server alarms. The server also internally assigns applicable ranges and limits to user sessions. The output of `showplan` and `statistics io` displays the optimizer’s cost estimate for a query. Set `allow resource limits` to 0 to disable resource limits.

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Valid values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>

### allow sendmsg

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
</tbody>
</table>
The `allow sendmsg` parameter enables or disables sending messages from Adaptive Server to a UDP (User Datagram Protocol) port. When `allow sendmsg` is set to 1, any user can send messages using `sp_sendmsg` or `syb_sendmsg`. To set the port number used by Adaptive Server, see “`syb_sendmsg port number`” on page 236.

**Note** Sending messages to UDP ports is not supported on Windows.

### `allow sql server async i/o`

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Valid values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

The `allow sql server async i/o` parameter enables Adaptive Server to run with asynchronous disk I/O. To use asynchronous disk I/O, you must enable it on both Adaptive Server and your operating system. See your operating system documentation for information on enabling asynchronous I/O at the operating system level.

In all circumstances, disk I/O runs faster asynchronously than synchronously. This is because when Adaptive Server issues an asynchronous I/O, it does not have to wait for a response before issuing further I/Os.
Configuration parameters

allow updates to system tables

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Valid values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

allow updates to system tables enables users with the system administrator role to make changes to the system tables and to create stored procedures that can modify system tables. A database administrator can update system tables in any tables that he or she owns if allow updates to system tables is enabled.

System tables include:

- All Sybase-supplied tables in the master database
- All tables in user databases that begin with “sys” and that have an ID value in the sysobjects table of less than or equal to 100

**Warning!** Incorrect alteration of a system table can result in database corruption and loss of data. Always use `begin transaction` when modifying a system table to protect against errors that might corrupt your databases. Immediately after finishing your modifications, disable allow updates to system tables.

Stored procedures and triggers you create while allow updates to system tables is set on can update the system tables, even after the parameter has been set off. When you set allow updates to system tables to on, you create a “window of vulnerability,” a period of time during which users can alter system tables or create a stored procedure with which the system tables can be altered in the future.
Because the system tables are so critical, Sybase suggests that you set this parameter to on only in highly controlled situations. To guarantee that no other users can access Adaptive Server while the system tables can be directly updated, restart Adaptive Server in single-user mode. For details, see startserver and dataserver in the Utility Guide.

Note The server-wide configuration option allow updates to system tables takes precedence over the stored procedure settings for allow updates to system tables. If you do not enable allow updates to system tables at the server level, individual, stored procedure settings determine whether you can modify system catalogs.

**audit queue size**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>

The in-memory audit queue holds audit records generated by user processes until the records can be processed and written to the audit trail. A system security officer can change the size of the audit queue using audit queue size. There is a trade-off between performance and risk that must be considered when you configure the queue size. If the queue is too large, records can remain in it for some time. As long as records are in the queue, they are at risk of being lost if the system crashes. However, if the queue is too small, it can become full repeatedly, which affects overall system performance; user processes that generate audit records sleep if the audit queue is full.

Following are some guidelines for determining how big your audit queue should be. You must also take into account the amount of auditing to be performed at your site.

- The memory requirement for a single audit record is 424 bytes; however a record can be as small as 22 bytes when it is written to a data page.
Configuration parameters

- The maximum number of audit records that can be lost in a system crash is the size of the audit queue (in records), plus 20. After records leave the audit queue they remain on a buffer page until they are written to the current audit table on the disk. The pages are flushed to disk every 20 records at the most (less if the audit process is not constantly busy).

- In the system audit tables, the `extrainfo` field and fields containing names are of variable length, so audit records that contain complete name information are generally larger.

The number of audit records that can fit on a page varies from 4 to as many as 80 or more. The memory requirement for the default audit queue size of 100 is approximately 42K.

**auditing**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

Auditing enables or disables auditing for Adaptive Server.

**check password for digit**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

The system security officer can tell the server to check for at least one character or digit in a password using the server-wide configuration parameter `check password for digit`. If set, this parameter does not affect existing passwords. By default, checking for digits is off.
To activate check password for digit functionality, enter:

```
sp_configure "check password for digit", 1
```

To deactivate check password for digit functionality, enter:

```
sp_configure "check password for digit", 0
```

### cis bulk insert array size

**Summary information**

<table>
<thead>
<tr>
<th>Default value</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range of values</td>
<td>0–2147483647</td>
</tr>
<tr>
<td>Status</td>
<td>Dynamic</td>
</tr>
<tr>
<td>Display level</td>
<td>Comprehensive</td>
</tr>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration group</td>
<td>Component Integration Services</td>
</tr>
</tbody>
</table>

When performing a bulk transfer of data from one Adaptive Server to another Adaptive Server, CIS buffers rows internally, and asks the Open Client bulk library to transfer them as a block. The size of the array is controlled by cis bulk insert array size. The default is 50 rows, and the property is dynamic, allowing it to be changed without restarting the server.

### cis bulk insert batch size

**Summary information**

<table>
<thead>
<tr>
<th>Default value</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range of values</td>
<td>0–2147483647</td>
</tr>
<tr>
<td>Status</td>
<td>Dynamic</td>
</tr>
<tr>
<td>Display level</td>
<td>Comprehensive</td>
</tr>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration group</td>
<td>Component Integration Services</td>
</tr>
</tbody>
</table>

The cis bulk insert batch size parameter determines how many rows from the source tables are to be bulk copied into the target table as a single batch using select into.

If the parameter is left at zero (the default), all rows are copied as a single batch. Otherwise, after the count of rows specified by this parameter has been copied to the target table, the server issues a bulk commit to the target server, causing the batch to be committed.
Configuration parameters

If a normal client-generated bulk copy operation (such as that produced by the bcp utility) is received, then the client is expected to control the size of the bulk batch, and the server ignores the value of this configuration parameter.

**cis connect timeout**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

The cis connect timeout parameter determines the wait time in seconds for a successful Client-Library connection. By default, no timeout is provided.

**cis cursor rows**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

The cis cursor rows parameter specifies the cursor row count for cursor open and cursor fetch operations. Increasing this value means more rows are fetched in one operation. This increases speed but requires more memory.

**cis idle connection timeout**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
</tbody>
</table>
**cis idle connection timeout** configures Adaptive Server to check for CIS connections to any remote server that have been unused longer than the specified number of seconds. Adaptive Server deletes the unused connections and reallocates their resources.

Although the number you specify is in seconds, the housekeeper task wakes up at most once a minute, so idle connections may be idle for much longer than the configured value. Adaptive Server does not drop idle connections if a transaction is active on the connection, and reestablishes the connection automatically if the user executes any command that accesses the connection.

**cis packet size**

The **cis packet size** parameter specifies the size of Tabular Data Stream™ (TDS) packets that are exchanged between the server and a remote server when a connection is initiated.

The default packet size on most systems is 512 bytes, and this may be adequate for most applications. However, larger packet sizes may result in significantly improved query performance, especially when text, unitext, and image or bulk data is involved.

If you specify a packet size larger than the default, and the requested server is a version 10 or later Adaptive Server, then the target server must be configured to allow variable-length packet sizes. Adaptive Server configuration parameters of interest in this case are:

- additional netmem
- maximum network packet size
Configuration parameters

cis rpc handling

The `cis rpc handling` parameter specifies the default method for remote procedural call (RPC) handling. Setting `cis rpc handling` to 0 sets the Adaptive Server site handler as the default RPC handling mechanism. Setting the parameter to 1 forces RPC handling to use Component Integration Service access methods. For more information, see `set cis rpc handling` in the Component Integration Services Users Guide.

configuration file

The `configuration file` parameter specifies the location of the configuration file currently in use. See “Using sp_configure with a configuration file” on page 67 for a complete description of configuration files.

In `sp_configure` output, the “Run Value” column displays only 10 characters. For this reason, the output may not display the entire path and name of your configuration file.

cost of a logical io

The `cost of a logical io` parameter specifies the cost of a logical I/O operation. For more information, see “Using sp_costs” on page 67 for a complete description of `sp_costs`.

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
</tbody>
</table>
cost of a logical io specifies the cost of a single logical IO.

### Summary information

<table>
<thead>
<tr>
<th>Status</th>
<th>Dynamic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display level</td>
<td>Comprehensive</td>
</tr>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration group</td>
<td></td>
</tr>
</tbody>
</table>

cost of a physical io

### Summary information

<table>
<thead>
<tr>
<th>Default value</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range of values</td>
<td>0 - 254</td>
</tr>
<tr>
<td>Status</td>
<td>Dynamic</td>
</tr>
<tr>
<td>Display level</td>
<td>Comprehensive</td>
</tr>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration group</td>
<td></td>
</tr>
</tbody>
</table>

cost of a physical io specifies the cost of a single physical IO.

cost of a cpu unit

### Summary information

<table>
<thead>
<tr>
<th>Default value</th>
<th>1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range of values</td>
<td>0 - 65534</td>
</tr>
<tr>
<td>Status</td>
<td>Dynamic</td>
</tr>
<tr>
<td>Display level</td>
<td>Comprehensive</td>
</tr>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration group</td>
<td></td>
</tr>
</tbody>
</table>

cost of a cpu unit specifies the cost of a single CPU operation.

The cost of a serial plan in the optimizer is described by this formula:

\[
\text{Cost} = \text{PIO} \times \text{estimated\_pio} + \text{LIO} \times \text{estimated\_lio} + 100 \times \text{estimated\_cpu} / \text{CPU}
\]

Where the default values of:

- PIO = 25
- LIO = 2
Configuration parameters

- CPU = 1000

If your Adaptive Server has sufficient memory, then all tables exist in memory, and a value of 0 for cost of a physical io is appropriate.

If you CPU is fast enough that the value for cost of a cpu unit is not an issue, then the formula for CPU combining with 2 LIO and 25 PIO (the default values) is:

\[ \text{CPU} \times \frac{100}{\text{configuration\_value}} \]

The default value for \text{configuration\_value} is 1000

As the you increase the value for cost of a cpu unit, this formula reduces the impact of CPU on cost

cpu accounting flush interval

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

cpu accounting flush interval specifies the amount of time, in machine clock ticks, that Adaptive Server waits before flushing CPU usage statistics for each user from sysprocesses to syslogins, a procedure used in chargeback accounting. (This is measured in machine clock ticks, not Adaptive Server clock ticks.)

When a user logs in to Adaptive Server, the server begins accumulating figures for CPU usage for that user process in sysprocesses. When a user logs off Adaptive Server, or when the value of cpu accounting flush interval is exceeded, the accumulated CPU usage statistics are flushed from sysprocesses to syslogins. These statistics continue accumulating in syslogins until you clear the totals using sp_clearstats. You can display the current totals from syslogins using sp_reportstats.

The value to which you set cpu accounting flush interval depends on the type of reporting you intend to do. If you run reports on a monthly basis, set cpu accounting flush interval to a relatively high value. With infrequent reporting, it is less critical that the data in syslogins be updated as often.
On the other hand, if you perform periodic ad hoc selects on the `totcpu` column in `syslogins` to determine CPU usage by process, set `cpu accounting flush interval` to a lower value. Doing so increases the likelihood of the data in `syslogins` being up-to-date when you execute your selects.

Setting `cpu accounting flush interval` to a low value may cause processes to be mistakenly identified as potential deadlock victims by the lock manager. When the lock manager detects a deadlock, it checks the amount of CPU time accumulated by each competing processes. The process with the lesser amount is chosen as the deadlock victim and is terminated by the lock manager. Additionally, when `cpu accounting flush interval` is set to a low value, the task handlers that store CPU usage information for processes are initialized more frequently, thus making processes appear as if they have accumulated less CPU time than they actually have. Because of this, the lock manager may select a process as the deadlock victim when, in fact, that process has more accumulated CPU time than the competing process.

If you do not intend to report on CPU usage at all, set `cpu accounting flush interval` to its maximum value. This reduces the number of times `syslogins` is updated, and reduces the number of times its pages need to be written to disk.

### `cpu grace time`

<table>
<thead>
<tr>
<th>Summary Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

cpu grace time, together with `timeslice`, specifies the maximum amount of time that a user process can run without yielding the CPU before Adaptive Server preempts it and terminates it with a timeslice error. The units for `cpu grace time` are time ticks, as defined by `sql server clock tick length`. See “sql server clock tick length” on page 226 for more information.
**Configuration parameters**

When a process exceeds cpu grace time Adaptive Server “infects” it by removing the process from the internal queues. The process is killed, but Adaptive Server is not affected. This prevents runaway processes from monopolizing the CPU. If any of your user processes become infected, you may be able to temporarily fix the problem by increasing the value of cpu grace time. However, be sure that the problem really is a process that takes more than the current value of cpu grace time to complete, rather than a runaway process.

Temporarily increasing the cpu grace time value is a workaround, not a permanent fix, since it may cause other complications; see “time slice” on page 240. Also, see Chapter 4, “Using Engines and CPUs” in the Performance and Tuning Series: Basics for a more detailed discussion of task scheduling.

**current audit table**

<table>
<thead>
<tr>
<th>Summary information</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
<td>1</td>
</tr>
<tr>
<td>Range of values</td>
<td>0–8</td>
</tr>
<tr>
<td>Status</td>
<td>Dynamic</td>
</tr>
<tr>
<td>Display level</td>
<td>Intermediate</td>
</tr>
<tr>
<td>Required role</td>
<td>System security officer</td>
</tr>
<tr>
<td>Configuration group</td>
<td>Security Related</td>
</tr>
</tbody>
</table>

current audit table establishes the table where Adaptive Server writes audit rows. A system security officer can change the current audit table, using:

```
sp_configure "current audit table", n
  [, "with truncate"]
```

where n is an integer that determines the new current audit table, as follows:

- 1 means sysaudits_01, 2 means sysaudits_02, and so forth, up to 8.
- 0 tells Adaptive Server to set the current audit table to the next table. For example, if your installation has three audit tables, sysaudits_01, sysaudits_02, and sysaudits_03, Adaptive Server sets the current audit table to:
  - 2 if the current audit table is sysaudits_01
  - 3 if the current audit table is sysaudits_02
  - 1 if the current audit table is sysaudits_03
"with truncate" specifies that Adaptive Server should truncate the new table if it is not already empty. sp_configure fails if this option is not specified and the table is not empty.

---

**Note** If Adaptive Server truncates the current audit table, and you have not archived the data, the table’s audit records are lost. Be sure that the audit data is archived before using the with truncate option.

---

To execute sp_configure to change the current audit table, you must have the sso_role active. You can write a threshold procedure to change the current audit table automatically.

---

### deadlock checking period

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

deadlock checking period specifies the minimum amount of time (in milliseconds) before Adaptive Server initiates a deadlock check for a process that is waiting on a lock to be released. Deadlock checking is time-consuming overhead for applications that experience no deadlocks or very few, and the overhead grows as the percentage of lock requests that must wait for a lock also increases.

If you set deadlock checking period to a nonzero value (n), Adaptive Server initiates a deadlock check after a process waits at least n milliseconds. For example, you can make a process wait at least 700 milliseconds for a lock before each deadlock check by entering:

```sql
sp_configure "deadlock checking period", 700
```

If you set deadlock checking period to 0, Adaptive Server initiates deadlock checking when each process begins to wait for a lock. Any value less than the number of milliseconds in a clock tick is treated as 0. See “sql server clock tick length” on page 226 for more information.
Configuring deadlock checking period to a higher value produces longer delays before deadlocks are detected. However, since Adaptive Server grants most lock requests before this time elapses, the deadlock checking overhead is avoided for those lock requests. If your applications deadlock infrequently, set deadlock checking period to a higher value to avoid the overhead of deadlock checking for most processes. Otherwise, the default value of 500 should suffice.

Use sp_sysmon to determine the frequency of deadlocks in your system and the best setting for deadlock checking period. See the Performance and Tuning Series: Monitoring Adaptive Server with sp_sysmon for more information.

deadlock pipe active

Summary information

| Default value | 0 |
| Range of values | 0–1 |
| Status | Dynamic |
| Display level | Comprehensive |
| Required role | System administrator |
| Configuration groups | Memory Use, Monitoring |

deadlock pipe active controls whether Adaptive Server collects deadlock messages. If both deadlock pipe active and deadlock pipe max messages are enabled, Adaptive Server collects the text for each deadlock. You can retrieve these deadlock messages using monDeadLock.

deadlock pipe max messages

Summary information

| Default value | 0 |
| Range of values | 0–2147483647 |
| Status | Dynamic |
| Display level | Comprehensive |
| Required role | System administrator |
| Configuration group | Monitoring |
deadlock pipe max messages determines the number of deadlock messages Adaptive Server stores per engine. The total number of messages in the monSQLText table will be the value of sql text pipe max messages times the number of engines running.

**deadlock retries**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>

deadlock retries specifies the number of times a transaction can attempt to acquire a lock when deadlocking occurs during an index page split or shrink.

For example, Figure 5-1 illustrates the following scenario:

- Transaction A locks page 1007 and needs to acquire a lock on page 1009 to update the page pointers for a page split.
- Transaction B is also inserting an index row that causes a page split, holds a lock on page 1009, and needs to acquire a lock on page 1007.

In this situation, rather than immediately choosing a process as a deadlock victim, Adaptive Server relinquishes the index locks for one of the transactions. This often allows the other transaction to complete and release its locks.

For the transaction that surrendered its locking attempt, the index is rescanned from the root page, and the page split operation is attempted again, up to the number of times specified by deadlock retries.
**Configuration parameters**

**Figure 5-1: Deadlocks during page splitting in a clustered index**

Transaction A: Splitting index page 1007; holds lock on 1007; needs to acquire a lock on 1009 to update its previous-page pointer.

Transaction B: Splitting index page 1009; holds lock on 1009; needs to acquire a lock on 1007 to update its next-page pointer.

*sp_sysmon* reports on deadlocks and retries. See the *Performance and Tuning Series: Locking and Concurrency Control* for more information.

**default character set id**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>
The default character set id parameter specifies the number of the default character set used by the server. The default is set at installation time, and can be changed later with the Sybase installation utilities. See Chapter 9, “Configuring Character Sets, Sort Orders, and Languages,” for a discussion of how to change character sets and sort orders.

**default database size**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

default database size sets the default number of megabytes allocated to a new user database if the create database statement is issued without any size parameters. A database size given in a create database statement takes precedence over the value set by this configuration parameter.

If most of the new databases on your Adaptive Server require more than one logical page size, you may want to increase the default.

**Note** If you alter the model database, you must also increase the default database size, because the create database command copies model to create a new user database.

**default exp_row_size percent**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>
**Configuration parameters**

`default exp_row_size percent` reserves space for expanding updates in data-only-locked tables, to reduce row forwarding. An “expanding update” is any update to a data row that increases the length of the row. Data rows that allow null values or that have variable-length columns may be subject to expanding updates. In data-only-locked tables, expanding updates can require row forwarding if the data row increases in size so that it no longer fits on the page.

The default value, 5, sets aside 5 percent of the available data page size for use by expanding updates. Since 2002 bytes are available for data storage on pages in data-only-locked tables, this leaves 100 bytes for expansion. This value is applied only to pages for tables that have variable-length columns.

Valid values are 0–100. Setting `default exp_row_size percent` to 0 means that all pages are completely filled and no space is left for expanding updates.

`default exp_row_size percent` is applied to data-only-locked tables with variable-length columns when `exp_row_size` is not explicitly provided with `create table` or set with `sp_chgattribute`. If a value is provided with `create table`, that value takes precedence over the configuration parameter setting. See the *Performance and Tuning Series: Locking and Concurrency Control* for more information.

**default fill factor percent**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

default fill factor percent determines how full Adaptive Server makes each index page when it is creating a new index on existing data, unless the fill factor is specified in the `create index` statement. The fillfactor percentage is relevant only at the time the index is created. As data changes, pages are not maintained at any particular level of fullness.

default fill factor percent affects:

- The amount of storage space used by your data – Adaptive Server redistributes the data as it creates the clustered index.
- Performance – splitting up pages uses Adaptive Server resources.
There is seldom a reason to change default fill factor percent, especially since you can override it in the `create index` command. For more information about the fill factor percentage, see “create index” in the Reference Manual: Commands.

**default language id**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

The default language id parameter is the number of the language that is used to display system messages unless a user has chosen another language from those available on the server. us_english always has an ID of NULL. Additional languages are assigned unique numbers as they are added.

**default network packet size**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>

default network packet size configures the default packet size for all Adaptive Server users. You can set default network packet size to any multiple of 512 bytes; values that are not even multiples of 512 are rounded down.

Memory for all users who log in with the default packet size is allocated from Adaptive Server’s memory pool, as set with total logical memory. This memory is allocated for network packets when Adaptive Server is started.

Each Adaptive Server user connection uses:
Configuration parameters

- One read buffer
- One buffer for messages
- One write buffer

Each of these buffers requires default network packet size bytes. The total amount of memory allocated for network packets is:

\[(\text{number of user connections} + \text{number of worker processes}) \times 3 \times \text{default network packet size}\]

For example, if you set the default network packet size to 1024 bytes, and you have 50 user connections and 20 worker processes, the amount of network memory required is:

\[(50 + 20) \times 3 \times 1024 = 215040 \text{ bytes}\]

If you increase the default network packet size, you must also increase the max network packet size to at least the same size. If the value of max network packet size is greater than the value of default network packet size, increase the value of additional network memory. See “additional network memory” on page 81 for further information.

Use \text{sp\_sysmon} to see how changing the default network packet size parameter affects network I/O management and task switching. For example, try increasing default network packet size and then checking \text{sp\_sysmon} output to see how this affects bcp for large batches. See the \textit{Performance and Tuning Series: Monitoring Adaptive Server with sp\_sysmon}.

Requesting a larger packet size at login

The default packet size for most client programs like bcp and isql is set to 512 bytes. If you change the default packet size, clients must request the larger packet size when they connect. Use the -A flag to Adaptive Server client programs to request a large packet size. For example:

\texttt{isql -A2048}

\textit{default sortorder id}

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
</tbody>
</table>

Adaptive Server Enterprise
The `default sortorder id` parameter is the number of the sort order that is installed as the default on the server. To change the default sort order, see Chapter 9, “Configuring Character Sets, Sort Orders, and Languages.”

### default unicode sortorder

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

The `default unicode sortorder` parameter is a string parameter that defines the default Unicode sort order installed on the server. A string parameter is used rather than a numeric parameter to guarantee a unique ID. To change the Unicode default sort order, see Chapter 9, “Configuring Character Sets, Sort Orders, and Languages.”

### default XML sortorder

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

The `default XML sortorder` parameter is a string parameter that defines the sort order used by the XML engine. A string parameter is used rather than a numeric parameter to guarantee a unique ID. For more information, see Chapter 6, “XML Support for I18N” in *XML Services in Adaptive Server Enterprise.*
Configuration parameters

disable character set conversions

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Default value</strong></td>
</tr>
<tr>
<td><strong>Valid values</strong></td>
</tr>
<tr>
<td><strong>Status</strong></td>
</tr>
<tr>
<td><strong>Display level</strong></td>
</tr>
<tr>
<td><strong>Required role</strong></td>
</tr>
<tr>
<td><strong>Configuration group</strong></td>
</tr>
</tbody>
</table>

Changing disable character set conversions to 1 turns off character set conversion for data moving between clients and Adaptive Server. By default, Adaptive Server performs conversion on data moving to and from clients that use character sets that are different than the server’s. For example, if some clients use Latin-1 (iso_1) and Adaptive Server uses Roman-8 (roman8) as its default character set, data from the clients is converted to Roman-8 when being loaded into Adaptive Server. For clients using Latin-1, the data is reconverted when it is sent to the client; for clients using the same character set as Adaptive Server, the data is not converted.

By setting disable character set conversions, you can request that no conversion take place. For example, if all clients are using a given character set, and you want Adaptive Server to store all data in that character set, you can set disable character set conversions to 1, and no conversion takes place.

disable disk mirroring

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Default value</strong></td>
</tr>
<tr>
<td><strong>Valid values</strong></td>
</tr>
<tr>
<td><strong>Status</strong></td>
</tr>
<tr>
<td><strong>Display level</strong></td>
</tr>
<tr>
<td><strong>Required role</strong></td>
</tr>
<tr>
<td><strong>Configuration group</strong></td>
</tr>
</tbody>
</table>
disable disk mirroring enables or disables disk mirroring for Adaptive Server. This is a global variable; Adaptive Server does not perform any disk mirroring after this configuration parameter is set to 1 and Adaptive Server is restarted. Setting disable disk mirroring to 0 enables disk mirroring.

**Note** Disk mirroring must be disabled if you configure Adaptive Server for Failover in a high availability system.

### disk i/o structures

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

The *disk i/o structures* parameter specifies the initial number of disk I/O control blocks Adaptive Server allocates at start-up.

User processes require a disk I/O control block before Adaptive Server can initiate an I/O request for the process. The memory for disk I/O control blocks is preallocated when Adaptive Server starts. You should configure *disk i/o structures* to as high a value as your operating system allows, to minimize the chance of running out of disk I/O structures. See your operating system documentation for information on concurrent disk I/Os.

Use `sp_sysmon` to determine whether to allocate more disk I/O structures. See the *Performance and Tuning Series: Monitoring Adaptive Server with sp_sysmon*. You can set the `max async i/os per server` configuration parameter to the same value as *disk i/o structures*. See “`max async i/os per server`” on page 149 for more information.

### dtm detach timeout period

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Valid values</td>
</tr>
</tbody>
</table>
Configuration parameters

**Summary information**

<table>
<thead>
<tr>
<th>Status</th>
<th>Dynamic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display level</td>
<td>10</td>
</tr>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration group</td>
<td>DTM Administration</td>
</tr>
</tbody>
</table>

*dtm detach timeout period* sets the amount of time, in minutes, that a distributed transaction branch can remain in the detached state. In some X/Open XA environments, a transaction may become detached from its thread of control (usually to become attached to a different thread of control). Adaptive Server permits transactions to remain in a detached state for the length of time specified by *dtm detach timeout period*. After this time has passed, Adaptive Server rolls back the detached transaction.

**Summary information**

<table>
<thead>
<tr>
<th>Default value</th>
<th>300 (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valid values</td>
<td>1 – 2147483647 (seconds)</td>
</tr>
<tr>
<td>Status</td>
<td>Dynamic</td>
</tr>
<tr>
<td>Display level</td>
<td>10</td>
</tr>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration group</td>
<td>DTM Administration</td>
</tr>
</tbody>
</table>

*dtm lock timeout period* sets the maximum amount of time, in seconds, that a distributed transaction branch waits for lock resources to become available. After this time has passed, Adaptive Server considers the transaction to be in a deadlock situation, and rolls back the transaction branch that triggered the deadlock. This ultimately rolls back the entire distributed transaction.

Distributed transactions may potentially deadlock themselves if they propagate a transaction to a remote server, and in turn, the remote server propagates a transaction back to the originating server. This situation is shown in Figure 5-2: the work of distributed transaction “dxact1” is propagated to Adaptive Server 2 via “rpc1.” Adaptive Server 2 then propagates the transaction back to the coordinating server via “rpc2.” “rpc2” and “dxact1” share the same gtrid but have different branch qualifiers, so they cannot share the same transaction resources. If “rpc2” is awaiting a lock held by “dxact1,” a deadlock situation exists.
Adaptive Server does not attempt to detect interserver deadlocks. Instead, it relies on dtm lock timeout period. In Figure 5-2, after dtm lock timeout period has expired, the transaction created for “rpc2” is aborted. This causes Adaptive Server 2 to report a failure in its work, and “dxact1” is ultimately aborted as well.

The value of dtm lock timeout period applies only to distributed transactions. Local transactions may use a lock timeout period with the server-wide lock wait period parameter.

**Note** Adaptive Server does not use dtm lock timeout period to detect deadlocks on system tables.

### dump on conditions

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Default value</strong></td>
</tr>
<tr>
<td><strong>Range of values</strong></td>
</tr>
<tr>
<td><strong>Status</strong></td>
</tr>
<tr>
<td><strong>Display level</strong></td>
</tr>
<tr>
<td><strong>Required role</strong></td>
</tr>
<tr>
<td><strong>Configuration group</strong></td>
</tr>
</tbody>
</table>
dump on conditions determines whether Adaptive Server generates a dump of data in shared memory when it encounters the conditions specified in maximum dump conditions.

**Note** The dump on conditions parameter is included for use only by Sybase Technical Support. Do not modify it unless you are instructed to do so by Sybase Technical Support.

**dynamic allocation on demand**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>

Determines when memory is allocated for changes to dynamic memory configuration parameters.

If you set dynamic allocation on demand to 1, memory is allocated only as it is needed. That is, if you change the configuration for number of user connections from 100 to 200, the memory for each user is added only when the user connects to the server. Adaptive Server continues to add memory until it reaches the new maximum for user connections.

If dynamic allocation on demand is set to 0, all the memory required for any dynamic configuration changes is allocated immediately. That is, when you change the number of user connections from 100 to 200, the memory required for the extra 100 user connections is immediately allocated.

**enable cis**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Valid values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
</tbody>
</table>

Adaptive Server Enterprise
enable cis

The `enable cis` parameter enables or disables Component Integration Service.

### enable DTM

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Valid values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>

`enable DTM` enables or disables the Adaptive Server Distributed Transaction Management (DTM) feature. When the DTM feature is enabled, you can use Adaptive Server as a resource manager in X/Open XA and MSDTC systems. You must reboot the server for this parameter to take effect. See the XA Interface Integration Guide for CICS, Encina, and TUXEDO for more information about using Adaptive Server in an X/Open XA environment. See Using Adaptive Server Distributed Transaction Management Features for information about transactions in MSDTC environments, and for information about Adaptive Server native transaction coordination services.

**Note**  The license information and the run value for `enable DTM` are independent of each other. Whether or not you have a license for DTM, the Run value and the configuration value are set to 1 after you reboot Adaptive Server. Until you have a license, you cannot run DTM. If you have not installed a valid license, Adaptive Server logs an error message and does not activate the feature. See the installation guide for your platform for information about installing license keys.

### enable encrypted columns

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
</tbody>
</table>

`enable encrypted columns`
Configuration parameters

### Summary information

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range of values</td>
<td>0 - 1</td>
</tr>
<tr>
<td>Status</td>
<td>Dynamic</td>
</tr>
<tr>
<td>Display level</td>
<td>Comprehensive</td>
</tr>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
</tbody>
</table>

The configuration parameter “enable encrypted columns” must be set to 1 (1=on) to use the encryption functionality. If the configuration option is turned off (0=off) in a server that contains encrypted columns, any commands against those columns fail with an error. You must set both the configuration parameter and the license option to enable encryption.

```
1> sp_configure 'enable encrypted columns', 1
2> go
```

This version of Adaptive Server changes `sp_configure "enable encrypted columns"` from a static parameter to a dynamic configuration option. In other words, you do not need to restart Adaptive Server for the parameter to take effect.

### enable enterprise java beans

### Summary information

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
<td>0 (disabled)</td>
</tr>
<tr>
<td>Range of values</td>
<td>0 (disabled), 1 (enabled)</td>
</tr>
<tr>
<td>Status</td>
<td>Static</td>
</tr>
<tr>
<td>Display level</td>
<td>Comprehensive</td>
</tr>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration group</td>
<td>Java Services</td>
</tr>
</tbody>
</table>

...
The `enable enterprise java beans` parameter enables and disables EJB Server in the Adaptive Server database. You cannot use EJB Server until the Adaptive Server is enabled for EJB Server.

**Note** The license information and the Run value for `enable java beans` are independent of each other. Whether or not you have a license for `java`, the Run value and the Config value are set to 1 after you restart Adaptive Server. You cannot run EJB Server until you have a license. If you have not installed a valid license, Adaptive Server logs an error message and does not activate the feature. See the installation guide for your platform for information about installing license keys.

### enable file access

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Valid values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

Enables access through proxy tables to the External File System. Requires a license for ASE_XFS.

### enable full-text search

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Valid values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

Enables Enhanced Full-Text Search services. Requires a license for ASE_EFTS.
**Configuration parameters**

**enable HA**

**Summary information**

<table>
<thead>
<tr>
<th></th>
<th>Default value</th>
<th>Range of values</th>
<th>Status</th>
<th>Display level</th>
<th>Required role</th>
<th>Configuration group</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 (off)</td>
<td>0 – 2</td>
<td>Static</td>
<td>Comprehensive</td>
<td>System administrator</td>
<td>SQL Server Administration</td>
</tr>
</tbody>
</table>

Setting `enable HA` is set to 1 allows you to configure Adaptive Server as a companion server in an active-active high availability subsystem. Setting `enable HA` is set to 2 allows you to configure Adaptive Server as a companion server in an active-passive high availability subsystem.

Adaptive Server uses Sybase Failover to interact with the high availability subsystem. You must set `enable HA` to 1 before you run the `installhasvss` script (or `installhasv` on Windows), which installs the system procedures for Sybase Failover.

**Note** The license information and the Run value for `enable HA` are independent of each other. Whether or not you have a license for Sybase Failover, the Run value and the Config value are set to 1 after you reboot Adaptive Server. And until you have a license, you cannot run Sybase Failover. If you have not installed a valid license, Adaptive Server logs an error message and does not activate the feature. See the installation guide for your platform for information about installing license keys.

Setting `enable HA` to 1 or 2 does not mean that Adaptive Server is configured to work in a high availability system. You must perform the steps described in *Using Sybase Failover in A High Availability System* to configure Adaptive Server to be a companion server in a high availability system.

When `enable HA` is set to 0, you cannot configure for Sybase Failover, and you cannot run `installhasvss` (or `installhasv` on Windows).

**enable housekeeper GC**

**Summary information**

<table>
<thead>
<tr>
<th></th>
<th>Default value</th>
<th>Range of values</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 (on)</td>
<td>0 – 4</td>
</tr>
</tbody>
</table>

Adaptive Server Enterprise
The housekeeper garbage collection task performs space reclamation on data-only-locked tables. When a user task deletes a row from a data-only-locked table, a task is queued to the housekeeper to check the data and index pages for committed deletes.

The housekeeper garbage collection task is controlled by `enable housekeeper GC`. For more information on the housekeeper garbage collection task see Chapter 3, “Using Engines and CPUs” in the *Performance and Tuning Series: Basics*.

These are valid values for `enable housekeeper GC`:

- **0** – disables the housekeeper garbage collection task, but enables the `delete` command’s lazy garbage collection. 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. Sybase does not recommend using this value.
- **1** – enables lazy garbage collection for the housekeeper garbage collection task and the `delete` command. 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.
- **3** – reserved for future.
- **4** – enables aggressive garbage collection for the housekeeper garbage collection task and the `delete` command. This option is the most effective, but the `delete` command is expensive. This option is ideal if the deletes on your DOL tables are in a batch.
- **5** – enables aggressive garbage collection for the housekeeper, and lazy garbage collection for the `delete` command. This option is less expensive for deletes than option 4. This option is suitable when deletes are caused by concurrent transactions.

`sp_sysmon` reports on how often the housekeeper garbage collection task performed space reclamation and how many pages were reclaimed. See the *Performance and Tuning Series: Monitoring Adaptive Server with sp_sysmon*. 

### Summary information

<table>
<thead>
<tr>
<th>Status</th>
<th>Dynamic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display level</td>
<td>Intermediate</td>
</tr>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration group</td>
<td>SQL Server Administration</td>
</tr>
</tbody>
</table>

*sp_sysmon* reports on how often the housekeeper garbage collection task performed space reclamation and how many pages were reclaimed. See the *Performance and Tuning Series: Monitoring Adaptive Server with sp_sysmon*. 

System Administration Guide: Volume 1 117
**Configuration parameters**

**enable java**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

The *enable java* parameter enables and disables Java in the Adaptive Server database. You cannot install Java classes or perform any Java operations until the server is enabled for Java.

**Note** The license information and the Run value for *enable java* are independent of each other. Whether or not you have a license for java, the Run value and the Config value are set to 1 after you restart Adaptive Server. You cannot run Java until you have a license. If you have not installed a valid license, Adaptive Server logs an error message and does not activate the feature. See the installation guide for your platform for information about installing license keys.

**enable job scheduler**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

Determines whether Job Scheduler starts when Adaptive Server starts.

**enable ldap user auth**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
</tbody>
</table>
When `enable ldap user auth` is 1, Adaptive Server searches the LDAP server to authenticate each user. If the LDAP authentication fails, Adaptive Server searches `syslogins` to authenticate the user. Use this level when migrating users from Adaptive Server authentication to LDAP authentication.

**enable literal autoparam**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

`enable literal autoparam` enables and disables literal server-wide parameterization.

**enable logins during recovery**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>
enable logins during recovery determines whether non-system administrator logins are allowed during database recovery. A value of 1 indicates that logins are allowed during recovery, and a value of 0 indicates that logins are not allowed during recovery, that is, only the system administrator can log in to Adaptive Server.

**enable merge join**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

enable merge join enables or disables merge join at the server level. Setting enable merge join to:

- 0 – disables merge joins at the server level
- 1 – enables merge joins at the server level.
- 2 – sets merge joins to their default values at the server level

The default value for merge join depends on current value of the optimization goal configuration parameter:

<table>
<thead>
<tr>
<th>Value for optimization goal</th>
<th>Default value for merge join</th>
</tr>
</thead>
<tbody>
<tr>
<td>allrows_mix</td>
<td>on</td>
</tr>
<tr>
<td>allrows_dss</td>
<td>on</td>
</tr>
<tr>
<td>allrows_oltp</td>
<td>off</td>
</tr>
</tbody>
</table>

**enable metrics capture**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
</tbody>
</table>
enable metrics capture enables Adaptive Server to capture metrics at the server level. Metrics for ad hoc statements are captured in the system catalogs; metrics for statements in a stored procedure are saved in the procedure cache.

**enable monitoring**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

enable monitoring controls whether Adaptive Server collects the monitoring table data. Data is not collected if enable monitoring is set to 0. enable monitoring acts as a master switch that determines whether any of the following configuration parameters are enabled.

**enable pam user auth**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

enable pam user auth controls the ability to authenticate users using pluggable authentication modules (PAM).
Configuration parameters

When `enable pam user auth` is set to 1, Adaptive Server uses the PAM provider to authenticate each user. If the PAM authentication fails, Adaptive Server searches `syslogins` to authenticate the user. Use this level when migrating users from Adaptive Server authentication to PAM authentication.

`enable real time messaging`

Summary information

<table>
<thead>
<tr>
<th>Default value</th>
<th>1 (on)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range of values</td>
<td>0 (off), 1 (on)</td>
</tr>
<tr>
<td>Status</td>
<td>Dynamic</td>
</tr>
<tr>
<td>Display level</td>
<td>Comprehensive</td>
</tr>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration group</td>
<td>SQL Server Administration</td>
</tr>
</tbody>
</table>

Use enables the real time messaging services.

`enable rep agent threads`

Summary information

<table>
<thead>
<tr>
<th>Default value</th>
<th>1 (on)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range of values</td>
<td>0 (off), 1 (on)</td>
</tr>
<tr>
<td>Status</td>
<td>Dynamic</td>
</tr>
<tr>
<td>Display level</td>
<td>Basic</td>
</tr>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration groups</td>
<td>Memory Use, Rep Agent Thread Administration</td>
</tr>
</tbody>
</table>

`enable rep agent threads` enables the RepAgent thread within Adaptive Server.

Through version 11.0.3 of Replication Server, the Log Transfer Manager (LTM), a replication system component, transfers replication data to the Replication Server. Beginning with Replication Server versions later than 11.0.3, transfer of replication data is handled by RepAgent, which runs as a thread under Adaptive Server. Setting `enable rep agent threads` enables this feature.

Other steps are also required to enable replication. For more information, see the Replication Server documentation.
**enable row level access control**

**Summary information**

<table>
<thead>
<tr>
<th>Default value</th>
<th>0 (off)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valid values</td>
<td>0 (off), 1 (on)</td>
</tr>
<tr>
<td>Status</td>
<td>Dynamic</td>
</tr>
<tr>
<td>Display level</td>
<td>Comprehensive</td>
</tr>
<tr>
<td>Required role</td>
<td>System security officer</td>
</tr>
<tr>
<td>Configuration group</td>
<td>Security Related</td>
</tr>
</tbody>
</table>

Enables row level access control. You must have the security services license key enabled before you can configure **enable row level access control**.

**enable ssl**

**Summary information**

<table>
<thead>
<tr>
<th>Default value</th>
<th>0 (off)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valid values</td>
<td>0 (off), 1 (on)</td>
</tr>
<tr>
<td>Status</td>
<td>Static</td>
</tr>
<tr>
<td>Display level</td>
<td>Comprehensive</td>
</tr>
<tr>
<td>Required role</td>
<td>System security officer</td>
</tr>
<tr>
<td>Configuration group</td>
<td>Security Related</td>
</tr>
</tbody>
</table>

The **enable ssl** parameter enables or disables Secure Sockets Layer session-based security.

**enable semantic partitioning**

**Summary information**

<table>
<thead>
<tr>
<th>Default value</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range of values</td>
<td>1 (enabled), 0 (disabled)</td>
</tr>
<tr>
<td>Status</td>
<td>Dynamic</td>
</tr>
<tr>
<td>Display level</td>
<td>Comprehensive</td>
</tr>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration group</td>
<td>SQL Server Administration</td>
</tr>
</tbody>
</table>

Enables partitioning other than round-robin (for example list, hash, and range partitioning) in Adaptive Server. Before you use any of these partitioning schemes, you must first have the appropriate license.
Configuration parameters

**enable surrogate processing**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

Activates the processing and maintains the integrity of surrogate pairs in Unicode data. Set `enable surrogate processing` to 1 to enable surrogate processing. If this is disabled, the server ignores the presence of surrogate pairs in the Unicode data, and all code that maintains the integrity of surrogate pairs is skipped. This enhances performance, but restricts the range of Unicode characters that can appear in the data.

**enable unicode conversion**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>

Activates character conversion using Unilib for the `char`, `varchar`, and `text` datatypes. Set `enable unicode conversion` to 1 to use the built-in conversion. If it cannot find a built-in conversion, Adaptive Server uses the Unilib character conversion. Set `enable unicode conversion` to 2 to use the appropriate Unilib conversion. Set the parameter to 0 to use only the built-in character-set conversion.

**enable unicode normalization**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
</tbody>
</table>
Activates Unilib character normalization. The normalization process modifies the data so there is only a single representation in the database for a given sequence of abstract characters. Often, characters followed by combined diacritics are replaced by precombined forms.

Set enable unicode normalization to 1 to use the built-in process that enforces normalization on all incoming Unicode data. If this parameter is disabled (set to 0), the normalization step is bypassed and the client code is responsible for normalization rather than the server. If normalization is disabled, performance is improved—but only if all clients present Unicode data to the server using the same representation.

**Note** Once disabled, normalization cannot be turned on again. This one-way change prevents non-normalized data from entering the database.

---

### enable webservice

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Default value</strong></td>
</tr>
<tr>
<td><strong>Range of values</strong></td>
</tr>
<tr>
<td><strong>Status</strong></td>
</tr>
<tr>
<td><strong>Display level</strong></td>
</tr>
<tr>
<td><strong>Required role</strong></td>
</tr>
<tr>
<td><strong>Configuration group</strong></td>
</tr>
</tbody>
</table>

Enables the webservice. A value of one enables webservice, and a value of 0 disables webservice.

---

### enable xact coordination

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Default value</strong></td>
</tr>
</tbody>
</table>

Enables XACT coordination.
**Configuration parameters**

**enable xact coordination** enables or disables Adaptive Server transaction coordination services. When this parameter is set to 1 (on), coordination services are enabled, and the server can propagate transactions to other Adaptive Servers. This may occur when a transaction executes a remote procedure call (RPC) to update data in another server, or updates data in another server using Component Integration Services (CIS). Transaction coordination services ensure that updates to remote Adaptive Server data commit or roll back with the original transaction.

If this parameter is set to 0 (off), Adaptive Server does not coordinate the work of remote servers. Transactions can still execute RPCs and update data using CIS, but Adaptive Server cannot ensure that remote transactions are rolled back with the original transaction or that remote work is committed along with an original transaction, if remote servers experience a system failure. This corresponds to the behavior of Adaptive Server versions earlier than version 12.x.

**enable xml**

Enables the XML services. A value of one enables XML services, and a value of 0 disables XML services.

---

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Valid values</strong></td>
</tr>
<tr>
<td><strong>Status</strong></td>
</tr>
<tr>
<td><strong>Display level</strong></td>
</tr>
<tr>
<td><strong>Required role</strong></td>
</tr>
<tr>
<td><strong>Configuration group</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Summary information</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Default value</strong></td>
</tr>
<tr>
<td><strong>Range of values</strong></td>
</tr>
<tr>
<td><strong>Status</strong></td>
</tr>
<tr>
<td><strong>Display level</strong></td>
</tr>
<tr>
<td><strong>Required role</strong></td>
</tr>
<tr>
<td><strong>Configuration group</strong></td>
</tr>
</tbody>
</table>
errorlog pipe active

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

errorlog pipe active controls whether Adaptive Server collects error log messages. If both errorlog pipe active and errorlog pipe max messages are enabled, Adaptive Server collects all the messages sent to the error log. You can retrieve these error log messages using monErrorLog.

errorlog pipe max messages

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>

errorlog pipe max messages determines the number of error log messages Adaptive Server stores per engine. The total number of messages in the monSQLText table will be the value of sql text pipe max messages times the number of engines running.

esp execution priority

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>
Configuration parameters

The esp execution priority parameter sets the priority of the XP Server thread for ESP execution. ESPs can be CPU-intensive over long periods of time. Also, since XP Server resides on the same machine as Adaptive Server, XP Server can impact Adaptive Server’s performance.

Use esp execution priority to set the priority of the XP Server thread for ESP execution. See the Open Server Server-Library/C Reference Manual for information about scheduling Open Server threads.

esp execution stacksize

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

The esp execution stacksize parameter sets the size of the stack, in bytes, to be allocated for ESP execution.

Use this parameter if you have your own ESP functions that require a larger stack size than the default, 34816.

esp unload dll

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

The esp unload dll parameter specifies whether DLLs that support ESPs should be automatically unloaded from XP Server memory after the ESP call has completed.

If esp unload dll is set to 0, DLLs are not automatically unloaded. If it is set to 1, they are automatically unloaded.
If `esp unload dll` is set to 0, you can still unload individual DLLs explicitly at runtime, using `sp_freedll`.

**event buffers per engine**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>100</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>1–2147483647</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Static</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Comprehensive</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
<tr>
<td>Memory Use, SQL Server Administration</td>
</tr>
</tbody>
</table>

The *event buffers per engine* parameter specifies the number of events per Adaptive Server engine that can be monitored simultaneously by Adaptive Server Monitor. Events are used by Adaptive Server Monitor for observing Adaptive Server performance; if you are not using Adaptive Server Monitor, set this parameter to 1.

The value to which you set *event buffers per engine* depends on the number of engines in your configuration, the level of activity on your Adaptive Server, and the types of applications you are running.

Setting *event buffers per engine* to a low value may result in the loss of event information. The default value is likely to be too low for most sites. Values of 2000 and above may be more reasonable for general monitoring. However, you should experiment to determine the appropriate value for your site.

In general, setting *event buffers per engine* to a high value may reduce the amount of performance degradation that Adaptive Server Monitor causes Adaptive Server.

Each event buffer uses 100 bytes of memory. To determine the total amount of memory used by a particular value for *event buffers per engine*, multiply the value by the number of Adaptive Server engines in your configuration.

**event log computer name (Windows only)**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>'LocalSystem'</td>
</tr>
</tbody>
</table>
**Configuration parameters**

### Summary information

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
</table>
| Valid values               | Name of an Windows machine on the network configured to record Adaptive Server messages  
|                            | ‘LocalSystem’                                                        |
|                            | ‘NULL’                                                                |
| Status                     | Dynamic                                                              |
| Display level              | Comprehensive                                                        |
| Required role              | System administrator                                                 |
| Configuration group        | Error Log                                                             |

The `event log computer name` parameter specifies the name of the Windows PC that logs Adaptive Server messages in its Windows Event Log. You can use this parameter to log Adaptive Server messages logged to a remote machine. This feature is available on Windows servers only.

A value of ‘LocalSystem’ or ‘NULL’ specifies the default local system.

You can also use the Server Config utility to set the `event log computer name` parameter by specifying the Event Log Computer Name under Event Logging. See the configuration guide for information about the Server Config utility.

Setting the `event log computer name` parameter with `sp_configure` or specifying the Event Log Computer Name under Event Logging overwrites the effects of the command line `-G` option, if it was specified. If Adaptive Server was started with the `-G` option, you can change the destination remote machine by setting the `event log computer name` parameter.

For more information about logging Adaptive Server messages to a remote site, see the *Configuration Guide for Windows*.

---

**event logging (Windows only)**

### Summary information

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
<td>1</td>
</tr>
<tr>
<td>Valid values</td>
<td>0 (off), 1 (on)</td>
</tr>
<tr>
<td>Status</td>
<td>Dynamic</td>
</tr>
<tr>
<td>Display level</td>
<td>Comprehensive</td>
</tr>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration group</td>
<td>Error Log</td>
</tr>
</tbody>
</table>

---
The event logging parameter enables and disables the logging of Adaptive Server messages in the Windows Event Log. This feature is available on Windows servers only.

The default value of 1 enables Adaptive Server message logging in the Windows Event Log; a value of 0 disables it.

You use the Server Config utility to set the event logging parameter by selecting “Use Windows Event Logging” under Event Logging. See the configuration guide for information about the Server Config utility.

Setting the event logging parameter or selecting “Use Windows Event Logging” overwrites the effects of the command line -g option, if it was specified.

**executable codesize + overhead**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

executable codesize + overhead reports the combined size (in kilobytes) of the Adaptive Server executable and overhead. It is a calculated value and is not user-configurable.

**extended cache size**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

extended cache size specify the size of the secondary cache.
Configuration parameters

**global async prefetch limit**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

The `global async prefetch limit` parameter specifies the percentage of a buffer pool that can hold the pages brought in by asynchronous prefetch that have not yet been read. This parameter sets the limit for all pools in all caches for which the limit has not been set explicitly with `sp_poolconfig`.

If the limit for a pool is exceeded, asynchronous prefetch is temporarily disabled until the percentage of unread pages falls below the limit. For more information, see Chapter 6, “Tuning Asynchronous Prefetch” in the *Performance and Tuning Series: Basics*.

**global cache partition number**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

global cache partition number sets the default number of cache partitions for all data caches. The number of partitions for a particular cache can be set using `sp_cacheconfig`; the local value takes precedence over the global value.

Use cache partitioning to reduce cache spinlock contention; in general, if spinlock contention exceeds 10 percent, partitioning the cache should improve performance. Doubling the number of partitions cuts spinlock contention by about one-half.

See “Adding cache partitions” on page 113 for information on configuring cache partitions. For more information, see Chapter 6, “Tuning Asynchronous Prefetch” in the *Performance and Tuning Series: Basics*.  

Adaptive Server Enterprise
heap memory per user

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Valid values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>

Heap memory per user configures the amount of heap memory per user. A heap memory pool is an internal memory created at start-up that tasks use to dynamically allocate memory as needed. This memory pool is important if you are running tasks that use wide columns, which require a lot of memory from the stack. The heap memory allocates a temporary buffer that enables these wide column tasks to finish. The heap memory the task uses is returned to the heap memory pool when the task is finished.

The size of the memory pool depends on the number of user connections. Sybase recommends that you set heap memory per user to three times the size of your logical page.

histogram tuning factor

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

Histogram tuning factor controls the number of steps Adaptive Server analyzes per histogram for update statistics, update index statistics, update all statistics, and create index. A value of 1 disables the parameter.

Note: For Adaptive Server releases 15.0.2 ESD #2 and later, if you set histogram tuning factor to the default value of 20 and a large number of steps are requested for the histogram, the actual step count used for the histogram is limited to:
Configuration parameters

\[ \text{min} \left( \text{max} \left( 400, \text{requested\_steps} \right), \text{histogram\_tuning\_factor} \times \text{requested\_steps} \right) \]

which reduces the procedure cache usage.

In the following example, Adaptive Server generates an intermediate 20-step histogram with 30 values:

```sql
sp_configure 'histogram tuning factor', 20
update statistics tab using 30 values
```

Adaptive Server analyzes the histogram and compresses it into the resulting histogram according to the following parameters:

- The first step is copied unchanged.
- The high-frequency steps are copied unchanged.
- The consecutive range steps are collapsed into the resulting step, so the total weight of the collapsed step would not be bigger than one-thirtieth of the value.

The final histogram in `sysstatistics`:

- Has range steps generated in a way similar for a 30-step `update statistics`, and high frequency ranges are isolated as if the histogram were created with 600 steps.
- The total number of steps in the resulting histogram may differ between 30 and 600 values.
- For equally distributed data, the value should be very close to 30.
- More “frequent” values in the table means more steps in the histogram.
- If a column has few different values, all those values may appear as high-frequency cells.

You could achieve the same result by increasing the number of steps to 600 as using histogram tuning factor, but this would use more resources in the buffer and procedure cache.

`histogram tuning factor` minimizes the resources histograms consume, and only increases resource usage when it is in the best interest for optimization. For example, when there is non-uniform distribution of data in a column, or highly duplicated values within a column. In this situation, up to 600 histogram steps are used. However, in most cases, it uses the default value (30 in the example above).
**housekeeper free write percent**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

*housekeeper free write percent* specifies the maximum percentage by which the housekeeper wash task can increase database writes.

For example, to stop the housekeeper task from working when the frequency of database writes reaches 5 percent above normal, set *housekeeper free write percent* to 5:

```
sp_configure "housekeeper free write percent", 5
```

When Adaptive Server has no user tasks to process, the housekeeper wash task automatically begins writing changed pages from cache to disk. These writes result in improved CPU utilization, decreased need for buffer washing during transaction processing, and shorter checkpoints.

In applications that repeatedly update the same database page, the housekeeper wash may initiate some unnecessary database writes. Although these writes occur only during the server’s idle cycles, they may be unacceptable on systems with overloaded disks.

The table and index statistics that are used to optimize queries are maintained in memory structures during query processing. When these statistics change, the changes are not written to the *systabstats* table immediately, to reduce I/O contention and improve performance. Instead, the housekeeper chores task periodically flushes statistics to disk.

The default value allows the housekeeper wash task to increase disk I/O by a maximum of 1 percent. This results in improved performance and recovery speed on most systems.

To disable the housekeeper wash task, set the value of *housekeeper free write percent* to 0:

```
sp_configure "housekeeper free write percent", 0
```

Set this value to 0 only if disk contention on your system is high, and it cannot tolerate the extra I/O generated by the housekeeper wash task.
If you disable the housekeeper tasks, keep statistics current. Commands that write statistics to disk are:

- **update statistics**
- **dbcc checkdb** (for all tables in a database) or **dbcc checktable** (for a single table)
- **sp_flushstats**

Run one of these commands on any tables that have been updated since the last time statistics were written to disk, at the following times:

- Before dumping a database
- Before an orderly shutdown
- After rebooting, following a failure or orderly shutdown; in these cases, you cannot use **sp_flushstats**—you must use **update statistics** or **dbcc** commands
- After any significant changes to a table, such as a large bulk copy operation, altering the locking scheme, deleting or inserting large numbers of rows, or performing a **truncate table** command

To allow the housekeeper wash task to work continuously, regardless of the percentage of additional database writes, set **housekeeper free write percent** to 100:

```
sp_configure "housekeeper free write percent", 100
```

Use **sp_sysmon** to monitor housekeeper performance. See the *Performance and Tuning Series: Monitoring Adaptive Server with sp_sysmon* for more information.

You might also want to look at the number of free checkpoints initiated by the housekeeper task. The *Performance and Tuning Series: Basics* describes this output.

### i/o accounting flush interval

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
</tbody>
</table>
CHAPTER 5  Setting Configuration Parameters

**Summary information**

<table>
<thead>
<tr>
<th>Configuration group</th>
<th>SQL Server Administration</th>
</tr>
</thead>
</table>

**i/o accounting flush interval** specifies the amount of time, in *machine* clock ticks, that Adaptive Server waits before flushing I/O statistics for each user from `sysprocesses` to `syslogins`. This is used for chargeback accounting.

When a user logs in to Adaptive Server, the server begins accumulating I/O statistics for that user process in `sysprocesses`. When the value of **i/o accounting statistics interval** is exceeded, or a user logs off Adaptive Server, the accumulated I/O statistics for that user are flushed from `sysprocesses` to `syslogins`. These statistics continue accumulating in `syslogins` until you clear the totals by using `sp_clearstats`. You can display the current totals from `syslogins` by using `sp_reportstats`.

The value to which you set **i/o accounting flush interval** depends on the type of reporting you intend to do. If you run reports on a monthly basis, set **i/o accounting flush interval** to a relatively high value. With infrequent reporting, it is less critical that the data in `syslogins` be updated frequently.

If you perform periodic ad hoc selects on the `totio` column in `syslogins` to determine I/O volume by process, set **i/o accounting flush interval** to a lower value. Doing so increases the likelihood of the data in `syslogins` being up to date when you execute your selects.

If you do not report on I/O statistics at all, set **i/o accounting flush interval** to its maximum value. This reduces the number of times `syslogins` is updated and the number of times its pages must be written to disk.

---

**i/o batch size**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>
Configuration parameters

**i/o batch size** sets the number of writes issued in a batch before the task goes to sleep. Once this batch is completed, the task is woken up, and the next batch of writes are issued, ensuring that the I/O subsystem is not flooded with many simultaneous writes. Setting **i/o batch size** to the appropriate value can improve the performance of operations like checkpoint, dump database, select into, and so on.

**i/o polling process count**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

**i/o polling process count** specifies the maximum number of processes that Adaptive Server can run before the scheduler checks for disk and/or network I/O completions. Tuning **i/o polling process count** affects both the response time and throughput of Adaptive Server.

Adaptive Server checks for disk or network I/O completions:

- If the number of tasks run since the last time Adaptive Server checked for I/O completions equals the value for **i/o polling process count**, and
- At every Adaptive Server clock tick.

As a general rule, increasing the value of **i/o polling process count** increases throughput for applications that generate a lot of disk and network I/O. Conversely, decreasing the value improves process response time in these applications, possibly at the risk of lowering throughput.

If your applications create both I/O and CPU-bound tasks, tuning **i/o polling process count** to a low value (1–2) ensures that I/O-bound tasks get access to CPU cycles.

For OLTP applications (or any I/O-bound application with user connections and short transactions), tuning **i/o polling process count** to a value in the range of 20–30 may increase throughput, but may also increase response time.

When tuning **i/o polling process count**, consider three other parameters:
• sql server clock tick length, which specifies the duration of Adaptive Server’s clock tick in microseconds. See “sql server clock tick length” on page 226.

• time slice, which specifies the number of clock ticks the Adaptive Server’s scheduler allows a user process to run. See “time slice” on page 240.

• cpu grace time, which specifies the maximum amount of time (in clock ticks) a user process can run without yielding the CPU before Adaptive Server preempts it and terminates it with a timeslice error. See “cpu grace time” on page 97.

Use sp_sysmon to determine the effect of changing the i/o polling process count parameter. See the Performance and Tuning Series: Monitoring Adaptive Server with sp_sysmon for more information.

**identity burning set factor**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

IDENTITY columns are of type numeric and scale zero whose values are generated by Adaptive Server. Column values can range from a low of 1 to a high determined by the column precision.

For each table with an IDENTITY column, Adaptive Server divides the set of possible column values into blocks of consecutive numbers, and makes one block at a time available in memory. Each time you insert a row into a table, Adaptive Server assigns the IDENTITY column the next available value from the block. When all the numbers in a block have been used, the next block becomes available.
This method of choosing IDENTITY column values improves server performance. When Adaptive Server assigns a new column value, it reads the current maximum value from memory and adds 1. Disk access becomes necessary only after all values within the block have been used. Because all remaining numbers in a block are discarded in the event of server failure (or shutdown with nowait), this method can lead to gaps in IDENTITY column values.

Use identity burning set factor to change the percentage of potential column values that is made available in each block. This number should be high enough for good performance, but not so high that gaps in column values are unacceptably large. The default value, 5000, releases .05 percent of the potential IDENTITY column values for use at one time.

To get the correct value for sp_configure, express the percentage in decimal form, and then multiply it by 10^7 (10,000,000). For example, to release 15 percent (.15) of the potential IDENTITY column values at a time, specify a value of .15 times 10^7 (or 1,500,000) in sp_configure:

```
sp_configure "identity burning set factor", 1500000
```

### identity grab size

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

Identity grab size allows each Adaptive Server process to reserve a block of IDENTITY column values for inserts into tables that have an IDENTITY column.

This is useful if you are performing inserts, and you want all the inserted data to have contiguous IDENTITY numbers. For instance, if you are entering payroll data, and you want all records associated with a particular department to be located within the same block of rows, set identity grab size to the number of records for that department.
CHAPTER 5  Setting Configuration Parameters

identity grab size applies to all users on Adaptive Server. Large identity grab size values result in large gaps in the IDENTITY column when many users insert data into tables with IDENTITY columns.

Sybase recommends that you set identity grab size to a value large enough to accommodate the largest group of records you want to insert into contiguous rows.

job scheduler interval

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

Sets the interval when the Job Scheduler checks which scheduled job are due to be executed

job scheduler tasks

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

Maximum number of jobs that can run at the same time through Job Scheduler.

license information

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Valid values</td>
</tr>
<tr>
<td>Status</td>
</tr>
</tbody>
</table>
Configuration parameters

### Summary information

<table>
<thead>
<tr>
<th>Display level</th>
<th>Comprehensive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration group</td>
<td>SQL Server Administration</td>
</tr>
</tbody>
</table>

license information allows Sybase system administrators to monitor the number of user licenses used in Adaptive Server. Enabling this parameter only monitors the number of licenses issued; it does not enforce the license agreement.

If license information is set to 0, Adaptive Server does not monitor license use. If license information is set to a number greater than 0, the housekeeper chores task monitors the number of licenses used during the idle cycles in Adaptive Server. Set license information to the number of licenses specified in your license agreement.

license information is set to 25, by default. To disable license information, issue:

```sql
sp_configure "license information", 0
```

If the number of licenses used is greater than the number to which license information is set, Adaptive Server writes the following error message to the error log:

```
WARNING: Exceeded configured number of user licenses
```

At the end of each 24-hour period, the maximum number of licenses used during that time is added to the syblicenseslog table. The 24-hour period restarts if Adaptive Server is restarted.

See “Monitoring license use” on page 477 for more information.

### lock address spinlock ratio

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>
For Adaptive Servers running with multiple engines, the `address lock spinlock ratio` sets the number of rows in the internal address locks hash table that are protected by one spinlock.

Adaptive Server manages the acquiring and releasing of address locks using an internal hash table with 1031 rows (known as hash buckets). This table can use one or more spinlocks to serialize access between processes running on different engines.

Adaptive Server’s default value for `address lock spinlock ratio` is 100, which defines 11 spinlocks for the address locks hash table. The first 10 spinlocks protect 100 rows each, and the eleventh spinlock protects the remaining 31 rows. If you specify a value of 1031 or greater for `address lock spinlock ratio`, Adaptive Server uses only 1 spinlock for the entire table.

**lock hashtable size**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>

`lock hashtable size` specifies the number of `hash buckets` in the lock hash table. This table manages all row, page, and table locks, and all lock requests. Each time a task acquires a lock, the lock is assigned to a hash bucket, and each lock request for that lock checks the same hash bucket. Setting this value too low results in large numbers of locks in each hash bucket and slows the searches. On Adaptive Servers with multiple engines, setting this value too low can also lead to increased spinlock contention. Do not set the value to less than the default value, 2048.

`lock hashtable size` must be a power of 2. If the value you specify is not a power of 2, `sp_configure` rounds the value to the next highest power of 2 and prints an informational message.
Configuration parameters

The optimal hash table size is a function of the number of distinct objects (pages, tables, and rows) that will be locked concurrently. The optimal hash table size is at least 20 percent of the number of distinct objects that need to be locked concurrently. See the Performance and Tuning Series: Locking and Concurrency Control for more information on configuring the lock hash table size.

lock scheme

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

lock scheme sets the default locking scheme to be used by create table and select into commands when a lock scheme is not specified in the command.

The values for lock scheme are character data, so you must use 0 as a placeholder for the second parameter, which must be numeric, and specify allpages, datapages, or datarows as the third parameter:

```
sp_configure "lock scheme", 0, datapages
```

lock shared memory

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Valid values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

lock shared memory disallows swapping of Adaptive Server pages to disk and allows the operating system kernel to avoid the server’s internal page locking code. This can reduce disk reads, which are expensive.
Not all platforms support shared memory locking. Even if your platform does, lock shared memory may fail due to incorrectly set permissions, insufficient physical memory, or for other reasons. See the configuration documentation for your platform for information on shared memory locking.

**lock spinlock ratio**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>

Adaptive Server manages the acquiring and releasing of locks using an internal hash table with a configurable number of hash buckets. On SMP systems, this hash table can use one or more spinlocks to serialize access between processes running on different engines. To set the number of hash buckets, use lock hashtable size.

For Adaptive Servers running with multiple engines, lock spinlock ratio sets a ratio that determines the number of lock hash buckets that are protected by one spinlock. If you increase lock hashtable size, the number of spinlocks increases, so the number of hash buckets protected by one spinlock remains the same.

The Adaptive Server default value for lock spinlock ratio is 85. With lock hashtable size set to the default value of 2048, the default spinlock ratio defines 26 spinlocks for the lock hash table. For more information about configuring spinlock ratios, see “Configuring spinlock ratio parameters” on page 134 in System Administration Guide: Volume 2.

sp_sysmon reports on the average length of the hash chains in the lock hash table. See the Performance and Tuning Series: Monitoring Adaptive Server with sp_sysmon for more information.

**lock table spinlock ratio**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
</tbody>
</table>
For Adaptive Servers running with multiple engines, the **table lock spinlock ratio** configuration parameter sets the number of rows in the internal table locks hash table that are protected by one spinlock.

Adaptive Server manages the acquiring and releasing of table locks using an internal hash table with 101 rows (known as hash buckets). This table can use one or more spinlocks to serialize access between processes running on different engines.

The Adaptive Server default value for **table lock spinlock ratio** is 20, which defines 6 spinlocks for the table locks hash table. The first 5 spinlocks protect 20 rows each; the sixth spinlock protects the last row. If you specify a value of 101 or greater for **table lock spinlock ratio**, Adaptive Server uses only 1 spinlock for the entire table.

**lock wait period**

**Summary information**

<table>
<thead>
<tr>
<th></th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
<td>2147483647</td>
</tr>
<tr>
<td>Range of values</td>
<td>0–2147483647</td>
</tr>
<tr>
<td>Status</td>
<td>Dynamic</td>
</tr>
<tr>
<td>Display level</td>
<td>Comprehensive</td>
</tr>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration group</td>
<td>Lock Manager</td>
</tr>
</tbody>
</table>

**lock wait period** limits the number of seconds that tasks wait to acquire a lock on a table, data page, or data row. If the task does not acquire the lock within the specified time period, Adaptive Server returns error message 12205 to the user and rolls back the transaction.

The **lock wait** option of the **set** command sets a session-level number of seconds that a task waits for a lock. It overrides the server-level setting for the session.
lock wait period, used with the session-level setting set lock wait nnn, is applicable only to user-defined tables. These settings have no influence on system tables.

At the default value, all processes wait indefinitely for locks. To restore the default value, reset the value to 2147483647 or enter:

    sp_configure "lock wait period", 0, "default"

**log audit logon failure**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

The log audit logon failure parameter specifies whether to log unsuccessful Adaptive Server logins to the Adaptive Server error log and, on Windows servers, to the Windows Event Log, if event logging is enabled.

A value of 1 requests logging of unsuccessful logins; a value of 0 specifies no logging.

**log audit logon success**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

The log audit logon success parameter specifies whether to log successful Adaptive Server logins to the Adaptive Server error log and, on Windows servers, to the Windows Event Log, if event logging is enabled.
**Configuration parameters**

A value of 1 requests logging of successful logins; a value of 0 specifies no logging.

**max async i/os per engine**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

max async i/os per engine specifies the maximum number of outstanding asynchronous disk I/O requests for a single engine at one time.

**On the Linux platform**

On the Linux platform, max async i/os per engine controls the number of asynchronous I/Os each engine reserves from the operating system when the machine starts. Your system may benefit from using a number greater than the default value.

You can use sp_sysmon to help tune the max async i/os per engine parameter. sp_sysmon’s disk i/o section contains information about the maximum number of outstanding I/Os for each engine during the sample period and the number of I/Os that were delayed because of engine or operating system limits. Generally, any I/Os delayed by engine limits indicate that you should increase the value of max async i/os per engine.

Whether Adaptive Server can perform asynchronous IO on a device depends on whether or not this device support KAIO. The Linux kernel requires that you implement KAIO support at the filesystem level. Most major filesystems provide support for KAIO, including ext3, xfs, jfs, and raw devices (The tmpfs file system does not support KAIO). If the device does not support KAIO, Adaptive Server cannot perform asynchronous IO on that device, and instead reverts to standard synchronous IO for all reads and writes to that device. Adaptive Server prints a message similar to the following in the errorlog indicating that the device has switched to synchronous IO:

```
00:0000:00001:2006/12/15 11:47:17.98 kernel  Virtual device
'/dev/shm/tempdb.dat' does not support kernel asynchronous i/o. Synchronous i/o will be used for this device.
```
**max async i/os per server**

<table>
<thead>
<tr>
<th>Summary information</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
<td>Platform dependent</td>
</tr>
<tr>
<td>Range of values</td>
<td>1 – platform dependent value</td>
</tr>
<tr>
<td>Status</td>
<td>Static</td>
</tr>
<tr>
<td>Display level</td>
<td>Comprehensive</td>
</tr>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration group</td>
<td>O/S Resources</td>
</tr>
</tbody>
</table>

The `max async i/os per server` parameter specifies the maximum number of asynchronous disk I/O requests that can be outstanding for Adaptive Server at one time. This limit is not affected by the number of online engines per Adaptive Server; `max async i/os per server` limits the total number of asynchronous I/Os a server can issue at one time, regardless of how many online engines it has. `max async i/os per engine` limits the number of outstanding I/Os per engine.

Most operating systems limit the number of asynchronous disk I/Os that can be processed at any one time; some operating systems limit the number per operating system process, some limit the number per system, and some do both. If an application exceeds these limits, the operating system returns an error message. Because operating system calls are relatively expensive, it is inefficient for Adaptive Server to attempt to perform asynchronous I/Os that get rejected by the operating system.

To avoid this, Adaptive Server maintains a count of the outstanding asynchronous I/Os per engine and per server; if an engine issues an asynchronous I/O that would exceed either `max async i/os per engine` or `max async i/os per server`, Adaptive Server delays the I/O until enough outstanding I/Os have completed to fall below the exceeded limit.

For example, assume an operating system limit of 200 asynchronous I/Os per system and 75 per process and an Adaptive Server with three online engines. The engines currently have a total of 200 asynchronous I/Os pending, distributed according to the following table:

<table>
<thead>
<tr>
<th>Engine</th>
<th>Number of I/Os pending</th>
<th>Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>60</td>
<td>Engine 0 delays any further asynchronous I/Os until the total for the server is under the operating system <code>per-system</code> limit and then continues issuing asynchronous I/Os.</td>
</tr>
<tr>
<td>1</td>
<td>75</td>
<td>Engine 1 delays any further asynchronous I/Os until the per-engine total is under the operating system <code>per-process</code> limit and then continues issuing asynchronous I/Os.</td>
</tr>
</tbody>
</table>
Configuration parameters

<table>
<thead>
<tr>
<th>Engine</th>
<th>Number of I/Os pending</th>
<th>Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>65</td>
<td>Engine 2 delays any further asynchronous I/Os until the total for server is under the operating system per-system limit and then continues issuing asynchronous I/Os.</td>
</tr>
</tbody>
</table>

All I/Os (both asynchronous and synchronous) require a disk I/O structure, so the total number of outstanding disk I/Os is limited by the value of disk i/o structures. It is slightly more efficient for Adaptive Server to delay the I/O because it cannot get a disk I/O structure than because the I/O request exceeds max i/os per server. Set max async i/os per server equal to the value of disk i/o structures. See “disk i/o structures” on page 109.

If the limits for asynchronous I/O can be tuned on your operating system, make sure they are set high enough for Adaptive Server. There is no penalty for setting them as high as needed.

Use sp_sysmon to see if the per server or per engine limits are delaying I/O on your system. If sp_sysmon shows that Adaptive Server exceeded the limit for outstanding requests per engine or per server, raise the value of the corresponding parameter. See the Performance and Tuning Series: Monitoring Adaptive Server with sp_sysmon for more information.

max cis remote connections

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

The max cis remote connections parameter specifies the maximum number of concurrent Client-Library connections that can be made to remote servers by Component Integration Services.

By default, Component Integration Services allows up to four connections per user to be made simultaneously to remote servers. If you set the maximum number of users to 25, as many as 100 simultaneous Client-Library connections are allowed by Component Integration Services.
If this number does not meet the needs of your installation, you can override the setting by specifying exactly how many outgoing Client-Library connections you want the server to be able to make at one time.

**max concurrently recovered db**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Valid values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

**max concurrently recovered db** determines the degree of parallelism. The minimum value is 1, but you can also use the default value of 0, directing Adaptive Server to use a self-tuning approach. The maximum value is the number of engines at startup minus 1. max concurrently recovered db is also limited by the value of the configuration parameter number of open databases.

The default value is 0, which indicates automatic self-tuning by the server to determine the appropriate number of recovery tasks. A value of 1 indicates serial recovery.

**max memory**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>

Specifies the maximum amount of total physical memory that you can configure Adaptive Server to allocate. **max memory** must be greater than the total logical memory consumed by the current configuration of Adaptive Server.
Configuration parameters

There is no performance penalty for configuring Adaptive Server to use the maximum memory available to it on your computer. However, assess the other memory needs on your system, or Adaptive Server may not be able to acquire enough memory to start.


If Adaptive Server cannot start

When allocate max shared memory is set to 1, Adaptive Server must have the amount of memory available that is specified by max memory. If the memory is not available, Adaptive Server does not start. If this occurs, reduce the memory requirements for Adaptive Server by manually changing the value of max memory in the server’s configuration file. You can also change the value of allocate max shared memory to 0 so that not all memory required by max memory is required at start-up.

You may also want to reduce the values for other configuration parameters that require large amounts of memory. Then restart Adaptive Server to use the memory specified by the new values. If Adaptive Server fails to start because the total of other configuration parameter values is higher than the max memory value, see Chapter 3, “Configuring Memory,” in System Administration Guide: Volume 2 for information about configuration parameters that use memory.

max native threads per engine

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Maximum values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

max native threads per engine defines the maximum number of native threads the server spawns per engine. When the limit for the native threads is reached, Adaptive Server sessions that require a native thread, sleep until another session releases a native thread.
**max network packet size**

<table>
<thead>
<tr>
<th><strong>Summary information</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
<td>512</td>
</tr>
<tr>
<td>Range of values</td>
<td>512–65024</td>
</tr>
<tr>
<td>Status</td>
<td>Static</td>
</tr>
<tr>
<td>Display level</td>
<td>Intermediate</td>
</tr>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration group</td>
<td>Network Communication</td>
</tr>
</tbody>
</table>

`max network packet size` specifies the maximum network packet size that can be requested by clients communicating with Adaptive Server.

If some of your applications send or receive large amounts of data across the network, these applications can achieve significant performance improvement by using larger packet sizes. Two examples are large bulk copy operations and applications that read or write large text, unitext, and image values.

Generally, you want:

- The value of `default network packet size` to be small for users who perform short queries
- `max network packet size` to be large enough to allow users who send or receive large volumes of data to request larger packet sizes

`max network packet size` must always be as large as, or larger than, the `default network packet size`. Values that are not even multiples of 512 are rounded down.

For client applications that explicitly request a larger network packet size to receive it, you must also configure additional network memory. See “additional network memory” on page 81 for more information.

Open Client Server cannot accept a network packet size greater than 64K. See `bcp` and `isql` in the *Utility Guide* for information on using larger packet sizes from these programs. Open Client Client-Library documentation includes information on using variable packet sizes.

### Choosing packet sizes

For best performance, choose a server packet size that works efficiently with the underlying packet size on your network. The goals are:

- Reducing the number of server reads and writes to the network
• Reducing unused space in network packets (increasing network throughput)

For example, if your network packet size carries 1500 bytes of data, setting Adaptive Server’s packet size to 1024 (512*2) will probably achieve better performance than setting it to 1536 (512*3). Figure 5-3 shows how four different packet size configurations would perform in such a scenario.

Figure 5-3: Factors in determining packet size

Underlying network packets: 1500 bytes after overhead

Packet size 512
Used 1024 bytes
Unused 476 bytes
% Used: 68%
2 server reads

Packet size 1024
Used 1024 bytes
Unused 476 bytes
% Used: 68%
1 server read

Packet size 1536
Used 1536 bytes
Unused 1464 bytes
% Used: 51%
2 server reads

Packet size 2560
Used 2560 bytes
Unused 440 bytes
% Used: 85%
2 server reads

Depending on amount of data, network packets may have 1 or 2 packets

Should yield improved performance over default of 512

Possibly the best option of illustrated choices

Probably the worst option of illustrated choices

Key:
Overhead  Data  Unused
After you determine the available data space of the underlying packets on your network, perform your own benchmark tests to determine the optimum size for your configuration.

Use `sp_sysmon` to see how changing `max network packet size` affects network I/O management and task switching. For example, try increasing `max network packet size` and then checking `sp_sysmon` output to see how this affects `bcp` for large batches. See the *Performance and Tuning Series: Monitoring Adaptive Server with sp_sysmon*.

**max number network listeners**

<table>
<thead>
<tr>
<th>Summary information</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
<td>5</td>
</tr>
<tr>
<td>Range of values</td>
<td>0–2147483647</td>
</tr>
<tr>
<td>Status</td>
<td>Static</td>
</tr>
<tr>
<td>Display level</td>
<td>Comprehensive</td>
</tr>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration groups</td>
<td>Memory Use, Network Communication</td>
</tr>
</tbody>
</table>

`max number network listeners` specifies the maximum number of network listeners allowed by Adaptive Server at one time.

Each master port has one network listener. Generally, there is no need to have multiple master ports, unless your Adaptive Server must communicate over more than one network type. Some platforms support both socket and TLI (Transport Layer Interface) network interfaces. See the configuration documentation for your platform for information on supported network types.

**max online engines**

<table>
<thead>
<tr>
<th>Summary information</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
<td>1</td>
</tr>
<tr>
<td>Range of values</td>
<td>1–128</td>
</tr>
<tr>
<td>Status</td>
<td>Static</td>
</tr>
<tr>
<td>Display level</td>
<td>Intermediate</td>
</tr>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration groups</td>
<td>Memory Use, Processors</td>
</tr>
</tbody>
</table>
The role of **max online engines** is to set a high value of engines to be taken online at any one time in an SMP environment. It does not take the number of CPUs available at start-up into account, and allows users to add CPUs at a later date.

**max engines online** specifies the maximum number of Adaptive Server engines that can be online at any one time in an SMP environment. See Chapter 5, “Managing Multiprocessor Servers,” in *System Administration Guide: Volume 2* for a detailed discussion of how to set this parameter for your SMP environment.

At start-up, Adaptive Server starts with a single engine and completes its initialization, including recovery of all databases. Its final task is to allocate additional server engines. Each engine accesses common data structures in shared memory.

**When tuning the max engines online parameter:**

- Never have more online engines than there are CPUs.
- Depending on overall system load (including applications other than Adaptive Server), you may achieve optimal throughput by leaving some CPUs free to run non-Adaptive Server processes.
- You can achieve better throughput by running fewer engines with high CPU use, rather than by running more engines with low CPU use.
- Scalability is application-dependent. Conduct extensive benchmarks on your application to determine the best configuration of online engines.
- You can use `sp_engine` to take engines offline or to bring them online. You can take all engines offline except engine zero.

See Chapter 3, “Using Engines and CPUs” in the *Performance and Tuning Series: Basics* for more information on performance and engine tuning.

### max online Q engines

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Default value</strong></td>
</tr>
<tr>
<td><strong>Range of values</strong></td>
</tr>
<tr>
<td><strong>Status</strong></td>
</tr>
<tr>
<td><strong>Display level</strong></td>
</tr>
<tr>
<td><strong>Required role</strong></td>
</tr>
</tbody>
</table>

Adaptive Server Enterprise
'max online Q engines' required for MQ. Specifies the maximum number of Q engines you can have online. You may need to increase 'max online engines' to accommodate the number of 'max online Q engines'.

Add 2 more to 'max online engines' assuming 'max online engines' is at 4.

```
sp_configure 'max online engines' 6
go
sp_configure 'max online Q engines'
```

Restart Adaptive Server for 'max online Q engines’ to take effect.

### max parallel degree

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

max parallel degree specifies the server-wide maximum number of worker processes allowed per query. This is called the “maximum degree of parallelism.”

If this number is too low, the performance gain for a given query may not be as significant as it could be; if the number is too high, the server may compile plans that require more processes than are actually available at execution time, or the system may become saturated, resulting in decreased throughput. To enable parallel partition scans, set this parameter to be equal to or greater than the number of partitions in the table you are querying.

The value of this parameter must be less than or equal to the current value of number of worker processes.

If you set max parallel degree to 1:

- Adaptive Server scans all tables or indexes serially.
- Adaptive Server forces serial query execution and the optimizer may select plans with a higher parallel degree than if it is disabled.

Changing max parallel degree causes all query plans in the procedure cache to be invalidated, and new plans are compiled the next time you execute a stored procedure or trigger.
For more information on parallel sorting, see Chapter 9, “Parallel Sorting” in the *Performance and Tuning Series: Query Processing and Abstract Plans.*

**max repartition degree**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

*max repartition degree* configures the amount of dynamic repartitioning Adaptive Server requires, which enables Adaptive Server to use horizontal parallelism. However, if the number of partitions is too large, the system is flooded with worker processes that compete for resources, which degrades performance. The value for *max repartition degree* enforces the maximum number of partitions created for these resources. If all of the tables and indices are unpartitioned, Adaptive Server uses the value for *max repartition degree* to provide the number of partitions to create as a result of re-partitioning the data.

**max resource granularity**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

*max resource granularity* indicates the maximum percentage of the system’s resources a query can use. It is set to 10 percent by default. However, this parameter is not enforced at execution time but is only a guide for the query optimizer, and does not prevent the query processor from running queries in parallel. The query engine can avoid some memory intensive strategies by using *max resource granularity* as a guide.
max scan parallel degree

**Summary information**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
<td>1</td>
</tr>
<tr>
<td>Range of values</td>
<td>1–255</td>
</tr>
<tr>
<td>Status</td>
<td>Dynamic</td>
</tr>
<tr>
<td>Display level</td>
<td>Basic</td>
</tr>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration group</td>
<td>Query Tuning</td>
</tr>
</tbody>
</table>

max scan parallel degree specifies the server-wide maximum degree of parallelism for hash-based scans. Hash-based scans may be used for the following access methods:

- Parallel index scans for partitioned and nonpartitioned tables
- Parallel table scans for nonpartitioned tables

max scan parallel degree applies per table or index; that is, if max scan parallel degree is 3, and one table in a join query is scanned using a hash-based table scan and the second can best be accessed by a hash-based index scan, the query can use 9 worker processes (as long as max scan parallel degree is set to 9 or higher.)

The optimizer uses this parameter as a guideline when it selects the number of processes to use for parallel, nonpartition-based scan operations. It does not apply to parallel sort. Because there is no partitioning to spread the data across devices, parallel processes can be accessing the same device during the scan. This can cause additional disk contention and head movement, which can degrade performance. To prevent multiple disk accesses from becoming a problem, use this parameter to reduce the maximum number of processes that can access the table in parallel.

If this number is too low, the performance gain for a given query is not as significant as it could be; if the number is too large, the server may compile plans that use enough processes to make disk access less efficient. A general rule of thumb is to set this parameter to no more than 2 or 3, because it takes only 2 to 3 worker processes to fully utilize the I/O of a given physical device.

Set the value of this parameter to less than or equal to the current value of max parallel degree. Adaptive Server returns an error if you specify a number larger than the max parallel degree value.

If you set max scan parallel degree to 1, Adaptive Server does not perform hash-based scans.
Configuration parameters

Changing `max scan parallel degree` causes all query plans in the procedure cache to be invalidated, and new plans are compiled the next time you execute a stored procedure or trigger.

**max SQL text monitored**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>

`max SQL text monitored` specifies the amount of memory allocated per user connection for saving SQL text to memory shared by Adaptive Server Monitor. Initially, the amount of memory allocated for saving text is 0, and since this parameter is static, you must restart Adaptive Server before you can start saving SQL text.

If you do not allocate enough memory for the batch statements, the text you want to view may be in the section of the batch that is truncated. Sybase recommends an initial value of 1024 bytes of memory per user connection.

The total memory allocated from shared memory for the SQL text is the product of `max SQL text monitored` multiplied by the currently configured number of user connections.

For more information on `max SQL text monitored`, see “Configuring Adaptive Server to save SQL batch text” on page 363.

**maximum dump conditions**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>
The maximum dump conditions parameter sets the maximum number of conditions you can specify under which Adaptive Server generates a dump of data in shared memory.

**Note** This parameter is included for use only by Sybase Technical Support. Do not modify it unless you are instructed to do so by Sybase Technical Support.

### max buffers per lava operator

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

**maximum buffers per lava operator** sets an upper limit for the number of buffers used by Lava operators that perform sorting or hashing (which are “expensive” in terms of processing). These Lava operators use buffers from the session’s `tempdb` data cache pool as a work area for processing rows.

These Lava operators often recurse through their input streams. Sorting requires subsequent merge passes until there are enough buffers available to merge all of the remaining runs. Hashing requires subsequent passes to build hash tables on any spilled sets until all of the remaining data can fit into an in-memory hash table. Some queries require less I/O if you increase `max buffers per lava operator`. This is particularly true for queries that use the `HASH DISTINCT`, `HASH VECTOR AGGREGATE`, and `HASH UNION` operators.

Be careful when you increase the default value of `max buffers per lava operator` for servers with many concurrent users: Adaptive Server may allocate more buffers solely for expensive operators, reducing the number of buffers available for caching user’s tables and other session’s worktables. Use `sp_sysmon` to analyze `tempdb`’s data caching effectiveness.

`max buffers per lava operator` works with `max resource granularity` to limit the number of buffers used. The limit is set to the minimum of:

- The value of `max buffers per lava operator`, or,
Configuration parameters

- (max resource granularity) X (the number of data buffers in tempdb’s pagesize pool)

See “number of sort buffers” on page 191 for information about setting the amount of memory allocated for sort buffers.

maximum failed logins

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

maximum failed logins allows you to set the server-wide maximum number of failed login attempts for logins and roles.

A value of -1 indicates that the failed login count in the syslogins column logincount is updated whenever an authentication failure occurs, but that the account is not locked. Compare with a 0 (zero) value, which avoids incrementing the column for every failed authentication and avoids locking the account due to authentication failures.

To set the system-wide maximum failed logins to 5, enter:

```
sp_configure "maximum failed logins", 5
```

Use `create role` to set maximum failed logins for a specific role or creation. To create the `intern_role` role with the password “temp244”, and set maximum failed logins for `intern_role` to 20, enter:

```
create role intern_role with passwd "temp244", maximum failed logins 20
```

Use `sp_modifylogin` to set or change maximum failed logins for an existing login. To change maximum failed logins for the login “joe” to 40, enter:

```
sp_modifylogin "joe", @option="maximum failed logins", @value="40"
```

Note: The `value` parameter is a character datatype; therefore, quotes are required for numeric values.
To change the overrides for maximum failed logins for all logins to 3, enter:

```
sp_modifylogin "all overrides", "maximum failed logins", "3"
```

To remove the overrides for maximum failed logins option for all logins, enter:

```
sp_modifylogin "all overrides", @option="maximum failed logins", @value="-1"
```

Use `alter role` to set or change the maximum failed logins for an existing role. For example, to change the maximum failed logins allowed for `physician_role` to 5, enter:

```
alter role physician_role set maximum failed logins 5
```

To remove the overrides for maximum failed logins for all roles, enter:

```
alter role "all overrides" set maximum failed logins -1
```

### maximum job output

**Summary information**

<table>
<thead>
<tr>
<th>Default value</th>
<th>32768</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range of values</td>
<td>0–2147483647</td>
</tr>
<tr>
<td>Status</td>
<td>Dynamic</td>
</tr>
<tr>
<td>Display level</td>
<td>Comprehensive</td>
</tr>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration group</td>
<td>SQL Server Administration</td>
</tr>
</tbody>
</table>

Sets limit, in bytes, on the maximum output a single job can produce. If a job produces more output than specified in this parameter, all the data returned above this value is discarded.

### memory alignment boundary

**Summary information**

<table>
<thead>
<tr>
<th>Default value</th>
<th>Logical page size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range of values</td>
<td>2048–16384</td>
</tr>
<tr>
<td>a. Minimum determined by server’s logical page size</td>
<td></td>
</tr>
<tr>
<td>Status</td>
<td>Static</td>
</tr>
<tr>
<td>Display level</td>
<td>Comprehensive</td>
</tr>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
</tbody>
</table>
### Summary information

| Configuration group | Cache Manager |

The memory alignment boundary parameter determines the memory address boundary on which data caches are aligned.

Some machines perform I/O more efficiently when structures are aligned on a particular memory address boundary. To preserve this alignment, values for memory alignment boundary should always be powers of two between the logical page size and 2048K.

**Note** The memory alignment boundary parameter is included for support of certain hardware platforms. Do not modify it unless you are instructed to do so by Sybase Technical Support.

### memory per worker process

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

memory per worker process specifies the amount of memory (in bytes) used by worker processes. Each worker process requires memory for messaging during query processing. This memory is allocated from a shared memory pool; the size of this pool is memory per worker process multiplied by number of worker processes. For most query processing, the default size is more than adequate. If you use dbcc checkstorage, and have set number of worker processes to 1, you may need to increase memory per worker process to 1792 bytes.

For more information on Adaptive Server's memory allocation, see Chapter 3, “Configuring Memory.”
**CHAPTER 5  Setting Configuration Parameters**

**messaging memory**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

Configures the amount of memory available for Sybase messaging.

**metrics elap max**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
</tbody>
</table>

metrics elap max configures maximum elapsed time and thresholds for QP metrics

**metrics exec max**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
</tbody>
</table>

metrics exec max configures maximum execution time and thresholds for QP metrics.
Configuration parameters

metrics lio max

**Summary information**
- Default value: 0
- Range of values: 0 - 2147483647
- Status: dynamic
- Display level: Comprehensive
- Required role: System administrator

metrics lio max configures maximum logical I/O and thresholds for QP metrics.

metrics pio max

**Summary information**
- Default value: 0
- Range of values: 0 - 2147483647
- Status: dynamic
- Display level: Comprehensive
- Required role: System administrator

metrics pio max configures maximum physical I/O and thresholds for QP metrics.

min pages for parallel scan

**Summary information**
- Default value: 200
- Range of values: 0 - 2147483647
- Status: dynamic
- Display level: Comprehensive
- Required role: System administrator

"parallel scan" is the way of running a query in parallel. The 'min pages for parallel scan' parameter is a lower limit in order to avoid having small queries produce overhead because of parallel processing. There is no benefit in having small queries processed in parallel.
minimum password length

Summary information

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
<td>6</td>
</tr>
<tr>
<td>Range of values</td>
<td>0 – 30</td>
</tr>
<tr>
<td>Status</td>
<td>Dynamic</td>
</tr>
<tr>
<td>Display level</td>
<td>10</td>
</tr>
<tr>
<td>Required role</td>
<td>System security officer</td>
</tr>
<tr>
<td>Configuration group</td>
<td>Security Related</td>
</tr>
</tbody>
</table>

minimum password length allows you to customize the length of server-wide password values or per-login or per-role password values to fit your personal needs. The per-login or per-role minimum password length value overrides the server-wide value. Setting minimum password length affects only the passwords you create after you have set the value; existing password lengths are not changed.

Use minimum password length to specify a server-wide value for minimum password length for both logins and roles. For example, to set the minimum password length for all logins and roles to 4 characters, enter:

```
sp_configure "minimum password length", 4
```

To set minimum password length for a specific login at creation, use `sp_addlogin`. For example, to create the new login “joe” with the password “Djdiek3”, and set minimum password length for “joe” to 4, enter:

```
sp_addlogin joe, "Djdiek3", minimum password length=4
```

To set minimum password length for a specific role at creation, use `create role`. To create the new role “intern_role” with the password “temp244” and set the minimum password length for “intern_role” to 0, enter:

```
create role intern_role with passwd "temp244", minimum password length 0
```

The original password is seven characters, but the password can be changed to one of any length because the minimum password length is set to 0.

Use `sp_modifylogin` to set or change minimum password length for an existing login. `sp_modifylogin` only effects user roles, not system roles. For example, to change minimum password length for the login “joe” to 8 characters, enter:

```
sp_modifylogin "joe", @option=“minimum password length”, @value="8"
```
Note  The value parameter is a character datatype; therefore, quotes are required for numeric values.

To change the value of the overrides for minimum password length for all logins to 2 characters, enter:

sp_modifylogin "all overrides", "minimum password length", @value="2"

To remove the overrides for minimum password length for all logins, enter:

sp_modifylogin "all overrides", @option="minimum password length", @value="-1"

Use alter role to set or change the minimum password length for an existing role. For example, to set the minimum password length for “physician_role”, an existing role, to 5 characters, enter:

alter role physician_role set minimum password length 5

To override the minimum password length for all roles, enter:

alter role "all overrides" set minimum password length -1

msg confidentiality reqd

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

The msg confidentiality reqd parameter requires that all messages into and out of Adaptive Server be encrypted. The use security services parameter must be 1 for messages to be encrypted.

msg integrity reqd

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
</tbody>
</table>
**msg integrity reqd** requires that all messages be checked for data integrity. Use security services must be 1 for message integrity checks to occur. If msg integrity reqd is set to 1, Adaptive Server allows the client connection to succeed unless the client is using one of the following security services: message integrity, replay detection, origin checks, or out-of-seq checks.

**net password encryption reqd** restricts login authentication to use only RSA encryption algorithm or the Sybase proprietary algorithm. Table 5-3 describes valids values for net password encryption reqd.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Allows the client to choose the encryption algorithm used for login passwords on the network, including no password encryption. This is the default value for this configuration parameter and provides functionality most similar to earlier releases. This allows the choice of network password encryption to be established by the client application.</td>
</tr>
<tr>
<td>1</td>
<td>Restricts clients to use either RSA or Sybase proprietary encryption algorithms to encrypt login passwords on the network. This provides an incrementally restrictive setting that allows older clients to connect with the Sybase proprietary algorithm and new clients to connect with the stronger RSA algorithm. A client that attempts to connect without using password encryption will fail.</td>
</tr>
<tr>
<td>2</td>
<td>Restricts clients to use only the RSA encryption algorithms to encrypt login passwords on the network. This provides strong RSA encryption of passwords and requires use of newer clients. A client that attempts to connect without using the RSA encryption will fail.</td>
</tr>
</tbody>
</table>

The client receives the following error message when a connection is refused because network password encryption is required:
Configuration parameters

Msg 1640, Level 16, State 2:
Adaptive Server requires encryption of the login password on the network.

number of alarms

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>

number of alarms specifies the number of alarm structures allocated by Adaptive Server.

The Transact-SQL command `waitfor` defines a specific time, time interval, or event for the execution of a statement block, stored procedure, or transaction. Adaptive Server uses alarms to execute `waitfor` commands correctly. Other internal processes require alarms.

When Adaptive Server needs more alarms than are currently allocated, this message is written to the error log:

    uasetalarm: no more alarms available

The number of bytes of memory required for each is small. If you raise the number of alarms value significantly, you should adjust max memory accordingly.

number of aux scan descriptors

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>
number of aux scan descriptors sets the number of auxiliary scan descriptors available in a pool shared by all users on a server.

Each user connection and each worker process has 48 scan descriptors exclusively allocated to it. Of these, 16 are reserved for user tables, 12 are reserved for worktables, and 20 are reserved for system tables (with 4 of these set aside for rollback conditions). A descriptor is needed for each table referenced, directly or indirectly, by a query. For user tables, a table reference includes the following:

- All tables referenced in the from clause of the query
- All tables referenced in a view named in the query (the view itself is not counted)
- All tables referenced in a subquery
- All tables that need to be checked for referential integrity (these are used only for inserts, updates, and deletes)
- A table created with select...into
- All worktables created for the query

If a table is referenced more than once (for example, in a self-join, in more than one view, or in more than one subquery) the table is counted each time. If the query includes a union, each select statement in the union query is a separate scan. If a query runs in parallel, the coordinating process and each worker process needs a scan descriptor for each table reference.

When the number of user tables referenced by a query scan exceeds 16, or the number of worktables exceeds 12, scan descriptors from the shared pool are allocated. Data-only-locked tables also require a system table descriptor for each data-only-locked table accessed via a table scan (but not those accessed via an index scan). If more than 16 data-only-locked tables are scanned using table scans in a query, auxiliary scan descriptors are allocated for them.

If a scan needs auxiliary scan descriptors after it has used its allotted number, and there are no descriptors available in the shared pool, Adaptive Server displays an error message and rolls back the user transaction.

If none of your queries need additional scan descriptors, you may still want to leave number of aux scan descriptors set to the default value in case your system requirements grow. Set it to 0 only if you are sure that users on your system will not run queries on more than 16 tables and that your tables have few or no referential integrity constraints. See “Monitoring scan descriptor usage” on page 172 for more information.
If your queries need more scan descriptors, use one of the following methods to remedy the problem:

- Rewrite the query, or break it into steps using temporary tables. For data-only-locked tables, consider adding indexes if there are many table scans.

- Redesign the table’s schema so that it uses fewer scan descriptors, if it uses a large number of referential integrity constraints. You can find how many scan descriptors a query would use by enabling `set showplan, noexec on` before running the query.

- Increase the number of aux scan descriptors setting.

The following sections describe how to monitor the current and high-watermark usage with `sp_monitorconfig` to avoid running out of descriptors and how to estimate the number of scan descriptors you need.

**Monitoring scan descriptor usage**

`sp_monitorconfig` reports the number of unused (free) scan descriptors, the number of auxiliary scan descriptors currently being used, the percentage that is active, and the maximum number of scan descriptors used since the server was last started. Run it periodically, at peak periods, to monitor scan descriptor use.

This example output shows scan descriptor use with 500 descriptors configured:

```
sp_monitorconfig "aux scan descriptors"
Usage information at date and time: Apr 22 2002 2:49PM.
Name num_free num_active pct_act Max_Used Reused
-------------- -------- --------- -------- -------- ------
number of aux 260 240 48.00 427 NA
```

Only 240 auxiliary scan descriptors are being used, leaving 260 free. However, the maximum number of scan descriptors used at any one time since the last time Adaptive Server was started is 427, leaving about 20 percent for growth in use and exceptionally heavy use periods. “Re-used” does not apply to scan descriptors.

**Estimating and configuring auxiliary scan descriptors**

To get an estimate of scan descriptor use:
1 Determine the number of table references for any query referencing more than 16 user tables or those that have a large number of referential constraints, by running the query with `set showplan` and `set noexec` enabled. If auxiliary scan descriptors are required, `showplan` reports the number needed:

   Auxiliary scan descriptors required: 17

The reported number includes all auxiliary scan descriptors required for the query, including those for all worker processes. If your queries involve only referential constraints, you can also use `sp_helpconstraint`, which displays a count of the number of referential constraints per table.

2 For each query that uses auxiliary scan descriptors, estimate the number of users who would run the query simultaneously and multiply. If 10 users are expected to run a query that requires 8 auxiliary descriptors, a total of 80 will be needed at any one time.

3 Add the per-query results to calculate the number of needed auxiliary scan descriptors.

### number of checkpoint tasks

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Valid values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

`number of checkpoint tasks` configures parallel checkpoints. The value of `number of checkpoint tasks` must be less than or equal to the value of `number of engines at startup`. The maximum value is limited by the value of the configuration parameters `number of engines online at startup` and `number of open databases`, with an absolute ceiling of 8.

The default value is 1, which implies serial checkpoints is the default behavior.
**Configuration parameters**

### number of devices

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>

The number of devices parameter controls the number of database devices Adaptive Server can use. It does not include devices used for database or transaction log dumps.

When you execute disk init, you can also assign the virtual device number (the *vdevno*), although this value is optional. If you do not assign the *vdevno*, Adaptive Server assigns the next available virtual device number.

If you do assign the virtual device number, each device number must be unique among the device numbers used by Adaptive Server. The number 0 is reserved for the master device. Otherwise, valid numbers are 1–2,147,483,647. You can enter any unused device number.

To determine which numbers are currently in use, enter:

```sql
select vdevno from master..sysdevices
where status & 2 = 2
```

Here, “*status 2*” specifies physical disk.

**Note** On UNIX platforms: If you are using a large number of devices, Sybase recommends that you set the appropriate number of devices and user connections in the configuration file and then restart Adaptive Server. Attempting to configure a large number of devices dynamically using `sp_configure` may fail.

### number of dtx participants

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Valid values</td>
</tr>
<tr>
<td>Status</td>
</tr>
</tbody>
</table>

Adaptive Server Enterprise
number of dtx participants sets the total number of remote transactions that the Adaptive Server transaction coordination service can propagate and coordinate at one time. A DTX participant is an internal memory structure that the coordination service uses to manage a remote transaction branch. As transactions are propagated to remote servers, the coordination service must obtain new DTX participants to manage those branches.

By default, Adaptive Server can coordinate 500 remote transactions. Setting number of dtx participants to a smaller number reduces the number of remote transactions that the server can manage. If no DTX participants are available, new distributed transactions cannot start. In-progress distributed transactions may abort if no DTX participants are available to propagate a new remote transaction.

Setting number of dtx participants to a larger number increases the number of remote transaction branches that Adaptive Server can handle, but also consumes more memory.

**Optimizing the number of dtx participants for your system**

During a peak period, use sp_monitorconfig to examine the use of DTX participants:

```
sp_monitorconfig "number of dtx participants"
```

<table>
<thead>
<tr>
<th>Name</th>
<th>num_free</th>
<th>num_active</th>
<th>pct_act</th>
<th>Max_Used</th>
<th>Reused</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of dtx</td>
<td>80</td>
<td>20</td>
<td>4.00</td>
<td>210</td>
<td>NA</td>
</tr>
</tbody>
</table>

If the num_free value is zero or very low, new distributed transactions may be unable to start due to a lack of DTX participants. Consider increasing the number of dtx participants value.

If the Max_used value is too low, unused DTX participants may be consuming memory that could be used by other server functions. Consider reducing the value of number of dtx participants.
Configuration parameters

**number of dump threads**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

**number of dump threads** controls the number of threads that Adaptive Server spawns to perform a memory dump. Using the appropriate value for number of dump threads can reduce the amount of time the engines are halted during the memory dump.

Consider the following when you are determining the number of threads for memory:

- Use a value of 8 if the machine has enough free memory for the file system cache to hold the entire memory dump.

- If you do not know whether the machine has enough free memory, the value for number of dump threads depends on many factors, including the speed of the I/O system, the speed of the disks, the controller’s cache, whether the dump file lives in a logical volume manager created on several disks, and so on.

- Use a value of 1 (no parallel processing) if you do not halt the engines when performing memory dumps, described below.

When Adaptive Server performs a memory dump, the number of files it creates is the sum of the number of memory segments that it has allocated multiplied by the number of threads configured. Adaptive Server uses separate threads to write on separate files. When this job completes, the engines are restarted, and the files are merged into the target dump file. Because of this, the time to dump the shared memory in parallel is greater than doing it serially.

- If you halt the engines during the memory dump, a value other than 1 may reduce the amount of time the engines spend stopped while dumping the memory.
number of engines at startup

**Summary information**

<table>
<thead>
<tr>
<th>Default value</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range of values</td>
<td>1 – number of CPUs on machine</td>
</tr>
<tr>
<td>Status</td>
<td>Static</td>
</tr>
<tr>
<td>Display level</td>
<td>Basic</td>
</tr>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration groups</td>
<td>Java Services, Memory Use, Processors</td>
</tr>
</tbody>
</table>

Adaptive Server allows users to take all engines offline, except engine zero. number of engines at startup is used exclusively during start-up to set the number of engines brought online. It is designed to allow users the greatest flexibility in the number of engines brought online, subject to the restriction that you cannot set the value of number of engines at startup to a value greater than the number of CPUs on your machine, or to a value greater than the configuration of max online engines. Users who do not intend to bring engines online after start-up should set max online engines and number of engines at startup to the same value. A difference between number of engines at startup and max online engines wastes approximately 1.8 MB of memory per engine.

number of histogram steps

**Summary information**

<table>
<thead>
<tr>
<th>Default value</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range of values</td>
<td>3 – 2147483647</td>
</tr>
<tr>
<td>Status</td>
<td>Dynamic</td>
</tr>
<tr>
<td>Display level</td>
<td>Comprehensive</td>
</tr>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration group</td>
<td>Query Tuning</td>
</tr>
</tbody>
</table>

number of histogram steps specifies the number of steps in a histogram.

number of index trips

**Summary information**

<table>
<thead>
<tr>
<th>Default value</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range of values</td>
<td>0–65535</td>
</tr>
</tbody>
</table>
The number of index trips parameter specifies the number of times an aged index page traverses the most recently used/least recently used (MRU/LRU) chain before it is considered for swapping out. As you increase the value of number of index trips, index pages stay in cache for longer periods of time.

A data cache is implemented as an MRU/LRU chain. As the user threads access data and index pages, these pages are placed on the MRU end of the cache’s MRU/LRU chain. In some high transaction environments (and in some benchmarks), it is desirable to keep index pages in cache, since they will probably be needed again soon. Setting number of index trips higher keeps index pages in cache longer; setting it lower allows index pages to be swapped out of cache sooner.

You do need not set the number of index pages parameter for relaxed LRU pages. For more information, see Chapter 4, “Configuring Data Caches.”

Note If the cache used by an index is relatively small (especially if it shares space with other objects) and you have a high transaction volume, do not set number of index trips too high. The cache can flood with pages that do not age out, and this may lead to the timing out of processes that are waiting for cache space.

### number of java sockets

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Valid values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>
The new `number of java sockets` parameter is necessary to enable the Java VM and the `java.net` classes Sybase supports. To open 10 sockets, for example, enter:

```
sp_configure "number of java sockets", 10
```

### number of large i/o buffers

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Valid values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>

The `number of large i/o buffers` parameter sets the number of allocation unit-sized buffers reserved for performing large I/O for certain Adaptive Server utilities. These large I/O buffers are used primarily by the `load database` command, which uses one buffer to load the database, regardless of the number of stripes it specifies. `load database` then uses up to 32 buffers to clear the pages for the database it is loading. These buffers are not used by `load transaction`. To perform more than six `load database` commands concurrently, configure one large I/O buffer for each `load database` command.

`create database` and `alter database` use these buffers for large I/O while clearing database pages. Each instance of `create database` or `load database` can use up to 32 large I/O buffers.

These buffers are also used by disk mirroring and by some `dbcc` commands.

**Note** In Adaptive Server version 12.5.0.3 and later, the size of the large I/O buffers is one allocation (256 pages), not one extent (8 pages). The server thus requires more memory allocation for large buffers. For example, a disk buffer that required memory for 8 pages in earlier versions now requires memory for 256 pages.
Configuration parameters

**number of locks**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>

The number of locks parameter sets the total number of available locks for all users on Adaptive Server.

The total number of locks needed by Adaptive Server depends on the number and nature of the queries that are running. The number of locks required by a query can vary widely, depending on the number of concurrent and parallel processes and the types of actions performed by the transactions. To see how many locks are in use at a particular time, use `sp_lock`.

For serial operation, we suggest that you can start with an arbitrary number of 20 locks for each active, concurrent connection.

Parallel execution requires more locks than serial execution. For example, if you find that queries use an average of five worker processes, try increasing, by one-third, the number of locks configured for serial operation.

If the system runs out of locks, Adaptive Server displays a server-level error message. If users report lock errors, it typically indicates that you need to increase **number of locks**; but remember that locks use memory. See “Number of locks” on page 68 for information.

**Note** Datarows locking may require that you change the value for number of locks. See the Performance and Tuning Guide for more information.

**number of mailboxes**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
</tbody>
</table>
number of mailboxes specifies the number of mailbox structures allocated by Adaptive Server. Mailboxes, which are used in conjunction with messages, are used internally by Adaptive Server for communication and synchronization between kernel service processes. Mailboxes are not used by user processes. Do not modify this parameter unless instructed to do so by Sybase Technical Support.

number of messages specifies the number of message structures allocated by Adaptive Server. Messages, which are used in conjunction with mailboxes, are used internally by Adaptive Server for communication and synchronization between kernel service processes. Messages are also used for coordination between a family of processes in parallel processing. Do not modify this parameter unless instructed to do so by Sybase Technical Support.

number of oam trips
The **number of oam trips** parameter specifies the number of times an **object allocation map (OAM)** page traverses the MRU/LRU chain before it is considered for swapping out. The higher the value of **number of oam trips**, the longer aged OAM pages stay in cache.

Each table, and each index on a table, has an OAM page, which holds information on pages allocated to the table or index and is checked when a new page is needed for the index or table. (See “page utilization percent” on page 202 for further information.) A single OAM page can hold allocation mapping for between 2,000 and 63,750 data or index pages.

The OAM pages point to the allocation page for each allocation unit where the object uses space. The allocation pages, in turn, track the information about extent and page usage within the allocation unit.

In some environments and benchmarks that involve significant allocations of space (that is, massive bulk copy operations), keeping OAM pages in cache longer improves performance. Setting **number of oam trips** higher keeps OAM pages in cache.

**Note** If the cache is relatively small and used by a large number of objects, do not set **number of oam trips** too high. This may result in the cache being flooded with OAM pages that do not age out, and user threads may begin to time out.

---

**number of open databases**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>

**number of open databases** sets the maximum number of databases that can be open simultaneously on Adaptive Server.

When you calculate a value, include the system databases `master`, `model`, `sybsystemprocs`, and `tempdb`. If you have installed auditing, include the `sybsecurity` database. Also, count the sample databases `pubs2` and `pubs3`, the syntax database `sybsyntax`, and the `dbcc` database `dbccdb` if they are installed.
If you are planning to make a substantial change, such as loading a large database from another server, you can calculate an estimated metadata cache size by using `sp_helpconfig`. `sp_helpconfig` displays the amount of memory required for a given number of metadata descriptors, as well as the number of descriptors that can be accommodated by a given amount of memory. A database metadata descriptor represents the state of the database while it is in use or cached between uses.

If Adaptive Server displays a message saying that you have exceeded the allowable number of open databases, adjust the value.

To set the `number of open databases` parameter optimally:

- Step 1: Determine the total number of databases (database metadata descriptors).
- Step 2: Reset `number of open databases` to that number.
- Step 3: Find the number of active databases (active metadata descriptors) during a peak period.
- Step 4: Reset `number of open databases` to that number, plus 10 percent.

The following section details the basic steps listed above.

1. Use `sp_countmetadata` to find the total number of database metadata descriptors. For example:
   ```
   sp_countmetadata "open databases"
   ```
   The best time to run `sp_countmetadata` is when there is little activity on the server. Running `sp_countmetadata` during a peak time can cause contention with other processes.

   Suppose Adaptive Server reports the following information:
   ```
   There are 50 databases, requiring 1719 Kbytes of memory. The 'open databases' configuration parameter is currently set to 500.
   ```

2. Configure `number of open databases` with the value of 50:
   ```
   sp_configure "number of open databases", 50
   ```
   This new configuration number is only a start; the ideal size should be based on the number of active metadata database cache descriptors, not the total number of databases.

3. During a peak period, find the number of active metadata descriptors. For example:
Configuration parameters

sp_monitorconfig "open databases"

Usage information at date and time: Apr 22 2002 2:49PM.

<table>
<thead>
<tr>
<th>Name</th>
<th>num_free</th>
<th>num_active</th>
<th>pct_act</th>
<th>Max_Used</th>
<th>Reused</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of open</td>
<td>50</td>
<td>20</td>
<td>40.00</td>
<td>26</td>
<td>No</td>
</tr>
</tbody>
</table>

At this peak period, 20 metadata database descriptors are active; the maximum number of descriptors that have been active since the server was last started is 26.


4 Configure *number of open databases* to 26, plus additional space for 10 percent more (about 3), for a total of 29:

```
sp_configure "number of open databases", 29
```

If there is a lot of activity on the server, for example, if databases are being added or dropped, run `sp_monitorconfig` periodically. You must reset the cache size as the number of active descriptors changes.

number of open indexes

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>

`number of open indexes` sets the maximum number of indexes that can be used simultaneously on Adaptive Server.

If you are planning to make a substantial change, such as loading databases with a large number of indexes from another server, you can calculate an estimated metadata cache size by using `sp_helpconfig`. `sp_helpconfig` displays the amount of memory required for a given number of metadata descriptors, as well as the number of descriptors that can be accommodated by a given amount of memory. An index metadata descriptor represents the state of an index while it is in use or cached between uses.
The default run value is 500. If this number is insufficient, Adaptive Server displays a message after trying to reuse active index descriptors, and you must adjust this value.

To configure the number of open indexes parameter optimally, perform the following steps:

1. Use `sp_countmetadata` to find the total number of index metadata descriptors. For example:
   ```sql
   sp_countmetadata "open indexes"
   ```
   The best time to run `sp_countmetadata` is when there is little activity in the server. Running `sp_countmetadata` during a peak time can cause contention with other processes.

   Suppose Adaptive Server reports the following information:
   - There are 698 user indexes in all database(s), requiring 286.289 Kbytes of memory. The 'open indexes' configuration parameter is currently set to 500.

2. Configure the number of open indexes parameter to 698 as follows:
   ```sql
   sp_configure "number of open indexes", 698
   ```
   This new configuration is only a start; the ideal size should be based on the number of active index metadata cache descriptors, not the total number of indexes.

3. During a peak period, find the number of active index metadata descriptors. For example:
   ```sql
   sp_monitorconfig "open indexes"
   ```
   Usage information at date and time: Apr 22 2002 2:49PM.

<table>
<thead>
<tr>
<th>Name</th>
<th>num_free</th>
<th>num_active</th>
<th>pct_act</th>
<th>Max_Used</th>
<th>Reused</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of open</td>
<td>182</td>
<td>516</td>
<td>73.92</td>
<td>590</td>
<td>No</td>
</tr>
</tbody>
</table>

   In this example, 590 is the maximum number of index descriptors that have been used since the server was last started.

   See `sp_monitorconfig` in the Reference Manual for more information.

4. Configure the number of open indexes configuration parameter to 590, plus additional space for 10 percent more (59), for a total of 649:
   ```sql
   sp_configure "number of open indexes", 649
   ```
If there is a lot of activity on the server, for example, if tables are being added or dropped, run `sp_monitorconfig` periodically. You must reset the cache size as the number of active descriptors changes.

### number of open objects

<table>
<thead>
<tr>
<th>Summary information</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
<td>500</td>
</tr>
<tr>
<td>Range of values</td>
<td>100–2147483647</td>
</tr>
<tr>
<td>Status</td>
<td>Dynamic</td>
</tr>
<tr>
<td>Display level</td>
<td>Basic</td>
</tr>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration groups</td>
<td>Memory Use, Meta-Data Caches, SQL Server Administration</td>
</tr>
</tbody>
</table>

**number of open objects** sets the maximum number of objects that can be open simultaneously on Adaptive Server.

If you are planning to make a substantial change, such as loading databases with a large number of objects from another server, you can calculate an estimated metadata cache size by using `sp_helpconfig`. `sp_helpconfig` displays the amount of memory required for a given number of metadata descriptors, as well as the number of descriptors that can be accommodated by a given amount of memory. An object metadata descriptor represents the state of an object while it is in use, or cached between uses.

The default run value is 500. If this number is insufficient, Adaptive Server displays a message after trying to reuse active object descriptors. You must adjust this value.

**To set the number of open objects parameter optimally:**

1. Use `sp_countmetadata` to find the total number of object metadata cache descriptors. For example:
   ```
   sp_countmetadata "open objects"
   ```

   The best time to run `sp_countmetadata` is when there is little activity in the server. Running `sp_countmetadata` during a peak time can cause contention with other processes.

   Suppose Adaptive Server reports the following information:
   ```
   There are 1042 user partitions in all database(s),
   requiring 1003 Kbytes of memory. The 'open objects'
configuration parameter is currently set to 500.

2 Configure the number of open objects parameter to that value, as follows:

   sp_configure "number of open objects", 357

357 covers the 340 user objects, plus 5 percent to accommodate temporary tables.

This new configuration is only a start; the ideal size should be based on the number of active object metadata cache descriptors, not the total number of objects.

3 During a peak period, find the number of active metadata cache descriptors, for example:

   sp_monitorconfig "open objects"

   Usage information at date and time: Aug 20 2007  1:32PM..
   Name                  Num_free Num_active Pct_act Max_Used Num_reuse
   number of open objects 160  357 71.40 397 0

   In this example, 397 is the maximum number of object descriptors that have been used since the server was last started.

4 Configure the number of open objects to 397, plus 10 percent (40), for a total of 437:

   sp_configure "number of open objects", 437

   If there is a lot of activity on the server, for example, if tables are being added or dropped, run sp_monitorconfig periodically. You must reset the cache size as the number of active descriptors changes. See sp_monitorconfig in the Reference Manual for more information.

---

**number of open partitions**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
</tbody>
</table>

---
Configuration parameters

Summary information

<table>
<thead>
<tr>
<th>Configuration groups</th>
<th>Memory Use, Meta-Data Caches</th>
</tr>
</thead>
</table>
| **Number of open partitions** | Specifies the number of partitions that Adaptive Server can access at one time. The default value is 500.

The default run value is 500. If this number is insufficient, Adaptive Server displays a message after trying to reuse active partition descriptors. You must adjust this value.

To set the number of open partitions parameter optimally:

1. Use `sp_countmetadata` to find the total number of open partitions. For example:
   ```sql
   sp_countmetadata "open partitions"
   ```
   
The best time to run `sp_countmetadata` is when there is little activity in the server. Running `sp_countmetadata` during a peak time can cause contention with other processes.

Suppose Adaptive Server reports the following information:

   There are 42 user partitions in all database(s), requiring 109 Kbytes of memory. The 'open partitions' configuration parameter is currently set to 110.

2. Configure number of open partitions to that value, as follows:
   ```sql
   sp_configure "number of open partitions", 110
   ```

3. During a peak period, find the number of active metadata cache descriptors, for example:
   ```sql
   sp_monitorconfig "open partitions"
   ```
   Usage information at date and time: Jun 30 2008  3:15PM.

<table>
<thead>
<tr>
<th>Name</th>
<th>Num_free</th>
<th>Num_active</th>
<th>Pct_act</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max_Use</td>
<td>Reuse_cnt</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>----------</td>
<td>------------</td>
<td>---------</td>
</tr>
<tr>
<td>number of open partitions</td>
<td>27</td>
<td>57</td>
<td>51.8</td>
</tr>
<tr>
<td>83</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

   In this example, 83 is the maximum number of partition descriptors that have been used since the server was last started.

4. Configure the number of open partitions to 83, plus 10 percent (8), for a total of 91:
sp_configure "number of open partitions", 91

If there is a lot of activity on the server, for example, if tables are being added or dropped, run `sp_monitorconfig` periodically. You must reset the cache size as the number of active descriptors changes. See `sp_monitorconfig` in the Reference Manual for more information.

**number of pre-allocated extents**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

`number of pre-allocated extents` specifies the number of extents (eight pages) allocated in a single trip to the page manager. Currently, this parameter is used only by `bcp` to improve performance when copying in large amounts of data. By default, `bcp` allocates two extents at a time and writes an allocation record to the log each time.

Setting `number of pre-allocated extents` means that `bcp` allocates the specified number of extents each time it requires more space, and writes a single log record for the event.

An object may be allocated more pages than actually needed, so the value of `number of pre-allocated extents` should be low if you are using `bcp` for small batches. If you are using `bcp` for large batches, increase the value of `number of pre-allocated extents` to reduce the amount of overhead required to allocate pages and to reduce the number of log records.

**number of Q engines at startup**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
</tbody>
</table>

Configuration parameters

Summary information

'number of Q engines at startup' is required for MQ. Specifies the number of Q engines that are online when the server starts. You may need to increase 'max online engines' to accommodate the number of 'max online Q engines'. You must restart Adaptive Server for 'number of Q engines at startup' to take effect.

Add 2 more to 'max online engines' assuming current 'max online engines' is 4.

    sp_configure 'number of Q engines at startup', 2
    go

number of remote connections

Summary information

Default value  20
Range of values  5–32767
Status  Static
Display level  Intermediate
Required role  System administrator
Configuration groups  Memory Use, Network Communication

number of remote connections specifies the number of logical connections that can be open to and from an Adaptive Server at one time. Each simultaneous connection to XP Server for ESP execution uses up to one remote connection each. For more information, see Chapter 15, “Managing Remote Servers.”

number of remote logins

Summary information

Default value  20
Range of values  0–32767
Status  Static
Display level  Intermediate
Required role  System administrator
Configuration groups  Memory Use, Network Communication
number of remote logins controls the number of active user connections from Adaptive Server to remote servers. Each simultaneous connection to XP Server for ESP execution uses up to one remote login each. Set this parameter to the same (or a lower) value as number of remote connections. For more information, see Chapter 15, “Managing Remote Servers.”

**number of remote sites**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>

number of remote sites determines the maximum number of remote sites that can access Adaptive Server simultaneously. Each Adaptive Server-to-XP Server connection uses one remote site connection.

Internally, number of remote sites determines the number of site handlers that can be active at any one time; all server accesses from a single site are managed with a single site handler. For more information, see Chapter 15, “Managing Remote Servers.”

**number of sort buffers**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

number of sort buffers specifies the amount of memory allocated for buffers used to hold pages read from input tables and perform index merges during sorts. number of sort buffers is used only for parallel sorting. Parallel sorts are used when you:
Configuration parameters

- Run updates statistics
- Create indexes

For more information about running update statistics and creating indexes, see Chapter 10, “Using Statistics to Improve Performance,” in the Performance and Tuning Series: Query Processing and Abstract Plans.

The value you use for number of sort buffers depends on the page size of the server.

Sybase recommends that you leave this parameter set to the default except when you are creating indexes in parallel.

Setting the value too high can rob non-sorting processes of access to the buffer pool in caches being used to perform sorts.

If you configure a high number of sort buffers, a sort on a large table may require more procedure cache. The effect is more pronounced with tables that have smaller row sizes, because the number of rows per page is higher.

This equation estimates the amount of proc cache required (in bytes):

\[
(Number \ of \ sort \ buffers) \times (\text{rows per page}) \times 100
\]

If you do not configure enough procedure cache for the number of sort buffers, the sort may fail with error message 701. If this occurs, reconfigure Adaptive Server with a lower number of sort buffers and retry the sort.

See “max buffers per lava operator” on page 161 for information about setting an upper limit for the number of buffers used by an operator.

number of user connections

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>

number of user connections sets the maximum number of user connections that can be connected to Adaptive Server at the same time. It does not refer to the maximum number of processes; that number depends not only on the value of this parameter but also on other system activity.
The maximum allowable number of file descriptors per process is operating-system-dependent; see the configuration documentation for your platform.

The number of file descriptors available for Adaptive Server connections is stored in the global variable `@@max_connections`. You can report the maximum number of file descriptors your system can use with:

```
select @@max_connections
```

The return value represents the maximum number of file descriptors allowed by the system for your processes, minus overhead. Overhead increases with the number of engines. For more information on how multiprocessing affects the number file descriptors available for Adaptive Server connections, see “Managing user connections” on page 132.

In addition, you must reserve a number of connections for the following items, which you also set with configuration parameters:

- The database devices, including mirror devices
- Site handlers
- Network listeners

The following formula determines how high you can set the number of user connections, number of devices, max online engines, number of remote sites, and max number network listeners:

\[
\text{number of user connections} + (\text{number of devices} \times \text{max online engines} \times 2) + \text{number of remote sites} + \text{max number network listeners} \leq \text{value of } \text{@@max_connections}.
\]

Reserved connections

One connection from the configured number of connections is reserved for temporary administrative tasks to make sure that database administrators can connect to Adaptive Server to increase the number of user connections and there are no free connections. A reserved connection is allocated only to a user who has the `sa_role` and has a total login time of 15 minutes. After this, Adaptive Server terminates the connection to ensure the availability of the reserved connection at an installation with multiple database administrators.

Adaptive Server uses this reserved connection automatically when a client uses the last resource for connecting to Adaptive Server.

If Adaptive Server is using a reserved connection, the following informational message is displayed when the user logs in to Adaptive Server:

```
There are not enough user connections available; you are being connected using a temporary administrative connection which will time out after '15' minutes. Increase the value of the 'number of user connections' parameter
```
Adaptive Server also prints a message similar to the following to the error log when the final connection to Adaptive Server terminates due to a timeout:

```
00:00000:00008:2003/03/14 11:25:31.36 server Process '16' has been
terminated as it exceeded the maximum login time allowed for such processes.
This process used a connection reserved for system administrators and has a
maximum login period of '15' minutes
```

There is no formula for determining how many connections to allow for each user. You must estimate this number, based on the system and user requirements described here. You must also take into account that on a system with many users, there is more likelihood that connections needed only occasionally or transiently can be shared among users. The following processes require user connections:

- One connection is needed for each user running `isql`.
- Application developers use one connection for each editing session.
- The number of connections required by users running an application depends on how the application has been programmed. Users executing Open Client programs need one connection for each open DB-Library `dbprocess` or Client-Library `cs_connection`.

**Note** Sybase suggests that you estimate the maximum number of connections used by Adaptive Server and update `number of user connections` as you add physical devices or users to the system. Use `sp_who` periodically to determine the number of active user connections on your Adaptive Server.

Certain other configuration parameters, including `stack size` and `default network packet size`, affect the amount of memory for each user connection.

Adaptive Server uses the value of the `number of user connections` parameter to establish the number of shared-memory connections for EJB Server. Thus, if `number of user connections` is 30, Adaptive Server establishes 10 shared-memory connections for EJB Server. Shared-memory connections are not a subset of user connections, and are not subtracted from the number of user connections.

To increase the number of user connections for shared memory, you must:

1. Increase `number of user connections` to a number one-third of which is the number of desired shared-memory connections.
2. Restart Adaptive Server.
Although number of user connections is a dynamic configuration parameter, you must restart the server to change the number of user connections for shared memory. See the EJB Server User’s Guide for more information.

With Adaptive Server version 12.5.3, ESD #2, no sockets are automatically reserved for EJB. However, you can enable traceflag 1642 to revert to the previous functionality, reserving one-third of the sockets for EJB. You must enable traceflag 1642 to setup the EJB server. For this release of Adaptive Server, if the message, "hbc_ninit: No sockets available for HBC", is in the errorlog, but the EJB server is not configured, the message can be ignored.

In Adaptive Server version 12.5.3, if the EJB server is enabled and HBC sockets are not available, the message "hbc_ninit: No sockets available for HBC" is reported. If traceflag 1642 is not enabled, then Adaptive Server must be booted with the 1642 traceflag. If the EJB server is not enabled, then no message is reported and Adaptive Server automatically disables the sockets reserved for EJB server.

**number of worker processes**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>

*number of worker processes* specifies the maximum number of worker processes that Adaptive Server can use at any one time for all simultaneously running parallel queries combined.

Adaptive Server issues a warning message at start-up if there is insufficient memory to create the specified number of worker processes. *memory per worker process* controls the memory allocated to each worker process.

If you have not configured *number of worker processes* for a sufficient number of threads from the worker thread pool, Adaptive Server adjusts query plans at runtime to use fewer worker threads. If Adaptive Server cannot adjust the queries at runtime, the queries recompile serially. However, *alter table* and *execute immediate* commands are aborted if they do not have sufficient worker threads.
Configuration parameters

**o/s file descriptors**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

`o/s file descriptors` indicates the maximum per-process number of file descriptors configured for your operating system. This parameter is read-only and cannot be configured through Adaptive Server.

Many operating systems allow you to configure the number of file descriptors available per process. See your operating system documentation for further information on this.

The number of file descriptors available for Adaptive Server connections, which is less than the value of `o/s file descriptors`, is stored in the variable `@@max_connections`. For more information on the number of file descriptors available for connections, see “Upper limit to the maximum number of user connections” on page 193.

**object lockwait timing**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

`object lockwait timing` controls whether Adaptive Server collects timing statistics for requests of locks on objects.
**open index hash spinlock ratio**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>

open index hash spinlock ratio sets the number of index metadata descriptor hash tables that are protected by one spinlock. This parameter is used for multiprocessing systems only.

All the index descriptors belonging to the table are accessible through a hash table. When a query is run on the table, Adaptive Server uses hash tables to look up the necessary index information in its sysindexes rows. A hash table is an internal mechanism used by Adaptive Server to retrieve information quickly.

Usually, you do not need to change this parameter. In rare instances, however, you may need to reset it if Adaptive Server demonstrates contention from hash spinlocks. You can get information about spinlock contention by using sp_sysmon. See the Performance and Tuning Series: Monitoring Adaptive Server with sp_sysmon.

For more information about configuring spinlock ratios, see “Configuring spinlock ratio parameters” on page 134 in System Administration Guide: Volume 2.

**open index spinlock ratio**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>

open index spinlock ratio specifies the number of index metadata descriptors that are protected by one spinlock.
Adaptive Server uses a spinlock to protect an index descriptor, since more than one process can access the contents of the index descriptor. This parameter is used for multiprocessing systems only.

The value specified for this parameter defines the ratio of index descriptors per spinlock.

If one spinlock is shared by too many index descriptors, it can cause spinlock contention. Use `sp_sysmon` to get a report on spinlock contention. See the "Performance and Tuning Series: Monitoring Adaptive Server with sp_sysmon".

If `sp_sysmon` output indicates an index descriptor spinlock contention of more than 3 percent, try decreasing the value of `open index spinlock ratio`.

For more information about configuring spinlock ratios, see see “Configuring spinlock ratio parameters” on page 134 in System Administration Guide: Volume 2.

### open object spinlock ratio

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

`open object spinlock ratio` specifies the number of object descriptors that are protected by one spinlock. Adaptive Server uses a spinlock to protect an object descriptor, since more than one process can access the contents of the object descriptor. This configuration parameter is used for multiprocessing systems only.

The default value for this parameter is 100; 1 spinlock for each 100 object descriptors configured for your server. If your server is configured with only one engine, Adaptive Server sets only 1 object descriptor spinlock, regardless of the number of object descriptors.
If one spinlock is shared by too many object descriptors, it causes spinlock contention. Use `sp_sysmon` to get a report on spinlock contention. See the *Performance and Tuning Series: Monitoring Adaptive Server with sp_sysmon* for more information on spinlock contention. If `sp_sysmon` output indicates an object descriptor spinlock contention of more than 3 percent, try decreasing the value of the `open object spinlock ratio` parameter.

For more information about configuring spinlock ratios, see see “Configuring spinlock ratio parameters” on page 134 in *System Administration Guide: Volume 2*.

### optimization goal

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

Optimization goals are a convenient way of matching the user’s query demands with the best optimization techniques, ensuring optimal use of the optimizer's time and resources. Adaptive Server allows users to configure for two optimization goals, which you can specify at three tiers: server level, session level, and query level.

The server-level optimization goal is overridden at the session level, which is overridden at the query level.

These optimization goals allow you to choose an optimization strategy that best fits your query environment:

- `allrows.oltp` – the most useful goal for purely OLTP queries.
- `allrows.dss` – the most useful goal for operational DSS queries of medium-to-high complexity.

### optimization timeout limit

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
</tbody>
</table>
**Configuration parameters**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

The optimization timeout limit parameter specifies the amount of time Adaptive Server can spend optimizing a query as a fraction of the estimated execution time of the query.

A value of 0 indicates there is no optimization timeout.

**page lock promotion HWM**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>

Page lock promotion HWM (high-water mark), together with the page lock promotion LWM (low-water mark) and page lock promotion PCT (percentage), specifies the number of page locks permitted during a single scan session of a page-locked table or index before Adaptive Server attempts to escalate from page locks to a table lock.

Page lock promotion HWM sets a maximum number of page locks allowed on a table before Adaptive Server attempts to escalate to a table lock. When the number of page locks acquired during a scan session exceeds page lock promotion HWM, Adaptive Server attempts to acquire a table lock. The page lock promotion HWM value cannot be higher than number of locks value.

For more detailed information on scan sessions and setting up page lock promotion limits, see Chapter 2, “Locking Configuration and Tuning,” in the *Performance and Tuning Series: Locking and Concurrency Control.*
The default value for page lock promotion HWM is appropriate for most applications. You may want to raise the value to avoid table locking. For example, if you know that there are regular updates to 500 pages of an allpages-locked or datapages-locked table containing thousands of pages, you can increase concurrency for the tables by setting page lock promotion HWM to 500 so that lock promotion does not occur at the default setting of 200.

You can also configure lock promotion of page-locked tables and views at the per-object level. See `sp_setrowlockpromote` in the Reference Manual.

Use `sp_sysmon` to see how changing page lock promotion HWM affects the number of lock promotions. `sp_sysmon` reports the ratio of exclusive page to exclusive table lock promotions and the ratio of shared page to shared table lock promotions. See the Performance and Tuning Series: Monitoring Adaptive Server with `sp_sysmon`.

### page lock promotion LWM

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>

The page lock promotion LWM (low-water mark) parameter, together with the page lock promotion HWM (high-water mark) and the page lock promotion PCT, specify the number of page locks permitted during a single scan session of a page locked table or an index before Adaptive Server attempts to promote from page locks to a table lock.

The page lock promotion LWM sets the number of page locks below which Adaptive Server does not attempt to issue a table lock on an object. The page lock promotion LWM must be less than or equal to page lock promotion HWM.

The default value for page lock promotion LWM is sufficient for most applications. If Adaptive Server runs out of locks (except for an isolated incident), increase number of locks.

For more information about locking, see the Performance and Tuning Series: Locking and Concurrency Control.
Configuration parameters

You can also configure page lock promotion at the per-object level. See `sp_setpglockpromote` in the Reference Manual: Procedures.

**page lock promotion PCT**

<table>
<thead>
<tr>
<th><strong>Summary information</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>

If the number of locks held on an object is between page lock promotion LWM (low-water mark) and page lock promotion HWM (high-water mark), page lock promotion PCT sets the percentage of page locks (based on the table size) above which Adaptive Server attempts to acquire a table lock.

For more detailed information on setting up page lock promotion limits, see Chapter 2, “Locking Configuration and Tuning,” in the Performance and Tuning Series: Locking and Concurrency Control.

The default value for page lock promotion PCT is appropriate for most applications.

You can also configure lock promotion at the per-object level for page locked objects. See `sp_setpglockpromote` in the Reference Manual.

**page utilization percent**

<table>
<thead>
<tr>
<th><strong>Summary information</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>
The page utilization percent parameter is used during page allocations to control whether Adaptive Server scans a table’s object allocation map (OAM) to find unused pages or simply allocates a new extent to the table. (See “number of oam trips” on page 181 for more information on the OAM.) The page utilization percent parameter is a performance optimization for servers with very large tables; it reduces the time needed to add new space.

If you set page utilization percent to 100, Adaptive Server scans through all OAM pages to find unused pages allocated to the object before allocating a new extent. When this parameter is set lower than 100, Adaptive Server compares the page utilization percent setting to the ratio of used and unused pages allocated to the table, as follows:

\[ 100 \times \frac{\text{used pages}}{\text{used pages} + \text{unused pages}} \]

If the page utilization percent setting is lower than the ratio, Adaptive Server allocates a new extent instead of searching for the unused pages.

For example, when inserting data into a 10GB table that has 120 OAM pages and only 1 unused data page:

- A page utilization percent of 100 tells Adaptive Server to scan through all 120 OAM pages to locate an unused data page.
- A page utilization percent of 95 allows Adaptive Server to allocate a new extent to the object, because 95 is lower than the ratio of used pages to used and unused pages.

A low page utilization percent value results in more unused pages. A high page utilization percent value slows page allocations in very large tables, as Adaptive Server performs an OAM scan to locate each unused page before allocating a new extent. This increases logical and physical I/O.

If page allocations (especially in the case of large inserts) seem to be slow, you can lower the value of page utilization percent, but reset it after inserting the data. A lower setting affects all tables on the server and results in unused pages in all tables.

Fast bulk copy ignores the page utilization percent setting and always allocates new extents until there are no more extents available in the database.

### Summary information

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
</tbody>
</table>
Configuration parameters

Summary information

<table>
<thead>
<tr>
<th>Status</th>
<th>Dynamic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display level</td>
<td>Comprehensive</td>
</tr>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration groups</td>
<td>Memory Use, Meta-Data Cache</td>
</tr>
</tbody>
</table>

Partition groups specifies the maximum number of partition groups that can be allocated by Adaptive Server. Partition groups are internal structures used by Adaptive Server to control access to individual partitions of a table. Partition groups are used during upgrade or during a load database upgrade to unpartition Adaptive Server 12.5.x and earlier partitions.

The default value allows a maximum 1024 open partition groups and a maximum of 16384 (1024 times 16) open partitions. The actual number of partitions may be slightly less, due to the grouping of partitions.

Partition spinlock ratio

Summary information

<table>
<thead>
<tr>
<th>Default value</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range of values</td>
<td>1–2147483647</td>
</tr>
<tr>
<td>Status</td>
<td>Dynamic</td>
</tr>
<tr>
<td>Display level</td>
<td>Comprehensive</td>
</tr>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration groups</td>
<td>Memory Use, Meta-Data Cache</td>
</tr>
</tbody>
</table>

For Adaptive Servers running with multiple engines, partition spinlock ratio sets the number of rows in the partition descriptors that are protected by one spinlock.

Adaptive Server manages access to table partitions using partition descriptors. Each partition descriptor stores information about a partition (for example, the last page of the partition) that processes must use when accessing that partition. Partition descriptors are configured using the configuration parameter number of open partitions.

By default, Adaptive Server systems are configured with partition spinlock ratio set to 10, or 1 spinlock for every 10 partition caches. Decreasing the value of partition spinlock ratio may have little impact on the performance of Adaptive Server. The default setting is correct for most servers.
For more information about configuring spinlock ratios, see “Managing Multiprocessor Servers” on page 125 in *System Administration Guide: Volume 2*.

**per object statistics active**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

*per object statistic active* controls whether Adaptive Server collects statistics for each object.

**percent database for history**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Valid values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

Specifies the amount of space reserved for the *js_history* table, as a percentage of the total space available in *sybmgmtdb*. Increase *percent database for history* if there are more jobs running, or if you need to store historical records about executed jobs for future queries.

**percent database for output**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Valid values</td>
</tr>
<tr>
<td>Status</td>
</tr>
</tbody>
</table>
Configuration parameters

**Summary information**

<table>
<thead>
<tr>
<th>Display level</th>
<th>Intermediate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration group</td>
<td>SQL Server Administration</td>
</tr>
</tbody>
</table>

This specifies the amount of space reserved for jobs' output in percentage of the total space available in `sybmgmtdb`. Legal values are between 0 and 100. Default value is 30. Increase this if there are more jobs running or jobs which produce lot of output information are running and that output needs to be stored for querying.

**percent history free**

**Summary information**

<table>
<thead>
<tr>
<th>Default value</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valid values</td>
<td>0 – 100</td>
</tr>
<tr>
<td>Status</td>
<td>Dynamic</td>
</tr>
<tr>
<td>Display level</td>
<td>Intermediate</td>
</tr>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration group</td>
<td>SQL Server Administration</td>
</tr>
</tbody>
</table>

Specifies what percentage of reserved space in `sybmgmtdb` should be kept free. For example, for the default value of 30 percent, when 70 percent of the space reserved for history in `sybmgmtdb` is occupied, Adaptive Server starts purging the oldest history records to make room for new records.

**percent output free**

**Summary information**

<table>
<thead>
<tr>
<th>Default value</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valid values</td>
<td>0 – 100</td>
</tr>
<tr>
<td>Status</td>
<td>Dynamic</td>
</tr>
<tr>
<td>Display level</td>
<td>Intermediate</td>
</tr>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration group</td>
<td>SQL Server Administration</td>
</tr>
</tbody>
</table>
Specifies the percentage of reserved space kept free in `sybmgmtdb` that is reserved for the Job Scheduler’s output. For example, for the default value of 30 percent, when 70 percent of the space reserved for history in `sybmgmtdb` is occupied, Adaptive Server starts purging the oldest history records to make room for new records.

**performance monitoring option**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

**permission cache entries**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>

**permission cache entries** determines the number of cache protectors per task. This parameter increases the amount of memory for each user connection and worker process.

Information about user permissions is held in the permission cache. When Adaptive Server checks permissions, it looks first in the permission cache; if it does not find what it needs, it looks in the `sysprotects` table. This process is significantly faster if Adaptive Server finds the information it needs in the permission cache and does not have to read `sysprotects`. 
Configuration parameters

However, Adaptive Server looks in the permission cache only when it is checking user permissions, not when permissions are being granted or revoked. When a permission is granted or revoked, the entire permission cache is flushed. This is because existing permissions have timestamps that become outdated when new permissions are granted or revoked.

If users on your Adaptive Server frequently perform operations that require their permissions to be checked, you may see a small performance gain by increasing the value of permission cache entries. This effect is not likely to be significant enough to warrant extensive tuning.

If users on your Adaptive Server frequently grant or revoke permissions, avoid setting permission cache entries to a large value. The space used for the permission cache would be wasted, since the cache is flushed with each grant and revoke command.

**plan text pipe active**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

Plan text pipe active determines whether Adaptive Server collects query plan text. If both plan text pipe active and plan text pipe max messages are enabled, Adaptive Server collects the plan text for each query. You can use monSysPlanText to retrieve the query plan text for all user tasks.

**plan text pipe max messages**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>
plan text pipe max messages determines the number of query plan text messages Adaptive Server stores per engine. The total number of messages in the monSQLText table will be the value of sql text pipe max messages times the number of engines running.

print deadlock information

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Valid values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>

print deadlock information enables the printing of deadlock information to the error log.

If you are experiencing recurring deadlocks, setting print deadlock information to 1 provides you with information that can be useful in tracing the cause of the deadlocks. However, setting print deadlock information to 1 can seriously degrade Adaptive Server performance. For this reason, you should use it only when you are trying to determine the cause of deadlocks.

Use sp_sysmon output to determine whether deadlocks are occurring in your application. If they are, set print deadlock information to 1 to learn more about why they are occurring. See the Performance and Tuning Series: Monitoring Adaptive Server with sp_sysmon.

A value of 2, which allows you to print a summary of deadlock information to the errorlog. For example:

Deadlock Id 34: Process (Familyid 0, Spid 70) was waiting for a 'exclusive page' lock on page 10858346 of the 'equineline_job' table in database 18 but process (Familyid 0, Spid 88) already held a 'exclusive page' lock on it.
Deadlock Id 34: Process (Familyid 0, Spid 88) was waiting for a 'exclusive page' lock on page 11540986 of the 'equineline_job' table in database 18 but process (Familyid 0, Spid 70) already held a 'update page' lock on it.
Configuration parameters

print recovery information

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Valid values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

The `print recovery information` parameter determines what information Adaptive Server displays on the console during recovery. (Recovery is performed on each database at Adaptive Server start-up and when a database dump is loaded.) The default value is 0, which means that Adaptive Server displays only the database name and a message saying that recovery is in progress. The other value is 1, which means that Adaptive Server displays information about each individual transaction processed during recovery, including whether it was aborted or committed.

procedure cache size

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>

Specifies the size of the procedure cache in 2K pages. Adaptive Server uses the procedure cache while running stored procedures. If the server finds a copy of a procedure already in the cache, it does not need to read it from the disk. Adaptive Server also uses space in the procedure cache to compile queries while creating stored procedures.
Since the optimum value for procedure cache size differs from application to application, resetting it may improve Adaptive Server’s performance. For example, if you run many different procedures or ad hoc queries, your application uses the procedure cache more heavily, so you may want to increase this value.

**Warning!** If procedure cache size is too small, Adaptive Server’s performance is greatly affected.

**If you are upgrading**

If you are upgrading, procedure cache size is set to the size of the original procedure cache at the time of upgrade. procedure cache size is dynamically configurable, subject to the amount of max memory currently configured.

**process wait events**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>

process wait events controls whether Adaptive Server collect statistics for each wait event for every task. You can get wait information for a specific task using monProcessWaits.

**prod-consumer overlap factor**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>
Configuration parameters

'prod-consumer overlap factor' is a configuration parameter that affects optimization. Adaptive Server changes the group by algorithm, and you cannot use set statistics I/O with parallel plans.

1> sp_configure 'prod-consumer overlap factor'
2> go

read committed with lock

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Valid values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

read committed with lock determines whether an Adaptive Server using transaction isolation level 1 (read committed) holds shared locks on rows or pages of data-only-locked tables during select queries. For cursors, the option applies only to cursors declared as read-only. By default, this parameter is turned off to reduce lock contention and blocking. This parameter affects only queries on data-only locked tables.

For transaction isolation level 1, select queries on allpages-locked tables continue to hold locks on the page at the current position. Any updatable cursor on a data-only-locked table also holds locks on the current page or row. See the Performance and Tuning Series: Basics for more information.

recovery interval in minutes

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>
The recovery interval in minutes parameter sets the maximum number of minutes per database that Adaptive Server uses to complete its recovery procedures in case of a system failure. The recovery procedure rolls transactions backward or forward, starting from the transaction that the checkpoint process indicates as the oldest active transaction. The recovery process has more or less work to do, depending on the value of recovery interval in minutes.

Adaptive Server estimates that 6000 rows in the transaction log require 1 minute of recovery time. However, different types of log records can take more or less time to recover. If you set recovery interval in minutes to 3, the checkpoint process writes changed pages to disk only when syslogs contains more than 18,000 rows since the last checkpoint.

**Note**  The recovery interval has no effect on long-running, minimally logged transactions (such as create index) that are active at the time Adaptive Server fails. It may take as much time to reverse these transactions as it took to run them. To avoid lengthy delays, dump each database after index maintenance operations.

Adaptive Server uses the recovery interval in minutes setting and the amount of activity on each database to decide when to checkpoint each database. When Adaptive Server checkpoints a database, it writes all dirty pages (data pages in cache that have been modified) to disk. This may create a brief period of high I/O, called a checkpoint spike. The checkpoint also performs other maintenance tasks, including truncating the transaction log for each database for which the truncate log on chkpt option has been set. About once per minute, the sleeping checkpoint process “wakes up,” checks the truncate log on chkpt setting, and checks the recovery interval to determine if a checkpoint is needed. Figure 5-4 shows the logic used by Adaptive Server during this process.
You may want to change the recovery interval if your application and its use change. For example, you may want to shorten the recovery interval when there is an increase in update activity on Adaptive Server. Shortening the recovery interval causes more frequent checkpoints, with smaller, more frequent checkpoint spikes, and slows the system slightly. On the other hand, setting the recovery interval too high may cause the recovery time to be unacceptably long. The spikes caused by checkpointing can be reduced by reconfiguring the housekeeper free write percent parameter. See “housekeeper free write percent” on page 135 for further information. For more information on the performance implications of recovery interval in minutes, see Chapter 5, “Memory Use and Performance,” in the Performance and Tuning SeriesBasics.

Use sp_sysmon to determine how a particular recovery interval affects the system. See the Performance and Tuning Series: Monitoring Adaptive Server with sp_sysmon.
**remote server pre-read packets**

<table>
<thead>
<tr>
<th><strong>Summary information</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
<td>3</td>
</tr>
<tr>
<td>Range of values</td>
<td>3–255</td>
</tr>
<tr>
<td>Status</td>
<td>Static</td>
</tr>
<tr>
<td>Display level</td>
<td>Intermediate</td>
</tr>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration groups</td>
<td>Memory Use, Network Communication</td>
</tr>
</tbody>
</table>

remote server pre-read packets determines the number of packets that are “pre-read” by a site handler during connections with remote servers.

All communication between two servers is managed through a single site handler, to reduce the required number of connections. The site handler can pre-read and keep track of data packets for each user process before the receiving process is ready to accept them.

The default value for remote server pre-read packets is appropriate for most servers. Increasing the value uses more memory; decreasing the value can slow network traffic between servers. For more information, see Chapter 15, “Managing Remote Servers.”

**row lock promotion HWM**

<table>
<thead>
<tr>
<th><strong>Summary information</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
<td>200</td>
</tr>
<tr>
<td>Range of values</td>
<td>2–2147483647</td>
</tr>
<tr>
<td>Status</td>
<td>Dynamic</td>
</tr>
<tr>
<td>Display level</td>
<td>Intermediate</td>
</tr>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration groups</td>
<td>Lock Manager, SQL Server Administration</td>
</tr>
</tbody>
</table>

row lock promotion HWM (high-water mark), together with row lock promotion LWM (low-water mark) and row lock promotion PCT specifies the number of row locks permitted during a single scan session of a table or an index before Adaptive Server attempts to escalate from row locks to a table lock.
row lock promotion HWM sets a maximum number of row locks allowed on a table before Adaptive Server attempts to escalate to a table lock. When the number of locks acquired during a scan session exceeds row lock promotion HWM, Adaptive Server attempts to acquire a table lock. The lock promotion HWM value cannot be higher than the number of locks value.

For more information on scan sessions and setting up lock promotion limits, see Chapter 2, “Locking Configuration and Tuning,” in Performance and Tuning Series: Locking and Concurrency Control.

The default value for row lock promotion HWM is appropriate for most applications. You may want to raise the value to avoid table locking. For example, if you know that there are regular updates to 500 rows on a table that has thousands of rows, you can increase concurrency for the tables by setting row lock promotion HWM to around 500.

You can also configure row lock promotion at the per-object level. See sp_setpglockpromote in the Reference Manual.

**Summary information**

<table>
<thead>
<tr>
<th>Default value</th>
<th>200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range of values</td>
<td>2–value of row lock promotion HWM</td>
</tr>
<tr>
<td>Status</td>
<td>Dynamic</td>
</tr>
<tr>
<td>Display level</td>
<td>Intermediate</td>
</tr>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration groups</td>
<td>Lock Manager, SQL Server Administration</td>
</tr>
</tbody>
</table>

row lock promotion LWM (low-water mark), together with the row lock promotion HWM (high-water mark) and row lock promotion PCT specifies the number of row locks permitted during a single scan session of a table or an index before Adaptive Server attempts to promote from row locks to a table lock.

row lock promotion LWM sets the number of locks below which Adaptive Server does not attempt to acquire a table lock on the object. The row lock promotion LWM must be less than or equal to row lock promotion HWM.

For more detailed information on scan sessions and setting up lock promotion limits, see Chapter 2, “Locking Configuration and Tuning,” in Performance and Tuning Series: Locking and Concurrency Control.
The default value for row lock promotion LWM is sufficient for most applications. If Adaptive Server runs out of locks (except for an isolated incident), increase number of locks. See the Performance and Tuning Series: Locking and Concurrency Control.

You can also configure lock promotion at the per-object level. See sp_setpglockpromote in the Reference Manual.

**row lock promotion PCT**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>

If the number of locks held on an object is between row lock promotion LWM (low-water mark) and row lock promotion HWM (high-water mark), row lock promotion PCT sets the percentage of row locks (based on the number of rows in the table) above which Adaptive Server attempts to acquire a table lock.

For more information on setting up lock promotion limits, see Chapter 2, “Locking Configuration and Tuning,” in Performance and Tuning Series: Locking and Concurrency Control.

The default value for row lock promotion PCT is appropriate for most applications.

You can also configure row lock promotion at the per-object level. See sp_sterowlockpromote in the Reference Manual.

**rtm thread idle wait period**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
</tbody>
</table>
Configuration parameters

rtm thread idle wait period defines the time a native thread used by Adaptive Server waits when it has no work to do. When the time set for a native thread is reached, the thread automatically fades out.

Runnable process search count

Runnable process search count specifies the number of times an engine loops while looking for a runnable task before relinquishing the CPU to the operating system.

Adaptive Server engines check the run queue for runnable tasks whenever a task completes or exceeds its allotted time on the engine. At times, there are not any tasks in the run queues. An engine can either relinquish the CPU to the operating system or continue to check for a task to run. Setting runnable process search count higher causes the engine to loop more times, thus holding the CPU for a longer time. Setting the runnable process search count lower causes the engine to release the CPU sooner.

If your machine is a uniprocessor that depends on helper threads to perform I/O, you may see some performance benefit from setting runnable process search to perform network I/O, disk I/O, or other operating system tasks. If a client, such as a bulk copy operation, is running on the same machine as a single CPU server that uses helper threads, it can be especially important to allow both the server and the client access to the CPU.

Note If you are having performance problems, try setting runnable process search count to 3.
For Adaptive Servers running on uniprocessor machines that do not use helper threads, and for multiprocessor machines, the default value provides good performance.

Use `sp_sysmon` to determine how the `runnable process search count` parameter affects the Adaptive Server use of CPU cycles, engine yields to the operating system, and blocking network checks. See the Performance and Tuning Series: Monitoring Adaptive Server with `sp_sysmon` for information.

**sampling percent**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

`sampling percent` is the numeric value of the sampling percentage, such as 05 for 5%, 10 for 10%, and so on. The sampling integer is between zero (0) and one hundred (100).

To reduce I/O contention and resources, run `update statistics` using a sampling method, which can reduce the I/O and time when your maintenance window is small and the data set is large. If you are updating a large data set or table that is in constant use, being truncated and repopulated, you may want to perform a statistical sampling to reduce the time and the size of the I/O.

You must use caution with sampling since the results are not fully accurate. Balance changes to histogram values against the savings in I/O.

Although a sampling of the data set may not be completely accurate, usually the histograms and density values are reasonable within an acceptable range.

When you are deciding whether or not to use sampling, consider the size of the data set, the time constraints you are working with, and if the histogram produced is as accurate as needed.

The percentage to use when sampling depends on your needs. Test various percentages until you receive a result that reflects the most accurate information on a particular data set.

Statistics are stored in the system tables `systabstats` and `sysstatistics`. 
secure default login

Summary information

<table>
<thead>
<tr>
<th>Summary information</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
<td>0</td>
</tr>
<tr>
<td>Range of values</td>
<td>0 (followed by another parameter naming the default login)</td>
</tr>
<tr>
<td>Status</td>
<td>Dynamic</td>
</tr>
<tr>
<td>Display level</td>
<td>Intermediate</td>
</tr>
<tr>
<td>Required role</td>
<td>System security officer</td>
</tr>
<tr>
<td>Configuration group</td>
<td>Security Related</td>
</tr>
</tbody>
</table>

secure default login specifies a default login for all users who are preauthenticated but who do not have a login in master..syslogins.

Establish the secure default login with:

```
sp_configure "secure default login", 0, default_login_name
```

where:

- `secure default login` – is the name of the parameter.
- `0` – is a required parameter because the second parameter of `sp_configure` must be a numeric value.
- `default_login_name` – is the name of the default login for a user who is unknown to Adaptive Server, but who has already been authenticated by a security mechanism. The login name must be a valid login in master..syslogins.

For example, to specify “dlogin” as the secure default login, type:

```
sp_configure "secure default login", 0, dlogin
```

select on syscomments.text column

Summary information

<table>
<thead>
<tr>
<th>Summary information</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
<td>1</td>
</tr>
<tr>
<td>Range of values</td>
<td>0–1</td>
</tr>
<tr>
<td>Status</td>
<td>Dynamic</td>
</tr>
<tr>
<td>Display level</td>
<td>Comprehensive</td>
</tr>
<tr>
<td>Required role</td>
<td>System security officer</td>
</tr>
<tr>
<td>Configuration group</td>
<td>Security Related</td>
</tr>
</tbody>
</table>
This parameter enables protection of the text of database objects through restriction of the `select` permission on the `text` column of the `syscomments` table. The default value of 1 allows `select` permission to “public.” Set the option to 0 to restrict `select` permission to the object owner and the system administrator.

### `send doneinproc tokens`

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
</tbody>
</table>

'send doneinproc tokens' is a persistent tuning option for 'doneinproc'. This configuration option replaces earlier dbcc tune option 'doneinproc' and the trace flag 292. Any queries currently running will immediately take note of any change in the option.

```
1> sp_config 'send doneinproc tokens'
2> go
```

Using the value of 1 is safe in most cases, however care should be taken for cases where sprocs are executed using asynchronous commands from ct-lib. Using the value of 0 may cause state-machine errors in some ct-lib applications.

**Note** Changing the default configuration for this option must be done with care since the ‘doneinproc’ tokens are crucial to the ct-lib and db-lib based client behavior.

### `session tempdb log cache size`

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
</tbody>
</table>
Configuration parameters

**Summary information**

<table>
<thead>
<tr>
<th>Display level</th>
<th>Comprehensive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration group</td>
<td>User Environment</td>
</tr>
</tbody>
</table>

`session tempdb log cache size` configures the size of the ULC, helping to determine how often it needs flushing.

**shared memory starting address**

**Summary information**

<table>
<thead>
<tr>
<th>Default value</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range of values</td>
<td>Platform-specific</td>
</tr>
<tr>
<td>Status</td>
<td>Static</td>
</tr>
<tr>
<td>Display level</td>
<td>Comprehensive</td>
</tr>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration group</td>
<td>Physical Memory</td>
</tr>
</tbody>
</table>

`shared memory starting address` determines the virtual address where Adaptive Server starts its shared memory region.

It is unlikely that you will ever have to reconfigure `shared memory starting address`. You should do so only after consulting with Sybase Technical Support.

`number of worker processes`, `max parallel degree`, and `max scan parallel degree` control parallel query processing at the server level. Using the `parallel_degree`, `process_limit_action`, and `scan_parallel_degree` options to the `set` command can limit parallel optimization at the session level, and using the `parallel` keyword of the `select` command can limit parallel optimization of specific queries.

**size of auto identity column**

**Summary information**

<table>
<thead>
<tr>
<th>Default value</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range of values</td>
<td>1–38</td>
</tr>
<tr>
<td>Status</td>
<td>Dynamic</td>
</tr>
<tr>
<td>Display level</td>
<td>Intermediate</td>
</tr>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration group</td>
<td>SQL Server Administration</td>
</tr>
</tbody>
</table>
size of auto identity column sets the precision of IDENTITY columns that are automatically created with the `sp_dboption auto identity` and `unique auto_identity` index options.

The maximum value that can be inserted into an IDENTITY column is $10^{\text{precision} - 1}$. After an IDENTITY column reaches its maximum value, all further `insert` statements return an error that aborts the current transaction.

If you reach the maximum value of an IDENTITY column, you can increase it with a modify operation in the `alter table` command. See Transact-SQL User's Guide for examples.

You can also use the `create table` command to create a table that is identical to the old one, but with a larger precision for the IDENTITY column. After you have created the new table, use the `insert` command or `bcp` to copy data from the old table to the new one.

**size of global fixed heap**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default values</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Minimum values</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>

The `size of global fixed heap` parameter specifies the memory space for internal data structures and other needs.

If you change the size of the global fixed heap, you must also change the total logical memory by the same amount.

**size of process object heap**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default values</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
Configuration parameters

**Summary information**

Minimum values
- 45 pages (32-bit version)
- 90 pages (64-bit version)

Status: Dynamic

Display level: Basic

Required role: System administrator

Configuration groups: Java Services, Memory Use

The size of process object fixed heap parameter specifies the total memory space for all processes using the Java VM.

If you change the size of process object fixed heap, you must change the total logical memory by that amount.

**size of shared class heap**

**Summary information**

Default values
- 1536 pages (32-bit version)
- 3072 pages (64-bit version)

Minimum values
- 650 pages (32-bit version)
- 1300 pages (64-bit version)

Status: Dynamic

Display level: Basic

Required role: System administrator

Configuration groups: Java Services, Memory Use

The size of shared class heap parameter specifies the shared memory space for all Java classes called into the Java VM. Adaptive Server maintains the shared class heap server-wide for both user-defined and system-provided Java classes.

If you change the size of shared class heap, you must change the total logical memory by the same amount.

**size of unilib cache**

**Summary information**

Default value: 0

Range of values: 0–2147483647

Status: Dynamic
### CHAPTER 5  Setting Configuration Parameters

**Summary information**

<table>
<thead>
<tr>
<th>Display level</th>
<th>Comprehensive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration groups</td>
<td>Memory Use, Unicode</td>
</tr>
</tbody>
</table>

Determines the size of the Unilib cache. \textit{size of unilib cache} specifies the memory used in bytes rounded up to the nearest 1K in addition to the minimum overhead size, which provides enough memory to load a single copy of the largest Unilib conversion table plus the largest Unilib sort table. Asian clients may need to increase \textit{size of unilib cache} by an extra 100K for every additional character set they want to support via Unicode-based conversion.

**sproc optimize timeout limit**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

\textit{sproc optimize timeout limit} specifies the amount of time Adaptive Server can spend optimizing a stored procedure as a fraction of the estimated execution time.

**SQL batch capture**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

\textit{SQL batch capture} controls whether Adaptive Server collects SQL text. If both \textit{SQL batch capture} and \textit{max SQL text monitored} are enabled, Adaptive Server collects the SQL text for each batch for each user task.
**Configuration parameters**

**SQL Perfmon Integration** *(Windows only)*

**Summary information**

<table>
<thead>
<tr>
<th>Default value</th>
<th>1 (on)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valid values</td>
<td>0 (off), 1 (on)</td>
</tr>
<tr>
<td>Status</td>
<td>Static</td>
</tr>
<tr>
<td>Display level</td>
<td>Intermediate</td>
</tr>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration group</td>
<td>SQL Server Administration</td>
</tr>
</tbody>
</table>

SQL Perfmon Integration enables and disables the ability to monitor Adaptive Server statistics from the Windows Performance Monitor.

Adaptive Server must be registered as an Windows Service to support monitor integration. This occurs automatically when:

- You start Adaptive Server using the Services Manager in the Sybase for the Windows program group.
- You use the Services option in the Control Panel.
- You have configured Windows to start Adaptive Server as an automatic service.

See *Configuring Adaptive Server for Windows* for a list of the Adaptive Server counters you can monitor.

**sql server clock tick length**

**Summary information**

<table>
<thead>
<tr>
<th>Default value</th>
<th>Platform-specific</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range of values</td>
<td>Platform-specific minimum–1000000, in multiples of default value</td>
</tr>
<tr>
<td>Status</td>
<td>Static</td>
</tr>
<tr>
<td>Display level</td>
<td>Comprehensive</td>
</tr>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration group</td>
<td>SQL Server Administration</td>
</tr>
</tbody>
</table>

*sql server clock tick length* specifies the duration of the server’s clock tick, in microseconds. Both the default value and the minimum value are platform-specific. Adaptive Server rounds values up to an even multiple of \( n \), where \( n \) is the platform-specific clock-tick default value. You can find the current values for *sql server clock tick length* by using `sp_helpconfig` or `sp_configure`. 
In mixed-use applications with some CPU-bound tasks, decreasing the value of `sql server clock tick length` helps I/O-bound tasks. A value of 20,000 is reasonable for this. Shortening the clock tick length means that CPU-bound tasks exceed the allotted time on the engine more frequently per unit of time, which allows other tasks greater access to the CPU. This may also marginally increase response times, because Adaptive Server runs its service tasks once per clock tick. Decreasing the clock tick length means that the service tasks are run more frequently per unit of time.

Increasing `sql server clock tick length` favors CPU-bound tasks, because they execute longer between context switches. The maximum value of 1,000,000 may be appropriate for primarily CPU-bound applications. However, any I/O-bound tasks may suffer as a result. This can be mitigated somewhat by tuning `cpu grace time` (see “cpu grace time” on page 97) and `time slice` (see “time slice” on page 240).

**Note** Changing the value of `sql server clock tick length` can have serious effects on Adaptive Server performance. Consult with Sybase Technical Support before resetting this value.

---

### `sql text pipe active`

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

`sql text pipe active` controls whether Adaptive Server collects SQL text. If this option is enabled and `sql text pipe max messages` is set, Adaptive Server collects the SQL text for each query. You can use `monSysSQLText` to retrieve the SQL text for all user tasks.

### `sql text pipe max messages`

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
</tbody>
</table>
**Configuration parameters**

### sql text pipe max messages

Specifies the number of SQL text messages Adaptive Server stores per engine. The total number of messages in the monSQLText table will be the value of sql text pipe max messages times the number of engines running.

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>

### stack guard size

Sets the size (in bytes) of the stack guard area. The stack guard area is an overflow stack of configurable size at the end of each stack. Adaptive Server allocates one stack for each user connection and worker process when it starts. These stacks are located contiguously in the same area of memory, with a guard area at the end of each stack. At the end of each stack guard area is a guardword, which is a 4-byte structure with a known pattern. Figure 5-5 illustrates how a process can corrupt a stack guardword.

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>
Adaptive Server periodically checks to see whether the stack pointer for a user connection has entered the stack guard area associated with that user connection’s stack. If it has, Adaptive Server aborts the transaction, returns control to the application that generated the transaction, and generates Error 3626:

The transaction was aborted because it used too much stack space. Either use sp_configure to increase the stack size, or break the query into smaller pieces. spid: %d, suid: %d, hostname: %.*s, application name: %.*s
Adaptive Server also periodically checks the guardword pattern to see if it has changed, thus indicating that a process has overflowed the stack boundary. When this occurs, Adaptive Server prints these messages to the error log and shuts down:

```
kernel: *** Stack overflow detected: limit: 0x%lx sp: 0x%lx
kernel: *** Stack Guardword corrupted
kernel: *** Stack corrupted, server aborting
```

In the first message, “limit” is the address of the end of the stack guard area, and “sp” is the current value of the stack pointer.

In addition, Adaptive Server periodically checks the stack pointer to see whether it is completely outside both the stack and the stack guard area for the pointer’s process. If it is, Adaptive Server shuts down, even if the guardword is not corrupted. When this happens, Adaptive Server prints the following messages to the error log:

```
kernel: *** Stack overflow detected: limit: 0x%lx sp: 0x%lx
kernel: *** Stack corrupted, server aborting
```

The default value for stack guard size is appropriate for most applications. However, if you experience server shutdown from either stack guardword corruption or stack overflow, increase stack guard size by a 2K increment. Each configured user connection and worker process has a stack guard area; thus, when you increase stack guard size, you use up that amount of memory, multiplied by the number of user connections and worker processes you have configured.

Rather than increasing stack guard size to avoid stack overflow problems, consider increasing stack size (see “stack size” on page 230). The stack guard area is intended as an overflow area, not as an extension to the regular stack.

Adaptive Server allocates stack space for each task by adding the values of the stack size and stack guard size parameters. stack guard size must be configured in multiples of 2K. If the value you specify is not a multiple of 2K, sp_configure verification routines round the value up to the next highest multiple.

---

**stack size**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
</tbody>
</table>
CHAPTER 5  Setting Configuration Parameters

stack size specifies the size (in bytes) of the execution stacks used by each user process on Adaptive Server. To find the stack size values for your platform, use sp_helpconfig or sp_configure. stack size must be configured in multiples of 2K. If the value you specify is not a multiple of 2K, sp_configure verification routines round the value up to the next highest multiple.

An execution stack is an area of Adaptive Server memory where user processes keep track of their process context and store local data.

Certain queries can contribute to the probability of a stack overflow. Examples include queries with extremely long where clauses, long select lists, deeply nested stored procedures, and multiple selects and updates using holdlock. When a stack overflow occurs, Adaptive Server prints an error message and rolls back the transaction. See “stack guard size” on page 228 for more information on stack overflows. See the Troubleshooting and Error Messages Guide for more information on specific error messages.

The two options for remedying stack overflows are to break the large queries into smaller queries and to increase stack size. Changing stack size affects the amount of memory required for each configured user connection and worker process. See “total logical memory” on page 242 for further information.

If you have queries that exceed the size of the execution stack, you may want to rewrite them as a series of smaller queries. This is particularly true if there are only a small number of such queries or if you run them infrequently.

There is no way to determine how much stack space a query requires without actually running the query. Stack space for each user connection and worker process is preallocated at start-up.

Therefore, determining the appropriate value for stack size is an empirical process. Test your largest and most complex queries using the default value for stack size. If they run without generating error messages, the default is probably sufficient. If they generate error messages, begin by increasing stack size by a small amount (2K). Re-run your queries and see if the amount you have added is sufficient. If it is not, continue to increase stack size until queries run without generating error messages.

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

System Administration Guide: Volume 1 231
If you are using CIS, or if Java is enabled in the database and you want to use methods that call JDBC, Sybase recommends that you increase the default by 50 percent. If you are not using JDBC or CIS, the standard default value is usually sufficient.

**start mail session (Windows only)**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Valid values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

The `start mail session` parameter enables and disables the automatic initiation of an Adaptive Server mail session when you start Adaptive Server. This feature is available only on Windows servers.

A value of 1 configures Adaptive Server to start a mail session the next time Adaptive Server is started. A value of 0 configures Adaptive Server not to start a mail session at the next restart.

If `start mail session` is 0, you can start an Adaptive Server mail session explicitly, using the `xp_startmail` system ESP.

Before setting the `start mail session` parameter, you must prepare your Windows system by creating a mailbox and mail profile for Adaptive Server. Then, you must create an Adaptive Server account for Sybmail. See the *Configuration Guide for Windows* for information about preparing your system for Sybmail.

**startup delay**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>
startup delay controls when RepAgent is started during the server start. By default, RepAgent starts at the same time as Adaptive Server. Adaptive Server writes a message to the error log stating the wait time.

For example, this configures RepAgent to wait 50 seconds before starting:

```
sp_config_rep_agent pubs2,'startup delay','50'
```

**statement cache size**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Default value</strong></td>
</tr>
<tr>
<td><strong>Valid values</strong></td>
</tr>
<tr>
<td><strong>Status</strong></td>
</tr>
<tr>
<td><strong>Display level</strong></td>
</tr>
<tr>
<td><strong>Required role</strong></td>
</tr>
<tr>
<td><strong>Configuration groups</strong></td>
</tr>
</tbody>
</table>

The statement cache size parameter increases the server allocation of procedure cache memory and limits the amount of memory from the procedure cache pool used for cached statements. The statement cache feature is enabled server-wide:

```
statement cache size size_of_cache
```

**Note** You must configure `set chained on/off` in its own batch if you enable the statement cache.

Because cached statements are transformed into lightweight stored procedures, statement caching requires additional open object descriptors.

**statement pipe active**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Default value</strong></td>
</tr>
<tr>
<td><strong>Range of values</strong></td>
</tr>
<tr>
<td><strong>Status</strong></td>
</tr>
<tr>
<td><strong>Display level</strong></td>
</tr>
<tr>
<td><strong>Required role</strong></td>
</tr>
<tr>
<td><strong>Configuration group</strong></td>
</tr>
</tbody>
</table>
Configuration parameters

**statement pipe active** controls whether Adaptive Server collects statement-level statistics. If both statement pipe active and statement pipe max messages are enabled, Adaptive Server collects the statement statistics for each query. You can retrieve the statistics for all executed statements using monSysStatement.

**statement pipe max messages**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>

statement pipe max messages determines the number of statement statistics messages Adaptive Server stores per engine. The total number of messages in the monSQLText table will be the value of sql text pipe max messages times the number of engines running.

**statement statistics active**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

statement statistic active controls whether Adaptive Server collects the monitoring tables statement-level statistics. You can use monProcessStatement to get statement statistics for a specific task.

**strict dtm enforcement**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
</tbody>
</table>
strict dtm enforcement determines whether or not Adaptive Server transaction coordination services strictly enforce the ACID properties of distributed transactions.

In environments where Adaptive Server should propagate and coordinate transactions only to other Adaptive Servers that support transaction coordination, set strict dtm enforcement to 1 (on). This ensures that transactions are propagated only to servers that can participate in Adaptive Server-coordinated transactions, and transactions complete in a consistent manner. If a transaction attempts to update data in a server that does not support transaction coordination services, Adaptive Server aborts the transaction.

In heterogeneous environments, you may want to make use of servers that do not support transaction coordination. This includes older versions of Adaptive Server and non-Sybase database stores configured using CIS. Under these circumstances, you can set strict dtm enforcement to 0 (off). This allows Adaptive Server to propagate transactions to legacy Adaptive Servers and other data stores, but does not ensure that the remote work of these servers is rolled back or committed with the original transaction.

### suspend audit when device full

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valid values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

suspend audit when device full
**Configuration parameters**

**suspend audit when device full** determines what Adaptive Server does when an audit device becomes completely full.

**Note** If you have two or more audit tables, each on a separate device other than the master device, and you have a threshold procedure for each audit table segment, the audit devices should never become full. Only if a threshold procedure is not functioning properly would the “full” condition occur.

Choose one of these values:

- **0** – truncates the next audit table and starts using it as the current audit table when the current audit table becomes full. If you set the parameter to 0, you ensure that the audit process is never suspended. However, you incur the risk that older audit records are lost if they have not been archived.

- **1** – suspends the audit process and all user processes that cause an auditable event. To resume normal operation, the system security officer must log in and set up an empty table as the current audit table. During this period, the system security officer is exempt from normal auditing. If the system security officer’s actions would generate audit records under normal operation, Adaptive Server sends an error message and information about the event to the error log.

**syb_sendmsg port number**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Valid values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

The **syb_sendmsg port number** parameter specifies the port number that Adaptive Server uses to send messages to a UDP (User Datagram Protocol) port with `sp_sendmsg` or `syb_sendmsg`.

236  Adaptive Server Enterprise
If more than one engine is configured, a port is used for each engine, numbered consecutively from the port number specified. If the port number is set to the default value, 0 Adaptive Server assigns port numbers.

**Note** Sending messages to UDP ports is not supported on Windows.

A system security officer must set the `allow sendmsg` configuration parameter to 1 to enable sending messages to UDP ports. To enable UDP messaging, a system administrator must set `allow sendmsg` to 1. See “allow sendmsg” on page 86. For more information on UDP messaging, see `sp_sendmsg` in the Reference Manual.

### `sysstatistics flush interval`

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Default value</strong></td>
</tr>
<tr>
<td><strong>Valid values</strong></td>
</tr>
<tr>
<td><strong>Status</strong></td>
</tr>
<tr>
<td><strong>Display level</strong></td>
</tr>
<tr>
<td><strong>Required role</strong></td>
</tr>
<tr>
<td><strong>Configuration group</strong></td>
</tr>
</tbody>
</table>

The `sysstatistics flush interval` parameter determines the length of the interval (in minutes) between flushes of `sysstatistics`.

Adaptive Server dynamically maintains the statistics for the number of rows and columns modified in a table as part of any DML statement and flushes them according to the value of `sysstatistics flush interval`.

Adaptive Server uses these statistics for query optimization since they are more accurate. The `datachange` function determines the amount of data that is changed at the table, column, or partition level since the last `update statistics`, and initiates updating statistics on the object.

The in-memory statistics are always flushed to disk during a polite shutdown of the server. You can configure `sysstatistics flush interval` to flush these in-memory statistics to disk by the housekeeper task at regular intervals. Set `sysstatistics flush interval` to 0 to disable this housekeeper task.
**systemwide password expiration**

<table>
<thead>
<tr>
<th><strong>Summary information</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
<td>0</td>
</tr>
<tr>
<td>Range of values</td>
<td>0–32767</td>
</tr>
<tr>
<td>Status</td>
<td>Dynamic</td>
</tr>
<tr>
<td>Display level</td>
<td>Intermediate</td>
</tr>
<tr>
<td>Required role</td>
<td>System security officer</td>
</tr>
<tr>
<td>Configuration group</td>
<td>Security Related</td>
</tr>
</tbody>
</table>

Systemwide password expiration, which can be set only by a system security officer, sets the number of days that passwords remain in effect after they are changed. If systemwide password expiration is set to 0, passwords do not expire.

Systemwide password expiration can be any value between 0 and 32767, inclusive. The password expires when the number of specified days passes. For example, if you create a new login on August 1, 2007 at 10:30 AM, with a password expiration interval of 30 days, the password expires on August 31, 2007 at 10:30 AM.

An account’s password is considered expired if an interval greater than number_of_days has passed since the last time the password for that account was changed.

When the number of days remaining before expiration is less than 25 percent of the value of systemwide password expiration or 7 days, whichever is greater, each time the user logs in, a message displays, giving the number of days remaining before expiration. Users can change their passwords anytime before expiration.

When an account’s password has expired, the user can still log in to Adaptive Server but cannot execute any commands until he or she has used sp_password to change his or her password. If the system security officer changes the user’s password while the account is in sp_password-only mode, the account returns to normal after the new password is assigned.

This restriction applies only to login sessions established after the password has expired. Users who are logged in at the time their passwords expire are not affected until the next time they log in.
**tape retention in days**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

The *tape retention in days* parameter specifies the number of days you intend to retain each tape after it has been used for either a database or a transaction log dump. This parameter can keep you from accidentally overwriting a dump tape.

For example, if you have set *tape retention in days* to 7 days, and you attempt to use the tape before 7 days have elapsed since the last time you dumped to that tape, Backup Server issues a warning message.

You can override the warning using the `with init` option when executing the `dump` command. Doing this causes the tape to be overwritten and all data on the tape to be lost.

Both the `dump database` and `dump transaction` commands provide a *retaindays* option, which overrides the *tape retention in days* value for a particular dump. See “Protecting dump files from being overwritten” on page 388 in *System Administration Guide: Volume 2* for more information.

**tcp no delay**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Valid values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>

The *tcp no delay* parameter controls TCP (Transmission Control Protocol) packet batching. The default value is 1, which means that TCP packets are not batched.
TCP normally batches small logical packets into single larger physical packets (by briefly delaying packets) fill physical network frames with as much data as possible. This is intended to improve network throughput in terminal emulation environments where there are mostly keystrokes being sent across the network.

However, applications that use small TDS (Tabular Data Stream) packets may benefit from disabling TCP packet batching. To disable TCP packet batching, set `tcp no delay` to 1.

**Note** Disabling TCP packet batching means that packets are sent, regardless of size; this increases the volume of network traffic.

### text prefetch size

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Valid values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
</tbody>
</table>

The `text prefetch size` parameter limits the number of pages of text, unitext, and image data that can be prefetched into an existing buffer pool. Adaptive Server prefetches only text, unitext, and image data that was created with Adaptive Server 12.x or was upgraded using `dbcc rebuild_text`.

### time slice

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>
time slice sets the number of milliseconds that the Adaptive Server scheduler allows a task to run. If time slice is set too low, Adaptive Server may spend too much time switching between tasks, which increases response time. If it is set too high, CPU-intensive tasks may monopolize engines, which also increases response time. The default value, 100 milliseconds, allows each task to run for 1/10 of a second before relinquishing the CPU to another task.


Use sp_sysmon to determine how time slice affects voluntary yields by Adaptive Server engines. See the Performance and Tuning Series: Monitoring Adaptive Server with sp_sysmon.

**total data cache size**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>

The total data cache size parameter reports the amount of memory, in kilobytes, that is currently available for data, index, and log pages. This parameter is a calculated value that is not directly user-configurable.

The amount of memory available for the data cache can be affected by a number of factors, including:

- The amount of physical memory available on your machine
- The values to which the following parameters are set:
  - total logical memory
  - number of user connections
  - total procedure cache percent
  - number of open databases
  - number of open objects
  - number of open indexes
Configuration parameters

- number of devices

A number of other parameters also affect the amount of available memory, but to a lesser extent.

For information on how Adaptive Server allocates memory and for information on data caches, see “Configuration parameters” on page 79.

total logical memory

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>

total logical memory displays the total logical memory for the current configuration of Adaptive Server. The total logical memory is the amount of memory that Adaptive Server’s current configuration uses. total logical memory displays the memory that is required to be available, but which may or may not be in use at any given moment. For information about the amount of memory in use at a given moment, see the configuration parameter total physical memory. You cannot use total logical memory to set any of the memory configuration parameters.

total physical memory

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>
total physical memory is a read-only configuration parameter that displays the total physical memory for the current configuration of Adaptive Server. The total physical memory is the amount of memory that Adaptive Server is using at a given moment in time. Configure Adaptive Server so that the value for max memory is larger than the value for total logical memory, and the value for total logical memory is larger than the value for total physical memory.

**txn to pss ratio**

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Valid values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration groups</td>
</tr>
</tbody>
</table>

Adaptive Server manages transactions as configurable server resources. Each time a new transaction begins, Adaptive Server must obtain a free transaction descriptor from a global pool that is created when the server is started. Transaction descriptors are internal memory structures that Adaptive Server uses to represent active transactions.

Adaptive Server requires one free transaction descriptor for:

- The outer block of each server transaction. The outer block of a transaction may be created explicitly when a client executes a new `begin transaction` command. Adaptive Server may also implicitly create an outer transaction block when clients use Transact-SQL to modify data without using `begin transaction` to define the transaction.

  **Note** Subsequent, nested transaction blocks, created with additional `begin transaction` commands, do not require additional transaction descriptors.

- Each database accessed in a multi-database transaction. Adaptive Server must obtain a new transaction descriptor each time a transaction uses or modifies data in a new database.

`txn to pss ratio` determines the total number of transaction descriptors available to the server. At start-up, this ratio is multiplied by the number of PSS structures to create the transaction descriptor pool:

\[
\text{# of transaction descriptors} = \text{PSS structures} \times \text{txn to pss ratio}
\]
Configuration parameters

The default value, 16, ensures compatibility with earlier versions of Adaptive Server. Prior to version 12.x, Adaptive Server allocated 16 transaction descriptors for each user connection. In version 12.x and later, the number of simultaneous transactions is limited only by the number of transaction descriptors available in the server.

Note You can have as many databases in a user transaction as there are in your Adaptive Server installation. For example, if your Adaptive Server has 25 databases, you can include 25 databases in your user transactions.

Optimizing the txn to pss ratio for your system

During a peak period, use `sp_monitorconfig` to examine the use of transaction descriptors:

```
sp_monitorconfig "txn to pss ratio"
```

<table>
<thead>
<tr>
<th>Name</th>
<th>num_free</th>
<th>num_active</th>
<th>pct_act</th>
<th>Max_Used</th>
<th>Reused</th>
</tr>
</thead>
<tbody>
<tr>
<td>txn to pss ratio</td>
<td>784</td>
<td>80</td>
<td>10.20</td>
<td>523</td>
<td>NA</td>
</tr>
</tbody>
</table>

If the `num_used` value is zero or very low, transactions may be delayed as Adaptive Server waits for transaction descriptors to become free in the server. In this case, consider increasing the value of `txn to pss ratio`.

If the `Max_used` value is too low, unused transaction descriptors may be consuming memory that can be used by other server functions. Consider reducing the value of `txn to pss ratio`.

unified login required

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Range of values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>
unified login required requires that all users who log in to Adaptive Server be authenticated by a security mechanism. The use security services parameter must be 1 to use the unified login security service.

**upgrade version**

**Summary information**

- Default value: 1100
- Range of values: 0–2147483647
- Status: Dynamic
- Display level: Comprehensive
- Required role: System administrator
- Configuration group: SQL Server Administration

**upgrade version** reports the version of the upgrade utility that upgraded your master device. The upgrade utility checks and modifies this parameter during an upgrade.

**Warning!** Although this parameter is configurable, do not reset it. Doing so may cause serious problems with Adaptive Server.

You can determine whether an upgrade has been done on your master device by using **upgrade version** without specifying a value:

```
sp_configure "upgrade version"
```

**use security services**

**Summary information**

- Default value: 0 (off)
- Range of values: 0 (off), 1 (on)
- Status: Static
- Display level: Intermediate
- Required role: System security officer
- Configuration group: Security Related

**use security services** specifies that Adaptive Server uses network-based security services. If the parameter is set to 0, none of the network-based security services can be used.
**user log cache size**

**Summary information**

<table>
<thead>
<tr>
<th></th>
<th>Logical page size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
<td>Logical page size</td>
</tr>
<tr>
<td>Range of values</td>
<td>2048^a –2147483647</td>
</tr>
<tr>
<td>a. Minimum determined by server’s logical page size</td>
<td></td>
</tr>
<tr>
<td>Status</td>
<td>Static</td>
</tr>
<tr>
<td>Display level</td>
<td>Intermediate</td>
</tr>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration groups</td>
<td>Memory Use, User Environment</td>
</tr>
</tbody>
</table>

`user log cache size` specifies the size (in bytes) for each user’s log cache. Its size is determined by the server’s logical page size. There is one user log cache for each configured user connection and worker process. Adaptive Server uses these caches to buffer the user transaction log records, which reduces the contention at the end of the transaction log.

When a user log cache becomes full or another event occurs (such as when the transaction completes), Adaptive Server “flushes” all log records from the user log cache to the database transaction log. By first consolidating the log records in each user’s log cache, rather than immediately adding each record to the database’s transaction log, Adaptive Server reduces contention of processes writing to the log, especially for SMP systems configured with more than one engine.

**Note**  For transactions using a database with mixed data and log segments, the user log cache is flushed to the transaction log after each log record. No buffering takes place. If your databases do not have dedicated log segments, do not increase the user log cache size.

Do not configure `user log cache size` to be larger than the maximum amount of log information written by an application’s transaction. Since Adaptive Server flushes the user log cache when the transaction completes, any additional memory allocated to the user log cache is wasted. If no transaction in your server generates more than 4000 bytes of transaction log records, set `user log cache size` no higher than that value. For example:

```
sp_configure "user log cache size", 4000
```

Setting `user log cache size` too high wastes memory. Setting it too low can cause the user log cache to fill up and flush more than once per transaction, increasing the contention for the transaction log. If the volume of transactions is low, the amount of contention for the transaction log may not be significant.
Use `sp_sysmon` to understand how this parameter affects cache behavior. See the *Performance and Tuning Series: Monitoring Adaptive Server with sp_sysmon*.

**user log cache spinlock ratio**

<table>
<thead>
<tr>
<th>Summary information</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
<td>20</td>
</tr>
<tr>
<td>Range of values</td>
<td>1–2147483647</td>
</tr>
<tr>
<td>Status</td>
<td>Dynamic</td>
</tr>
<tr>
<td>Display level</td>
<td>Intermediate</td>
</tr>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration groups</td>
<td>Memory Use, User Environment</td>
</tr>
</tbody>
</table>

For Adaptive Servers running with multiple engines, the `user log cache spinlock ratio` parameter specifies the ratio of user log caches per user log cache spinlock. There is one user log cache for each configured user connection.

The default value for this parameter is 20, or one spinlock for each 20 user connections configured for your server.

Use `sp_sysmon` to understand how this parameter affects cache behavior. See the *Performance and Tuning Series: Monitoring Adaptive Server with sp_sysmon*.

**wait event timing**

<table>
<thead>
<tr>
<th>Summary information</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
<td>0</td>
</tr>
<tr>
<td>Range of values</td>
<td>0–1</td>
</tr>
<tr>
<td>Status</td>
<td>Dynamic</td>
</tr>
<tr>
<td>Display level</td>
<td>Comprehensive</td>
</tr>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration groups</td>
<td>Memory Use, Monitoring</td>
</tr>
</tbody>
</table>

`wait event timing` controls whether Adaptive Server collects statistics for individual wait events. A task may have to wait for a variety of reasons (for example, waiting for a buffer read to complete). The `monSysWaits` table contains the statistics for each wait event. The `monWaitEventInfo` table contains a complete list of wait events.
**xact coordination interval**

**Summary information**

<table>
<thead>
<tr>
<th>Summary information</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
<td>60 (seconds)</td>
</tr>
<tr>
<td>Valid values</td>
<td>1 – 2147483647 (seconds)</td>
</tr>
<tr>
<td>Status</td>
<td>Dynamic</td>
</tr>
<tr>
<td>Display level</td>
<td>10</td>
</tr>
<tr>
<td>Required role</td>
<td>System administrator</td>
</tr>
<tr>
<td>Configuration group</td>
<td>DTM Administration</td>
</tr>
</tbody>
</table>

*xact coordination interval* defines the length of time between attempts to resolve transaction branches that were propagated to remote servers.

The coordinating Adaptive Server makes regular attempts to resolve the work of remote servers participating in a distributed transaction. The coordinating server contacts each remote server participating in the distributed transaction in a serial manner, as shown in Figure 5-6. The coordination service may be unable to resolve a transaction branch for a variety of reasons. For example, if the remote server is not reachable due to network problems, the coordinating server reattempts the connection after the time specified by *xact coordination level*.

*Figure 5-6: Resolving remote transaction branches*

With the default value of *xact coordination interval*, 60, Adaptive Server attempts to resolve remote transactions once every minute. Decreasing the value may speed the completion of distributed transactions, but only if the transactions are themselves resolved in less than a minute. Under normal circumstances, there is no performance penalty to decreasing the value of *xact coordination interval*.
Setting `xact coordination interval` to a higher number can slow the completion of distributed transactions, and cause transaction branches to hold resources longer than they normally would. Under normal circumstances, do not increase the value of `xact coordination interval` beyond its default.

### `xp_cmdshell context`

<table>
<thead>
<tr>
<th>Summary information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default value</td>
</tr>
<tr>
<td>Valid values</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Display level</td>
</tr>
<tr>
<td>Required role</td>
</tr>
<tr>
<td>Configuration group</td>
</tr>
</tbody>
</table>

The `xp_cmdshell context` parameter sets the security context for the operating system command to be executed using the `xp_cmdshell` system ESP. The values for the context determines under which account the command runs:

- 0 – command runs under XP Server’s account
- 1 – command runs under user’s account
- 2 - command runs under XP Server’s account only if the user has administrator privileges.

Setting `xp_cmdshell context` to 1 restricts the `xp_cmdshell` security context to users who have accounts at the operating system level. Its behavior is platform-specific. If `xp_cmdshell context` is set to 1, to use an `xp_cmdshell` ESP, an operating system user account must exist for the Adaptive Server user name. For example, an Adaptive Server user named “sa” cannot use `xp_cmdshell` unless he or she has an operating-system-level user account named “sa”.

On Windows, when `xp_cmdshell context` is set to 1, `xp_cmdshell` succeeds only if the user name of the user logging in to Adaptive Server is a valid Windows user name with Windows system administration privileges on the system on which Adaptive Server is running.

On other platforms, when `xp_cmdshell context` is set to 1, `xp_cmdshell` succeeds only if Adaptive Server was started by a user with “superuser” privileges at the operating system level. When Adaptive Server gets a request to execute `xp_cmdshell`, it checks the `uid` of the user name of the ESP requestor and runs the operating system command with the permissions of that `uid`. 
If \texttt{xp\_cmdshell\ context} is 0, the permissions of the operating system account under which Adaptive Server is running are the permissions used to execute an operating system command from \texttt{xp\_cmdshell}. This allows users to execute operating commands that they would not ordinarily be able to execute under the security context of their own operating system accounts.
Overview of Disk Resource Issues

This chapter discusses some basic issues that determine how you allocate and use disk resources with Adaptive Server.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device allocation and object placement</td>
<td>251</td>
</tr>
<tr>
<td>Commands for managing disk resources</td>
<td>252</td>
</tr>
<tr>
<td>Considerations in storage management</td>
<td>254</td>
</tr>
<tr>
<td>Status and defaults at installation time</td>
<td>255</td>
</tr>
<tr>
<td>System tables that manage storage</td>
<td>256</td>
</tr>
</tbody>
</table>

Many Adaptive Server defaults are set to reasonable values for aspects of storage management, such as where databases, tables, and indexes are placed 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 Adaptive Server and the physical placement of databases, tables, and indexes on those resources.

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 Adaptive Server. Device allocation topics are described in the chapters shown in Table 6-1.

<table>
<thead>
<tr>
<th>Task</th>
<th>Chapter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialize and allocate a default pool of database devices</td>
<td>Chapter 7, “Initializing Database Devices”</td>
</tr>
</tbody>
</table>
After the initial disk resources have been allocated to Adaptive Server, 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. Object placement tasks are discussed throughout this manual, including the chapters shown in Table 6-2.

**Table 6-2: Object placement topics**

<table>
<thead>
<tr>
<th>Task</th>
<th>Chapter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Place databases on specific database devices</td>
<td>Chapter 6, “Creating and Managing User Databases”</td>
</tr>
<tr>
<td>Place tables and indexes on specific database devices</td>
<td>Chapter 8, “Creating and Using Segments” in <em>System Administration Guide: Volume 2</em></td>
</tr>
</tbody>
</table>

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, you must first allocate those devices to Adaptive Server. The remaining sections in this chapter provide an overview that spans both device allocation and object placement issues, providing pointers to chapters where appropriate.

**Commands for managing disk resources**

Table 6-3 lists the major commands a system administrator uses to allocate disk resources to Adaptive Server and provides references to the chapters that discuss those commands.

**Table 6-3: Commands for allocating disk resources**

<table>
<thead>
<tr>
<th>Command</th>
<th>Task</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>disk init name = &quot;dev_name&quot; physname = &quot;phys_name&quot;...</code></td>
<td>Makes a physical device available to a particular Adaptive Server. Assigns a database device name (<code>dev_name</code>) that is used to identify the device in other Adaptive Server commands.</td>
<td>Chapter 7, “Initializing Database Devices”</td>
</tr>
<tr>
<td><code>sp_deviceattr logicalname, optname, optvalue</code></td>
<td>Changes the <code>dsync</code> setting of an existing database device file.</td>
<td>Chapter 7, “Initializing Database Devices”</td>
</tr>
<tr>
<td><code>sp_diskdefault &quot;dev_name&quot;...</code></td>
<td>Adds <code>dev_name</code> to the general pool of default database space.</td>
<td>Chapter 7, “Initializing Database Devices”</td>
</tr>
<tr>
<td><code>disk resize name = &quot;device_name&quot;, size = additional_space</code></td>
<td>Dynamically increases the size of database devices.</td>
<td>Chapter 7, “Initializing Database Devices”</td>
</tr>
</tbody>
</table>
Table 6-4 lists the commands used in object placement. For information about how object placement affects performance, see Chapter 1, “Controlling Physical Data Placement,” in the Performance and Tuning Series: Physical Database Tuning.

**Table 6-4: Commands for placing objects on disk resources**

<table>
<thead>
<tr>
<th>Command</th>
<th>Task</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>disk mirror name = &quot;dev_name&quot; mirror = &quot;phys_name&quot;...</td>
<td>Mirrors a database device on a specific physical device.</td>
<td>Chapter 2, “Mirroring Database Devices” in System Administration Guide: Volume 2</td>
</tr>
<tr>
<td>create database...on dev_name or alter database...on dev_name</td>
<td>Makes database devices available to a particular Adaptive Server database. The log on clause to create database places the database’s logs on a particular database device.</td>
<td>Chapter 6, “Creating and Managing User Databases”</td>
</tr>
<tr>
<td>create database... or alter database...</td>
<td>When used without the on dev_name clause, these commands allocate space on the default database devices.</td>
<td>Chapter 6, “Creating and Managing User Databases” in System Administration Guide: Volume 2</td>
</tr>
<tr>
<td>sp_addsegment seg_name, dbname, devname and sp_extendsegment seg_name, dbname, devname</td>
<td>Creates a segment – a named collection of space – from the devices available to a particular database.</td>
<td>Chapter 8, “Creating and Using Segments” in System Administration Guide: Volume 2</td>
</tr>
<tr>
<td>create table...on seg_name or create index...on seg_name</td>
<td>Creates database objects, placing them on a specific segment of the database’s assigned disk space.</td>
<td>Chapter 8, “Creating and Using Segments” in System Administration Guide: Volume 2</td>
</tr>
<tr>
<td>create table... or create index...</td>
<td>When used without on seg_name, tables and indexes occupy the general pool of space allocated to the database (the default devices).</td>
<td>Chapter 8, “Creating and Using Segments” in System Administration Guide: Volume 2</td>
</tr>
</tbody>
</table>
Considerations in storage management decisions

The system administrator must make many decisions regarding the physical allocation of space to Adaptive Server databases. The major considerations in these choices are:

- **Recovery** – disk mirroring and maintaining logs on a separate physical device provide two mechanisms for full recovery in the event of physical disk crashes.
- **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.

**Recovery**

Recovery is the key motivation for using several disk devices. Nonstop recovery can be accomplished by mirroring database devices. Full recovery can also be ensured by storing a database’s log on a separate physical device.

**Keeping 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 crash, 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 Chapter 6, “Creating and Managing User Databases,” in *System Administration Guide: Volume 2* for information about the log on clause of create database.

**Mirroring**

Nonstop recovery in the event of a hard disk crash is guaranteed by mirroring all Adaptive Server devices to a separate physical disk. Chapter 2, “Mirroring Database Devices,” describes the process of mirroring devices.
Performance

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

- Placing a table on one hard disk and nonclustered indexes on another ensures that physical reads and writes are faster, since the work is split between two disk drives.
- Splitting large tables across two disks can improve performance, particularly for multiuser applications.
- When log and data share devices, user log cache buffering of transaction log records is disabled.
- Partitioning provides multiple insertion points for a heap table, adds a degree of parallelism to systems configured to perform parallel query processing, and makes it possible to distribute a table’s I/O across multiple database devices.


Status and defaults at installation time

You can find instructions for installing Adaptive Server in the installation documentation for your platform. 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 Adaptive Server, 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.
System tables that manage storage

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

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. See “Designating default devices” on page 272 for more information.

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

System tables that manage storage

Two system tables in the master database and two more in each user database track the placement of databases, tables (including the transaction log table, `syslogs`), and indexes. The relationship between the tables is illustrated in Figure 6-1.
Figure 6-1: System tables that manage storage

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 Adaptive Server.

The `disk init` command adds entries for database devices to `master.sysdevices`. Dump devices, added using `sp_addumpdevice`, are discussed in Chapter 11, “Developing a Backup and Recovery Plan.”
System tables that manage storage

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 Adaptive Server 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 Chapter 6, “Creating and Managing User Databases,” in System Administration Guide: Volume 2 for information about creating databases on specific database devices.

The sysusages table

The sysusages table in the master database keeps track of all of the space that you assign to all Adaptive Server 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. Chapter 8, “Creating and Using Segments,” in System Administration Guide: Volume 2 discusses these procedures in detail.
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`.

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.

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`.
System tables that manage storage
CHAPTER 7

Initializing Database Devices

This chapter explains how to initialize database devices and how to assign devices to the default pool of devices.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>What are database devices?</td>
<td>261</td>
</tr>
<tr>
<td>Using the disk init command</td>
<td>262</td>
</tr>
<tr>
<td>disk init syntax</td>
<td>262</td>
</tr>
<tr>
<td>Getting information about devices</td>
<td>269</td>
</tr>
<tr>
<td>Dropping devices</td>
<td>271</td>
</tr>
<tr>
<td>Designating default devices</td>
<td>272</td>
</tr>
<tr>
<td>Increasing the size of devices with disk resize</td>
<td>273</td>
</tr>
</tbody>
</table>

What are 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 Adaptive Server 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
Using the **disk init** command

A system administrator initializes new database devices with the **disk init** command, which:

- 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

**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 Adaptive Server. You may want to repartition the disks on your computer to provide maximum performance for your Sybase 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.

**Warning!** After you run the **disk init** command, dump the *master* database. This makes recovery easier and safer in case *master* is damaged. See Chapter 13, “Restoring the System Databases.”

disk init syntax

The syntax of **disk init** is:

```plaintext
disk init
  name = "device_name",
  physname = "physicalname",
  [vdevno = virtual_device_number,]
  size = number_of_blocks
  [, vstart = virtual_address
    , cntrltpe = controller_number ]
  [, contiguous]
```

Using the **disk init** command

A system administrator initializes new database devices with the **disk init** command, which:

- 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

**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 Adaptive Server. You may want to repartition the disks on your computer to provide maximum performance for your Sybase 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.

**Warning!** After you run the **disk init** command, dump the *master* database. This makes recovery easier and safer in case *master* is damaged. See Chapter 13, “Restoring the System Databases.”

disk init syntax

The syntax of **disk init** is:

```plaintext
disk init
  name = "device_name",
  physname = "physicalname",
  [vdevno = virtual_device_number,]
  size = number_of_blocks
  [, vstart = virtual_address
    , cntrltpe = controller_number ]
  [, contiguous]
```
 CHAPTER 7  Initializing Database Devices

[, dsync = {true | false}]
[, directio = {true | false}]

disk init examples

On UNIX:

disk init
  name = "user_disk",
  physname = "/dev/rxy1a",
  size = "64G"

On Windows NT:

disk init
  name = "user_disk",
  physname = "d:\devices\userdisk.dat",
  size = "64G"

Specifying a logical device name with disk init

The device_name must be a valid identifier. This name is used in the create
database and alter database commands, and in the system procedures that
manage segments. The logical device name is known only to Adaptive Server,
not to the operating system on which the server runs.

Specifying a physical device name with disk init

The physname of the database device gives the name of a raw disk partition
(UNIX), foreign device, or the name of an operating system file. On PC
platforms, you typically use operating system file names for physname.

Choosing a device number for disk init

Adaptive Server accepts, but does not require, the disk init vdevno parameter. If
you specify a vdevno, you may choose any currently unused identifier from 1
to 2,147,483,647 (virtual device ID 0 is used by the master device). For
example, specifying vdevno = 33 assigns virtual device ID 33 to a device. If
you do not specify a vdevno, Adaptive Server chooses a number higher than the
highest vdevno currently listed in sysdevices.
The number of database devices you can create is limited by the `number of devices` configuration parameter. Adaptive Server is initially configured for 10 devices. Use `sp_configure` to change this parameter if you need more devices. For more information about `sp_configure`, see Chapter 5, “Setting Configuration Parameters.”

Your operating system may also limit the number of devices your installation can use concurrently. Each Sybase device counts as one open file to the operating system.

Adaptive Server automatically specifies the next available identifying number for the database device. This is the virtual device number (`vdevno`). You need not specify this number when you issue the `disk init` command.

If you choose to select the `vdevno` manually, it must be unique among the devices used by Adaptive Server. Device number 0 represents the master device. Legal numbers are 1 – 2,147,483,647. You can choose any unused `devno` within that range.

To see the numbers already in use for `vdevno`, look in the `vdevno` column of the report from `sp_helpdevice`, or use the following query to list all the device numbers currently in use:

```
select vdevno from master..sysdevices
   where status & 2=2
```

Here, “status 2” specifies physical disk.

### Specifying the device size with `disk init`

You can use the following unit specifiers to indicate the size of the device: ‘k’ or ‘K’ indicates kilobytes, ‘m’ or ‘M’ indicates megabytes, ‘g’ or ‘G’ indicates gigabytes, and ‘t’ or ‘T’ indicates terabytes. Although it is optional, Sybase recommends that you always include the unit specifier in both the `disk init` and `create database` commands to avoid confusion in the actual number of pages allocated. You must enclose the unit specifier in single or double quotes or in brackets.

Theoretically, you can create as many as 2,147,483,647 disk devices, each of which can be as large as 2,147,483,648 2K-blocks. Because the number and size of possible devices are effectively unlimited, the maximum installation size becomes a function of database size, hardware, and operating system limits.

The following guidelines apply to the syntax for `disk init`:
• If you do not include a unit specifier for the size argument of disk init or disk reinit, size is measured, by default, in number of virtual pages. Thus, if you enter size = 15000, Adaptive Server assumes 15,000 virtual pages. A virtual page is 2048 bytes.

• You can increase, but not decrease, the size of an existing database device using the disk resize command.

• If you are planning to use the new device for the creation of a new database, the minimum size depends on the logical page size used by the server, described in Table 7-1:

<table>
<thead>
<tr>
<th>Logical page size</th>
<th>Minimum database size</th>
</tr>
</thead>
<tbody>
<tr>
<td>2K</td>
<td>3 Megabytes</td>
</tr>
<tr>
<td>4K</td>
<td>6 Megabytes</td>
</tr>
<tr>
<td>8K</td>
<td>12 Megabytes</td>
</tr>
<tr>
<td>16K</td>
<td>24 Megabytes</td>
</tr>
</tbody>
</table>

You cannot have a database smaller than the model database. If your model database is larger than the minimums listed above, then this is the minimum database size.

Adaptive Server allocates and manages database space in allocation units, which are groups of 256 logical pages. Because the smallest size create database permits you to specify is one megabyte, the size of the smallest usable database device is the larger of one MB or 256 logical pages (for a 2k or 4k logical page size, this is one megabyte, for an 8k logical page size, this is 2MB, for a 16k logical page size, this is 4MB.

It is helpful to keep this grouping of 256 pages in mind when you decide how large to make a device to avoid wasting space. For example, if your installation uses a 16k logical page size, specifying a device as size = '31M' leaves three megabytes wasted at the end of the device, since an allocation unit would be 4 MB.

If you are initializing a raw device, determine the size of the device from your operating system, as described in the installation documentation for your platform. Use the total size available, up to the maximum for your platform. After you have initialized the disk for use by Adaptive Server, you cannot use any space on that raw device for any other purpose.
disk init syntax

disk init uses size to compute the value for the high virtual page number in sysdevices.high.

---

Note  The numbers in sysdevices.high and sysdevices.low are virtual page numbers with blocks of 2k bytes, which is Adaptive Server’s unit of physical disk management. This may not be the same as your installation’s logical page size.

---

Warning! If the physical device does not contain the number of blocks specified by the size parameter, disk init fails. If you use the optional vstart parameter, the physical device must contain the sum of the blocks specified by both the vstart and size parameters, or the command fails.

---

Specifying the dsync setting with disk init (optional)

For devices initialized on UNIX operating system files, the dsync setting controls whether or not writes to those files are buffered. When the dsync setting is on, Adaptive Server opens a database device file using the UNIX dsync flag. The dsync flag ensures that writes to the device file occur directly on the physical storage media, and Adaptive Server can recover data on the device in the event of a system failure.

When dsync is off (false), writes to the device file may be buffered by the UNIX file system, and the recovery of data on the device cannot be ensured. Turn off dsync only when data integrity is not required, or when the system administrator requires performance and behavior similar to earlier Adaptive Server versions.

---

Note  The dsync setting is ignored for devices initialized on raw partitions. Instead, writes to the database device take place directly to the physical media.

---

Performance implications of dsync

The use of the dsync setting with database device files incurs the following performance trade-offs:

- Adaptive Server does not support asynchronous I/O on operating system files for HP-UX.
CHAPTER 7  Initializing Database Devices

• If database device files on these platforms use the dsync option, the Adaptive Server engine writing to the device file blocks until the write operation completes. This can cause poor performance during update operations.

• When dsync is on (true), write operations to database device files may be slower compared to earlier versions of Adaptive Server (where dsync is not supported). This is because Adaptive Server must write data to disk instead of simply copying cached data to the UNIX file system buffer. In cases where highest write performance is required (but data integrity after a system failure is not required) turning dsync off yields device file performance similar to earlier Adaptive Server versions. For example, you may consider storing tempdb on a dedicated device file with dsync disabled, if performance is not acceptable while using dsync.

• Response time for read operations is generally better for devices stored on UNIX operating system files as compared to devices stored on raw partitions. Data from device files can benefit from the UNIX file system cache as well as the Adaptive Server cache, and more reads may take place without requiring physical disk access.

Limitations and restrictions of dsync

The following limitations and restrictions apply to using the dsync setting:

• dsync is always set to true for the master device file. You cannot change the dsync setting for the master device. If you attempt to turn dsync off for the master device, Adaptive Server displays a warning message.

• If you change a device file’s dsync setting using the sp_deviceattr procedure, you must restart Adaptive Server before the change takes effect.

• When you upgrade from an Adaptive Server earlier than version 12.x, dsync is set to true for the master device file only. Use sp_deviceattr to change the dsync setting for any other device files.

• Adaptive Server ignores the dsync setting for database devices stored on raw partitions. Writes to devices stored on raw partitions are always done directly to the physical media.

• The directio and dsync parameters are mutually exclusive. If a device has dsync set to “true,” you cannot set directio to “true” for this device. To enable directio for a device, you must first reset dsync to “false.”
**Using directio to bypass operating system buffer**

The `directio` parameter for `disk init`, `disk reinit`, and `sp_deviceattr` allows you to configure Adaptive Server to transfer data directly to disk, bypassing the operating system buffer cache. `directio` performs IO in the same manner as raw devices and provides the same performance benefit as raw devices, but has the ease of use and manageability of file system devices. You cannot set the `directio` option for the master device. `directio` is a static parameter that requires a restart of Adaptive Server to take effect.

**Note** `directio` is not available on all platforms. If you issue `disk init` with the `directio` parameter on a platform on which it is not supported, Adaptive Server issues the message `No such parameter: 'directio'`.

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

The `directio` and `dsync` parameters are mutually exclusive. If a device has `dsync` set to “true,” you cannot set `directio` to “true” for this device. To enable `directio` for a device, you must first reset `dsync` to “false.”

**Note** Devices used for databases for which recovery is not important (for example, `tempdb`), may have `dsync` set to “false.” For these devices, enabling `directio` may have an adverse performance effect, so you should carefully review device use before you enable `directio`.

The following creates a device named “user_disk” that uses `directio` to write data directly to disk:

```sql
  disk init
  name = "user_disk",
  physname = "/usr/u/sybase/data/userfile1.dat",
  size = 5120, directio = true
```

Initializes 10MB of a disk on a UNIX operating system file:

```sql
  disk reinit
  name = "user_disk",
  physname = "/usr/u/sybase/data/userfile1.dat",
  size = 5120, directio = true
```

By default, `directio` is disabled for all existing devices, and you enable it with `sp_deviceattr`. The syntax for `sp_deviceattr` is:

```sql
  sp_deviceattr device_name, directio, [true | false]
```
For example, the following enables directio disk writes for the “user_disk” device:

```sql
sp_deviceattr user_disk, directio, true
```

You must reboot the server for this change to take effect.

**Other optional parameters for `disk init`**

`vstart` is the starting virtual address, or the offset, for Adaptive Server to begin using the database device. `vstart` accepts the following optional unit specifiers: `k` or `K` (kilobytes), `m` or `M` (megabytes), `g` or `G` (gigabytes) and `t` or `T` (terabytes). The size of the offset depends on how you enter the value for `vstart`:

- If you do not specify a unit size, `vstart` uses 2K pages for its starting address. For example, if you specify `vstart = 13`, Adaptive Server uses 13 * 2K pages as the offset for the starting address.
- If you specify a unit value, `vstart` uses this as the starting address. For example, if you specify `vstart = "13M"`, Adaptive Server sets the starting address offset at 13 megabytes.

The default value (and usually the preferred value) of `vstart` is 0. If the specified device does not have the sum of `vstart + size` blocks available, the `disk init` command fails.

The optional `cntrtype` keyword specifies the disk controller. Its default value is 0. Reset it only if instructed to do so by your system administrator.

---

**Note** To perform disk initialization, the user who started Adaptive Server must have the appropriate operating system permissions on the device that is being initialized.

---

**Getting information about devices**

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

When used without a device name, `sp_helpdevice` lists all the devices available on Adaptive Server. 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
```

<table>
<thead>
<tr>
<th>device_name</th>
<th>physical_name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>master</td>
<td>d_master</td>
<td>special, default disk, physical disk, 30 MB</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>status</th>
<th>cntrltype</th>
<th>vdevno</th>
<th>vp_low</th>
<th>vpn_high</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10239</td>
</tr>
</tbody>
</table>

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. Entries in `sysdevices` 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` fields represent the page numbers that have been assigned to the device. For dump devices, they represent the media capacity of the device.

The `status` field in `sysdevices` is a bitmap that 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. The status bits and their meanings are listed in Table 7-2:

<table>
<thead>
<tr>
<th>Table 7-2: Status bits in sysdevices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit</td>
</tr>
<tr>
<td>------</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>
For more information about dump devices and `sp_addumpdevice`, see Chapter 11, “Developing a Backup and Recovery Plan.”

### Dropping devices

To drop database and dump devices, use `sp_dropdevice`. The syntax is:

```sql
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 Adaptive Server. You must use operating system commands to delete a file after using `sp_dropdevice`.

<table>
<thead>
<tr>
<th>Bit</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Physical disk</td>
</tr>
<tr>
<td>4</td>
<td>Logical disk (not used)</td>
</tr>
<tr>
<td>8</td>
<td>Skip header (used with tape dump devices)</td>
</tr>
<tr>
<td>16</td>
<td>Dump device</td>
</tr>
<tr>
<td>32</td>
<td>Serial writes</td>
</tr>
<tr>
<td>64</td>
<td>Device mirrored</td>
</tr>
<tr>
<td>128</td>
<td>Reads mirrored</td>
</tr>
<tr>
<td>256</td>
<td>Secondary mirror side only</td>
</tr>
<tr>
<td>512</td>
<td>Mirror enabled</td>
</tr>
<tr>
<td>2048</td>
<td>Used internally; set after <code>disk unmirror, side = retain</code></td>
</tr>
<tr>
<td>4096</td>
<td>Primary device needs to be unmirrored (used internally)</td>
</tr>
<tr>
<td>8192</td>
<td>Secondary device needs to be unmirrored (used internally)</td>
</tr>
<tr>
<td>16384</td>
<td>UNIX file device uses <code>dsync</code> setting (writes occur directly to physical media)</td>
</tr>
</tbody>
</table>

For more information about dump devices and `sp_addumpdevice`, see Chapter 11, “Developing a Backup and Recovery Plan.”
Designating default devices

To create a pool of default database devices to be used by all Adaptive Server 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}
```

You are most likely to 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
```

Adaptive Server 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.

**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.

Choosing default and nondefault devices

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

Make sure these devices are *not* default devices:

- The master device (use `sp_diskdefault` to set `defaultoff` after adding user devices)
- The device 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.

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

---

### 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 `/sybase/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.

You can 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.

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.

You can 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.

Increasing the size of devices with disk resize

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 before the error occurs.

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.

Device shrinkage

You cannot decrease the size of a device with disk resize.

disk resize syntax

disk resize has the following syntax:

```sql
disk resize
  name = "device_name",
  size = additional_space
```

Where `device_name` is the name of the device you are increasing and `additional_space` is the additional disk space you are adding to this device.

- You must have already initialized the device with disk init.
- `device_name` must refer to a valid logical device name.
- The minimum size for disk resize is 1MB or one allocation unit, whichever is greater.
- You must disable mirroring while the resize operation is in progress. You can reestablish mirroring when the resize operation is complete.

<table>
<thead>
<tr>
<th>Page size</th>
<th>Allocation unit size</th>
<th>Minimum incremental size</th>
</tr>
</thead>
<tbody>
<tr>
<td>2K</td>
<td>0.5MB</td>
<td>1MB</td>
</tr>
<tr>
<td>4K</td>
<td>1MB</td>
<td>1MB</td>
</tr>
<tr>
<td>8K</td>
<td>2MB</td>
<td>2MB</td>
</tr>
<tr>
<td>16K</td>
<td>4MB</td>
<td>4MB</td>
</tr>
</tbody>
</table>

Note The new size of the device is the sum of the old device size plus the size specified in the disk resize command.
**Disk resize example**

For example, the configuration of the device `testdev` from `isql`:

```
sp_helpdevice testdev
```

<table>
<thead>
<tr>
<th>device_name</th>
<th>physical_name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>status</td>
<td>cntrlttype</td>
<td>vdevno</td>
</tr>
<tr>
<td>------------</td>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>-------------</td>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>-------------</td>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>testdev</td>
<td>/sybase/dev/testdev.dat</td>
<td>special, dsync on, directio off, physical disk, 10.00MB</td>
</tr>
<tr>
<td>16386</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

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
```

<table>
<thead>
<tr>
<th>device_name</th>
<th>physical_name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>status</td>
<td>cntrlttype</td>
<td>vdevno</td>
</tr>
<tr>
<td>------------</td>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>testdev</td>
<td>/sybase/dev/testdev.dat</td>
<td>special, dsync on, directio off, physical disk, 14.00MB</td>
</tr>
<tr>
<td>16386</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

**Specifying a logical device name with `disk resize`**

The `device_name` must have a valid identifier. The device should have already been initialized using the `disk init` command and it must refer to a valid Adaptive Server device.

**Specifying the device size with `disk resize`**

You can use the following unit specifiers to indicate the size of the device: “k” or “K” to indicate kilobytes, “m” or “M” to indicate megabytes, “g” or “G” to indicate gigabytes, and “t” or “T” to indicate terabytes.
Increasing the size of devices with disk resize

Although it is optional, Sybase recommends that you always include the unit specifier with the disk resize command to avoid confusion in the actual number of pages allocated. You must enclose the unit specifier in single or double quotes. If you do not use a unit specifier, the size defaults to the number of disk pages.

To verify the new size, use sp_helpdevice.
Setting Database Options

This chapter describes how to use database options.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>What are database options?</td>
<td>277</td>
</tr>
<tr>
<td>Using the sp_dboption procedure</td>
<td>277</td>
</tr>
<tr>
<td>Database option descriptions</td>
<td>278</td>
</tr>
<tr>
<td>Changing database options</td>
<td>286</td>
</tr>
<tr>
<td>Viewing the options on a database</td>
<td>287</td>
</tr>
</tbody>
</table>

What are database options?

Database options control:
- The behavior of transactions
- Defaults for table columns
- Restrictions to user access
- Performance of recovery and bcp operations
- Log behavior

The system administrator and the database owner can use database options to configure the settings for an entire database. Database options differ from sp_configure parameters, which affect the entire server, and set options, which affect only the current session or stored procedure.

Using the *sp_dboption* procedure

Use *sp_dboption* to change settings for an entire database. The options remain in effect until they are changed. *sp_dboption*:
Database option descriptions

- 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 for user databases only. 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}]
```

To make an option or options take effect for every new database, change the option in the model database.

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
async log service
auto identity
dbo use only
ddl in tran
delayed commit
disable alias access
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.

The following sections describe each database option in detail.

**abort tran on log full**

`abort tran on log full` determines the fate of a transaction that is running when the last-chance threshold is crossed. The default value is `false`, meaning that the transaction is suspended and is awakened only when space has been freed. If you change the setting to `true`, all user queries that must write to the transaction log are killed until space in the log has been freed.

**allow nulls by default**

Setting `allow nulls by default` to `true` changes the default null type of a column from `not null` to `null`, in compliance with the SQL standard. The Transact-SQL default value for a column is `not null`, meaning that null values are not allowed in a column unless `null` is specified in the `create table` or `alter table` column definition.

You cannot use `allow nulls by default` to change the nullibility of a column during `select into` statements. Instead, use `convert` to specify the nullibility of the resulting columns.

**asynch log service**

Enabling `asynch log service` (ALS) allows for greater scalability in Adaptive Server, providing higher throughput in logging subsystems for high-end symmetric multiprocessor systems. You can enable ALS on any specified database that has at least one of the following performance issues, so long as your systems runs 4 or more online engines.
auto identity

While the auto identity option is true, a 10-digit IDENTITY column is defined in each new table that is created without specifying either a primary key, a unique constraint, or an IDENTITY column. This IDENTITY column is created only when you issue a create table command, not when you issue a select into. The column is not visible when you select all columns with the select * statement. To retrieve it, you must explicitly mention the column name, SYB_IDENTITY_COL, in the select list.

To set the precision of the automatic IDENTITY column, use the size of auto identity configuration parameter.

Though you can set auto identity to true in tempdb, it is not recognized or used, and temporary tables created there do not automatically include an IDENTITY column.

dbo use only

While dbo use only is set to true (on), only the database owner can use the database.

ddl in tran

Setting ddl in tran to true allows these commands to be used inside a user-defined transaction:

- alter table (clauses other than partition and unpartition are allowed)
- create default
- create index
- create procedure
- create rule
- create schema
- create table
- create trigger
- create view
- drop default
Data definition statements lock system tables for the duration of a transaction, which can result in performance problems. Use them only in short transactions.

These commands cannot be used in a user-defined transaction under any circumstances:

- alter database
- alter table...partition
- alter table...unpartition
- create database
- disk init
- dump database
- dump transaction
- drop database
- load transaction
- load database
- select into
- truncate table
- update statistics
**Database option descriptions**

**delayed commit**

The `delayed_commit` parameter allows you to determine when log records are written to disk. With the `delayed_commit` parameter set to true, the log records are asynchronously written to the disk and control is returned to the client without waiting for the IO to complete. This improves the response time for the transactions for which the `delayed_commit` parameter is enabled.

**identity in nonunique index**

`identity in nonunique index` automatically includes an IDENTITY column in a table’s index keys so that all indexes created on the table are unique. This database option makes logically nonunique indexes internally unique and allows those indexes to be used to process updatable cursors and isolation level 0 reads.

The table must already have an IDENTITY column for the `identity in nonunique index` option to work either from a `create table` statement or from setting the `auto_identity` database option to true before creating the table.

Use `identity in nonunique index` if you plan to use cursors and isolation level 0 reads on tables that have nonunique indexes. A unique index ensures that the cursor is positioned at the correct row the next time a `fetch` is performed on that cursor.

Do not confuse the `identity in nonunique index` option with unique `auto_identity index`, which is used to add an IDENTITY column with a unique, nonclustered index to new tables.

**no chkpt on recovery**

`no chkpt on recovery` is set to true (on) when an up-to-date copy of a database is kept. In these situations, there is a “primary” database and a “secondary” database. Initially, the primary database is dumped and loaded into the secondary database. Then, at intervals, the transaction log of the primary database is dumped and loaded into the secondary database.
If this option is set to false (off)—the default—a checkpoint record is added to the database after it is recovered by restarting Adaptive Server. This checkpoint, which ensures that the recovery mechanism is not re-run unnecessarily, changes the sequence number of the database. If the sequence number of the secondary database has been changed, a subsequent dump of the transaction log from the primary database cannot be loaded into it.

Turning this option on for the secondary database causes it to not get a checkpoint from the recovery process so that subsequent transaction log dumps from the primary database can be loaded into it.

**no free space acctg**

`no free space acctg` suppresses free-space accounting and execution of threshold actions for the non-log segments. This speeds recovery time because the free-space counts are not recomputed for those segments. It disables updating the rows-per-page value stored for each table, so system procedures that estimate space usage may report inaccurate values.

**read only**

`read only` means that users can retrieve data from the database, but cannot modify anything.

**select into/bulkcopy/pllsort**

`select into/bulkcopy/pllsort` must be set to `on` to perform operations that do not keep a complete record of the transaction in the log, which include:

- Using the `writetext` utility.
- Doing a `select` into a permanent table.
- Doing a “fast” `bulk copy` with `bcp`. By default, `fast bcp` is used on tables that do not have indexes.
- Performing a parallel sort.
Adaptive Server performs minimal logging for these commands, recording only page allocations and deallocations, but not the actual changes made to the data pages.

You do not have to set `select into /bulkcopy/pllsort` on to `select into` a user database when you issue the `select into` command to a temporary table. This is because temporary tables are created on `tempdb` and `tempdb` is never recovered. Additionally, you need not set the option to run `bcp` on a table that has indexes, because inserts are logged.

After you have run `select into` or performed a bulk copy in a database, you cannot perform a regular transaction log dump. After you have made minimally logged changes to your database, you must perform a `dump database`, since changes are not recoverable from transaction logs.

Setting `select into/bulkcopy/pllsort` does not block log dumping, but making minimally logged changes to data does block the use of a regular `dump transaction`. However, you can still use `dump transaction...with no_log` and `dump transaction...with truncate_only`.

By default, `select into/bulkcopy/pllsort` is turned off in newly created databases. To change the default, turn this option on in the `model` database.

**single user**

When `single user` is set to true, only one user at a time can access the database. You cannot set `single user` to true in `tempdb`.

**trunc log on chkpt**

When `trunc log on chkpt` is true (on), the transaction log is truncated (committed transactions are removed) when the checkpoint checking process occurs (usually once per minute), if 50 or more rows have been written to the log. The log is not truncated if less than 50 rows were written to the log, or if the database owner runs the `checkpoint` command manually.

You may want to turn this option on while doing development work during which backups of the transaction log are not needed. If this option is off (the default), and the transaction log is never dumped, the transaction log continues to grow, and you may run out of space in your database.
When `trunc log on chkpt` is on, you cannot dump the transaction log because changes to your data are not recoverable from transaction log dumps. Use `dump database` instead.

By default, the `trunc log on chkpt` option is off in newly created databases. To change the default, turn this option on in the `model` database.

---

**Warning!** If you set `trunc log on chkpt` on in `model`, and you need to load a set of database and transaction logs into a newly created database, be sure to turn the option off in the new database.

---

**unique auto_identity index**

When the `unique auto_identity index` option is set to `true`, it adds an IDENTITY column with a unique, nonclustered index to new tables. By default, the IDENTITY column is a 10-digit numeric datatype, but you can change this default with the `size of auto identity column` configuration parameter.

Though you can set `unique auto_identity index` to `true` in `tempdb`, it is not recognized or used, and temporary tables created there do not automatically include an IDENTITY column with a unique index.

The `unique auto_identity index` option provides a mechanism for creating tables that have an automatic IDENTITY column with a unique index that can be used with updatable cursors. The unique index on the table ensures that the cursor is positioned at the correct row after a `fetch`. (If you are using isolation level 0 reads and need to make logically nonunique indexes internally unique so that they can process updatable cursors, use the `identity in nonunique index` option.)

In some cases, the `unique auto_identity index` option can avoid the Halloween problem for the following reasons:

- Users cannot update an IDENTITY column; hence, it cannot be used in the cursor update.
- The IDENTITY column is automatically created with a unique, nonclustered index so that it can be used for the updatable cursor scan.

For more information about the Halloween Problem, IDENTITY columns, and cursors, see the Transact-SQL User's Guide.
Do not confuse the unique auto_identity index option with the identity in nonunique index option, which is used to make all indexes in a table unique by including an IDENTITY column in the table’s index keys.

Changing database options

Only a system administrator or the database owner can change a user’s database options by executing sp_dboption. Users aliased to the database owner cannot change database options with sp_dboption.

You must be using the master database to execute sp_dboption. Then, for the change to take effect, you must issue the checkpoint command while using the database for which the option was changed.

Remember that you cannot change any master database options.

To change pubs2 to read-only:

```bash
use master
sp_dboption pubs2, "read only", true
sp_dboption run checkpoint automatically.
```

For the optname parameter of sp_dboption, Adaptive Server understands any unique string that is part of the option name. To set the trunc log on chkpt option:

```bash
use master
sp_dboption pubs2, trunc, true
```

If you enter an ambiguous value for optname, an error message is displayed. For example, two of the database options are dbo use only and read only. Using “only” for the optname parameter generates a message because it matches both names. The complete names that match the string supplied are printed out so that you can see how to make the optname more specific.

You can turn on more than one database option at a time. You cannot change database options inside a user-defined transaction.
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
```
Viewing the options on a database
CHAPTER 9
Configuring Character Sets, Sort Orders, and Languages

This chapter discusses Adaptive Server internationalization and localization support issues.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Understanding internationalization and localization</td>
<td>289</td>
</tr>
<tr>
<td>Advantages of internationalized systems</td>
<td>290</td>
</tr>
<tr>
<td>A sample internationalized system</td>
<td>291</td>
</tr>
<tr>
<td>Elements of an internationalized system</td>
<td>293</td>
</tr>
<tr>
<td>Selecting the character set for your server</td>
<td>293</td>
</tr>
<tr>
<td>Selecting the sort order</td>
<td>303</td>
</tr>
<tr>
<td>Selecting a language for system messages</td>
<td>311</td>
</tr>
<tr>
<td>Setting up your server: examples</td>
<td>313</td>
</tr>
<tr>
<td>Changing the character set, sort order, or message language</td>
<td>315</td>
</tr>
<tr>
<td>Installing date strings for unsupported languages</td>
<td>326</td>
</tr>
<tr>
<td>Internationalization and localization files</td>
<td>327</td>
</tr>
</tbody>
</table>

Understanding internationalization and localization

Internationalization is the process of 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.
Localizatio

Localization is the process of 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.

Sybase provides both internationalization and localization support. Adaptive Server 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.

Sybase Language Modules provide translated system messages and formats for Chinese (Simplified), French, German, Japanese, Korean, Brazilian Portuguese, and Spanish. By default, Adaptive Server 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 Adaptive Server.

Advantages of internationalized systems

The task of 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.
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 (Adaptive Server)
- An order fulfillment server in France (Adaptive Server)
- A central accounting application in Japan (an Open Server application working with an Adaptive Server)

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:
Figure 9-1: Example of an international system

In this example, all applications and servers use local languages and character sets to accept input and output messages.
Elements of an internationalized system

There are three elements that you can manipulate to configure your server language in an internationalized environment. Sybase suggests that you review these three 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 Sybase. If your server language is not one of the languages provided, your system messages display in English, the default.

The following sections provide details about each of these elements.

Selecting the character set for your server

All data is encoded in your server 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 Adaptive Server, 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 Group 1 in Table 9-1 on page 295). 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 Table 9-1 on page 295). 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. See “Unicode” on page 296 for more information.

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.

Adaptive Server supports the following languages and character sets:
### Table 9-1: Supported languages and character sets

<table>
<thead>
<tr>
<th>Language group</th>
<th>Languages</th>
<th>Character sets</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Group 1</strong></td>
<td>Western European: Albanian, Catalan, Danish, Dutch, English, Faeroese, Finnish, French, Galician, German, Icelandic, Irish, Italian, Norwegian, Portuguese, Spanish, Swedish</td>
<td>ASCII 8, CP 437, CP 850, CP 860, CP 863, CP 1252&lt;sup&gt;a&lt;/sup&gt;, ISO 8859-1, ISO 8859-15, Macintosh Roman, ROMAN8, ROMAN9, ISO-15, CP 858</td>
</tr>
<tr>
<td><strong>Group 2</strong></td>
<td>Eastern European: Croatian, Czech, Estonian, Hungarian, Latvian, Lithuanian, Polish, Romanian, Slovak, Slovene (and English)</td>
<td>CP 852, CP 1250, ISO 8859-2, Macintosh Central European</td>
</tr>
<tr>
<td><strong>Group 4</strong></td>
<td>Baltic (and English)</td>
<td>CP 1257</td>
</tr>
<tr>
<td><strong>Group 5</strong></td>
<td>Cyrillic: Bulgarian, Byelorussian, Macedonian, Russian, Serbian, Ukrainian (and English)</td>
<td>CP 855, CP 866, CP 1251, ISO 8859-5, Koi8, Macintosh Cyrillic</td>
</tr>
<tr>
<td><strong>Group 6</strong></td>
<td>Arabic (and English)</td>
<td>CP 864, CP 1256, ISO 8859-6</td>
</tr>
<tr>
<td><strong>Group 7</strong></td>
<td>Greek (and English)</td>
<td>CP 869, CP 1253, GREEK8, ISO 8859-7, Macintosh Greek</td>
</tr>
<tr>
<td><strong>Group 8</strong></td>
<td>Hebrew (and English)</td>
<td>CP 1255, ISO 8859-8</td>
</tr>
<tr>
<td><strong>Group 9</strong></td>
<td>Turkish (and English)</td>
<td>CP 857, CP 1254, ISO 8859-9, Macintosh Turkish, TURKISH8</td>
</tr>
<tr>
<td><strong>Group 101</strong></td>
<td>Japanese (and English)</td>
<td>CP 932 DEC Kanji, EUC-JIS, Shift-JIS</td>
</tr>
<tr>
<td><strong>Group 102</strong></td>
<td>Simplified Chinese (PRC) (and English)</td>
<td>CP 936, EUC-GB, GB18030</td>
</tr>
<tr>
<td><strong>Group 103</strong></td>
<td>Traditional Chinese (ROC) (and English)</td>
<td>Big 5, CP 950&lt;sup&gt;b&lt;/sup&gt;, EUC-CNS, Big 5 HKSCS</td>
</tr>
<tr>
<td><strong>Group 104</strong></td>
<td>Korean (and English)</td>
<td>EUC-KSC, cp949</td>
</tr>
<tr>
<td><strong>Group 105</strong></td>
<td>Thai (and English)</td>
<td>CP 874, TIS 620</td>
</tr>
<tr>
<td><strong>Group 106</strong></td>
<td>Vietnamese (and English)</td>
<td>CP 1258</td>
</tr>
<tr>
<td><strong>Unicode</strong></td>
<td>Over 650 languages</td>
<td>UTF-8</td>
</tr>
</tbody>
</table>

<sup>a</sup> CP 1252 is identical to ISO 8859-1 except for the 0x80–0x9F code points which are mapped to characters in CP 1252.

<sup>b</sup> CP 950 is identical to Big 5.
Selecting the character set for your server

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.

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

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

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.

Unicode

Unicode is the first character set that 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.

Sybase 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.

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

The character set model used by Adaptive Server is based on a single, configurable, server-wide character set. All data stored in Adaptive Server, 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.

Character set installation

Adaptive Server version 12.5.1 and later supports the 4-byte form of UTF-8. This form is used to represent the same rare Unicode characters that are represented in UTF-16 by pairs of 16-bit values (“surrogate pairs”). Prior to Adaptive Server version 12.5.1, only the 3-byte forms of UTF-8 were supported. If you have installed the UTF-8 character set in an Adaptive Server server earlier than version 12.5.1, you should reinstall it to enable the use of the 4-byte form of UTF-8.
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 Adaptive Server to ensure the integrity of surrogate pairs. You can switch this checking off by setting the configuration parameter “enable surrogate processing” 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:

\[
\begin{align*}
0x00E9 & \quad \text{é (LATIN SMALL LETTER E WITH ACUTE)} \\
0x00650301 & \quad \text{e (LATIN SMALL LETTER E), ´ (COMBINING ACUTE ACCENT)}
\end{align*}
\]

The enable unicode normalization configuration parameter controls whether or not Adaptive Server 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 Adaptive Server to make several assumptions about the nature of the Unicode data, and code has been implemented to take advantage of these assumptions.

Functions

All built-in functions taking char parameters have been overloaded to accept unichar as well. Built-in 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 are modified to fall at the beginning of a surrogate pair.

Several functions have been added to 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.
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:

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

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

whereas:

```sql
% 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.

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 Adaptive Server default character set.

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.
Selecting the character set for your server

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.

Limitations

Due to the lack of a Unicode-based language parser in previous releases of Adaptive Server, a restriction was imposed on the use of the new Unicode datatypes. To use the new datatypes, the server required its default character set to be configured as UTF-8. This restriction has been removed in Adaptive Server release 12.5.1 and later. Unicode datatypes can be used regardless of the server’s default character set.

Selecting the server default character set

When you configure your server, you must specify a default character set for the server. The default character set is the character set in which the server stores and manipulates data. 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 Adaptive Server as the default on your server (see Table 9-1 on page 295).

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 nonconverted 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).

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

In the example below (Figure 9-2), 175 clients all access the same Adaptive Server. 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 (see Table 9-1 on page 295). The default language for the Adaptive Server 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.
Selecting the sort order

Different languages sort the same characters differently. For example, in English, *Cho* would be 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.
Selecting the sort order

Each character set comes with one or more sort orders that Adaptive Server 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. For a list of character sets and their available sort orders, see Table 9-5 on page 306.

Using sort orders

Sort orders are used to:

- Create indexes
- Store data into indexed tables
- Specify an order by clause

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.

Selecting the default sort order

Sybase servers can support only one default sort order at a time. If your users are using the same language or their languages use the same sort order, then select the desired sort order. 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.
Selecting the sort order

You can use the `sortkey` function to setup 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.

Table 9-5: Available sort orders

<table>
<thead>
<tr>
<th>Language or script</th>
<th>Character sets</th>
<th>Sort orders</th>
</tr>
</thead>
<tbody>
<tr>
<td>All languages</td>
<td>UTF-8</td>
<td>Multiple sort orders, see Table 9-7 for list</td>
</tr>
<tr>
<td>Cyrillic: Bulgarian, Byelorussian, Macedonian, Russian, Serbian, Ukrainian</td>
<td>CP 855, CP 866, CP 1251, ISO 8859-5, Koi8, Macintosh Cyrillic</td>
<td>Dictionary order, case sensitive, accent sensitive</td>
</tr>
<tr>
<td>Eastern European: Czech, Slovak</td>
<td>CP 852, ISO 8859-2, CP 1250</td>
<td>Dictionary order, case sensitive, accent sensitive, with preference</td>
</tr>
<tr>
<td>Eastern European: Czech, Slovak</td>
<td>CP 852, ISO 8859-2, CP 1250</td>
<td>Dictionary order, case insensitive, accent sensitive</td>
</tr>
<tr>
<td>English, French, German</td>
<td>ASCII 8, CP 437, CP850, CP 860, CP 863, CP 1252a, ISO 8859-1, ISO 8859-15, Macintosh Roman, ROMAN8, ROMAN9, ISO 15</td>
<td>Dictionary order, case sensitive, accent sensitive, with preference</td>
</tr>
<tr>
<td>English, French, German</td>
<td>CP 850, CP 858</td>
<td>Alternate dictionary order, case sensitive, accent insensitive</td>
</tr>
<tr>
<td>Greek</td>
<td>ISO 8859-7</td>
<td>Dictionary order, case sensitive, accent sensitive, with preference</td>
</tr>
<tr>
<td>Hungarian</td>
<td>ISO 8859-2</td>
<td>Dictionary order, case sensitive, accent sensitive, with preference</td>
</tr>
<tr>
<td>Japanese</td>
<td>EUCJIS, SJIS, DECKANJI</td>
<td>General purpose case-insensitive dictionary ordering</td>
</tr>
<tr>
<td>Russian</td>
<td>CP 866, CP 1251, ISO 8859-5, Koi8, Macintosh Cyrillic</td>
<td>Dictionary order, case sensitive, accent sensitive, with preference</td>
</tr>
<tr>
<td>Scandinavian</td>
<td>CP 850</td>
<td>Dictionary order, case sensitive, accent sensitive, with preference</td>
</tr>
</tbody>
</table>
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.

### Selecting case-insensitive sort orders for Chinese and Japanese character sets

Use two stored procedures to select case-insensitive sort orders:

- `sp_helpsort`
- `sp_configure`

**sp_helpsort**

`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
```
Selecting the sort order

sp_configure

To switch to a case-insensitive sort order, enter:

sp_configure 'default sortorder id', 52

Selecting the default Unicode sort order

The default Unicode sort order is distinctly different from 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.

Table 9-6 lists the available default Unicode sort orders.
Table 9-6: Default Unicode sort orders

<table>
<thead>
<tr>
<th>Name</th>
<th>ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>defaultml</td>
<td>20</td>
<td>Default Unicode multi-lingual ordering</td>
</tr>
<tr>
<td>thaidict</td>
<td>21</td>
<td>Thai dictionary ordering</td>
</tr>
<tr>
<td>iso14651</td>
<td>22</td>
<td>Ordering as per ISO14651 standard</td>
</tr>
<tr>
<td>utf8bin</td>
<td>24</td>
<td>Ordering for UTF-16 that matches the UTF-8 binary</td>
</tr>
<tr>
<td>binary</td>
<td>25</td>
<td>Binary sort</td>
</tr>
<tr>
<td>altnoacc</td>
<td>39</td>
<td>Alternate accent-insensitive</td>
</tr>
<tr>
<td>alldict</td>
<td>45</td>
<td>Alternate dictionary ordering</td>
</tr>
<tr>
<td>altnocp</td>
<td>46</td>
<td>Alternate case-insensitive with preference</td>
</tr>
<tr>
<td>scandict</td>
<td>47</td>
<td>Scandinavian dictionary ordering</td>
</tr>
<tr>
<td>scannocp</td>
<td>48</td>
<td>Scandinavian case-insensitive with preference</td>
</tr>
<tr>
<td>bin_utf8</td>
<td>50</td>
<td>UTF-8 binary sort order</td>
</tr>
<tr>
<td>dict</td>
<td>51</td>
<td>General-purpose dictionary ordering</td>
</tr>
<tr>
<td>nocase</td>
<td>52</td>
<td>General-purpose case-insensitive dictionary ordering</td>
</tr>
<tr>
<td>nocasep</td>
<td>53</td>
<td>General-purpose case-insensitive with preference</td>
</tr>
<tr>
<td>noaccent</td>
<td>54</td>
<td>General-purpose accent-insensitive dictionary ordering</td>
</tr>
<tr>
<td>espidict</td>
<td>55</td>
<td>Spanish dictionary ordering</td>
</tr>
<tr>
<td>espnocs</td>
<td>56</td>
<td>Spanish case-insensitive dictionary ordering</td>
</tr>
<tr>
<td>espnocac</td>
<td>57</td>
<td>Spanish accent-insensitive dictionary ordering</td>
</tr>
<tr>
<td>rsnocs</td>
<td>59</td>
<td>Russian case-insensitive dictionary ordering</td>
</tr>
<tr>
<td>cyrnocs</td>
<td>64</td>
<td>Cyrillic case-insensitive dictionary ordering</td>
</tr>
<tr>
<td>elldict</td>
<td>65</td>
<td>Greek dictionary ordering</td>
</tr>
<tr>
<td>hundict</td>
<td>69</td>
<td>Hungarian dictionary ordering</td>
</tr>
<tr>
<td>hunnoac</td>
<td>70</td>
<td>Hungarian accent-insensitive dictionary ordering</td>
</tr>
<tr>
<td>hunnocs</td>
<td>71</td>
<td>Hungarian case-insensitive dictionary ordering</td>
</tr>
<tr>
<td>turknoac</td>
<td>73</td>
<td>Turkish accent-insensitive dictionary ordering</td>
</tr>
</tbody>
</table>
Table 9-7 lists the loadable sort orders.

**Table 9-7: Loadable sort orders**

<table>
<thead>
<tr>
<th>Name</th>
<th>ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cp932bin</td>
<td>129</td>
<td>Ordering that matches the binary ordering of CP932</td>
</tr>
<tr>
<td>dynix</td>
<td>130</td>
<td>Chinese phonetic ordering</td>
</tr>
<tr>
<td>gb3213bn</td>
<td>137</td>
<td>Ordering that matches the binary ordering of GB2312</td>
</tr>
<tr>
<td>cyrdict</td>
<td>140</td>
<td>Common cyrillic dictionary ordering</td>
</tr>
<tr>
<td>turdict</td>
<td>155</td>
<td>Turkish Dictionary ordering</td>
</tr>
<tr>
<td>euckscbn</td>
<td>161</td>
<td>Ordering that matches the binary ordering of EUCKSC</td>
</tr>
<tr>
<td>gbpinyn</td>
<td>163</td>
<td>Chinese phonetic ordering</td>
</tr>
<tr>
<td>rusdict</td>
<td>165</td>
<td>Russian dictionary ordering</td>
</tr>
<tr>
<td>sjisbin</td>
<td>179</td>
<td>Ordering that matches the binary ordering of SJIS</td>
</tr>
<tr>
<td>eucjisbn</td>
<td>192</td>
<td>Ordering that matches the binary ordering of EUCJIS</td>
</tr>
<tr>
<td>big5bin</td>
<td>194</td>
<td>Ordering that matches the binary ordering of BIG5</td>
</tr>
</tbody>
</table>


You can add sort orders using external files in the `$SYBASE/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.

**Note** External Unicode sort orders are provided by Sybase. 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 Adaptive Server 12.5.1, 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.

Selecting a language for system messages

Any installation of Adaptive Server can use Language Modules containing files of messages in different languages. Adaptive Server 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

Table 9-8 displays the supported system message languages and their language
groups. Each user can select only one language per session for system
messages.

<table>
<thead>
<tr>
<th>Language group</th>
<th>System message languages</th>
<th>Character sets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group 1</td>
<td>French, German, Spanish, Brazilian Portuguese</td>
<td>ASCII 8, CP 437, CP 850, CP 860, CP 863, CP 1252, ISO 8859-1, ISO 8859-15, Macintosh Roman, ROMAN8</td>
</tr>
<tr>
<td>Group 2</td>
<td>Polish</td>
<td>Cp 1250, CP 852, ISO 8859-2</td>
</tr>
<tr>
<td>Group 101</td>
<td>Japanese</td>
<td>CP 932, DEC Kanji, EUC-JIS, Shift-JIS</td>
</tr>
<tr>
<td>Group 102</td>
<td>Simplified Chinese (PRC)</td>
<td>CP 936, EUC-GB, GB18030</td>
</tr>
<tr>
<td>Group 104</td>
<td>Korean</td>
<td>EUC-KSC, CP 949</td>
</tr>
<tr>
<td>Group 105</td>
<td>Thai</td>
<td>CP 874, TIS 620</td>
</tr>
<tr>
<td>Unicode</td>
<td>French, German, Spanish, Brazilian Portuguese, Japanese, Simplified Chinese, Korean</td>
<td>UTF-8</td>
</tr>
<tr>
<td>All Other Language Groups</td>
<td>English</td>
<td></td>
</tr>
</tbody>
</table>

Install Language Modules for all languages in which clients will receive
messages. These Language Modules, located in the locales subdirectory of the
Adaptive Server installation directory, are part of a group of files called
localization files. For information about localization files and the software
message directory structure, see “Types of localization files” on page 329.
Setting up your server: examples

This section discusses setup options and the steps necessary to implement them. This is only a sample, and is meant to suggest ideas and methods for your own setup process.

A Spanish-version server

This examples shows how to set up a new server with all clients using the same language. To do this:

1. Select the server language, in this case, Spanish. By reviewing Table 9-1 on page 295, you see that Spanish is part of language group 1. Based on your platform, select a character set from language group 1. Sybase 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 (see “Selecting the character set for your server” on page 293).

2. Install the Spanish Language Module in the server. This allows clients to view system messages in Spanish.

3. Select the default sort order. By referring to Table 9-5 on page 306, you see that Spanish has three possible sort orders, in addition to binary sort order. Select a sort order.

4. Restart the server.

A U.S.-based company in Japan

This example involves clients in Japan, who want to enter data, sort data, and receive system messages in Japanese, while submitting data to a server that is accessed by English-only users:

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.
Setting up your server: examples

3 Select the sort order. By referring to Table 9-5 on page 306, you can see that a binary sort order is the only sort order available for Japanese. Therefore, 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.

A Japan-based company with multinational clients

This company is located in Japan, and has clients in France, Germany, and Spain. You need to mix European and Asian languages in the same server.

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. By reviewing Table 9-1 on page 295, you can see that 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.
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 Adaptive Server. 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.

To display Adaptive Server's default sort order, character set, and a table of its primary sort orders, enter:

```
sp_helpsort
```

Changing the default character set

Adaptive Server can have only one *default character set*, the character set in which data is stored in its databases. When you install Adaptive Server, you specify a default character set.

---

**Warning!** Read the following carefully, and exercise caution when changing the default character set in Adaptive Server. Sybase strongly recommends that you perform backups before you change a default character set.

---

When you change the default character set in Adaptive Server, 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.

In all other cases, you must convert the existing data as follows:

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.
Changing the character set, sort order, or message language

See the Utility Guide for more information about using bcp to copy data.

Warning! 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. See Chapter 10, “Configuring Client/Server Character Set Conversions,” for more information about supported character set conversions.

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.

Changing the sort order with a resources file

Adaptive Server character sets can be changed using the resource file. The sample resource file sqlloc.rs is located in $SYBASE/ASE-12_5/init/sample_resource_files/.

The resource file from the Adaptive Server 12.5.1 installation looks similar to the following:

```plaintext
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:
```
Changing the default sort order

Adaptive Server can have only one default sort order, the collating sequence it uses to order data. When you consider changing the sort order for character data on a particular Adaptive Server, keep this in mind: all of your organization’s Adaptive Servers 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. For more information, see “Reconfiguring the character set, sort order, or message language” on page 317.

Reconfiguring the character set, sort order, or message language

This section summarizes the steps to take before and after changing Adaptive Server’s default character set, sort order, or message language. 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 Adaptive Server 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. Adaptive Server 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 Adaptive Server aborts the load.
Changing the character set, sort order, or message language

• 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). See “Changing the default character set” on page 315 for more information.

Unicode examples

In the following example, a fictitious database named xpubs will be modified to use univarchar columns.

Schema

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”.

```sql
> 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.

Converting to UTF-8

The first step towards using Unicode is to extract the data and convert it to UTF-8 form.

```sql
% bcp xpubs..authors out authors.utf8.bcp -c -utf8 -Uusa -P
```

The next step to install UTF-8 as the default character set in the server:

```sql
% charset -Uusa -P binary.srt utf8
% isql -Uusa -P
> sp_configure 'default sortorder id', 50, 'utf8'
```
> go
> shutdown
> go

Restart the server to modify the default character set and re-create indexes on the system tables. Restart the server a second time, then reload the data:

```
% isql -Usa -P
> sp_dboption xpubs, 'select into', true
> go
> use xpubs
> go
> checkpoint
> go
> delete from authors
> go
> quit
% bcp xpubs..authors in authors.utf8.bcp -c -Jutf8 -Usa -P
```

**Migrating selected columns to univarchar**

With a working database running with UTF-8 as the default character set, it becomes a simple matter to convert select columns to univarchar:

```
% isql -Usa -P
> use xpubs
> go
> alter table authors modify au_lname univarchar(255),
    au_fname univarchar(255)
> go
```

The columns are modified to the new datatypes, the data is converted in place, and the index is re-created.

**Migrating to or from unitext**

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.
Preliminary steps

Before you run the installation program to reconfigure Adaptive Server:

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 Adaptive Server default character set, and your current databases contain non ASCII-7 data, use bcp to copy the existing data out of your databases.

Once you have loaded the Language Module, you can run the Adaptive Server installation program, which allows you to:

- Install or remove message languages and character sets included with Adaptive Server
- 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.

Note Before you change the character set or sort order, Adaptive Server must have as many open databases as there are databases managed by the server. If Adaptive Server does not have a sufficient number of open databases when it is re-started after a change in sort order, Adaptive Server 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 for UNIX Platforms. If you are using Windows, use the Server Config utility, described in Configuration Guide for Windows. If you are adding a new character set that is not included with Adaptive Server, see the Sybase Character Sets manual for complete instructions.

If you installed additional languages but did not change the Adaptive Server character set or sort order, you have completed the reconfiguration process.
If you changed the Adaptive Server default character set, and your current databases contain non ASCII-7 data, copy your data back into your databases, using bcp with the necessary flags to enable conversion.

If you changed the Adaptive Server default sort order or character set, see “Reconfiguring the character set, sort order, or message language” on page 317.

Setting the user’s default language

If you install an additional language, users running client programs can run sp_modifylogin to set that language as their default language, or set the LANG variable on the client machine, with the appropriate entries in locales.dat.

Recovery after reconfiguration


After recovery is complete, the new sort order and character set definitions are loaded.

If you have changed the sort order, Adaptive Server switches to single-user mode to allow the necessary updates to system tables and to prevent other users from using the server. Each table with a character-based index is automatically checked to see if any indexes have been corrupted by the sort order change. Character-based indexes in tables are automatically rebuilt, if necessary, using the new sort order definition.

After the system indexes are rebuilt, character-based user indexes are marked “suspect” in the sysindexes system table. User tables with suspect indexes are marked “read-only” in sysobjects to prevent updates to these tables and use of the “suspect” indexes until they have been checked and, if necessary, rebuilt.

Range-partitioned user tables are checked for character-based partition keys, and are marked “suspect” if the sort order change or character set change might cause partition corruption.
Changing the character set, sort order, or message language

Next, the new sort order information replaces the old information in the area of the disk that holds configuration information. Adaptive Server then shuts down so that it starts for the next session with a complete and accurate set of system information.

Using `sp_induspect` to find corrupt indexes

After Adaptive Server shuts down, restart it, and use `sp_induspect` to find the user tables that need to be reindexed. The following is the syntax, where `tab_name` is the optional name of a specific table:

```
sp_induspect [tab_name]
```

If `tab_name` is missing, `sp_induspect` creates a list of all tables in the current database that has indexes marked “suspect” when the sort order changes.

In this example, running `sp_induspect` in `mydb` database yields one suspect index:

```
sp_induspect
Suspect indexes in database mydb
Own.Tab.Ind (Obj_ID, Ind_ID) =
dbo.holdings.h_name_ix(160048003, 2)
```

Rebuilding indexes after changing the sort order

`dbcc reindex` checks the integrity of indexes on user tables by running a “fast” version of `dbcc checktable`. For details, see “dbcc checktable” on page 228 in System Administration Guide: Volume 2. `dbcc reindex` drops and rebuilds the indexes where the sort order used is not consistent with the new sort order. When `dbcc reindex` discovers the first index-related error, it displays a message, and then rebuilds the inconsistent indexes. The system administrator or table owner should run `dbcc reindex` after changing the sort order in Adaptive Server.

The syntax is:

```
dbcc reindex ({table_name | table_id})
```

Run this command on all tables listed by `sp_induspect` as containing suspect indexes. For example:

```
dbcc reindex(titles)
```

One or more indexes are corrupt. They will be rebuilt.

In the preceding example, `dbcc reindex` discovers one or more suspect indexes in the table `titles`; it drops and re-creates the appropriate indexes.
If the indexes for a table are already correct, or if there are no indexes for the table, `dbcc reindex` does not rebuild any indexes. It displays a message instead. If a table is suspected of containing corrupt data, the command is aborted. If that happens, an error message instructs the user to run `dbcc checktable`.

When `dbcc reindex` finishes successfully, all “suspect” marks on the table’s indexes are removed. The “read-only” mark on the table is also removed, and the table can be updated. These marks are removed whether or not any indexes have to be rebuilt.

`dbcc reindex` does not reindex system tables. System indexes are checked and rebuilt, if necessary, as an automatic part of recovery after Adaptive Server is restarted following a sort order change.

Upgrading text data after changing character sets

If you have changed an Adaptive Server’s character set to a multibyte character set, use `dbcc fix_text` to upgrade text values.

The syntax is:

```
    dbcc fix_text (table_name | table_id)
```

Changing to a multibyte character set makes the management of text data more complicated. A text value can be large enough to cover several pages; therefore, Adaptive Server must be able to handle characters that span page boundaries. To do so, Adaptive Server requires additional information on each of the text pages. The system administrator or table owner must run `dbcc fix_text` on each table that has text data to calculate the new values needed.

To see the names of all tables that contain text data, use:

```
select sysobjects.name
from sysobjects, syscolumns
where syscolumns.type = 35
    and sysobjects.id = syscolumns.id
```

The system administrator or table owner must run `dbcc fix_text` to calculate the new values needed.

The syntax of `dbcc fix_text` is:

```
    dbcc fix_text (table_name | table_id)
```

The table named must be in the current database.
Changing the character set, sort order, or message language

dbcc fix_text opens the specified table, calculates the character statistics required for each text value, and adds the statistics to the appropriate page header fields. This process can take a long time, depending on the number and size of the text values in a table. dbcc fix_text can generate a large number of log records, which may fill up the transaction log. dbcc fix_text performs updates in a series of small transactions so that if a log becomes full, only a small amount of work is lost.

If you run out of log space, clear out your log (see Chapter 12, “Backing Up and Restoring User Databases” in System Administration Guide: Volume 2). Then restart dbcc fix_text, using the same table that was being upgraded when the original dbcc fix_text halted. Each multibyte text value contains information that indicates whether it has been upgraded, so dbcc fix_text upgrades only the text values that were not processed in earlier passes.

If your database stores its log on a separate segment, you can use thresholds to manage clearing the log. See Chapter 16, “Managing Free Space with Thresholds,” in System Administration Guide: Volume 2.

If dbcc fix_text cannot acquire a needed lock on a text page, it reports the problem and continues with the work, like this:

Unable to acquire an exclusive lock on text page 408. This text value has not been recalculated. In order to recalculate those TEXT pages you must release the lock and reissue the dbcc fix_text command.

Retrieving text values after changing character sets

If you attempt to retrieve text values after changing to a multibyte character set, and you have not run dbcc fix_text, the command fails with this error message:

Adaptive Server is now running a multi-byte character set, and this TEXT column’s character counts have not been recalculated using this character set. Use dbcc fix_text before running this query again.

Note If you have changed the sort order or character set and errors occurred, see “How to Manually Change Sort Order or Default Character Set” in the Adaptive Server Enterprise Troubleshooting and Error Messages Guide.
Handling suspect partitions

Partitions are marked suspect for two reasons:

- A sort order or character set change on a range-partitioned table.
- 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.

Fixing tables 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 with 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.

Handling 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.
Installing date strings for unsupported languages

You can 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. With `sp_addlanguage`, 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

This example adds 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,gia,lug,ago,sett,ott,nov,dic",
"luned,mar,mercoled,giovener,sabato,domenica",
dmy, 1
```

`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.

Server versus client date interpretation

Generally, date values are resolved on the client. When a user selects date values, Adaptive Server sends them to the client in 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, Adaptive Server looks for the `common.loc` file in `/locales/spanish/char_set`. It uses the information in the file to display, for example, 12 febrero 1997.
Assume that the user’s default language is set to Italian, a language for which Adaptive Server 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
```

### Internationalization and localization files

#### Types of internationalization files

The files that support data processing in a particular language are called **internationalization files**. Several types of internationalization files come with Adaptive Server. Table 9-9 describes these files.

<table>
<thead>
<tr>
<th>File</th>
<th>Location</th>
<th>Purpose and contents</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>charset.loc</code></td>
<td>In each character set subdirectory of the <code>charsets</code> directory</td>
<td>Character set definition files that define the lexical properties of each character, such as alphanumeric, punctuation, operand, and uppercase or lowercase. Used by Adaptive Server to correctly process data.</td>
</tr>
<tr>
<td><code>*.srt</code></td>
<td>In each character set subdirectory of the <code>charsets</code> directory</td>
<td>Defines the sort order for alphanumeric and special characters, including ligatures, diacritics, and other language-specific considerations.</td>
</tr>
<tr>
<td><code>*.xlt</code></td>
<td>In each character set subdirectory of the <code>charsets</code> directory</td>
<td>Terminal-specific character translation files for use with utilities such as <code>bcp</code> and <code>isql</code>. For more information about how the <code>.xlt</code> files are used, see Chapter 10, “Configuring Client/Server Character Set Conversions,” and the Utility Guide.</td>
</tr>
</tbody>
</table>
Warning! Do not alter any of the internationalization files. If you need to install a new terminal definition or sort order, contact your local Sybase office or distributor.

Character sets directory structure

Figure 9-3 shows the directory structure for the Western European character sets that come with Adaptive Server. 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:

Figure 9-3: Structure of the charsets directory

The following global variables contain information about character sets:

Table 9-10: Global variables used for character sets

<table>
<thead>
<tr>
<th>Global variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>@@char_convert</td>
<td>Contains 0 if character set conversion is not in effect. Contains 1 if character set conversion is in effect.</td>
</tr>
<tr>
<td>@@client_csname</td>
<td>The client’s character set name. Set to NULL if client character set has never been initialized; otherwise, it contains the name of the character set for the connection.</td>
</tr>
<tr>
<td>@@client_csid</td>
<td>The client’s character set ID. Set to -1 if client character set has never been initialized; otherwise, it contains the client character set ID from syscharsets for the connection.</td>
</tr>
<tr>
<td>@@client_csexpansion</td>
<td>Returns the expansion factor used when converting from server’s character set to client’s character set.</td>
</tr>
<tr>
<td>@@maxcharlen</td>
<td>The maximum length, in bytes, of a character in the Adaptive Server default character set.</td>
</tr>
<tr>
<td>@@ncharsize</td>
<td>The maximum length, in bytes, of a character set in the current server default character set.</td>
</tr>
</tbody>
</table>
CHAPTER 9 Configuring Character Sets, Sort Orders, and Languages

Types of localization files

Adaptive Server includes several localization files for each Language Module, as shown in Table 9-11.

<table>
<thead>
<tr>
<th>File</th>
<th>Location</th>
<th>Purpose and contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>locales.dat</td>
<td>In the locales directory</td>
<td>Used by client applications to identify the default message language and character set.</td>
</tr>
<tr>
<td>server.loc</td>
<td>In the character set subdirectories under each language subdirectory in the $SYBASE/$SYBASE_ASE/locales directory</td>
<td>Software messages translated into the local language. Sybase products have product-specific *.loc files. If an entry is not translated, that software message or string appears in U.S. English instead of the local language.</td>
</tr>
<tr>
<td>common.loc</td>
<td>In each language and character set directory of the locales directory</td>
<td>Contains the local names of the months of the year and their abbreviations and information about the local date, time, and money formats.</td>
</tr>
</tbody>
</table>

All Adaptive Server-related locales files (used by dataserver, sqlloc, syconfig, and so on) are in $SYBASE/$SYBASE_ASE/locales. All Open Client/Server-related locales files (ctlib, ctisql, ctbcp, optdiag, installjava, and so on) are located in $SYBASE/locales.

Warning! Do not alter any of the localization files. If you need to alter any information in those files, contact your local Sybase office or distributor.
Software messages directory structure

Figure 9-4 shows how localization files are arranged. Within the locales directory is a subdirectory for each language installed. There is always a us_english subdirectory. (On PC platforms, this directory is called english.) During installation, when you are prompted to select the languages you want installed on Adaptive Server, the install program lists the supported software message languages. If you install Language Modules for additional languages, you see subdirectories for those languages. Within each language are subdirectories for the supported character sets; for example, cp850 is a supported character set for us_english. Software message files for each Sybase product reside in the character set subdirectories.

Figure 9-4: Messages directory structure

Message languages and global variables

The following global variables contain information about languages:

<table>
<thead>
<tr>
<th>@@langid</th>
<th>Contains the local language ID of the language currently in use (specified in syslanguages.langid)</th>
</tr>
</thead>
<tbody>
<tr>
<td>@@language</td>
<td>Contains the name of the language currently in use (specified in syslanguages.name)</td>
</tr>
</tbody>
</table>
CHAPTER 10

Configuring Client/Server Character Set Conversions

This chapter describes how to configure character set conversion when the client uses a different character set than Adaptive Server.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Character set conversion in Adaptive Server</td>
<td>331</td>
</tr>
<tr>
<td>Supported character set conversions</td>
<td>332</td>
</tr>
<tr>
<td>Types of character set conversion</td>
<td>334</td>
</tr>
<tr>
<td>Which type of conversion do I use?</td>
<td>335</td>
</tr>
<tr>
<td>Enabling and disabling character set conversion</td>
<td>337</td>
</tr>
<tr>
<td>Error handling in character set conversion</td>
<td>338</td>
</tr>
<tr>
<td>Conversions and changes to data lengths</td>
<td>339</td>
</tr>
<tr>
<td>Specifying the character set for utility programs</td>
<td>340</td>
</tr>
<tr>
<td>Display and file character set command line options</td>
<td>341</td>
</tr>
</tbody>
</table>

Character set conversion in Adaptive Server

In a heterogeneous environment, Adaptive Server 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 “à” remains an “a” even when crossing between machine and character set boundaries. This process is known as character set conversion.
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.

Conversion for native character sets

Adaptive Server supports character set conversion between native character sets belonging 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. Figure 10-1 is an example of a Western European client/server system. In this example, the clients’ character sets and the Adaptive Server default character set all belong to 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.

Figure 10-1: Character set conversion when server and client character sets belong to the same language group

For a list of the language groups and supported character sets, see Table 9-1 on page 295.
Conversion in a Unicode system

Adaptive Server also supports character set conversion between UTF-8 and any native character set that Sybase 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.

*Figure 10-2: 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.

*Figure 10-3: 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 Adaptive Server 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.
Sybase does not recommend that you configure a server for ASCII-7, but you can achieve the same benefits of compatibility by restricting each client to use only the first 128 characters of each native character set.

**Types of character set conversion**

Character set conversion is implemented on Adaptive Server in two different ways:

- Adaptive Server direct conversions
- Unicode conversions

**Adaptive Server direct conversions**

Adaptive Server direct conversions support conversions between two native character sets of the same language group. For example, Adaptive Server supports conversion between CP 437 and CP 850, because both belong to the group 1 language group. Adaptive Server direct conversions exist between many, but not all, native character sets of a language group (see Table 10-1 on page 336).

**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 Adaptive Server used direct conversions, and it is the default method for character set conversions. However, Unicode conversions allow easier and less complex character set conversion. Sybase continues to support existing Adaptive Server direct conversions, but Sybase now also uses Unicode conversions to provide complete conversion support for all character sets. Sybase has no plans to add new direct conversions.

Which type of conversion do I use?

To determine the conversion options that are available for your client/server system, see Table 10-1 on page 336.

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 Adaptive Server direct conversions.

However, there are some character sets for which there is no Adaptive Server direct conversion; in this situation, you must use Unicode conversions.

- If all character sets in your client/server system fall into column 1 of Table 10-1, use the Adaptive Server direct conversions. The character sets must all belong to the same language group.
- If the character sets in your client/server system fall into column 2 of Table 10-1, or some combination of columns 1 and 2, then you must 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. Table 10-1 shows that direct conversions exist between CP 850 and the client character sets. Now, suppose you add a client using CP 1252 to this configuration. 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 Adaptive Server direct conversions and others where you must use Unicode conversions—you can specify that a combination of Adaptive Server direct conversion and Unicode conversion be used.
**Which type of conversion do I use?**

**Unicode client/server systems**

If your server default is Unicode UTF-8, then all conversions are between UTF-8 and whatever native character set is being used on the client systems. Therefore, in a Unicode system, Unicode conversions are used exclusively.

**Table 10-1: Conversion methods for character sets**

<table>
<thead>
<tr>
<th>Language group</th>
<th>Column 1 – Adaptive Server direct conversions and Unicode conversions</th>
<th>Column 2 – Unicode conversions only</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group 1</td>
<td>CP 437, CP 850, ISO 8859-1, Macintosh Roman</td>
<td>CP 860, CP 1252, ISO 8859-15, CP 863</td>
</tr>
<tr>
<td>Group 2</td>
<td>CP 852, CP 1250, CP 8859-1, Macintosh Central European</td>
<td>ISO 8859-2</td>
</tr>
<tr>
<td>Group 5</td>
<td>No conversions needed (only one character set supported)</td>
<td></td>
</tr>
<tr>
<td>Group 6</td>
<td>CP 855, CP 866, CP 1251, ISO 8859-5, Koi8, Macintosh Cyrillic</td>
<td>CP 864, CP 1256, ISO 8859-6</td>
</tr>
<tr>
<td>Group 7</td>
<td>CP 869, CP 1253, GREEK8, ISO 8859-7, Macintosh Greek</td>
<td>CP 1255, ISO 8859-8</td>
</tr>
<tr>
<td>Group 8</td>
<td>CP 857, CP 1254, ISO 8859-9, Macintosh Turkish, TURKISH8</td>
<td></td>
</tr>
<tr>
<td>Group 101</td>
<td>DEC Kanji, EUC-JIS, Shift-JIS</td>
<td>CP 932</td>
</tr>
<tr>
<td>Group 102</td>
<td></td>
<td>CP 936, EUG-GB, GB18303</td>
</tr>
<tr>
<td>Group 103</td>
<td></td>
<td>Big 5, CP 950, EUC-CNS</td>
</tr>
<tr>
<td>Group 104</td>
<td></td>
<td>EUCKSC, CP 949</td>
</tr>
<tr>
<td>Group 105</td>
<td></td>
<td>CP 874, TIS 620</td>
</tr>
<tr>
<td>Group 106</td>
<td>No conversions needed (only one character set supported)</td>
<td></td>
</tr>
<tr>
<td>Unicode</td>
<td>No conversions needed (only one character set supported)</td>
<td></td>
</tr>
</tbody>
</table>

**Configuring the server**

By default, Adaptive Server uses direct conversions to convert data between different character sets. To use the Unicode conversions, you must configure the server with the `sp_configure` command. Set the `enable unicode conversions` option to either 1 or 2.

- If you set `sp_configure "enable unicode conversions"` to 1:

  This setting uses Adaptive Server direct conversions or Unicode conversions. Adaptive Server first checks to see if an Adaptive Server 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.
CHAPTER 10 Configuring Client/Server Character Set Conversions

Use this setting if the character sets in your client/server system fall into both columns 1 and 2 in Table 10-1.

- If you set `sp_configure “enable unicode conversions”` to 2:

  This setting uses Unicode conversions only. Adaptive Server uses Unicode conversions, without attempting to find an Adaptive Server direct conversion.

  Use this setting if the client/server conversions result in a change in the data length (see “Conversions and changes to data lengths” on page 339).

If all character sets fall into column 2 in Table 10-1, set `enable unicode conversions` to 2 to always use Unicode conversions.

For Adaptive Server release 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.

Enabling and disabling character set conversion

When a client requests a connection, the client identifies its character set to Adaptive Server. Adaptive Server 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, Adaptive Server determines whether it supports conversion between its default and the client’s character set. If it does not, it send an error message to the client and continues with the logon process. If it does, character set conversion is automatically enabled. If the default character set of the server is UTF-8, it automatically uses Unicode conversions. If the default is a native character set, the server uses Adaptive Server direct conversions, unless the user specifies that Unicode conversions be used.

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 converting the data, that is, without changing the encoding of the data.
To disable character set conversion at the server level, set the `disable character set conversion` parameter to 1. No conversion occurs for any client connecting to the server. By default this parameter is set to 0, which enables conversions.

You can also 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.

### Characters that cannot be converted

During the conversion process, some characters may not be converted. Here are two reasons:

- The character exists (is encoded) in the source character set, but it 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. You can avoid this limitation by configuring the `enable unicode conversion` option to 1 or 2.

### Error handling in character set conversion

The Adaptive Server 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. Adaptive Server 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, Adaptive Server must avoid putting suspect data into the database.
CHAPTER 10  Configuring Client/Server Character Set Conversions

When Adaptive Server 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 Adaptive Server encounters a conversion error while sending data to the client, it replaces the bytes of the suspect characters with ASCII question marks (?). However, the query batch continues to completion. When the statement is complete, Adaptive Server 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 (‘?’).

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, this occurs 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
Specifying the character set for utility programs

Configuring your system and application

If you are using UTF-8 anywhere in your client/server system, or using a Japanese character set, you are likely to encounter changes in data length as a result of character set conversion. If either of these conditions is true, you must configure your server to handle changes in data length. You may also need to set up your client to handle changes in data length.

1 Configure the server to use Unicode conversions. See “Configuring the server” on page 336. If the data length increases between the server and the client, then 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 is able to 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:

```c
CS_INT capval = CS_TRUE
call 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.

Specifying the character set for utility programs

The Sybase utility programs assume that the default character set of the client platform is the same character set the client is using. However, sometimes 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. Character set conversion can be controlled in the standalone utilities. 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.
Omitting the client character set’s command line flag causes the platform’s default character set to be used. See the Utility Guide for information.

### Display and file character set command line options

Although the focus of this chapter is on character set conversion between clients and Adaptive Server, there are two other places where you may need character set conversion:

- Between the client and a terminal
- Between the client and a file system

Figure 10-4 illustrates the paths and command line options that are available in the standalone utilities `isql`, `bcp`, and `defncopy`.

*Figure 10-4: Where character set conversion may be needed*

As described earlier, the `-J` or `/clientcharset` command line option specifies the character set used by the client when it sends and receives character data to and from Adaptive Server.
Display and file character set command line options

Setting the display character set

Use the -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 Figure 10-4, the -a option and the -J option are used together to identify the character set translation file (.xlt file) needed for the conversion.

Use -a without -J only if the client character set is the same as the default character set.

Setting the file character set

Use the -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 Figure 10-4, 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.
CHAPTER 11

Diagnosing System Problems

This chapter discusses diagnosing and fixing system problems.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>How Adaptive Server uses error messages</td>
<td>343</td>
</tr>
<tr>
<td>Adaptive Server error logging</td>
<td>346</td>
</tr>
<tr>
<td>Backup Server error logging</td>
<td>355</td>
</tr>
<tr>
<td>Killing processes</td>
<td>356</td>
</tr>
<tr>
<td>Housekeeper functionality</td>
<td>360</td>
</tr>
<tr>
<td>Configuring Adaptive Server to save SQL batch text</td>
<td>363</td>
</tr>
<tr>
<td>Shutting down servers</td>
<td>368</td>
</tr>
<tr>
<td>Learning about known problems</td>
<td>370</td>
</tr>
</tbody>
</table>

How Adaptive Server uses error messages

When Adaptive Server encounters a problem, it displays information—in an error message that describes whether the problem is caused by the user or the system—about the problem, how serious it is, and what you can do to fix it. The error message consists of:

- **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 Adaptive Server code at which the error was raised
- **An error message**, which tells you what the problem is, and may suggest how to fix it

For example, this is what happens if you try to access a table that does not exist:

```sql
select * from publisher
```

Msg 208, Level 16, State 1:
How Adaptive Server uses error messages

Publisher not found. Specify owner.objectname or use sp_help to check whether the object exists (sp_help may produce lots of output).

In some cases, there can be more than one error message for a single query. If there is more than one error in a batch or query, Adaptive Server usually reports only the first one. Subsequent errors are reported the next time you execute the batch or query.

The error messages are stored in master..sysmessages, which is updated with each new version of Adaptive Server. Here are the first few rows (from an Adaptive Server with us_english as the default language):

```sql
select error, severity, description
from sysmessages
where error >=101 and error <=106
and langid is null
```

<table>
<thead>
<tr>
<th>error</th>
<th>severity</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>15</td>
<td>Line %d: SQL syntax error.</td>
</tr>
<tr>
<td>102</td>
<td>15</td>
<td>Incorrect syntax near '%.*s'.</td>
</tr>
<tr>
<td>103</td>
<td>15</td>
<td>The %S_MSG that starts with '%.*s' is too long. Maximum length is %d.</td>
</tr>
<tr>
<td>104</td>
<td>15</td>
<td>Order-by items must appear in the select-list if the statement contains set operators.</td>
</tr>
<tr>
<td>105</td>
<td>15</td>
<td>Unclosed quote before the character string '%.*s'.</td>
</tr>
<tr>
<td>106</td>
<td>16</td>
<td>Too many table names in the query. The maximum allowable is %d.</td>
</tr>
</tbody>
</table>

(6 rows affected)

You can generate your own list by querying sysmessages. Here is some additional information for writing your query:

- 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. For information about languages on your server, use sp_helplanguage.

- The dlevel column in sysmessages is currently unused.

- 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.
Error messages and message numbers

The combination of message number (error) and language ID (langid) uniquely identifies each error message. Messages with the same message number but different language IDs are translations.

```sql
select error, description, langid
from sysmessages
where error = 101
```

<table>
<thead>
<tr>
<th>error</th>
<th>description</th>
<th>langid</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>Line %d: SQL syntax error.</td>
<td>NULL</td>
</tr>
<tr>
<td>101</td>
<td>Ligne %1!: erreur de syntaxe SQL.</td>
<td>1</td>
</tr>
<tr>
<td>101</td>
<td>Zeile %1!: SQL Syntaxfehler.</td>
<td>2</td>
</tr>
</tbody>
</table>

(3 rows affected)

The error message text is a description of 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 Adaptive Server 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 kind of database object; “%.s”—all within quotes—is a placeholder for the name of a particular database object. Table 11-1 on page 346 lists placeholders and what they represent.

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 as displayed 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. (See “Reporting errors” on page 354.)

Variables in error message text

Table 11-1 explains the symbols that appear in the text provided with each error message explanation:
Adaptive Server error logging

Table 11-1: Error text symbols key

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Stands for</th>
</tr>
</thead>
<tbody>
<tr>
<td>%d, %D</td>
<td>Decimal number</td>
</tr>
<tr>
<td>%x,%X,%.*x,%1x,%04x,%08x</td>
<td>Hexadecimal number</td>
</tr>
<tr>
<td>%s</td>
<td>Null-terminated string</td>
</tr>
<tr>
<td>%.*s, %<em>s, %</em>.s</td>
<td>String, usually the name of a particular database object</td>
</tr>
<tr>
<td>%S_type</td>
<td>Adaptive Server-defined structure</td>
</tr>
<tr>
<td>%c</td>
<td>Single character</td>
</tr>
<tr>
<td>%f</td>
<td>Floating-point number</td>
</tr>
<tr>
<td>%ld</td>
<td>Long decimal</td>
</tr>
<tr>
<td>%lf</td>
<td>Double floating-point number</td>
</tr>
</tbody>
</table>

Adaptive Server error logging

Error messages from Adaptive Server are sent only to the user’s screen. The stacktrace from fatal error messages (severity levels 19 and higher) and error messages from the kernel are also sent to an error log file. The name of this file varies; see the configuration documentation for your platform or the Utility Guide.

Note The error log file is owned by the user who installed Adaptive Server (or the person who started Adaptive Server 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 Adaptive Server.
Adaptive Server creates an error log for you if one does not already exist. You specify the location of the error log at start-up with the *errorlogfile* parameter in the runserver file or at the command line. The Sybase installation utility 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 Adaptive Server. For more information about specifying the location of the error log, see *dataserver* in the *Utility Guide*.

**Note** Always start Adaptive Server 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 Adaptive Server is shut down.

**Error log format**

Entries in the error log include the following information:

- The engine involved for each log entry. The engine number is indicated by a 2-digit number. If only one engine is online, the display is “00.”
- The family ID of the originating thread:
  - In serial processing, the display is “00000.”
  - In parallel processing, the display is the server process ID number of the parent of the originating thread.
- 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, then the display is “00000.”
  - In parallel processing, this is the server process ID number of the originating thread.
- The date, displayed in the format *yyyyMMdd*, which allows you to sort error messages by date.
Adaptive Server error logging

- The time, displayed in 24-hour format, which includes seconds and hundredths of a second.
- The word “server” or “kernel.” This entry is for Sybase Technical Support use only.
- The error message itself.

Figure 11-1 shows two examples of a line from an error log:

**Figure 11-1: Error log format**

**Single-engine server**

00:00000:00008:1997/05/16 15:11:46.58 server Process id 9
killed by Hostname danish, Host process id 3507.

**Multi-engine server**

00:00345:00023:1997/04/16 12:48:58.76 server The
configuration option 'allow updates to system tables' has
been changed by 'sa' from '1' to '0'.

**Severity levels**

The severity level of a message indicates information about the type and severity of the problem that Adaptive Server has encountered. For maximum integrity, when Adaptive Server 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 Adaptive Server messages and severity levels.

**Warning!** You can create your own error numbers and messages based on Adaptive Server error numbers (for example, by adding 20,000 to the Adaptive Server value). However, you cannot alter the Adaptive Server-supplied system messages in the `sysmessages` system table.

You can add user-defined error messages to `sysusermessages` with `sp_addmessage`. See the Reference Manual.
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 System Administration Guide: Volume 2:

- dbcc is discussed in Chapter 10, “Checking Database Consistency.”
- Loading a user database is discussed in Chapter 12, “Backing Up and Restoring User Databases.”
- Loading system databases is discussed in Chapter 13, “Restoring the System Databases.”

The following sections discuss each severity level.

**Severity levels 10–18**

Error messages with severity levels 10–16 are generated by problems that are caused by user errors. These problems can always 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.

**Level 10: Status information**

Messages with severity level 10 are not errors at all. They 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 has been run, Adaptive Server displays a message telling the user how much of the requested space has been allocated for the new database.

**Level 11: Specified database object not found**

Messages with severity level 11 indicate that Adaptive Server cannot find an object that was referenced in the command.
Adaptive Server error logging

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 the object names, use the owner names if the object is not owned by you or “dbo,” and make sure you are in the correct database.

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`. For information on `convert`, see the `Reference Manual` or the `Transact-SQL User's Guide`.

Level 13: User transaction syntax error

Messages with severity level 13 indicate that something is wrong with the current user-defined transaction. For example, you may have issued a `commit transaction` command without having issued a `begin transaction`, or you 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.

Level 14: Insufficient permission to execute command

Messages with severity level 14 mean that you do not have the necessary permission to execute the command or access the database object. You can ask the owner of the database object, the owner of the database, or the system administrator to grant you permission to use the command or object in question.

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.
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 can 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. Adaptive Server 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. When you are running the dbcc utility, check the Error Messages and Troubleshooting Guide for information about error messages between 2500 and 2599 with a severity level of 16.

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.

Level 17: Insufficient resources

Error messages with severity level 17 mean that the command has caused Adaptive Server to run out of resources or to exceed some limit set by the system administrator. You can continue with the work you are doing, although you may 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 Adaptive Server. They are stored in system tables and can be checked with sp_configure. See Chapter 5, “Setting Configuration Parameters,” for more information on changing configuration variables.

The database owner can correct the level 17 error messages indicating that you have run out of space. Other level 17 error messages should be corrected by the system administrator.
Level 18: Non-fatal internal error detected

Error messages with severity level 18 indicate some kind of internal software bug. However, the command runs to completion, and the connection to Adaptive Server is maintained. You can continue with the work you are doing, although you may not be able to execute a particular command. An example of a situation that generates severity level 18 is Adaptive Server 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. 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.

Severity levels 19–26

Fatal problems generate error messages with severity levels 19 and higher. They break the user’s connection to Adaptive Server (some of the higher severity levels shut down Adaptive Server). 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 was happening. It 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, it will be necessary to restart Adaptive Server. 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.

A backtrace of fatal error messages from the kernel is directed to the error log file, where the system administrator can review it.

Level 19: Adaptive Server fatal error in resource

Error messages with severity level 19 indicate that some non-configurable internal limit has been exceeded and that Adaptive Server cannot recover gracefully. You must reconnect to Adaptive Server.
Level 20: Adaptive Server fatal error in current process

Error messages with severity level 20 indicate that Adaptive Server has encountered a bug in a command. The problem has affected only the current process, and it is unlikely that the database itself has been damaged. Run dbcc diagnostics. You must reconnect to Adaptive Server.

Level 21: Adaptive Server fatal error in database processes

Error messages with severity level 21 indicate that Adaptive Server 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 Adaptive Server and run the dbcc diagnostics. You must reconnect to Adaptive Server.

Level 22: Adaptive Server fatal error: Table integrity suspect

Error messages with severity level 22 indicate that the table or index specified in the message was previously damaged by a software or hardware problem.

The first step is to restart Adaptive Server and run dbcc to determine whether other objects in the database are also damaged. Whatever the report from dbcc may be, it is possible that the problem is in the cache only and not on the disk itself. If so, restarting Adaptive Server 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 Adaptive Server has found a row with length 0 in a nonclustered index, the table owner can drop the index and re-create it.

Adaptive Server 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 for more information.

You must reconnect to Adaptive Server.

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 Adaptive Server 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 Adaptive Server with `startserver` fixes the problem.

**Level 24: Hardware error or system table corruption**

Error messages with severity level 24 reflect some kind of 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.

**Level 25: Adaptive Server internal error**

Level 25 errors are not displayed to the user; this level is only used for Adaptive Server internal errors.

**Level 26: Rule error**

Error messages with severity level 26 reflect that an internal locking or synchronization rule was broken. You must shut down and restart Adaptive Server.

**Reporting errors**

When you report an error, 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, which command was running at the time. You can help by providing a hard copy of the backtrace from the error log.
Backup Server error logging

Like Adaptive Server, Backup Server creates an error log if one does not already exist. You 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 Sybase installation utility 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 (`bcsrv -V` on Windows NT) to limit the messages printed to the error log. For more information about specifying the location of the error log, see the sections describing Backup Server in the Utility Guide.

Backup Server error messages are in the form:

```
MMM DD YYY: Backup Server:N.N.N.N: Message Text
```

Backup Server message numbers consist of 4 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.
Killing processes

- 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.

Killing processes

A process is a unit of execution carried out by Adaptive Server. 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 spids. Several processes create and assign spids: starting Adaptive Server, 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 kill command gets rid of an ongoing process. 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 processes that are:

- Waiting for an alarm, such as a waitfor command
- Waiting for network sends or receives
- Waiting for a lock
- Waiting for synchronization messages from another process in a family
- Most running or “runnable” processes
Adaptive Server 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`.

Table 11-2 shows the values that `sp_who` reports and when the `kill` command takes effect.

<table>
<thead>
<tr>
<th>Status</th>
<th>Indicates</th>
<th>Effects of <code>kill</code> command</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>recv sleep</code></td>
<td>Waiting on a network read</td>
<td>Immediate.</td>
</tr>
<tr>
<td><code>send sleep</code></td>
<td>Waiting on a network send</td>
<td>Immediate.</td>
</tr>
<tr>
<td><code>alarm sleep</code></td>
<td>Waiting on an alarm such as:</td>
<td>Immediate.</td>
</tr>
<tr>
<td></td>
<td><code>waitfor delay &quot;10:00&quot;</code></td>
<td></td>
</tr>
<tr>
<td><code>lock sleep</code></td>
<td>Waiting on a lock acquisition</td>
<td>Immediate.</td>
</tr>
<tr>
<td><code>sync sleep</code></td>
<td>Waiting on a synchronization message from another process in the family.</td>
<td>Immediate. Other processes in the family must also be brought to state in which they can be killed.</td>
</tr>
<tr>
<td><code>sleeping</code></td>
<td>Waiting on a disk I/O, or some other resource. Probably indicates a process that is running, but doing extensive disk I/O</td>
<td>Killed when it “wakes up,” usually immediate; a few sleeping processes do not wake up and require a Server restart to clear.</td>
</tr>
<tr>
<td><code>runnable</code></td>
<td>In the queue of runnable processes</td>
<td>Immediate.</td>
</tr>
<tr>
<td><code>running</code></td>
<td>Actively running on one of the server engines</td>
<td>Immediate.</td>
</tr>
<tr>
<td><code>infected</code></td>
<td>Server has detected serious error condition; extremely rare</td>
<td><code>kill</code> command not recommended. Server restart probably required to clear process.</td>
</tr>
<tr>
<td><code>background</code></td>
<td>A process, such as a threshold procedure, run by Adaptive Server rather than by a user process</td>
<td>Immediate; use <code>kill</code> with extreme care. Recommend a careful check of <code>sysprocesses</code> before killing a background process.</td>
</tr>
<tr>
<td><code>log suspend</code></td>
<td>Processes suspended by reaching the last-chance threshold on the log</td>
<td>Immediate.</td>
</tr>
</tbody>
</table>

Only a system administrator 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
```
Killing processes

2> kill 8
3> kill 9
4> go

A kill command is not reversible 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:

<table>
<thead>
<tr>
<th>fid</th>
<th>spid</th>
<th>status</th>
<th>loginame</th>
<th>origname</th>
<th>hostname</th>
<th>blk</th>
<th>dbname</th>
<th>cmd</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>recv sleep</td>
<td>howard</td>
<td>howard</td>
<td>svr30eng</td>
<td>0</td>
<td>master</td>
<td>AWAITING COMMAND</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>sleeping</td>
<td>NULL</td>
<td>NULL</td>
<td>svr30eng</td>
<td>0</td>
<td>master</td>
<td>NETWORK HANDLER</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>sleeping</td>
<td>NULL</td>
<td>NULL</td>
<td>svr30eng</td>
<td>0</td>
<td>master</td>
<td>DEADLOCK TUNE</td>
</tr>
<tr>
<td>0</td>
<td>4</td>
<td>sleeping</td>
<td>NULL</td>
<td>NULL</td>
<td>svr30eng</td>
<td>0</td>
<td>master</td>
<td>MIRROR HANDLER</td>
</tr>
<tr>
<td>0</td>
<td>5</td>
<td>sleeping</td>
<td>NULL</td>
<td>NULL</td>
<td>svr30eng</td>
<td>0</td>
<td>master</td>
<td>CHECKPOINT SLEEP</td>
</tr>
<tr>
<td>0</td>
<td>6</td>
<td>sleeping</td>
<td>NULL</td>
<td>NULL</td>
<td>svr30eng</td>
<td>0</td>
<td>master</td>
<td>HOUSEKEEPER</td>
</tr>
<tr>
<td>0</td>
<td>7</td>
<td>recv sleep</td>
<td>bill</td>
<td>bill</td>
<td>bigblue</td>
<td>0</td>
<td>master</td>
<td>AWAITING COMMAND</td>
</tr>
<tr>
<td>0</td>
<td>8</td>
<td>recv sleep</td>
<td>wilbur</td>
<td>wilbur</td>
<td>hazel</td>
<td>0</td>
<td>master</td>
<td>AWAITING COMMAND</td>
</tr>
<tr>
<td>0</td>
<td>9</td>
<td>recv sleep</td>
<td>joan</td>
<td>joan</td>
<td>luv2work</td>
<td>0</td>
<td>master</td>
<td>AWAITING COMMAND</td>
</tr>
<tr>
<td>0</td>
<td>10</td>
<td>running</td>
<td>foote</td>
<td>foote</td>
<td>svr47hum</td>
<td>0</td>
<td>master</td>
<td>SELECT</td>
</tr>
</tbody>
</table>

(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 them as system processes. You will always see NETWORK HANDLER, MIRROR HANDLER, HOUSEKEEPER, and CHECKPOINT SLEEP (or, rarely, CHECKPOINT). AUDIT PROCESS becomes activated if you enable auditing.

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 process whose status is “recv sleep” belongs to a user who is using Adaptive Server and may be pausing to examine the results of a command, or whether the process indicates that a user has restarted a PC or other terminal, and left a stranded process. You can learn more about a questionable process by querying the sysprocesses table for information. For example, this query shows the host process ID and client software used by process 8:

```sql
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.

### Using kill with statusonly

The `kill ...statusonly` command reports on the progress of a server process ID (spid) in rollback status. It does not terminate the spid. The `statusonly` report displays the percent of rollback completed and the estimated length of time in seconds before the rollback completes. To track the progress of a rollback, you must run `kill...with statusonly` multiple times. The syntax is:

```
kill spid with statusonly
```

Where `spid` is the number of the process you are terminating.

For example, the following reports on the process of the rollback of spid number 13:

```
kil 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 Adaptive Server is not rolling back the specified spid, `kill...statusonly` returns the following message:

```
Status report cannot be obtained. KILL spid:nn is not in progress.
```

### 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` 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, 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 Adaptive Server, see the *Performance and Tuning Series: Locking and Concurrency Control*. 
Housekeeper functionality

The housekeeper provides important functionalities:

- The housekeeper feature consists of three tasks: housekeeper wash, housekeeper garbage collection, and housekeeper chores. `sp_who` recognizes all three tasks, as the following output shows:

```
<table>
<thead>
<tr>
<th>fid</th>
<th>spid</th>
<th>status</th>
<th>logname</th>
<th>origname</th>
<th>hostname</th>
<th>blk_sp</th>
<th>block_xloid</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>8</td>
<td>sleeping</td>
<td>NULL</td>
<td>NULL</td>
<td>master</td>
<td>HK WASH</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>9</td>
<td>sleeping</td>
<td>NULL</td>
<td>NULL</td>
<td>master</td>
<td>HK GC</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>10</td>
<td>sleeping</td>
<td>NULL</td>
<td>NULL</td>
<td>master</td>
<td>HK CHORES</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>12</td>
<td>recv</td>
<td>sa</td>
<td>sa</td>
<td>chaucer</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
```

(11 rows affected, return status = 0)

- 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.

Three housekeepers

The housekeeper work is divided among three separate tasks:

- Housekeeper wash task
- Housekeeper chores task
- Housekeeper garbage collection task

The output for all three tasks appears in the output for `sp_who`. 
Housekeeper wash

Washing buffers is an optional task and runs at idle times only. 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.

Housekeeper chores

The housekeeper chores task runs at idle times only 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 chore using the configuration parameter `dtm detach timeout period`.
- Checking licence usage. You can turn this task off using the configuration parameter `license information`.

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. Performance is affected by the fragmentation of space allocated to a table, 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 that deleting transactions are committed.

Both the `delete` command and the housekeeper garbage collection task can be configured for aggressive or lazy garbage collection, through the configuration parameter `enable housekeeper GC`. 
Housekeeper functionality

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.

Running at user priority

The housekeeper garbage collection task operates at the priority level of an ordinary user, competing for CPU time with ordinary user tasks. This behavior prevents the list of empty pages from growing faster than the housekeeper can delete them.

Configuring enable housekeeper GC

To configure Adaptive Server for garbage collection task, use:

```
sp_configure "enable housekeeper GC", value
```

For example, enter:

```
sp_configure "enable housekeeper GC", 4
```

The following are the valid values for enable housekeeper GC configuration parameter:

- 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. Sybase does not recommend using 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 data-only 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.

**Using the reorg command**

Garbage collection is most effective when you set enable housekeeper GC to 4 or 5. Sybase 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.


**Configuring Adaptive Server to save SQL batch text**

Occasionally a query or procedure causes Adaptive Server Monitor to hang. Users with the system administrator role can configure Adaptive Server to grant Adaptive Server Monitor access to the text of the currently executing SQL batch. Viewing the SQL text of long-running batches helps you debug hung processes or fine-tune long statements that are heavy resource consumers.

Adaptive Server must be configured to collect the SQL batch text and write it to shared memory, where the text can be read by Adaptive Server Monitor Server (the server component of Adaptive Server Monitor). The client requests might come from Monitor Viewer, which is a plug-in to Sybase Central, or other Adaptive Server Monitor Server applications.
Configuring Adaptive Server to save SQL batch text also allows you to view the current query plan in showplan format (as you would see after setting showplan on). You can view the current query plan from within Adaptive Server; see “Viewing the query plan of a SQL statement” on page 367. SQL batches are viewable only through Adaptive Server Monitor Server. See the Adaptive Server Monitor Server documentation for more information about displaying the batch text.

Because the query or procedure you are viewing may be nested within a batch of SQL text, the sysprocesses table now includes columns for the line number, statement number and spid of a hung statement to view its query plan.

By default, Adaptive Server is not configured to save SQL batch text, so you must configure Adaptive Server to allocate memory for this feature. Adaptive Server Monitor access to SQL has no effect on performance if you have not configured any memory to save SQL batches.

Allocating memory for batch text

You can configure the amount of the SQL text batch you want to save. When text saving is enabled, Adaptive Server copies the subsequent SQL text batches to memory shared with SQL Server Monitor. Because each new batch clears the memory for the connection and overwrites the previous batch, you can view only currently executing SQL statements.

❖ Saving SQL text

1 Configure the amount of SQL text retained in memory (see “Configuring the amount of SQL text retained in memory” on page 364).

2 Enable Adaptive Server to start saving SQL text (see “Enabling Adaptive Server to start saving SQL text” on page 365).

Note You must have System Administration privileges to configure and save SQL text batches.

Configuring the amount of SQL text retained in memory

After installation, you must decide the maximum amount of SQL text that can be copied to shared memory. Consider the following to help you determine how much memory to allocate per user:
• SQL batches exceeding the allocated amount of memory are truncated without warning. If you do not allocate enough memory for the batch statements, the text you are interested in viewing might be the section of the batch that is truncated.

For example, if you configure Adaptive Server to save the amount of text designated by bracket A in the illustration, but the statement that is running occurs in the text designated by bracket B, Adaptive Server will not display the statement that is running.

• The more memory you allocate for SQL text from shared memory, the less chance the problem statement will be truncated from the batch copied to shared memory. However, Adaptive Server immediately rejects very large values because they do not leave enough memory for data and procedure caches.

Sybase recommends an initial value of 1024 bytes per user connection.

Use sp_configure with the max SQL text monitored configuration parameter to allocate shared memory, where bytes_per_connection (the maximum number of bytes saved for each client connection) is between 0 (the default) and 2,147,483,647 (the theoretical limit):

```sql
sp_configure "max SQL text monitored", bytes_per_connection
```

Since memory for SQL text is allocated by Adaptive Server at start-up, you must restart Adaptive Server for this parameter to take effect.

The total memory allocated for the SQL text from shared memory is the product of bytes_per_connection multiplied by the number of user connections.

**Enabling Adaptive Server to start saving SQL text**

After you allocate shared memory for SQL text, Adaptive Server saves a copy of each SQL batch whenever you enable an Adaptive Server Monitor event summary that includes SQL batches.

You may also have to reconfigure Adaptive Server Monitor’s event buffer scan interval for SQL text. See the Adaptive Server Monitor documentation for more information.
Configuring Adaptive Server to save SQL batch text

SQL commands not represented by text

If you use Client-Library™ functions not represented by text (such as `ct_cursor` or `ct_dynamic`) to issue SQL commands, Client-Library encodes the information for efficiency, and Adaptive Server generally decodes and displays key command information. For example, if you open a cursor with `ct_cursor` and the command is running, the Adaptive Server Monitor event summary displays the cursor name and the cursor declare statement.

Table 11-3 lists a complete list of the Client-Library functions not represented by text:

<table>
<thead>
<tr>
<th>Client-Library routine</th>
<th>DB-Library routine</th>
<th>Presentation name</th>
<th>Presentation data</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ct_cursor</code></td>
<td>N/A</td>
<td>CLOSE_CURSOR</td>
<td>Cursor name, statement</td>
</tr>
<tr>
<td><code>ct_cursor</code></td>
<td>N/A</td>
<td>DECLARE_CURSOR</td>
<td>Cursor name, statement</td>
</tr>
<tr>
<td><code>ct_cursor</code></td>
<td>N/A</td>
<td>DELETE_AT_CURSOR</td>
<td>Cursor name, statement</td>
</tr>
<tr>
<td><code>ct_cursor</code></td>
<td>N/A</td>
<td>FETCH_CURSOR</td>
<td>Cursor name, statement</td>
</tr>
<tr>
<td><code>ct_fetch</code> (when processing the results of <code>ct_cursor</code>)</td>
<td>N/A</td>
<td>FETCH_CURSOR</td>
<td>Cursor name, statement</td>
</tr>
<tr>
<td><code>ct_cursor CURSOR_ROWS, or ct_cancel when the connection has Client-Library cursors</code></td>
<td>N/A</td>
<td>CURSOR_INFO</td>
<td>Cursor name, statement</td>
</tr>
<tr>
<td><code>ct_cursor</code></td>
<td>N/A</td>
<td>OPEN_CURSOR</td>
<td>Cursor name, statement</td>
</tr>
<tr>
<td><code>ct_cursor</code></td>
<td>N/A</td>
<td>UPDATE_AT_CURSOR</td>
<td>Cursor name, statement</td>
</tr>
<tr>
<td><code>ct_command (CS_RPC_CMD) (default behavior)</code></td>
<td><code>dbrpcinit (only in version 10.0.1 or later)</code></td>
<td>DUBLIB_RPC</td>
<td>RPC name</td>
</tr>
<tr>
<td><code>ct_dynamic</code></td>
<td>N/A</td>
<td>DYNAMIC_SQL</td>
<td>Dynamic statement name, statement</td>
</tr>
<tr>
<td><code>ct_command (CS_MSG_CMD)</code></td>
<td>N/A</td>
<td>MESSAGE</td>
<td>None</td>
</tr>
<tr>
<td><code>ct_param</code></td>
<td><code>dbrpcparam</code></td>
<td>PARAM_FORMAT</td>
<td>None</td>
</tr>
<tr>
<td><code>ct_param</code></td>
<td><code>dbrpcparam</code></td>
<td>PARAMS</td>
<td>None</td>
</tr>
<tr>
<td><code>ct_command (CS_RPC_CMD) (only when a TDS version earlier than 5.0 is used)</code></td>
<td><code>dbrpcparam (in DB-Library versions earlier than 10.0.1)</code></td>
<td>RPC</td>
<td>RPC name</td>
</tr>
</tbody>
</table>
For more information about SQL commands not represented by text, see your Open Client documentation.

**Viewing the query plan of a SQL statement**

Use `sp_showplan` and the `spid` of the user connection in question to retrieve the query plan for the statement currently running on this connection. You can also use `sp_showplan` to view the query plan for a previous statement in the same batch.

The syntax is:

```sql
declare @batch int
declare @context int
declare @statement int
execute sp_showplan <spid_value>, @batch_id = @batch output,
@context_id = @context output, @stmt_num = @statement output
```

where:

- `batch_id` – is the unique number for a batch.
- `context_id` – is a unique number for every procedure (or trigger) executed in the batch.
- `stmt_num` – is the number of the current statement within a batch.

Adaptive Server uses the unique batch ID to synchronize the query plan with the batch text and other data retrieved by Adaptive Server Monitor.

---

**Note** You must be a system administrator to execute `sp_showplan`.

For example, to see the query plan for the current statement for `spid` 99, enter:

```sql
declare @batch int
declare @context int
declare @statement int
exec sp_showplan 99, @batch output, @context output, @statement output
```

You can run the query plan procedure independently of Adaptive Server Monitor, regardless of whether or not Adaptive Server has allocated shared memory for SQL text.
Viewing previous statements

To see the query plan for the previous statement in the same batch, issue `sp_showplan` with the same values as the original query, but subtract one from the statement number. Using this method, you can view all the statements in the statement batch back to query number one.

Viewing a nested procedure

Although `sp_showplan` allows you to view the query plan for the current statement, the actual statement that is running may exist within a procedure (or within a nested chain of procedures) called from the original SQL batch. Table 11-4 shows the columns in `sysprocesses` that contain information about these nested statements.

<table>
<thead>
<tr>
<th>Column</th>
<th>Datatype</th>
<th>Specifies</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Integer</td>
<td>The object ID of the running procedure (or 0 if no procedure is running)</td>
</tr>
<tr>
<td>stmtnum</td>
<td>Integer</td>
<td>The current statement number within the running procedure (or the SQL batch statement number if no procedure is running)</td>
</tr>
<tr>
<td>linenum</td>
<td>Integer</td>
<td>The line number of the current statement within the running stored procedure (or the line number of the current SQL batch statement if no procedure is running)</td>
</tr>
</tbody>
</table>

This information is saved in `sysprocesses`, regardless of whether SQL text is enabled or any memory is allocated for SQL text.

To display the `id`, `stmtnum`, and `linenum` columns, enter:

```sql
select id, stmtnum, linenum
from sysprocesses
where spid = spid_of_hung_session
```

**Note** You do not need the `sa_role` to run this `select` statement.

Shutting down servers

A system administrator can shut down Adaptive Server or Backup Server with the `shutdown` command. The syntax is:

```sql
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.

Shutting down Adaptive Server

If you do not provide a server name, shutdown shuts down the Adaptive Server you are using. When you issue a shutdown command, Adaptive Server:

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

In this way, shutdown minimizes the amount of work that automatic recovery must do when you restart Adaptive Server.

The with nowait option shuts down Adaptive Server 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.

Shutting down a Backup Server

To shut down a Backup Server, give the Backup Server’s name:

```
shutdown SYB_BACKUP
```

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 Adaptive Server, 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 Adaptive Server, and
- Listed in your local interfaces file.

Use sp_addserver to add a Backup Server to sysservers.
Learning about known problems

Checking for active dumps and loads

To see the activity on your Backup Server before executing a shutdown command, run sp_who on the 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
```

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.

Learning about known problems

The release bulletin is a valuable resource for learning about known problems or incompatibilities with Adaptive Server and Backup Server. Reading the release bulletin in advance can save you the time and guesswork of troubleshooting known problems.
The Adaptive Server installation program also installs files that list all system problem reports (SPRs) and closed problem reports (CPRs) for Adaptive Server. Problem reports are organized by functional areas of the product. For example, a file named \texttt{cpr\_bus} would contain a listing of closed (fixed) problem reports pertaining to the Backup Server, and the file \texttt{spr\_bus} would contain a list of currently open problem reports for the Backup Server.

See the release bulletin to learn the location of CPR and SPR files.
Learning about known problems
The following chapters discuss security administration in Adaptive Server:


- Chapter 15, “Managing Remote Servers,” discusses the steps the system administrator and system security officer of each Adaptive Server must execute to enable remote procedure calls (RPCs).

- Chapter 16, “External Authentication,” describes the network-based security services that enable you to authenticate users and protect data transmitted among machines on a network.

- Chapter 17, “Managing User Permissions,” describes the use and implementation of user permissions.

- Chapter 18, “Auditing,” describes how to set up auditing for your installation.

- Chapter 19, “Confidentiality of Data,” how to configure Adaptive Server to ensure that all data is secure and confidential.
CHAPTER 12

Introduction to Security

This chapter provides an overview of the security features available in Adaptive Server.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction to security</td>
<td>375</td>
</tr>
<tr>
<td>What is “information security?”</td>
<td>376</td>
</tr>
<tr>
<td>Information security standards</td>
<td>377</td>
</tr>
</tbody>
</table>

Introduction to security

Information is an asset to your company, and possibly your company's greatest asset. Information needs protection just like any other asset. Your company locks its doors at the end of the day, allowing only employees with a key to the building to enter. Similarly, your company needs to determine how best to protect the information contained in the databases, and who has access to the information.

In the past, organizations relied on physical separation and dedicated systems to ensure that sensitive information did not fall into the wrong hands. However, this approach is inadequate because of significant hardware and software costs, and the inability to meet operational requirements. As a counter-measure, individual database servers need strong, yet flexible, security support.

Users and the data they access can be anywhere in the world, connected by untrusted networks, and ensuring the confidentiality and integrity of sensitive data and transactions in this environment is critical. The same systems that allow users to access the data from anywhere in the world also open up the information for users who should not have access to the information.

Information is only useful if it gets to the people who need it, when they need it, regardless of where they are. With complex and dynamically changing business relationships, it is critical that information gets only to authorized users.
What is “information security?”

It is important for your organization to determine what “information security” means to the organization. This is not a one-size-fits-all concept. One organization’s acceptable level of security could be another organization’s worst nightmare. Although everybody has different definitions, these are guidelines for considering security:

- Sensitive information should be kept confidential – you need to determine who should have access to what information.
- The system should enforce integrity – the server should enforce the rules and constraints to ensure the information remains accurate and complete.
- The information should be available – even with all the safeguards in place, anybody who needs access to the information should have it available when the information is needed.

You should identify where your organization’s security requirements originate from. That is, what is it that your organization wants to protect and what does the outside world require of your organization:

- Identify the information assets and the security risks associated with them if they become vulnerable or compromised.
- Identify and understand any laws, statutes, regulations, and contractual agreements that apply to your organization and the information assets.
- Identify your organization’s business processes and the requirements they impose on information assets, to balance practical considerations with the security risks.

Remember that these requirements can change over time. You will probably have to revisit and reassess the security requirements to make sure they still reflect your organization’s needs.

After you and your organization determine what information security means, you must set up a series of controls and policies that meet the company’s security objectives. One desirable outcome of these efforts is an information security policy document that clarifies decisions made for information security.

For more information about security features in Adaptive Server, see Chapter 13, “Getting Started With Security Administration in Adaptive Server.”
Information security standards

Over the years Adaptive Server has been certified to various security standards, including the Common Criteria certification. Adaptive Server release 15.0 uses FIPS-140-2 certified modules for SSL encryption.

This section describes these certifications.

Adaptive Server version 15.0.1 available for Common Criteria configuration

Adaptive Server version 15.0.1 is available for the Common Criteria configuration (called the Evaluated Configuration). The Evaluated Configuration consists of Adaptive Server version 15.0.1 with the security and directory services options. Adaptive Server's evaluation was carried out in accordance with the Common Criteria Evaluation and Validation Scheme (CCEVS) process and scheme. The criteria against which the Adaptive Server Enterprise Target of Evaluation (TOE) was judged are described in the Common Criteria for Information Technology Security Evaluation, Version 2.3 and International Interpretations effective on August, 2005. If you configure Adaptive Server as specified in the Supplement for Installing Adaptive Server for Common Criteria Configuration, Adaptive Server satisfies all of the security functional requirements stated in the Sybase Adaptive Server Enterprise Security Target (Version 1.5).

Adaptive Server supports eight security functions:

- Cryptographic support – Adaptive Server supports transparent encryption of data at the column level. SQL statements and extensions provide secure key management.
- Security audit – an audit mechanism that checks access, authentication attempts, and administrator functions. The security audit records the date, time, responsible individual and other details describing the event in the audit trail.
- User data protection – Adaptive Server implements the discretionary access control policy over applicable database objects: databases, tables, views, stored procedures, and encryption keys.
- Identification and authentication – Adaptive Server provides its own identification and authentication mechanism in addition to the underlying operating system mechanism.
Information security standards

- Security management – functions that allow you to manage users and associated privileges, access permissions, and other security functions such as the audit trail. These functions are restricted based on discretionary access control policy rules, including role restrictions.

- Protection of the TSF – Adaptive Server protects itself by keeping its context separate from that of its users and by using operating system mechanisms to ensure that memory and files used by Adaptive Server have the appropriate access settings. Adaptive Server interacts with users through well-defined interfaces designed to ensure that its security policies are enforced.

- Resource utilization – Adaptive Server provides resource limits to prevent queries and transactions from monopolizing server resources.

- TOE access – Adaptive Server allows authorized administrators to construct login triggers that restrict logins to a specific number of sessions and restrict access based on time. Authorized administrators can also restrict access based on user identities.

Adaptive Server 15.0 contains all of the security features included in Adaptive Server version 12.5.2 and additional new security features. The additional security features are listed in What's New in Adaptive Server 15.0?

FIPS 140-2 Validated cryptographic module

SSL is the standard for securing the transmission of sensitive information, such as credit card numbers, stock trades, and banking transactions over the Internet. It relies on public key cryptography. SSL implementation uses FIPS 140-2 validated level 1 cryptographic modules using Certicom Security Builder GSE for products running on Windows, Solaris, AIX and HPUX operating systems. For more information, see validation certificate #542, dated June 2, 2005 at NIST website, http://csrc.nist.gov.

Adaptive Server encrypted columns rely on symmetric key cryptography, and use the same FIPS 140-2 validated cryptographic modules as SSL. For more information, see the User Guide for Encrypted Columns.
CHAPTER 13

Getting Started With Security Administration in Adaptive Server

This chapter provides an overview of the security features available in Adaptive Server.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>General process of security administration</td>
<td>379</td>
</tr>
<tr>
<td>Recommendations for setting up security</td>
<td>381</td>
</tr>
<tr>
<td>An example of setting up security</td>
<td>382</td>
</tr>
<tr>
<td>Discretionary access controls</td>
<td>385</td>
</tr>
<tr>
<td>Introduction to Security Features in Adaptive Server</td>
<td>383</td>
</tr>
<tr>
<td>Identification and authentication</td>
<td>384</td>
</tr>
<tr>
<td>External authentication</td>
<td>385</td>
</tr>
<tr>
<td>Managing remote servers</td>
<td>385</td>
</tr>
<tr>
<td>Discretionary access controls</td>
<td>385</td>
</tr>
<tr>
<td>Division of roles</td>
<td>387</td>
</tr>
<tr>
<td>Auditing for accountability</td>
<td>388</td>
</tr>
<tr>
<td>Confidentiality of data</td>
<td>389</td>
</tr>
</tbody>
</table>

General process of security administration

Table 13-1 describes the major tasks that are required to administer Adaptive Server in a secure manner and refers you to the documentation that contains the instructions for performing each task.
### General process of security administration

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Install Adaptive Server, including auditing.</td>
<td>This task includes preparing for installation, loading files from your distribution medium, performing the actual installation, and administering the physical resources that are required.</td>
<td>The installation documentation for your platform and Chapter 18, “Auditing.”</td>
</tr>
<tr>
<td>2. Set up a secure administrative environment.</td>
<td>This includes enabling auditing, granting roles to individual users to ensure individual accountability, assigning login names to system administrators and system security officers and establishing password and login policies.</td>
<td>Chapter 14, “Managing Adaptive Server Logins, Database Users, and Client Connections”</td>
</tr>
<tr>
<td>3. Add user logins to the server; add users to databases; establish groups and roles; set proxy authorization.</td>
<td>Add logins, create groups, add users to databases, drop and lock logins, and assign initial passwords. Assign roles to users, create user-defined roles, and define role hierarchies and mutual exclusivity of roles.</td>
<td>Chapter 14, “Managing Adaptive Server Logins, Database Users, and Client Connections”</td>
</tr>
<tr>
<td>4. Administer permissions for users, groups, and roles.</td>
<td>Grant and revoke permissions for certain SQL commands, executing certain system procedures, and accessing databases, tables, particular table columns, and views. Create access rules to enforce fine-grained access control.</td>
<td>Chapter 17, “Managing User Permissions”</td>
</tr>
<tr>
<td>5. Configure encryption in your database to encrypt sensitive data in tables. Encrypt sensitive data.</td>
<td>Configure Adaptive Server to use column-level encryption, decide which columnar data to encrypt, perform a one-time key creation operation, use alter table to perform initial data encryption.</td>
<td>User Guide for Encrypted Columns</td>
</tr>
<tr>
<td>7. Set up and maintain auditing.</td>
<td>Determine what is to be audited, audit the use of Adaptive Server, and use the audit trail to detect penetration of the system and misuse of resources.</td>
<td>Chapter 18, “Auditing,” and the Adaptive Server installation and configuration documentation for your platform</td>
</tr>
<tr>
<td>8. Set up your installation for advanced authentication mechanisms and network security.</td>
<td>Configure the server to use services, such as LDAP, PAM, or Kerberos- based User Authentication, data confidentiality with encryption, data integrity..</td>
<td>Chapter 16, “External Authentication” and Chapter 19, “Confidentiality of Data.”</td>
</tr>
</tbody>
</table>
Recommendations for setting up security

The following describes logins and how they relate to security.

Using the “sa” login

When Adaptive Server is installed, a single login called “sa” is configured with the system administrator and system security officer roles. This means that the “sa” login has unlimited power.

Use the “sa” login only during initial setup. Instead of allowing several users to use the “sa” account, establish individual accountability by assigning specific roles to individual administrators.

Changing the “sa” login password

The “sa” login is configured initially with a “NULL” password. Use `sp_password` to change the password immediately after installation.

Warning! When logging in to Adaptive Server, do not use the -P option of `isql` to specify your password because another user may have an opportunity to see it.

When to enable auditing

Enable auditing early in the administration process so that you have a record of privileged commands that are executed by system security officers and system administrators. You might also want to audit commands that are executed by those with other special roles, such as operators when they dump and load databases.
Assigning login names

Assign Adaptive Server login names that are the same as their respective operating system login names. This makes logging in to Adaptive Server easier, simplifies management of server and operating system login accounts, and makes it easier to correlate the audit data generated by Adaptive Server with that of the operating system.

An example of setting up security

Suppose you have decided to assign special roles to the users listed in Table 13-2.

Table 13-2: Users to whom you will assign roles

<table>
<thead>
<tr>
<th>Name</th>
<th>Privilege</th>
<th>Operating system login name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rajnish Smith</td>
<td>sso_role</td>
<td>rsmith</td>
</tr>
<tr>
<td>Catherine Macar-Swan</td>
<td>sa_role</td>
<td>cmacar</td>
</tr>
<tr>
<td>Soshi Ikedo</td>
<td>sa_role</td>
<td>sikedo</td>
</tr>
<tr>
<td>Julio Rozanski</td>
<td>oper_role</td>
<td>jrozan</td>
</tr>
<tr>
<td>Alan Johnson</td>
<td>dbo</td>
<td>ajohnson</td>
</tr>
</tbody>
</table>

Table 13-3 shows the sequence of commands you might use to set up a secure operating environment for Adaptive Server, based upon the role assignments shown in Table 13-2. After logging in to the operating system, you would issue these commands using the initial “sa” account.

Table 13-3: Examples of commands used to set up security

<table>
<thead>
<tr>
<th>Commands</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>isql -Usa</td>
<td>Logs in to Adaptive Server as “sa”. Both sa_role and sso_role are active.</td>
</tr>
<tr>
<td>sp_audit security, “all”, “all”, “on”</td>
<td>Sets auditing options for server-wide, security-relevant events and the auditing of all actions that have sa_role or sso_role active.</td>
</tr>
<tr>
<td>sp_audit “all”, “sa_role”, “all”, “on”</td>
<td></td>
</tr>
<tr>
<td>sp_audit “all”, “sso_role”, “all”, “on”</td>
<td></td>
</tr>
<tr>
<td>sp_configure “auditing”, 1</td>
<td>Enables auditing.</td>
</tr>
</tbody>
</table>

Note: Before you enable auditing, set up a threshold procedure for the audit trail and determine how to handle the transaction log in sybsecurity. For details, see Chapter 18, “Auditing.”

- sp_addlogin rsmith, js&2P3d, @fullname = “Rajnish Smith”
  Adds logins and passwords for Rajnish, Catherine, Soshi, and Julio.
CHAPTER 13  Getting Started With Security Administration in Adaptive Server

Introduction to Security Features in Adaptive Server

Table 13-4 describes the security features in Adaptive Server.

<table>
<thead>
<tr>
<th>Security feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification and authentication controls</td>
<td>Ensures that only authorized users can log into the system. In addition to password based login authentication, Adaptive Server supports external authentication using Kerberos, LDAP, or PAM.</td>
</tr>
</tbody>
</table>
Identification and authentication

<table>
<thead>
<tr>
<th>Security feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discretionary Access Controls (DAC)</td>
<td>Provides access controls that give object owners the ability to restrict access to objects, usually with the grant and revoke commands. This type of control is dependent upon an object owner's discretion.</td>
</tr>
<tr>
<td>Division of roles</td>
<td>Allows an administrator to grant privileged roles to specified users so only designated users can perform certain tasks. Adaptive Server has predefined roles, called “system roles,” such as system administrator and system security officer. In addition, Adaptive Server allows system security officers to define additional roles, called “user-defined roles.”</td>
</tr>
<tr>
<td>Accountability</td>
<td>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. Adaptive Server also provides a single option to audit a set of server-wide security-relevant events.</td>
</tr>
<tr>
<td>Confidentiality of data</td>
<td>Maintains confidentiality of data using encryption for Client-Server communications, available with Kerberos or SSL. Column-level encryption preserves confidentiality of data stored in the database. Data that is not active is kept confidential with password-protected database backup.</td>
</tr>
</tbody>
</table>

Identification and authentication

Adaptive Server uses the Server User Identity (SUID) to uniquely identify a user with a login account name. This identity is linked to a particular User Identity (UID) in each database. Access controls use the identity when determining whether to allow access for the user with this SUID to an object. Authentication verifies that a user is actually the person they claim to be. Adaptive Server allows both internal and external mechanisms for authentication.

Identification and authentication controls are discussed in Chapter 14, “Managing Adaptive Server Logins, Database Users, and Client Connections.” In addition, see “Using proxy authorization” on page 591 and Chapter 15, “Managing Remote Servers.”
External authentication

Security is often enhanced in large, heterogeneous applications by authenticating logins with a central repository. Adaptive Server supports a variety of external authentication methods:

- **Kerberos** – provides a centralized and secure authentication mechanism in enterprise environments that includes the Kerberos infrastructure. Authentication occurs with a trusted, third-party server called a Key Distribution Center (KDC) to verify both the client and the server.

- **LDAP User Authentication** – Lightweight Directory Access Protocol (LDAP) provides a centralized authentication mechanism based on a user’s login name and password.

- **PAM User Authentication** – Pluggable Authentication Module (PAM) provides a centralized authentication mechanism that uses interfaces provided by the operating system for both administration and runtime application operations.

For more information about each of these methods of external authentication, see Chapter 16, “External Authentication.”

Managing remote servers

Internal mechanisms for administering logins and users between Adaptive Servers are described in Chapter 15, “Managing Remote Servers.”

Discretionary access controls

Owners of objects can grant access to those objects to other users. Object owners can also grant other users the ability to pass the access permission to other users. With Adaptive Server’s discretionary access controls, you can give various kinds of permissions to users, groups, and roles with the `grant` command. Use the `revoke` command to rescind these permissions. The `grant` and `revoke` commands give users permission to execute specified commands and to access specified tables, procedures, views, encryption keys, and columns.
Some commands can be used at any time by any user, with no permission required. Others can be used only by users of a certain status such as a system administrator and are not transferable.

The ability to assign permissions for the commands that can be granted and revoked is determined by each user’s status (as system administrator, system security officer, database owner, or database object owner), and by whether or not a particular user has been granted a permission with the option to grant that permission to other users.

Discretionary access controls are discussed in Chapter 17, “Managing User Permissions.”

Row-level access control

The row-level access control provides a powerful and flexible means of protecting data, down to the row level. Administrators define access rules that are based on the value of individual data elements, and the server enforces these rules transparently. Once an administrator defines an access rule, it is automatically invoked whenever the affected data is queried through applications, ad hoc queries, stored procedures, views, and so on.

Using a rule-based access control simplifies both the security administration of an Adaptive Server installation and the application development process because it is the server, not the application, that enforces security. This allows developers to concentrate on implementing business functionality while administrators focus on defining a security policy to enforce consistently across the entire server. These are the features that allow you to implement row-level access control:

- Access Rules
- Application Context Facility
- Login Triggers
- Domain Integrity Rules

For more information on how to implement row-level access controls, see “Using row-level access control” on page 607.
Division of roles

An important feature in Adaptive Server is the division of roles. The roles supported by Adaptive Server enable you to enforce and maintain individual accountability. Adaptive Server provides system roles, such as system administrator and system security officer, and user-defined roles, which are created by a system security officer.

Roles provide individual accountability for users performing operational and administrative tasks. Their actions can be audited and attributed to them.

Role hierarchy

A system security officer can define role hierarchies such that if a user has one role, the user automatically has roles lower in the hierarchy. For example, the “chief_financial_officer” role might contain both the “financial_analyst” and the “salary_administrator” roles. The Chief Financial Officer can perform all tasks and see all data that can be viewed by the Salary Administrators and Financial Analysts.

Mutual exclusivity

Two roles can be defined to be mutually exclusive for:

- Membership – a single user cannot be granted both roles. For example, an installation might not want a single user to have both the “payment_requestor” and “payment_approver” roles to be granted to the same user.

- Activation – a single user cannot activate, or enable, both roles. For example, a user might be granted both the “senior_auditor” and the “equipment_buyer” roles, but the installation may not want to permit the user to have both roles enabled at the same time.

System roles, as well as user-defined roles, can be defined to be in a role hierarchy or to be mutually exclusive. For example, you might want a “super_user” role to contain the system administrator, Operator, and Tech Support roles. In addition, you might want to define the system administrator and system security officer roles to be mutually exclusive for membership; that is, a single user cannot be granted both roles.
See “Creating and assigning roles to users” on page 403 for information on administering and using roles.

Auditing for accountability

Adaptive Server includes a comprehensive audit system. The audit system consists of a system database called sybsecurity, configuration parameters for managing auditing, a system procedure, sp_audit, to set all auditing options, and a system procedure, sp_addauditrecord, to add user-defined records to the audit trail. When you install auditing, you can specify the number of audit tables that Adaptive Server will use for the audit trail. If you use two or more tables to store the audit trail, you can set up a smoothly running audit system with no manual intervention and no loss of records.

A system security officer manages the audit system and is the only user who can start and stop auditing, set up auditing options, and process the audit data. As a system security officer, you can establish auditing for events such as:

- Server-wide, security-relevant events
- Creating, dropping, and modifying database objects
- All actions by a particular user or all actions by users with a particular role active
- Granting or revoking database access
- Importing or exporting data
- Logins and logouts
- All actions related to encryption keys

Auditing functionality is discussed in Chapter 18, “Auditing.”
Confidentiality of data

Adaptive server allows you to maintain the confidentiality of data by encrypting client-server communications using the secure socket layer (SSL) standard or using Kerberos. You can also protect the confidentiality of data at rest by using column-level encryption in the database and encrypting backups for offline data.

For more information

- SSL – Chapter 19, “Confidentiality of Data”
- Kerberos – Chapter 16, “External Authentication”
- Encrypted columns – User Guide to Encrypted Columns

Password-Protected Database Backup

dump and load database include a password parameter that allows you to password-protect your database dumps. For more information, see Reference Manual: Commands and Chapter 12, “Backing Up and Restoring User Databases,” in Volume 2 of the System Administration Guide.
CHAPTER 14
Managing Adaptive Server Logins, Database Users, and Client Connections

This chapter describes how to manage Adaptive Server login accounts and database users.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>392</td>
</tr>
<tr>
<td>Choosing and creating a password</td>
<td>393</td>
</tr>
<tr>
<td>Adding logins to Adaptive Server</td>
<td>394</td>
</tr>
<tr>
<td>Login failure to Adaptive Server</td>
<td>395</td>
</tr>
<tr>
<td>Creating groups</td>
<td>396</td>
</tr>
<tr>
<td>Adding users to databases</td>
<td>396</td>
</tr>
<tr>
<td>Number of user and login IDs</td>
<td>400</td>
</tr>
<tr>
<td>Creating and assigning roles to users</td>
<td>403</td>
</tr>
<tr>
<td>Dropping users, groups, and user-defined roles</td>
<td>416</td>
</tr>
<tr>
<td>Locking or dropping Adaptive Server login accounts</td>
<td>418</td>
</tr>
<tr>
<td>Changing user information</td>
<td>420</td>
</tr>
<tr>
<td>Using aliases in databases</td>
<td>425</td>
</tr>
<tr>
<td>Getting information about users</td>
<td>428</td>
</tr>
<tr>
<td>Establishing a password and login policy</td>
<td>436</td>
</tr>
<tr>
<td>Monitoring license use</td>
<td>477</td>
</tr>
<tr>
<td>Getting information about usage: chargeback accounting</td>
<td>480</td>
</tr>
</tbody>
</table>
Overview

The responsibility of adding new logins to Adaptive Server, adding users to databases, and granting them permission to use commands and database objects is divided among the system security officer, system administrator, and database owner.

Note See “Adding new users” on page 392 for information about creating login accounts for a particular server using sp_addlogin, which stores account information in the syslogins table on that server. You can also create and store login accounts on a LDAP server.

❖ Adding new users

1. A system security officer uses sp_addlogin to create a server login account for a new user.

2. A system administrator or database owner uses sp_adduser to add a user to a database or assign a user to a group. For more information, see “Creating groups” on page 396. You can give a user access to a database using an alias. See “Adding aliases” on page 426.

3. A System Security officer grants specific roles to the user.

4. A system administrator, database owner, or object owner grants the user or group specific permissions on specific commands and database objects. Users or groups can also be granted permission to grant specific permissions on objects to other users or groups. See Chapter 17, “Managing User Permissions” for detailed information about permissions.

Table 14-1 summarizes the system procedures and commands used for these tasks.

<table>
<thead>
<tr>
<th>Task</th>
<th>Required role</th>
<th>Command or procedure</th>
<th>Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create new logins, assign passwords, default databases, default language, and full name</td>
<td>System security officer</td>
<td>sp_addlogin</td>
<td>Any database</td>
</tr>
<tr>
<td>Create groups</td>
<td>Database owner or system administrator</td>
<td>sp_addgroup</td>
<td>User database</td>
</tr>
<tr>
<td>Create and assign roles</td>
<td>System security officer</td>
<td>create role, grant role</td>
<td>Master database</td>
</tr>
</tbody>
</table>
Choosing and creating a password

Your password helps prevent access by unauthorized people. The system security officer assigns each user a password when adding the user as a login to Adaptive Server. Users can modify their passwords at any time using sp_password. For directions on changing a password, see “Changing passwords” on page 421.

When you create your password, follow these guidelines:

- Do not use information such as your birthday, street address, or any other word or number that has anything to do with your personal life.
- Do not use names of pets or loved ones.
- Do not use words that appear in the dictionary or words spelled backwards.

The most difficult passwords to guess are those that combine uppercase and lowercase letters and numbers. Never give anyone your password, and never write it down where anyone can see it.

Follow these rules to create a password:

- Passwords must be at least 6 bytes long.
- Passwords can consist of any printable letters, numbers, or symbols.
- A password must be enclosed in quotation marks in sp_addlogin if it:
  - Includes any character other than A–Z, a–z, 0–9, _, #, valid single-byte or multibyte alphabetic characters, or accented alphabetic characters
  - Begins with a number 0–9
Adding logins to Adaptive Server

For information about enforcing strong passwords, see “Password complexity checks” on page 445.

Adding logins to Adaptive Server

Use `sp_addlogin` to add a new login name to Adaptive Server. You do not use it to give the user permission to access user databases. Use `sp_adduser` for that purpose. Only the system security officer can execute `sp_addlogin`.

See the Adaptive Server Commands and Reference: System Procedures for the `sp_addlogin` syntax.

The following statement sets up an account for the user “maryd” with the password “100cents,” the default database (master), the default language, and no full name:

```
sp_addlogin "maryd", "100cents"
```

The password requires quotation marks because it begins with 1.

After this statement is executed, “maryd” can log in to Adaptive Server. She is automatically treated as a “guest” user in master, with limited permissions, unless she has been specifically given access to master.

The following statement sets up a login account (“omar_khayyam”) and password (“rubaiyat”) and makes pubs2 the default database for this user:

```
sp_addlogin omar_khayyam, rubaiyat, pubs2
```

To specify a full name for a user and use the default database and language, specify `null` in place of the `defdb` and `deflanguage` parameters. For example:

```
sp_addlogin omar, rubaiyat, null, null, "Omar Khayyam"
```

Alternatively, you can specify a parameter name, in which case you do not have to specify all the parameters. For example:

```
sp_addlogin omar, rubaiyat, @fullname = "Omar Khayyam"
```

When you execute `sp_addlogin`, Adaptive Server adds a row to `master.dbo.syslogins`, assigns a unique user ID (suid) for the new user, and fills in other information. When a user logs in, Adaptive Server looks in `syslogins` for the name and password provided by the user. The password column is encrypted with a one-way algorithm so it is not human-readable.
At login creation, the `crdate` column in `syslogins` is set to the current time.
The `suid` column in `syslogins` uniquely identifies each user on Adaptive Server.
A user’s `suid` remains the same, no matter what database he or she is using. The `suid` 1 is always assigned to the default “sa” account that is created when Adaptive Server is installed. Other users’ server user IDs are integers assigned consecutively by Adaptive Server each time `sp_addlogin` is executed.

Login failure to Adaptive Server

Adaptive Server must successfully authenticate a user before they are able to access data in Adaptive Server. If the authentication attempt fails, Adaptive Server returns the following message and the network connection is terminated:

```
isql -U bob -P badpass
Msg 4002, Level 14, State 1:
Server 'ACCOUNTING'
Login failed.
CT-LIBRARY error:
ct_connect(): protocol specific layer: external error:
The attempt to connect to the server failed
```

This message is a generic login failure message that does not tell the connecting user whether the failure resulted from a bad user name or a bad password. This generic message guards against malicious attempts to gain access to Adaptive Server.

Although the client sees a generic message for a login failure to avoid giving information to a malicious user, the system administrator may find the reason for the failure to be important to help detect intrusion attempts and diagnose user authentication problems.

Adaptive Server provides the reason for the login failure in the `Errornumber.Severity.State` of the `Other Information` section of `sysaudits.extrainfo` column. Login failure audits have event number 45 and `eventmod 2`.

Set the `sp_audit login` parameter to `on` or `fail` to enable auditing for login failure:

```
sp_audit "login", "all", "all", "fail"
sp_audit "login", "all", "all", "on"
```

For more information, see “Auditing login failures.”
Creating groups

Groups provide a convenient way to grant and revoke permissions to more than one user in a single statement. Groups enable you to provide a collective name to a group of users. They are especially useful if you administer an Adaptive Server installation that has a large numbers of users. Every user is a member of the group “public” and can also be a member of one other group. (Users remain in “public,” even when they belong to another group.)

It is probably most convenient to create groups before adding users to a database, since sp_adduser can assign users to groups as well as add them to the database.

You must have the system administrator or system security officer role, or be the database owner to create a group with sp_addgroup. The syntax is:

```
sp_addgroup grpname
```

The group name, a required parameter, must follow the rules for identifiers. The system administrator, system security officer, or the database owner can assign or reassign users to groups with sp_changeuser.

To set up the Senior Engineering group, use the following command while using the database to which you want to add the group:

```
sp_addgroup senioreng
```

sp_addgroup adds a row to sysusers in the current database. Therefore, each group in a database, as well as each user, has an entry in sysusers.

Adding users to databases

The database owner or a system administrator can use sp_adduser to add a user to a specific database. The user must already have an Adaptive Server login. The syntax is:

```
sp_adduser loginame [, name_in_db [, grpname]]
```

where:

- `loginame` – is the login name of an existing user.
- `name_in_db` – specifies a name that is different from the login name by which the user is to be known inside the database.
You can use this feature to accommodate users’ preferences. For example, if there are five Adaptive Server users named Mary, each must have a different login name. Mary Doe might log in as “maryd”, Mary Jones as “maryj”, and so on. However, if these users do not use the same databases, each might prefer to be known simply as “mary” inside a particular database.

If no name_in_db parameter is given, the name inside the database is the same as loginame.

Note This capability is different from the alias mechanism described in “Using aliases in databases” on page 425, which maps the identity and permissions of one user to another.

- grpname – is the name of an existing group in the database. If you do not specify a group name, the user is made a member of the default group “public.” Users remain in “public” even if they are a member of another group. See “Changing a user’s group membership” on page 423 for information about modifying a user’s group membership.

sp_adduser adds a row to the sysusers system table in the current database. When a user has an entry in the sysusers table of a database, he or she:

- Can issue use database_name to access that database
- Will use that database by default, if the default database parameter was issued as part of sp_addlogin
- Can use sp_modifylogin to make that database the default

This example shows how a database owner could give access permission to “maryh” of the engineering group “eng,” which already exists:

```
sp_adduser maryh, mary, eng
```

This example shows how to give “maryd” access to a database, keeping her name in the database the same as her login name:

```
sp_adduser maryd
```

This example shows how to add “maryj” to the existing “eng” group, keeping her name in the database the same as her login name by using null in place of a new user name:

```
sp_adduser maryj, null, eng
```
Adding users to databases

Users who have access to a database still need permissions to read data, modify data, and use certain commands. These permissions are granted with the grant and revoke commands, discussed in Chapter 17, “Managing User Permissions.”

Adding a “guest” user to a database

Creating a user named “guest” in a database enables any user with an Adaptive Server account to access the database as a guest user. If a user who has not been added to the database as a user or an aliased user issues the use database_name command, Adaptive Server looks for a guest user. If there is one, the user is allowed to access the database, with the permissions of the guest user.

The database owner can add a guest entry to the sysusers table of the database with sp_adduser:

```
sp_adduser guest
```

The guest user can be removed with sp_dropuser, as discussed in “Dropping users” on page 417.

If you drop the guest user from the master database, server users who have not yet been added to any databases cannot log in to Adaptive Server.

Note Although more than one individual can be a guest user in a database, Adaptive Server can still use the user’s server user ID, which is unique within the server, to audit each user’s activity. For more information about auditing, see Chapter 18, “Auditing.”

“guest” user permissions

“guest” inherits the privileges of “public.” The database owner and the owners of database objects can use grant and revoke to make the privileges of “guest” either more or less restrictive than those of “public.” See Chapter 17, “Managing User Permissions,” for a description of the “public” privileges.

When you install Adaptive Server, master..sysusers contains a guest entry.
“guest” user in user databases

In user databases, the database owner adds a guest user that permits all Adaptive Server users to use that database. This saves the owner from having to use `sp_adduser` to explicitly name each one as a database user.

You can use the guest mechanism to restrict access to database objects while allowing access to the database.

For example, the owner of the `titles` table could grant `select` permission on `titles` to all database users except “guest” by executing these commands:

```
grant select on titles to public
sp_adduser guest
revoke all on titles from guest
```

“guest” user in installed system databases

Adaptive Server creates the system `tempdb` database and user-created temporary databases with a guest user. Temporary objects and other objects created in `tempdb` are automatically owned by user “guest.” `sybsystemprocs`, `sybsystemdb`, and `sybsyntax` databases automatically include the “guest” user.

“guest” user in `pubs2` and `pubs3`

The “guest” user entry in the sample databases allows new Adaptive Server users to follow the examples in the `Transact-SQL User’s Guide`. The guest is given a wide range of privileges, including:

- `select` permission and data modification permission on all of the user tables
- `execute` permission on all of the procedures
- `create table`, `create view`, `create rule`, `create default`, and `create procedure` permissions
Adding a guest user to the server

The system security officer can use `sp_addlogin` to enter a login name and password that visiting users are instructed to use. Typically, such users are granted restricted permissions. A default database may be assigned.

Warning! A visitor user account is not the same as the “guest” user account. All users of the visitor account have the same server user ID; therefore, you cannot audit individual activity. Each “guest” user has a unique server ID, so you can audit individual activity and maintain individual accountability. Setting up a visitor account to be used by more than one user is not recommended because you lose individual accountability.

You can add a visitor user account named “guest” to `master..syslogins` using `sp_addlogin`. This “guest” user account takes precedence over the system “guest” user account. If you add a visitor user named “guest” with `sp_adduser`, this impacts system databases such as `sybsystemprocs` and `sybsystemdb`, which are designed to work with system “guest” user in them.

Adding remote users

You can allow users on another Adaptive Server to execute stored procedures on your server by enabling remote access. Working with the system administrator of the remote server, you can also allow users of your server to execute remote procedure calls to the remote server.

To enable remote procedure calls, both the local and the remote server must be configured. For information about setting up remote servers and adding remote users, see Chapter 15, “Managing Remote Servers.”

Number of user and login IDs

Adaptive Server supports over 2,000,000,000 logins per server and users per database. Adaptive Server uses negative numbers as well as positive numbers to increase the range of possible numbers available for IDs.
Limits and ranges of ID numbers

Table 14-2 describes the valid ranges for the ID types.

<table>
<thead>
<tr>
<th>ID type</th>
<th>Server limits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logins per server (suid)</td>
<td>2 billion plus 32K</td>
</tr>
<tr>
<td>Users per database (uid)</td>
<td>2 billion less 1032193</td>
</tr>
<tr>
<td>Groups or roles per database (gid)</td>
<td>16,384 to 1,048,576</td>
</tr>
</tbody>
</table>

Figure 14-1 illustrates the limits and ranges for logins, users, and groups.

Login connection limitations

Although Adaptive Server allows you to define over 2,000,000,000 logins per server, the actual number of users that can connect to Adaptive Server at one time is limited by the:

- Value of the number of user connections configuration parameter, and
Number of user and login IDs

- Number of file descriptors available for Adaptive Server. Each login uses one file descriptor for the connection.

Note: The maximum number of concurrent tasks running on the server is still 32,000.

Allowing the maximum number of logins and simultaneous connections

1. Configure the operating system on which Adaptive Server is running for at least 32,000 file descriptors.
2. Set the value of `number of user connections` to at least 32,000.

Note: Before Adaptive Server can have more than 64K logins and simultaneous connections, you must first configure the operating system for more than 64K file descriptors. See your operating system documentation for information about increasing the number of file descriptors.

Viewing server limits for logins, users, and groups

Table 14-3 lists the global variables for the server limits of logins, users, and groups:

<table>
<thead>
<tr>
<th>Name of variable</th>
<th>What it displays</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>@@invaliduserid</td>
<td>Invalid user ID</td>
<td>-1</td>
</tr>
<tr>
<td>@@minuserid</td>
<td>Lowest user ID</td>
<td>-32768</td>
</tr>
<tr>
<td>@@guestuserid</td>
<td>Guest user ID</td>
<td>2</td>
</tr>
<tr>
<td>@@mingroupid</td>
<td>Lowest group or role user ID</td>
<td>16384</td>
</tr>
<tr>
<td>@@maxgroupid</td>
<td>Highest group or role user ID</td>
<td>1048576</td>
</tr>
<tr>
<td>@@maxuserid</td>
<td>Highest user ID</td>
<td>2147483647</td>
</tr>
<tr>
<td>@@minsuid</td>
<td>Lowest server user ID</td>
<td>-32768</td>
</tr>
<tr>
<td>@@probesuid</td>
<td>Probe server user ID</td>
<td>2</td>
</tr>
<tr>
<td>@@maxsuid</td>
<td>Highest server user ID</td>
<td>2147483647</td>
</tr>
</tbody>
</table>

To issue a global variable, enter:

```
select variable_name
```

For example:
select @@minuserid
-----------
-32768

Creating and assigning roles to users

The final steps in adding database users are assigning them special roles, as required, and granting permissions. For more information on permissions, see Chapter 17, “Managing User Permissions.”

The roles supported by Adaptive Server enable you to enforce individual accountability. Adaptive Server provides system roles, such as system administrator and system security officer, and user-defined roles, which are created and granted to users or other roles by a system security officer. Object owners can grant database access as appropriate to each role.

System-defined roles

Table 14-4 lists the system roles, the value to use for the role_granted option of the grant role or revoke role command, and the tasks usually performed by a person with that role.

<table>
<thead>
<tr>
<th>Role</th>
<th>Value for role_granted</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>system administrator</td>
<td>sa_role</td>
<td>Manages and maintains Adaptive Server databases and disk storage</td>
</tr>
<tr>
<td>System security officer</td>
<td>sso_role</td>
<td>Performs security-related tasks</td>
</tr>
<tr>
<td>Operator</td>
<td>oper_role</td>
<td>Backs up and loads databases server-wide</td>
</tr>
<tr>
<td>Sybase technical support</td>
<td>sybase_ts_role</td>
<td>Analysis and repair of database structures</td>
</tr>
<tr>
<td>Replication</td>
<td>replication_role</td>
<td>Replicate user data</td>
</tr>
<tr>
<td>Distributed transaction manager</td>
<td>dtm_tm_role</td>
<td>Coordinate transactions across servers</td>
</tr>
<tr>
<td>High availability</td>
<td>ha_role</td>
<td>Administer and execute Failover</td>
</tr>
<tr>
<td>Monitor and diagnosis</td>
<td>mon_role</td>
<td>Administer and execute performance and diagnostic monitoring</td>
</tr>
<tr>
<td>Job Scheduler administration</td>
<td>js_admin_role</td>
<td>Administer Job Scheduler</td>
</tr>
</tbody>
</table>
Creating and assigning roles to users

System administrator privileges

This section describes the privileges of the system administrator in Adaptive Server.

System administrators:

- Handle tasks that are not specific to applications
- Work outside Adaptive Server’s discretionary access control system

The role of system administrator is usually granted to individual Adaptive Server logins. All actions taken by that user can be traced to his or her individual server user ID. If the server administration tasks at your site are performed by a single individual, you may instead choose to use the “sa” account that is installed with Adaptive Server. At installation, the “sa” account user has permission to assume the system administrator, system security officer, and Operator roles. Any user who knows the “sa” password can log in to that account and assume any or all of these roles.

The fact that a system administrator operates outside the protection system serves as a safety precaution. For example, if the database owner accidentally deletes all the entries in the sysusers table, the system administrator can restore the table (as long as backups exist). There are several commands that can be issued only by a system administrator. They include disk init, disk refit, disk reinit, shutdown, kill, disk mirror, mount, unmount and several monitoring commands.

In granting permissions, a system administrator is treated as the object owner. If a system administrator grants permission on another user’s object, the owner’s name appears as the grantor in sysprotects and in sp_helpprotect output.

<table>
<thead>
<tr>
<th>Role</th>
<th>Value for role Granted</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job Scheduler user</td>
<td>js_user_role, js_client_role</td>
<td>Create and run jobs through Job Scheduler</td>
</tr>
<tr>
<td>Real Time messaging</td>
<td>messaging_role</td>
<td>Administer and execute Real Time Messaging</td>
</tr>
<tr>
<td>Web Services</td>
<td>webservices_role</td>
<td>Administer Web Services</td>
</tr>
<tr>
<td>Key custodian</td>
<td>keycustodian_role</td>
<td>Creates and manages encryption keys</td>
</tr>
</tbody>
</table>
System administrators automatically assume the identity of a database owner when they log in to a database, and are conferred all database owner privileges. This automatic mapping occurs irrespective of any aliases assigned to the user. The system administrator can perform tasks usually reserved for the database owner such as dbcc commands, diagnostic functions, read data pages, recover data, or indexes.

**System security officer privileges**

System security officers perform security-sensitive tasks in Adaptive Server, including:

- Granting the system security officer, Operator, and Key Custodian roles
- Administering the audit system
- Changing passwords
- Adding new logins
- Dropping logins
- Locking and unlocking login accounts
- Creating and granting user-defined roles
- Administering network-based security
- Granting permission to use the set proxy or set session authorization commands

The system security officer can access any database—to enable auditing—but, in general, has no special permissions on database objects (except for encryption keys and decrypt permission on encrypted columns. See the User Guide for Encrypted Columns). An exception is the sybsecurity database, where only a system security officer can access the syasaudits table. There are also several system procedures that can be executed only by a system security officer.

System security officers can repair any damage inadvertently done to the protection system by a user. For example, if the database owner forgets his or her password, a system security officer can change the password to allow the database owner to log in.

The system security officers shares login management responsibilities with system administrators. System security officers are responsible for adding, locking, and unlocking logins.
System security officers can also create and grant user-defined roles to users, other roles, or groups. For information about creating and granting user-defined roles, see “Creating and assigning roles to users” on page 403.

**Operator privileges**

Users who have been granted the Operator role can back up and restore databases on a server-wide basis without having to be the owner of each database. The Operator role allows a user to use these commands on any database:

- `dump database`
- `dump transaction`
- `load database`
- `load transaction`
- `checkpoint`
- `online database`

The system security officer grants the operator role.

**Sybase technical support**

The Sybase Tech Support role allows access for a Sybase Technical Support engineer to display internal memory and on-disk data structures through trace output, consistency checking, and patching data structures. This role is used for analyzing problems and recovering data by hand. Some actions necessary for resolving these issues may require additional system roles for access. Sybase recommends that the system security officer grant this role to a knowledgeable Sybase engineer only while this analysis or repair is being done.

**Replication role**

The user maintaining Replication Server and ASE Replicator requires the Replication role. See the *Replication Server Administration Guide* and the *ASE Replicator User’s Guide* for information about this role.
Distributed Transaction Manager role

The Distributed Transaction Manager (DTM) transaction coordinator uses this role to allow system stored procedures to administer transactions across servers. Clients using the DTM XA interface require this role. See Using Adaptive Server Distributed Transaction Management Features for more information.

High availability role

You must have the high availability role to configure the high availability subsystem to administer primary and companion servers through commands and stored procedures. See Using Sybase Failover in a High Availability System for more information.

Monitoring and diagnosis

This role is required to administer the Adaptive Server Monitoring and Diagnostics (MDA) subsystem. You must have this role to execute a MDA remote procedure call and to administer the collection of monitored data. See the Performance and Tuning Series: Monitoring Tables for more information.

Job Scheduler roles

The Job Scheduler has three system roles to manage permissions for its operation:

- **js_admin_role** – required to administer Job Scheduler, and provides access to the stored procedures and allow you to modify, delete, and perform Job Scheduler administrative operations.

- **js_user_role** – required for a user to create, modify, delete, and run scheduled jobs using the Job Scheduler stored procedures.

- **js_client_role** – allows users to work with predefined jobs but not to create or alter jobs.

See the Job Scheduler User's Guide for more information.
Creating and assigning roles to users

Real-Time Messaging role
Used by the Real-Time Messaging (RTMS) subsystem execute msgsend, msgrecv, and certain sp_msgadmin commands. See the Messaging Services User’s Guide for more information.

Web Services role
Used by the Web Services subsystem to execute create service, create existing service, drop service, and alter service. See the Web Services User’s Guide for more information.

Key custodian role
The Key Custodian role is responsible for key management: creating and altering encryption keys, setting up the system encryption password, setting up key copies for users, and so on. See the User Guide for Encrypted Columns for more information.

User-defined roles
Planning user-defined roles
Before you implement user-defined roles, decide:
• The roles you want to create
• The responsibilities for each role
• The position of each in the role hierarchy
• Which roles in the hierarchy are mutually exclusive
• Whether such exclusivity is at the membership level or activation level

Avoid name conflicts when you create user-defined roles by following a naming convention. For example, you could use the “_role” suffix for role names. Adaptive Server does not check for such restrictions.

User-defined role names cannot duplicate user names. If a role must have the same name as a user, you can avoid conflict by creating a new role, having it contain the original role, and then granting the new role to the user.
After you have planned the roles to create and the relationships among them, decide how to allocate roles according to business requirements and the responsibilities of your users.

The maximum number of roles that a user can activate per user session is 127.

The minimum number of roles, 15, includes the system roles included with Adaptive Server.

The maximum number of user-defined roles that can be activated server-wide is 992. The first 32 roles are reserved for Sybase system roles.

Creating a user-defined role

Use the `create role` command to create a role. The syntax is:

```
create role role_name [with passwd "password"
   [, {passwd expiration | min passwd length | max failed_logins } option_value ]]
```

where:

- `role_name` – name of the new role.
- `password` – optional password. Must be specified by any user that is using the role.
- `passwd expiration` – specifies the password expiration interval in days. It can be any value between 0 and 32767, inclusive.
- `min passwd length` – specifies the minimum password length required for the specified role.
- `max failed_logins` – specifies the number of allowable failed login attempts for the specified login.
- `option_value` – specifies the value for `passwd expiration`, `min passwd length`, or `max failed_logins`.

For example, to create the `intern_role` without a password, enter:

```
create role intern_role
```

To create the `doctor_role` and assign the password “physician”, enter:

```
create role doctor_role with passwd "physician"
```

Only the system security officer can create user-defined roles.
Creating and assigning roles to users

Adding and removing passwords from a role

Only a system security officer can add or drop a password from a role.

Use the `alter role` command to add or drop a password from either a system or user-defined role. The syntax is:

```sql
alter role role_name
 [add passwd password | drop passwd]
```

For example, to require the password "oper8x" for the `oper_role`, enter:

```sql
alter role oper_role add passwd oper8x
```

To drop the password from the role, enter:

```sql
alter role oper_role drop passwd
```

Role hierarchies and mutual exclusivity

A system security officer can define role hierarchies such that if a user has one role, the user also has roles lower in the hierarchy. For example, the "chief_financial_officer" role might contain both the "financial_analyst" and the "salary_administrator" roles, as shown in Figure 14-2.

*Figure 14-2: Role hierarchy*

```
Chief Financial Officer
  Financial Analyst
  Salary Administrator
```

The Chief Financial Officer can perform all tasks and see all data that can be viewed by the Salary Administrators and Financial Analysts.

Additionally, you can define a role's mutual exclusivity to enforce static or dynamic separation of duty policies. Roles can be defined to be mutually exclusive for:

- Membership – one user cannot be granted two different roles. For example, you might not want the "payment_requestor" and "payment_approver" roles to be granted to the same user.
- Activation – one user cannot activate, or enable, two different roles. For example, a user might be granted both the "senior_auditor" and the "equipment_buyer" roles, but not permitted to have both roles enabled at the same time.
System roles, as well as user-defined roles, can be defined to be in a role hierarchy or to be mutually exclusive. For example, you might want a “super_user” role to contain the system administrator, Operator, and Technical Support roles. To enforce a separation of roles, you might also want to define the system administrator and system security officer roles to be mutually exclusive for membership; that is, one user cannot be granted both roles.

Role hierarchies and mutual exclusivity

This section describes how to set up role hierarchies and enforce a separation of roles.

Defining and changing mutual exclusivity of roles

To define mutual exclusivity between two roles, use:

```
alter role role1 { add | drop } exclusive { membership | activation } role2
```

For example, to define `intern_role` and `specialist_role` as mutually exclusive at the membership level, enter:

```
alter role intern_role add exclusive membership specialist_role
```

The example above restricts users who have membership in `intern_role` from also being members of `specialist_role`.

To define the `sso_role` and `sa_role` as mutually exclusive at the activation level, enter the following command, which prohibits a user who is a member of `sso_role` and `sa_role` from assuming both roles simultaneously:

```
alter role sso_role add exclusive activation sa_role
```

Defining and changing a role hierarchy

Defining a role hierarchy involves choosing the type of hierarchy and the roles, then implementing the hierarchy by granting roles to other roles.

For example:

```
grant role intern_role to specialist_role
grant role doctor_role to specialist_role
```
Creating and assigning roles to users

**Figure 14-3: Creating a role hierarchy**

In Figure 14-3, the “specialist” role contains the “doctor” and “intern” roles. This means that “specialist” has all the privileges of both “doctor” and “intern.”

To establish a hierarchy with a “super_user” role containing the sa_role and oper_role system roles, specify:

```sql
grant role sa_role to super_user
grant role oper_role to super_user
```

**Note** If a role that requires a password is contained within another role, the user with the role that contains the other does not need to use the password for the contained role. For example, in Figure 14-3, say the “doctor” role usually requires a password. The user who has the “specialist” role does not need to enter the “doctor” password because “doctor” is contained within “specialist.” Role passwords are only required for the highest level role.

When creating role hierarchies:

- You cannot grant a role to another role that directly contains it. This prevents duplication.
  
  For example, in Figure 14-3, you cannot grant “doctor” to “specialist” because “specialist” already contains “doctor.”

- You can grant a role to another role that does not directly contain it.
  
  For example, in Figure 14-4, you can grant the “intern” role to the “specialist” role, even though “specialist” already contains the “doctor” role, which contains “intern.” If you subsequently dropped “doctor” from “specialist,” then “specialist” still contains “intern.”

  In Figure 14-4, “doctor” has “consultant” role permissions because “consultant” has been granted to “doctor.” The “specialist” role also has “consultant” role permissions because “specialist” contains the “doctor” role, which in turn contains the “consultant.”

  However, “intern” does not have “consultant” role privileges, because “intern” does not contain the “consultant” role, either directly or indirectly.
You cannot grant a role to another role that is contained by the first role. This prevents “loops” within the hierarchy.

For example, in Figure 14-5, you cannot grant the “specialist” role to the “consultant” role; “consultant” is already contained in “specialist.”

When the system security officer grants a user a role that contains other roles, the user implicitly gets membership in all roles contained by the granted role. However, a role can only be activated or deactivated directly if the user has explicit membership in that role.

The system security officer cannot grant one role to another role that is explicitly or implicitly mutually exclusive at the membership level with the first role.

For example, in Figure 14-6, if the “intern” role is defined as mutually exclusive at the membership level with the “consultant” role, the system security officer cannot grant “intern” to the “doctor.”
Creating and assigning roles to users

**Figure 14-6: Mutual exclusivity at membership**

The user can activate or deactivate only directly granted roles.

For example, in the hierarchy shown in Figure 14-6, assume that you have been granted the “specialist” role. You have all the permissions of the “specialist” role, and, implicitly, because of the hierarchy, you have all the permissions of the “doctor” and “consultant” roles. However, you can activate only the “specialist” role. You cannot activate “doctor” or “consultant” because they were not directly granted to you. For information, see “Activating and deactivating roles” on page 415.

Revoking roles from other roles is similar to granting roles to other roles. It removes a containment relationship, and the containment relationship must be a direct one, as shown in Figure 14-7:

**Figure 14-7: Effect of revoking roles on role hierarchy**

For example, in Figure 14-7:

- If the system security officer revokes the “doctor” role from “specialist,” “specialist” no longer contains the “consultant” role or the “intern” role.

- The system security officer cannot revoke the “intern” role from “specialist” because “intern” is not directly contained by “specialist.”
Setting up default activation at login

A system security officer can change the default role for any user. Individual users can change only their own default settings.

When a user logs in to Adaptive Server, the user’s roles are not necessarily active, depending upon how the role is set up as a default role. If a role has a password associated with it, the user must use the set role command to activate the role.

The system security officer or user determines whether to activate any roles granted by default at login. sp_modifylogin sets the default status of user roles individually for each user. sp_modifylogin only affects user roles, not system roles.

By default, user-defined roles that are granted are not activated at login, but system roles that are granted are automatically activated, if they do not have passwords associated with them.

To set up a role to activate at login:

```
sp_modifylogin loginname, "add default role", role_name
```

To assign more than one default role to a user, use multiple sp_modifylogin commands.

To ensure that a role is inactive at login:

```
sp_modifylogin loginname, "drop default role", role_name
```

For example, to change the default setting for Ralph’s intern_role to be active automatically at login, execute:

```
sp_modifylogin ralph, "add default role", intern_role
```

Activating and deactivating roles

Roles must be active to have the access privileges of each role. A default role may not be active at login. If the role has a password, it is always inactive at login.

To immediately activate or deactivate a role:

```
set role role_name [on|off]
```

To activate or deactivate a role that has an attached password, use:

```
set role role_name with passwd "password" [on|off]
```
For example, to activate the “financial_analyst” role with the password “sailing19”, enter:

```
set role financial_analyst with passwd "sailing19" on
```

Activate roles only when you need them, and turn them off when you no longer need them. For example, when the sa_role is active, you assume the identity of database owner within any database that you use. To turn off the system administrator role and assume your “real” user identity, use:

```
set role sa_role off
```

If you are granted a role during a session, and you want to activate it immediately, use `set role` to turn it on.

### Setting up groups and adding users

The system security officer, the system administrator, or the database administrator creates a group using `sp_addgroup group_name`.

You can grant and revoke permissions at the group level. Group permissions are automatically passed to group members. Every database is created with a group named “public” to which all users automatically belong. Add a user to a group using `sp_adduser` and change a user’s group with `sp_changegroup`. See “Changing a user’s group membership” on page 423.

Groups are represented by an entry in the `sysusers` table. You cannot use the same name for creating a group and a user in the database (for example, you cannot have both a group and a user named “shirley”).

### Dropping users, groups, and user-defined roles

Table 14-5 list the system procedures that allow a system administrator, system security officer, or database owner to drop users and groups.

<table>
<thead>
<tr>
<th>Task</th>
<th>Required authority</th>
<th>System procedure</th>
<th>Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drop user from database</td>
<td>Database owner, system security officer, or system administrator</td>
<td><code>sp_dropuser</code></td>
<td>User database</td>
</tr>
</tbody>
</table>
A database owner, system security officer, or a system administrator can use `sp_dropuser` to deny an Adaptive Server user access to the database in which `sp_dropuser` is executed. (If a “guest” user is defined in that database, the user can still access that database as “guest.”)

The following is the syntax, where `<name_in_db>` is usually the login name, unless another name has been assigned with `sp_adduser`:

```
sp_dropuser <name_in_db>
```

You cannot drop a user who owns objects. Since there is no command to transfer ownership of objects, you must drop objects owned by a user before you drop the user with `sp_dropuser`. To deny access to a user who owns objects, use `sp_locklogin` to lock his or her account.

You also cannot drop a user who has granted permissions to other users. Use `revoke with cascade` to revoke permissions from all users who were granted permissions by the user to be dropped, then drop the user. You must then grant permissions to the users again, if appropriate.

### Dropping groups

The system security officer, the system administrator, or the database administrator uses `sp_dropgroup` to drop a group. The syntax is:

```
sp_dropgroup <grpname>
```

You cannot drop a group that has members. If you try to do so, the error report displays a list of the members of the group you are attempting to drop. To remove users from a group, use `sp_changegroup`, discussed in “Changing a user’s group membership” on page 423.

### Dropping user-defined roles

To drop a role, the system security officers uses the following, where `<role_name>` is the name of a user-defined role:
Locking or dropping Adaptive Server login accounts

To prevent a user from logging in to Adaptive Server, you can either lock or drop an Adaptive Server login account. Locking a login is safer than dropping it because locking a login account maintains the suid so that it cannot be reused.

Warning! Adaptive Server may reuse the server user ID (suid) of a dropped login account when the next login account is created. This occurs only when the dropped login holds the highest suid in syslogins; however, it can compromise accountability if execution of sp_droplogin is not being audited. Also, it is possible for a user with the reused suid to access database objects that were authorized for the old suid.

You cannot drop a login when:

- The user is in any database.
- The login is the last remaining user who holds the system security officer or system administrator roles.

Table 14-6: Locking or dropping login accounts

<table>
<thead>
<tr>
<th>Task</th>
<th>Required role</th>
<th>System procedure</th>
<th>Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lock login account, which maintains the suid so that it cannot be reused</td>
<td>System administrator or system security officer</td>
<td>sp_locklogin</td>
<td>Any. Must be in the master database if the system procedure is being logged for replication</td>
</tr>
</tbody>
</table>
### Locking and unlocking login accounts

Use `sp_locklogin` to lock and unlock accounts or to display a list of locked accounts. You must be a system security officer to use `sp_locklogin`.

The syntax is:

```sql
sp_locklogin  [ { loginame | "all"}, { "lock" | "unlock" } ]
```

where:

- `loginame` is the name of the account to be locked or unlocked. It must be an existing valid account.
- `all` indicates to lock or unlock all login accounts on an Adaptive Server, except those with `sa_role`.
- `lock` | `unlock` specifies whether the account is to be locked or unlocked.

To display a list of all locked logins, use `sp_locklogin` with no parameters.

You can lock an account that is currently logged in, and the user is not locked out of the account until he or she logs out. You can lock the account of a database owner, and a locked account can own objects in databases. In addition, you can use `sp_changedbowner` to specify a locked account as the owner of a database.

Adaptive Server ensures that there is always at least one unlocked system security officer’s account and one unlocked system administrator’s account.

### Dropping login accounts

A system security officer can use `sp_droplogin` to deny a user access to Adaptive Server. The syntax is:

```sql
sp_droplogin loginame
```
Changing user information

sp_droplogin fails if the user identified by loginame exists as a database user or alias in any database. Use sp_dropuser to drop the user from a database. For more information, see “Dropping users” on page 417.

Locking logins that own thresholds

This section discusses thresholds and how they are affected by locked user logins.

- As a security measure, threshold stored procedures are executed using the account name and roles of the login that created the procedure.
- You cannot drop the login of a user who owns a threshold.
- If you lock the login of a user who owns a threshold, the threshold cannot execute the stored procedure.
- The last-chance threshold, and thresholds created by the “sa” login are not affected by sp_locklogin. If you lock the “sa” login, the last chance threshold and thresholds created or modified by the “sa” user still fire.

Changing user information

Table 14-7 lists the system procedures you use to change passwords, default database, default language, full name, or group assignment.

Table 14-7: System procedures for changing user information

<table>
<thead>
<tr>
<th>Task</th>
<th>Required role</th>
<th>System procedure</th>
<th>Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change your password</td>
<td>None</td>
<td>sp_password</td>
<td>Any database</td>
</tr>
<tr>
<td>Change another user’s password</td>
<td>System security officer</td>
<td>sp_password</td>
<td>Any database</td>
</tr>
<tr>
<td>Change authentication mechanism</td>
<td>System security officer</td>
<td>sp_modifylogin</td>
<td>Any database</td>
</tr>
<tr>
<td>Change your default database, default language, or full name</td>
<td>None</td>
<td>sp_modifylogin</td>
<td>Any database</td>
</tr>
<tr>
<td>Change a login account’s default database, default language, or full name</td>
<td>System administrator or system security officer</td>
<td>sp_modifylogin</td>
<td>Any database</td>
</tr>
<tr>
<td>Change the group assignment of a user</td>
<td>System administrator, database owner, or system security officer</td>
<td>sp_changegroup</td>
<td>User database</td>
</tr>
</tbody>
</table>
Changing passwords

All users can change their passwords at any time using `sp_password`. The system security officer can use `sp_password` to change any user’s password.

See the *Adaptive Server Commands and Reference: System Procedures* for the `sp_password` syntax.

For example, a user can change his or her own password from “3blindmice” to “2mediumhot” using:

```
sp_password "3blindmice", "2mediumhot"
```

These passwords are enclosed in quotes because they begin with numbers.

In the following example, the system security officer whose password is “2tomato” changes Victoria’s password to “sesame1”:

```
sp_password "2tomato", sesame1, victoria
```

Requiring new passwords

You may choose to use the systemwide password expiration configuration parameter to establish a password expiration interval, which forces all Adaptive Server users to change their passwords on a regular basis. For information, see Chapter 5, “Setting Configuration Parameters.” Even if you do not use systemwide password expiration, it is important, for security reasons, that users change their passwords periodically.

The configuration parameter is superseded by the password policy settings. `password expiration interval` specifies the password expiration interval in days. It can be any value between 0 and 32767, inclusive. For example, if you create a new login on August 1, 2007 at 10:30 AM, with a password expiration interval of 30 days, the password expires on August 31, 2007 at 10:30 AM.

The column `pwdate` in the `syslogins` table records the date of the last password change. The following query selects all login names whose passwords have not changed since September 15, 2007:

```
select name, pwdate
from syslogins
where pwdate < "Sep 15 2007"
```
Changing user information

Null passwords

Do not assign a null password. When Adaptive Server is installed, the default “sa” account has a null password. The following example shows how to change a null password to a valid one:

```
sp_password null, "8M4LNCH"
```

**Note** “null” is not enclosed in quotes in the statement.

Logging in after lost password

If your site encounters any of these situations:

- All system administrator login accounts are locked.
- All system security officer login accounts are locked.
- The password for sa_role or sso_role has been lost,

You can restart Adaptive Server with the `dataserver -p login_name` parameter, which allows you to set a new password for these accounts and roles if there is no way to recover a lost password. `login_name` is the name of the user or the name of the role (sa_role or sso_role) whose password needs to be reset.

This allows you to set a new password for these account if there is no way to recover a lost password.

When you start with the `-p` parameter, Adaptive Server generates, displays, and encrypts a random password and saves it in `master..syslogins` or in `master..syssrvroles` as that account or role’s new password.

Sybase highly recommends that you change the password when the server restarts. For example, to reset the password for user rsmith who has sa_role:

```
dataserver -prsmith
```

To reset the password of the sso_role:

```
dataserver -psso_role
```
Changing user defaults

Any user can use sp_modifylogin to change his or her full name, default authentication method, default database, default language, and default role. Use sp_modifylogin to set password length and expiration, to limit failed login attempts, and to specify that a login script be run automatically when a user logs in. A system administrator can change these settings for any user. The syntax is:

```
sp_modifylogin login_name, option, value
```

where:
- `login_name` – is the name of the user whose account you are modifying.
- `option` – specifies the option that you are changing. See sp_modifylogin in the Reference Manual: Procedures for a list of available options.
- `value` – is the new value for the specified option.

After you execute sp_modifylogin to change the default database, the user is connected to the new default database the next time he or she logs in. However, sp_modifylogin does not automatically give the user access to the database. Unless the database owner has set up access with sp_adduser, sp_addalias, or with a guest user mechanism, the user is connected to master even after his or her default database has been changed.

This example changes the default database for “anna” to pubs2:

```
sp_modifylogin anna, defdb, pubs2
```

This example changes the default language for “claire” to French:

```
sp_modifylogin claire, deflanguage, french
```

This example changes the full name for “mtwain” to “Samuel Clemens.”

```
sp_modifylogin mtwain, fullname, "Samuel Clemens"
```

Changing a user’s group membership

A system administrator, system security officer, or the database owner can use sp_changegroup to change a user’s group affiliation. Each user can be a member of only one group other than “public,” of which all users are always members.

Before you execute sp_changegroup:
- The group must exist. Use sp_addgroup to create a group.
Changing user information

- The user must have access to the current database (must be listed in sysusers).

The syntax for `sp_changegroup` is:

`sp_changegroup grpname, username`

For example, to change the user “jim” from his current group to the group “management,” use:

`sp_changegroup management, jim`

To remove a user from a group without assigning the user to another group, you must change the group affiliation to “public”:

`sp_changegroup "public", jim`

The name “public” must be in quotes because it is a reserved word. This command reduces Jim’s group affiliation to “public” only.

When a user changes from one group to another, the user loses all permissions that he or she had as a result of belonging to the old group, but gains the permissions granted to the new group.

The assignment of users into groups can be changed at any time.

Changing the user process information

The `set` command includes options that allow you to assign each client an individual name, host name, and application name. This is useful for differentiating among clients in a system where many clients connect to Adaptive Server using the same name, host name, or application name.

The partial syntax for the `set` command is:

`set [clientname client_name | clienthostname host_name | clientapplname application_name]`

Where `client_name` is the name you are assigning the client, `host_name` is the name of the host from which the client is connecting, and `application_name` is the application that is connecting to Adaptive Server. These parameters are stored in the `clientname`, `clienthostname`, `clientapplname` columns of the `sysprocesses` table.

For example, if a user logs in to Adaptive Server as “client1,” you can assign them an individual client name, host name, and application name using commands similar to:

`set clientname 'alison'`
set clienthostname 'money1'
set clientapplname 'webserver2'

This user now appears in the sysprocesses table as user “alison” logging in from host “money1” and using the “webserver2” application. However, although the new names appear in sysprocesses, they are not used for permission checks, and sp_who still shows the client connection as belonging to the original login (in the case above, client1). set clientname does not perform the same function as set proxy, which allows you to assume the permissions, login name, and suid of another user.

You can set a client name, host name, or application name for only your current client session (although you can view the connection information for any client connection). Also, this information is lost when a user logs out. These parameters must be reassigned each time a user logs in. For example, the user “alison” cannot set the client name, host name, or application name for any other client connection.

Use the client’s spid to view their connection information. For example, if the user “alison” described above connects with a spid of 13, issue the following command to view all the connection information for this user:

```sql
select * from sysprocesses where spid = 13
```

To view the connection information for the current client connection (for example, if the user “alison” wanted to view her own connection information), enter:

```sql
select * from sysprocesses where spid = @@spid
```

### Using aliases in databases

The alias mechanism allows you to treat two or more users as the same user inside a database so that they all have the same privileges. This mechanism is often used so that more than one user can assume the role of database owner. A database owner can use the setuser command to impersonate another user in the database. You can also use the alias mechanism to set up a collective user identity.
Using aliases in databases

For example, suppose that several vice presidents want to use a database with identical privileges and ownerships. If you add the login “vp” to Adaptive Server and the database and have each vice president log in as “vp,” there is no way to tell the individual users apart. Instead, alias all the vice presidents, each of whom has his or her own Adaptive Server account, to the database user name “vp.”

Note: Although more than one individual can use the alias in a database, you can still maintain individual accountability by auditing the database operations performed by each user. For more information about auditing, see Chapter 18, “Auditing.”

The collective user identity from using aliases implies set-ownership for database objects. For example, if user “loginA” is aliased to dbo in in database db1, all objects created by “loginA” in db1 are owned by dbo. However, Adaptive Server concretely records an object’s ownership in terms of the login name and the creator’s database user ID. See “Concrete identification” on page 573. An alias cannot be dropped from a database if he concretely owns objects in that database.

Table 14-8 lists the system procedures used to manage aliases:

<table>
<thead>
<tr>
<th>Task</th>
<th>Require role</th>
<th>System procedure</th>
<th>Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add an alias for a user</td>
<td>Database owner or system administrator</td>
<td>sp_addalias</td>
<td>User database</td>
</tr>
<tr>
<td>Drop an alias</td>
<td>Database owner or system administrator</td>
<td>sp_dropalias</td>
<td>User database</td>
</tr>
</tbody>
</table>

Note: As of version 12.0, you cannot drop the alias of a login if that login created objects in the database. In most cases, you should use aliases only for users who do not own tables, procedures, views or triggers.

Adding aliases

To add an alias for a user, use sp_addalias. The syntax is:

```
sp_addalias loginame, name_in_db
```

where:
• *loginame* – is the name of the user who wants an alias in the current database. This user must have an account in Adaptive Server but cannot be a user in the current database.

• *name_in_db* – is the name of the database user to whom the user specified by *loginame* is to be linked. The *name_in_db* must exist in *sysusers* in the current database.

Executing `sp_addalias` maps the user name specified by *loginame* to the user name specified by *name_in_db*. It does this by adding a row to the system table *sysalternates*.

When a user tries to use a database, Adaptive Server checks for the user’s server user ID number (*suid*) in *sysusers*. If it is not found, Adaptive Server then checks *sysalternates*. If the user’s *suid* is found there, and it is mapped to a database user’s *suid*, the first user is treated as the second user while the first user is using the database.

For example, suppose that Mary owns a database. She wants to allow both Jane and Sarah to use the database as if they were its owner. Jane and Sarah have logins on Adaptive Server but are not authorized to use Mary’s database. Mary executes the following commands:

```sql
sp_addalias jane, dbo
exec sp_addalias sarah, dbo
```

**Warning!** Users who are aliased to the database owner have all the permissions and can perform all the actions that can be performed by the real database owner, with respect to the database in question. A database owner should carefully consider the implications of vesting another user with full access to a database.

---

**Dropping aliases**

Use `sp_dropalias` to drop the mapping of an alternate *suid* to a user ID. Doing this deletes the relevant row from *sysalternates*. The syntax is the following, where *loginame* is the name of the user specified by *loginame* when the name was mapped with `sp_addalias`:

```sql
sp_dropalias loginame
```

After a user’s alias is dropped, the user no longer has access to the database.
You cannot drop an alias if the aliased login created any objects or thresholds. Before using `sp_dropalias` to remove an alias that has performed these actions, remove the objects or procedures. If you still need them after dropping the alias, re-create them with a different owner.

**Getting information about aliases**

To display information about aliases, use `sp_helpuser`. For example, to find the aliases for “dbo,” execute:

```
sp_helpuser dbo
```

<table>
<thead>
<tr>
<th>Users_name</th>
<th>ID_in_db</th>
<th>Group_name</th>
<th>Login_name</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbo</td>
<td>1</td>
<td>public</td>
<td>sa</td>
</tr>
</tbody>
</table>

(1 row affected)

Users aliased to user.
Login_name

----------------------
andy
christa
howard
linda

(4 rows affected)

**Getting information about users**

Table 14-9 lists procedures you can use to obtain information about users, groups, and current Adaptive Server usage.
### Table 14-9: Reporting information about Adaptive Server users and groups

<table>
<thead>
<tr>
<th>Task</th>
<th>Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report current Adaptive Server users and processes</td>
<td><code>sp_who</code></td>
</tr>
<tr>
<td>Display information about login accounts</td>
<td><code>sp_displaylogin</code></td>
</tr>
<tr>
<td>Report users and aliases in a database</td>
<td><code>sp_helpuser</code></td>
</tr>
<tr>
<td>Report groups within a database</td>
<td><code>sp_helpgroup</code></td>
</tr>
</tbody>
</table>

### Getting reports on users and processes

Use `sp_who` to report information about current users and processes on Adaptive Server:

```
sp_who [loginame | "spid"]
```

where:

- `loginame` – is the user’s Adaptive Server login name. If you give a login name, `sp_who` reports information about processes being run by that user.
- `spid` – is the number of a specific process.

For each process run, `sp_who` reports the security-relevant information for the server process ID, its status, the login name of the process user, the real login name (if `loginame` is an alias), the name of the host computer, the server process ID of a process that is blocking this one (if any), the name of the database, and the command being run.

If you do not provide a login name or `spid`, `sp_who` reports on processes being run by all users.

The following example shows the security relevant results from executing `sp_who` without a parameter:

```
spid status loginame origname hostname blk dbname cmd
------- -------- -------- --------- -------- --- ------ --------
  1 running sa sa sunbird 0 pubs2 SELECT
  2 sleeping NULL NULL 0 master NETWORK HANDLER
  3 sleeping NULL NULL 0 master MIRROR HANDLER
  4 sleeping NULL NULL 0 master AUDIT PROCESS
  5 sleeping NULL NULL 0 master CHECKPOINT SLEEP
```

(5 rows affected, return status = 0)

`sp_who` reports NULL for the `loginame` for all system processes.
Getting information about login accounts

Use `sp_displaylogin` to display information about a specified login account—or login names matching a wild card pattern—including any roles granted, where `loginame` (or the wildcard matching pattern) is the user login name pattern about which you want information:

```
sp_displaylogin [loginame | wildcard]
```

If you are not a system security officer or system administrator, you can display information only about your own account. If you are a system security officer or system administrator, you can use the `loginame | wildcard` parameter to access information about any account.

`sp_displaylogin` displays your server user ID, login name, full name, any roles that have been granted to you, date of last password change, default database, default language, whether your account is locked, any auto login script, password expiration interval, whether password has expired, the login password encryption version used, and the authentication mechanism specified for the login.

The password expires at the time of day when the password was last changed after the number of days specified by `password expiration interval` has passed. `password expiration interval` specifies the password expiration interval in days. It can be any value between 0 and 32767, inclusive. For example, if you create a new login on August 1, 2007 at 10:30 AM, with a password expiration interval of 30 days, the password expires on August 31, 2007 at 10:30 AM.

`sp_displaylogin` displays all roles that have been granted to you, so even if you have made a role inactive with the `set` command, that role is displayed. For example, this displays the roles for the `sa`:

```
sp_displaylogin 'sa'
Suid: 121
Loginame: mylogin
Fullname: 
Default Database: master
Default Language:
Auto Login Script:
Configured Authorization:
  sa_role (default ON)
  sso_role (default ON)
  oper_role (default ON)
  sybase_ts_role (default ON)
Locked: NO
Date of Last Password Change: Aug 10 2006 11:17AM
Password expiration interval: 0
```
Password expired: NO
Minimum password length: 6
Maximum failed logins: 0
Current failed login attempts:
Authenticate with: NONE
Login password encryption: SYB-PROP, SHA-256
Last login date : Aug 17 2006 5:55PM
(return status = 0)

Getting information about database users

Use `sp_helpuser` to report information about authorized users of the current database, where `name_in_db` is the user’s name in the current database:

```
sp_helpuser [name_in_db]
```

If you give a user’s name, `sp_helpuser` reports information about that user. If you do not give a name, it reports information about all users.

The following example shows the results of executing `sp_helpuser` without a parameter in the database `pubs2`:

```
sp_helpuser

Users_name | ID_in_db | Group_name | Login_name
---------- | -------- | ---------- | ----------
dbo 1 | public | sa
marcy 4 | public | marcy
sandy 3 | public | sandy
judy 5 | public | judy
linda 6 | public | linda
anne 2 | public | anne
jim 7 | senioreng | jim
```

Finding user names and IDs

To find a user’s server user ID or login name, use `suser_id` and `suser_name`. 
Getting information about users

Table 14-10: System functions suser_id and suser_name

<table>
<thead>
<tr>
<th>To find</th>
<th>Use</th>
<th>With the argument</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server user ID</td>
<td>suser_id</td>
<td>(&quot;server_user_name&quot;)</td>
</tr>
<tr>
<td>Server user name (login name)</td>
<td>suser_name</td>
<td>([server_user_ID])</td>
</tr>
</tbody>
</table>

The arguments for these system functions are optional. If you do not provide one, Adaptive Server displays information about the current user.

This example shows how to find the server user ID for the user “sandy:”

```sql
select suser_id("sandy")
```

------

3
This example shows how a system administrator whose login name is “mary” issues the commands without arguments:

```sql
select suser_name(), suser_id()
---------------------------- ----
mary 4
```

To find a user's ID number or name inside a database, use `user_id` and `user_name`.

**Table 14-11: System functions `user_id` and `user_name`**

<table>
<thead>
<tr>
<th>To find</th>
<th>Use</th>
<th>With the argument</th>
</tr>
</thead>
<tbody>
<tr>
<td>User ID</td>
<td><code>user_id</code></td>
<td>[&quot;db_user_name&quot;]</td>
</tr>
<tr>
<td>User name</td>
<td><code>user_name</code></td>
<td>[db_user_id]</td>
</tr>
</tbody>
</table>

The arguments for these functions are optional. If you do not provide one, Adaptive Server displays information about the current user. For example:

```sql
select user_name(10)
---------------------------------------------
NULL
(1 row affected)
select user_name( )
---------------------------------------------
dbo
(1 row affected)
select user_id("joe")
---------------------------------------------
NULL
(1 row affected)
```

### Displaying information about roles

Table 14-12 lists the system procedures and functions to use to find information about roles, and the section in this chapter that provides details.

**Table 14-12: Finding information about roles**

<table>
<thead>
<tr>
<th>To display information about</th>
<th>Use</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>The role ID of a role name</td>
<td><code>role_id</code> system function</td>
<td>“Finding role IDs and names” on page 434</td>
</tr>
<tr>
<td>The role name of a role ID</td>
<td><code>role_name</code> system function</td>
<td>“Finding role IDs and names” on page 434</td>
</tr>
</tbody>
</table>
### Finding role IDs and names

To find a role ID when you know the role name, use `role_id`:

```
role_id(role_name)
```

Any user can execute `role_id`. If the role is valid, `role_id` returns the server-wide ID of the role (srid). The `sysrvroles` system table contains an `srid` column with the role ID and a `name` column with the role name. If the role is not valid, `role_id` returns NULL.

To find a role name when you know the role ID, use `role_name`:

```
role_name(role_id)
```

Any user can execute `role_name`.

### Viewing active system roles

Use `show_role` to display the currently active _system roles_ for the specified login:

```
show_role()
```

If you have not activated any system role, `show_role` returns NULL. If you are a database owner, and you execute `show_role` after using `setuser` to impersonate another user, `show_role` returns your own active system roles, not those for whom you are impersonating.
Any user can execute `show_role`.

**Note** The `show_role` function does not give information about user-defined roles.

### Displaying a role hierarchy
You can see all roles granted to your login name or see the entire hierarchy tree of roles displayed in table format using `sp_displayroles`:

```
sp_displayroles {login_name | rolename [, expand_up | expand_down]}
```

Any user can execute `sp_displayroles` to see his or her own roles. Only the system security officer can view information about roles granted to other users.

### Viewing user roles in a hierarchy
Use `role_contain` to determine whether any role you specify contains any other role you specify:

```
role_contain ("role1", "role2")
```

If `role1` is contained by `role2`, `role_contain` returns 1. Any user can execute `role_contain`.

### Determining mutual exclusivity
Use the `mut_excl_roles` function to determine whether any two roles assigned to you are mutually exclusive, and the level at which they are mutually exclusive:

```
mut_excl( role1, role2, {membership | activation} )
```

Any user can execute `mut_excl_roles`. If the specified roles, or any role contained by either specified role, are mutually exclusive, `mut_excl_roles` returns 1; if the roles are not mutually exclusive, `mut_excl_roles` returns 0.

### Determining role activation
To find all active roles for the current login session of Adaptive Server, use `sp_activeroles`:

```
sp_activeroles [expand_down]
```
Establishing a password and login policy

expand_down displays the hierarchy of all roles contained by any roles granted to you.

Any user can execute sp_activeroles.

Checking for roles in stored procedures

Use proc_role within a stored procedure to guarantee that only users with a specific role can execute the procedure. Only proc_role provides a fail-safe way to prevent inappropriate access to a particular stored procedure.

You can use grant execute to grant execute permission on a stored procedure to all users who have been granted a specified role. Similarly, revoke execute removes this permission.

However, grant execute permission does not prevent users who do not have the specified role from being granted execute permission on a stored procedure. If you want to ensure, for example, that all users who are not system administrators can never be granted permission to execute a stored procedure, use proc_role within the stored procedure itself. It checks to see whether the invoking user has the correct role to execute the procedure.

proc_role takes a string for the required role and returns 1 if the invoker possesses it. Otherwise, it returns 0.

For example, here is a procedure that uses proc_role to see if the user has the sa_role role:

```sql
create proc test_proc
as
if (proc_role("sa_role") = 0)
begin
    print "You don’t have the right role"
    return -1
end
else
    print "You have System Administrator role"
    return 0
```

Establishing a password and login policy

Adaptive Server includes several controls for setting policies for logins, roles, and passwords for internal authentication.
In Adaptive Server the system security officer can:

- Specifying the maximum allowable number of times an invalid password can be entered for a login or role before that login or role is automatically locked
- Logging in after a lost password
- Locking and unlocking logins and roles manually
- Displaying login password information
- Specify the minimum password length required server-wide or for a specific login or role
- Checking for password complexity of logins
- Enabling custom password checks of logins
- Setting the password expiration interval
- Securing login passwords stored on a disk and in memory
- Using only the SHA-256 algorithm for storing passwords on disk
- Login password character set considerations
- Upgrading and downgrading behavior
- Last login and locking inactive login accounts
- Using passwords in a high availability environment

Negative values may be used for user IDs ($uid$).

The server user ID ($suid$) associated with a group or a role in `sysusers` is not equal to the negation of their user ID ($uid$). Every $suid$ associated with a group or a role in `sysusers` is set to -2 (INVALID_SUID).

### Setting and changing the maximum login attempts

Setting the maximum number of login attempts allowed provides protection against “brute-force” or dictionary-based attempts to guess passwords. A system security officer can specify a maximum number of consecutive login attempts allowed, after which the login or role is automatically locked. The number of allowable failed login attempts can be set for the entire server or for individual logins and roles. Individual settings override the server-wide setting.
Establishing a password and login policy

The number of failed logins is stored in the logincount column in master.syslogins. A successful login resets the number of failed logins to 0.

❖ Setting the server-wide maximum failed logins

- By default, this value is turned off and this check is not applied to passwords. Use sp_passwordpolicy to set the server-wide setting for the maximum number of failed logins for logins and roles.

To set the number of failed logins allowed, enter:

```sql
sp_passwordpolicy 'set', 'maximum failed logins', number
```

For example:

```sql
sp_passwordpolicy 'set', 'maximum failed logins', 3
```


❖ Setting the maximum failed logins for specific logins

- To set the maximum failed logins for a specific login at creation, use sp_addlogin.

This example creates the new login “joe” with the password “Djdiek3” and sets the maximum number of failed login attempts for the login “joe” to 2:

```sql
sp_addlogin joe, "Djdiek3", pubs2, null, null, null, null, 2
```


❖ Setting the maximum failed logins for specific roles

- To set the maximum failed logins for a specific role at creation, use create role.

This example creates the intern_role role with the password “temp244”, and sets the maximum failed logins for intern_role to 20:

```sql
create role intern_role with passwd "temp244", maximum failed logins 20
```

For details on the syntax and rules for using maximum failed logins, see create role.

❖ Changing the maximum failed logins for specific logins

- Use sp_modifylogin to set or change the maximum failed logins for an existing login.
Example 1  Changes the maximum failed logins for the login “joe” to 40:

    sp_modifylogin "joe", "max failed_logins", "40"

Note  The value parameter is a character datatype; therefore, quotes are required for numeric values.

Example 2  Changes the overrides for maximum failed logins for all logins to 3:

    sp_modifylogin "all overrides", "max failed_logins", "3"

Example 3  Removes the overrides for maximum failed logins option for all logins:

    sp_modifylogin "all overrides", "max failed_logins", "-1"

    sp_modifylogin only effects user roles, not system roles. For details on the syntax and rules, see sp_modifylogin.

Changing the maximum failed logins for specific roles

- Use alter role to set or change the maximum failed logins for an existing role.

Example 1  Changes the maximum failed logins allowed for physician_role to 5:

    alter role "all overrides" set maximum failed logins -1

Example 2  Removes the overrides for the maximum failed logins for all roles:

    alter role physician_role set maximum failed logins 5

For details on the syntax and rules for using maximum failed logins, see alter role.

Logging in after lost password

Use the dataserver -p login_name parameter to specify the name of the system security officer or system administrator at the server startup. This allows you to set a new password for these account if there is no way to recover a lost password.
Establishing a password and login policy

When you start with the -p parameter, Adaptive Server generates, displays, and encrypts a random password and saves it in master.syslogins as that account’s new password.

You can use dataserver -p to reset the password for sa_role and sso_role. Use dataserver -p when you have lost the password for either of these roles, but they require a password to become active.

For example, if the server is started with:

```
dataserver -psa_role
```

Adaptive Server displays this message:

```
New password for role 'sa_role' : qjcdyrbfkxgc0
```

If sa_role does not have a password, and it is started with -psa_role, Adaptive Server prints an error message in the errorlog.

Sybase highly recommends that you change the password for the login or role when the server restarts.

Locking and unlocking logins and roles

A login or role can be locked when:

• Its password expires, or
• The maximum number of failed login attempts occur, or
• The system security officer locks the login or role manually.

❖ Locking and unlocking logins

• The system security officer can use sp_locklogin to lock or unlock a login manually.

For example:

```
sp_locklogin "joe" , "lock"
sp_locklogin "joe" , "unlock"
```

Information about the lock status of a login is stored in the status column of syslogins.

For details on the syntax and rules for using sp_locklogin, see sp_locklogin.
SMTP messages

- To send a message to a server, use the `sendmail` command.
- To receive a message, use the `acceptmail` command.
- To forward a message, use the `forwardmail` command.

The system administrator can use these commands to manage mail services.

Displaying password information

This section discusses displaying password information for logins and roles.

- **Displaying password information for specific logins**
  - Use `sp_displaylogin` to display the password settings for a login.

This example displays information about the login `joe`:

```bash
sp_displaylogin joe
Suid: 3
Loginame: joe
Fullname: 
Default Database: master
Default Language: 
Auto Login Script: 
Configured Authorization: 
Locked: NO
Date of Last Password Change: Sep 22 2008 3:50PM
Password expiration interval: 0
```
Establishing a password and login policy

Password expired: NO
Minimum password length: 6
Maximum failed logins: 1
Current failed login attempts: 2
Authenticate with: ANY
Login Password Encryption: SHA-256
Last login date: Sep 18 2008 10:48PM

For details on the syntax and rules, see sp_displaylogin in the Reference Manual: System Procedures.

❖ Displaying password information for specific roles

• Use sp_displayroles to display the password settings for a role.

This example displays information about the physician_role role:

    sp_displayroles physician_role, "display_info"
Role name = physician_role
Locked : NO
Date of Last Password Change : Nov 24 1997 3:35PM
Password expiration interval = 5
Password expired : NO
Minimum password length = 4
Maximum failed logins = 10
Current failed logins = 3

For details on the syntax and rules, see sp_displayroles in the Reference Manual: System Procedures.

Checking passwords for at least one digit

The system security officer can tell the server to check for at least one digit in a password using the server-wide configuration parameter, check password for digit. If set, this parameter does not affect existing passwords. By default, checking for digits is off.

This example activates the check password functionality:

    sp_configure "check password for digit", 1

This deactivates the check password functionality:

    sp_configure "check password for digit", 0

Setting and changing \textit{minimum password length}

In earlier versions of Adaptive Server, the minimum password length was a nonconfigurable, hard-coded value of six characters. The configurable password allows you to customize passwords to fit your needs such as using four-digit personal identification numbers (PINs) or anonymous logins with NULL passwords.

\textbf{Note} Adaptive Server uses a default value of 6 for \textit{minimum password length}. Sybase recommends that you use a value of 6 or more for this parameter.

The system security officer can specify:

- A globally enforced \textit{minimum password length}
- A per-login or per-role \textit{minimum password length}

The per-login or per-role value overrides the server-wide value. Setting \textit{minimum password length} affects only new passwords created after setting the value. It does not affect existing passwords.

\textbullet{} Setting \textit{minimum password length} for a specific login

\begin{itemize}
  \item To set the \textit{minimum password length} for a specific login at creation, use \texttt{sp_addlogin}.
  
  This example creates the new login “joe” with the password “Djdiek3”, and sets the \textit{minimum password length} for “joe” to 8:
  \begin{verbatim}
  sp_addlogin joe, "Djdiek3", @minpwdlen=8
  \end{verbatim}

  For details on the syntax and rules for using \textit{minimum password length}, see \texttt{sp_addlogin} in the Reference Manual: System Procedures.
\end{itemize}

\textbullet{} Setting \textit{minimum password length} for a specific role

\begin{itemize}
  \item To set the \textit{minimum password length} for a specific role at creation, use \texttt{create role}.
  
  This example creates the new role \texttt{intern\_role} with the password “temp244” and sets \textit{minimum password length} for \texttt{intern\_role} to 0:
  \begin{verbatim}
  create role intern\_role with passwd "temp244", min passwd length 0
  \end{verbatim}

  The original password is seven characters, but the password can be changed to one of any length because \textit{minimum password length} is set to 0.
\end{itemize}
Establishing a password and login policy

For details on the syntax and rules for using minimum password length, see create role in the Reference Manual: Commands.

❖ Changing minimum password length for a specific login
  • Use sp_modifylogin to set or change minimum password length for an existing login. sp_modifylogin effects only user roles, not system roles.

  **Example 1** Changes minimum password length for the login “joe” to 8 characters.

      sp_modifylogin "joe", @option="min passwd length", @value="8"

  **Note** The value parameter is a character datatype; therefore, quotes are required for numeric values.

  **Example 2** Changes the value of the overrides for minimum password length for all logins to eight characters.

      sp_modifylogin "all overrides", @option="min passwd length", @value="8"

  **Example 3** Removes the overrides for the minimum password length for all logins.

      sp_modifylogin "all overrides", "min passwd length", @value="-2"

  For details on the syntax and rules for using minimum password length, see sp_modifylogin.

❖ Changing minimum password length for a specific role
  • Use alter role to set or change minimum password length for an existing role.

  **Example 1** Sets the minimum length for physician_role, an existing role, to 5 characters:

      alter role physician_role set min passwd length 5

  **Example 2** Overrides the minimum password length for all roles:

      alter role "all overrides" set min passwd length -1

  For details on the syntax and rules for using minimum password length, see alter role in the Reference Manual: Commands.
Password complexity checks

You can use these options, which support password complexity checks, in a new stored procedure interface; their values are stored in the master.dbo.sysattributes table.

To turn off an individual option, enter:

```
sp_passwordpolicy 'clear', option
```

To turn off all password policy options, enter:

```
sp_passwordpolicy 'clear'
```


Disallowing simple passwords

disallow simple password checks to see if the password contains the login name as a substring. You can set it to:

- 0 – (default) turns off the option, and allows simple passwords.
- 1 – turns the option on, and disallows simple passwords.

To set this option, enter:

```
sp_passwordpolicy 'set', 'disallow simple passwords', 1
```

When you disallow simple passwords, you cannot use your login name as a substring in your password. You must set it to something complex. For example:

```
sp_password 'old_complex_password', BHotAcha789, johnd
```

The login johnd now has a password of BHotAcha789, which does not contain the login name as a substring.

However, if you change the login password entering the following, the login johnd is now a substring of the new password johnd123, and the command fails:

```
sp_password 'old_complex_password', johnd123, johnd
```

Custom password-complexity checks

Adaptive Server allows you to custom-configure password checking rules using sp_extrapwdchecks and sp_cleanpwdchecks.
These stored procedures are defined and located in the \texttt{master} database and are automatically invoked during Adaptive Server password complexity checks, and when dropping a login, respectively. See “Enabling custom password checks” on page 453 for an example of how to create these custom stored procedures.

### Specifying a minimum number of digits in a password

Use \texttt{min digits in password} to specify the minimum number of digits in a password. Valid values are:

- 0 through 16 – the minimum number of digits that must exist in a password.
- -1 – the password cannot contain digits.

By default, this password complexity option is turned off and this check is not applied to passwords.

To set this option, enter:

\begin{verbatim}
sp_passwordpolicy 'set', 'min digits in password', number
\end{verbatim}

For example, if you have set \texttt{min digits in password} to 4, you must have at least four digits in your password.

### Specifying a minimum number of alphabetic characters in a password

\texttt{min alpha in password} specifies the minimum number of alphabetic characters allowed in a password. This value must be at least the sum of minimum number of uppercase characters and minimum number of lowercase characters.

Valid values are:

- 0 through 16 – the number of alphabetic characters required to be in the password.
- -1 – the password cannot contain alphabetic characters.

By default, this password complexity option is turned off and this check is not applied to passwords.

To set the minimum number of alphabetic characters in a password, enter:

\begin{verbatim}
sp_passwordpolicy 'set', 'min alpha in password', number
\end{verbatim}

For example, if you have set \texttt{min alpha in password} to 4, you must have at least 4 alphabetic characters in your password.
CHAPTER 14 Managing Adaptive Server Logins, Database Users, and Client Connections

sp_addlogin 'johnd', 'sec123456'

Specifying a minimum number of special characters in a password

`min special char in password` specifies the minimum number of special characters for a password. Valid values are:

- 0 through 16 – the minimum number of special characters required for a password.
- -1 – the password cannot contain special characters.

By default, this password complexity option is turned off and this check is not applied to passwords.

To set the minimum number of special characters in a password, enter:

```
sp_passwordpolicy 'set', 'min special char in password', number
```

Specifying a minimum number of uppercase letters in a password

`min upper char in password` allows you to set the minimum number of uppercase letters for a password. Valid values are:

- 0 through 16 – the number of uppercase letters required for a password.
- -1 – the password cannot contain uppercase characters.

By default, this password complexity option is turned off and this check is not applied to passwords.

To set the minimum number of uppercase characters allowed in a password, enter:

```
sp_passwordpolicy 'set', 'min upper char in password', number
```

Specifying a minimum number of lowercase letters in a password

`min lower char in password` sets the minimum number of lowercase letters for a password. Valid values are:

- 0 through 16 – indicates the number of lowercase letters required for a password.
- -1 – indicates that the password cannot contain lowercase characters.

By default, this password complexity option is turned off and this check is not applied to passwords.
To set the minimum number of lowercase letters in a password, enter:

```
sp_passwordpolicy 'set', 'min lower char in password', number
```

For example, if you set `min lower char in password` to 3, you can create a login `johnd` with the password `abcdEF1#`, by entering:

```
sp_addlogin 'johnd', 'abcdEF1#'
```

However, if you attempt to add the following login and the minimum number of lowercase letters allowed is 3, the command fails:

```
sp_addlogin 'johnd', 'abCDE1'
```

### Specifying the minimum password length

`minimum password length` sets the minimum password length. You can set a minimum password length from 0 to 30. The value you specify with must be at least the sum of all other minimum requirements.

For example, `minimum password length` must be set to at least 10 if you have set:

- minimum digits in password to 3
- minimum special characters in password to 2
- minimum uppercase characters in password to 2
- minimum lowercase characters in password to 3

In this example, if the password length is less than 10, a warning message displays, but the setting of the password policy option succeeds.

To set the minimum password length, enter:

```
sp_passwordpolicy 'set', 'minimum password length', number
sp_password 'old_complex_password', 'joh12', 'johnd'
```

### Specifying the password expiration

`password expiration` specifies the number of days a password can exist before it expires. You specify this value on a global basis. Valid values include:

- 0 – the password will never expire.
- 1 through 32767 – the number of days the password can exist without expiring.
By default, this password complexity option is turned off and this check is not applied to passwords.

To specify the password expiration date, enter:

```
sp_passwordpolicy 'set', 'password expiration', number
```

**Specifying the password expiration warning interval**

`password exp warn interval` indicates the number of days before a password expires that the password expiration warning messages displays. These messages display with every successful login until the password is changed or it expires. This value must be less than or equal to the password expiration. Valid values are 0 to 365. This option is turned off by default.

To specify the password expiration warning interval, enter:

```
sp_passwordpolicy 'set', 'password exp warn interval', number
```

**Specifying the number of failed logins allowed**

`maximum failed logins` specifies the maximum number of failed logins that can occur before the login is locked. You specify this value globally. Valid values are:

- 0 – logins are never locked, regardless of the number of failed login attempts.
- 1 through 32767 – the number of failed logins that can occur before the login is locked.

By default, this value is turned off and this check is not applied to passwords.

To set the number of failed logins allowed, enter:

```
sp_passwordpolicy 'set', 'maximum failed logins', number
```

**Resetting the password at first login**

`expire login` changes the login status to expired when a system security officer creates or resets a login. The login is then required to change the password on the first login. Valid values are:

- 0 – new or reset logins will not expire.
- 1 – new or reset logins expire; you must reset your password at the first login.
Establishing a password and login policy

By default, this value is turned off and this check is not applied to passwords.

To require a change of password at first login, enter:

```
sp_password policy 'set', 'expire login', [ 1 | 0 ]
```

Password complexity option cross-checks

Some password complexity options have interaction implications:

- minimum password length **must be at least the sum of** min digits in password, min alpha in password, and min special characters in password.
- min alpha in password **must be at least the sum of** min upper char in password and min lower char in password.
- systemwide password expiration **must be greater than** password exp warn interval.

For the purpose of the above cross-checks, if Adaptive Server encounters a password complexity option value of -1, it interprets that as a value of 0. If an option is not set, Adaptive Server interprets the option value to be 0 as well.

Adaptive Server prints warnings for each new password complexity option that fails to satisfy the cross-checks. Option setting, however, is successful.

Setting old and new password complexity checks

**Table 14-13: Old and new password complexity checks**

<table>
<thead>
<tr>
<th>Password checks and policies for Adaptive Server authentication</th>
<th>Existing configuration parameters specified using sp_configure</th>
<th>New password complexity options specified using sp_passwordpolicy</th>
<th>Existing per-login overrides specified using sp_modifylogin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password expiration</td>
<td>system-wide password expiration</td>
<td>system-wide password expiration</td>
<td>password expiration</td>
</tr>
<tr>
<td>Digits in password</td>
<td>check password for digit</td>
<td>min digits in password</td>
<td>N/A</td>
</tr>
<tr>
<td>Alphabetic characters in password</td>
<td>N/A</td>
<td>min alpha in password</td>
<td>N/A</td>
</tr>
<tr>
<td>Password length</td>
<td>minimum password length</td>
<td>minimum password length</td>
<td>min passwd length</td>
</tr>
<tr>
<td>Failed logins lockout</td>
<td>maximum failed logins</td>
<td>maximum failed logins</td>
<td>max failed_logins</td>
</tr>
<tr>
<td>Disallow simple passwords</td>
<td>N/A</td>
<td>disallow simple passwords</td>
<td>N/A</td>
</tr>
<tr>
<td>Special characters in password</td>
<td>N/A</td>
<td>min special char in password</td>
<td>N/A</td>
</tr>
</tbody>
</table>
You can set the password complexity options at:

- The login level using `sp_addlogin` or `sp_modifylogin`.
- The global level using the new `sp_passwordpolicy` or `sp_configure`.

Because you can set password configuration options on a global and per-login basis, and using old and new parameters, the order of precedence in which the password options will be applied is important.

When applying password options, the order of precedence is:

1. Existing per-login parameters
2. New password complexity options
3. Existing global password options

### Examples

**Example 1** If you enter the following, you have set the minimum password length for “johnd” to 6:

```sql
sp_addlogin @login_name = 'johnd',
     @passwd = 'complex_password',
     @minpwdlen = 6
```

If you then enter the following existing global options for login “johnd”, you have created two different minimum password length requirements for login “johnd”, and you have also set restrictions about digits in the password:

```sql
sp_configure 'minimum password length', 8
sp_configure 'check password for digit', 'true'
sp_passwordpolicy 'set', 'min digits in password', 2
```

If you then try to create a password for login “johnd” as follows:
Establishing a password and login policy

sp_password @caller_password = 'old_complex_password',
@new_password = 'abcd123', @login_name = 'johnd'

Adaptive Server checks the password in the following order:

1. Per-login existing options check: minimum password length must be greater than 6. This is true and the check passes.
2. New options: minimum digits in password must be greater than 2. This is true and the check passes.
3. Existing global options: minimum password length specified here is not checked because there is already a per-login check for the login "johnd".
4. The check password for digit option is redundant because it is already checked when the minimum number of digits is turned on and set to 2.

Once these checks have been performed in the designated sequence, and the new password for login “johnd” passes these checks, the new password is successfully created.

Example 2  If, for the same login, you enter the following, Adaptive Server first checks the per-login existing options, and determines the minimum password length is set to 6, but that you have attempted to create a password with only 4 characters:

sp_password @caller_password = 'old_complex_password',
@new_password = 'abcd', @login_name = 'johnd'

The check fails, and Adaptive Server prints an error message. Once one password complexity check fails, no additional options are checked.

Example 3  If you attempt to create a new login with the following password configuration options, this sets the minimum password length for login johnd to 4:

sp_addlogin @login_name = 'johnd', @passwd = 'complex_password', @minpwdlen = 4

This is a per-login, existing option. If you then add the following, you have created a global requirement that the minimum number of digits for a password must be 1:

sp_passwordpolicy 'set', 'min digits in password', 1

If you then attempt to create the password for login johnd as follows:

sp_password @caller_password = 'old_complex_password',
@new_password = 'abcde', @login_name = 'johnd'

Adaptive Server performs the checks in the following order:
1 Per-login existing options check: the minimum password length of a new password is 4. The password “abcde” is greater than 4, so this check passes.

2 New global requirement check: the minimum digits in a password is set to 1, globally. This check fails.

Adaptive Server does not create a new password and prints an error message. To create a new password, all the checks must pass.

Enabling custom password checks

Adaptive Server allows a system security officer to write user-defined stored procedures that enable custom password checks.

For example, to implement password history checks, create a new user table to store password histories:

```sql
create table pwdhistory
(
    name varchar(30) not null,  -- Login name.
    password varbinary(30) not null,  -- old password.
    pwdate datetime not null,  -- datetime changed.
    changedby varchar(30) not null  -- Who changed.
) go
```

This user-defined stored procedure can be called when specifying a new password to save it in an encrypted form in the `pwdhistory` table:

```sql
create proc sp_extrapwdchecks
(
    @caller_password varchar(30), --the current password of caller
    @new_password varchar(30), -- the new password of the target acct
    @loginame varchar(30), -- user to change password on
)
begin
    declare @current_time datetime,
            @encrypted_pwd varbinary(30),
            @changedby varchar(30),
            @cutoffdate datetime
    select @changedby = suser_name()
```

```sql
```

```sql
```

```sql```
Establishing a password and login policy

-- Change this line according to your installation.
-- This keeps history of 12 months only.
select @current_time = getdate(),
       @cutoffdate = dateadd(month,-12,getdate())
select @encrypted_pwd = internal_encrypt(@new_password)

delete master..pwdhistory
  where name = @loginame
  and    pwdate < @cutoffdate
if not exists ( select 1 from master..pwdhistory
  where name = @loginame
  and   password = @encrypted_pwd )
begin
  insert master..pwdhistory
  select @loginame, internal_encrypt(@caller_password),
  @current_time, @changedby
  return (0)
end
else
begin
  raiserror 22001  --user defined error message
end
end

Use sp_addmessage to add the user-defined message 22001. A raiserror 22001 indicates a custom password-complexity check error and leads to a failure of sp_addlogin or sp_password.

The following user-defined stored procedure can be used for clean-up purposes after adding history using sp_extrapwdchecks.

create proc sp_cleanpwdchecks
  ( @loginame varchar(30) -- user to change password on )
as
begin
  delete master..pwdhistory
  where name = @loginame
  go
Once the two procedures above are defined and installed in the master database, they are called dynamically during the password complexity checks.

### Setting the login and role expiration interval for a password

System administrators and system security officers can:

<table>
<thead>
<tr>
<th>Use</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>sp_addlogin</td>
<td>Specify the expiration interval for a login password at creation</td>
</tr>
<tr>
<td>sp_modifylogin</td>
<td>Change the expiration interval for a login password. sp_modifylogin affects only user roles, not system roles.</td>
</tr>
<tr>
<td>create role</td>
<td>Specify the expiration interval for a role password at creation</td>
</tr>
<tr>
<td></td>
<td>(only the system security officer can issue create role)</td>
</tr>
<tr>
<td>alter role</td>
<td>Change the expiration interval for a role password (only the system security officer can issue alter role)</td>
</tr>
</tbody>
</table>

The following rules apply to password expiration for logins and roles:

- A password expiration interval assigned to individual login accounts or roles overrides the global password expiration value. This allows you to specify shorter expiration intervals for sensitive accounts or roles, such as system security officer passwords, and more relaxed intervals for less sensitive accounts such as an anonymous login.

- A login or role for which the password has expired is not directly activated.

- The password expires at the time of day when the password was last changed after the number of days specified by password expiration interval has passed. password expiration interval specifies the password expiration interval in days. It can be any value between 0 and 32767, inclusive. For example, if you create a new login on August 1, 2007 at 10:30 AM, with a password expiration interval of 30 days, the password expires on August 31, 2007 at 10:30 AM

For details on the syntax and rules for the commands and system procedures, see the Reference Manual.
Establishing a password and login policy

Password expiration turned off for pre-12.x passwords

Password expiration did not affect roles in versions prior to Adaptive Server 12.x. Therefore, in Adaptive Server 12.x and later, password expiration is deactivated for any existing user-defined role passwords. During the upgrade, all user-defined role passwords are stamped as having a password interval of 0.

Message for impending password expiration

When a password for a login or role is about to expire, a warning message asks the user to contact the system security officer.

Circumventing password protection

Circumventing the password-protection mechanism may be necessary in the case of automated login systems. You can create a role that could access other roles without passwords.

If a system security officer wants to bypass the password mechanism for certain users, the system security officer can grant the password-protected role to another role and grant this new role to one or more users. Activation of this role automatically activates the password-protected role without having to provide a password.

For example:

Jane is the system security officer for the fictitious company ABC Inc., which uses automated login systems. Jane creates the following roles:

- financial_assistant
  
  create role financial_assistant with passwd "L54K3j"

- accounts_officer
  
  create role accounts_officer with passwd "9sF6ae"

- chief_financial_officer
  
  create role chief_financial_officer

Jane grants the roles of financial_assistant and accounts_officer to the chief_financial_officer role:

  grant role financial_assistant, accounts_officer to chief_financial_officer

Jane then grants the chief_financial_officer role to Bob:

  grant role chief_financial_officer to bob
Bob logs in to Adaptive Server and activates the `chief_financial_officer` role:

```sql
set role chief_financial_officer on
```

The roles of `financial_assistant` and `accounts_officer` are automatically activated without Bob providing a password. Bob can now access everything under the `financial_assistant` and `accounts_officer` roles without having to enter the passwords for those roles.

**Creating a password expiration interval for a new login**

Use `sp_addlogin` to set the password expiration interval for a new login.

This example creates the new login “joe” with the password “Djdiek3”, and sets the password expiration interval for “joe” to 2 days:

```sql
sp_addlogin joe, "Djdiek3", null, null, null, 2
```

The password for “joe” expires after 2 days from the time of day the login account was created, or 2 days from when the password was last changed.

*password expiration interval* specifies the password expiration interval in days. It can be any value between 0 and 32767, inclusive. For example, if you create a new login on August 1, 2007 at 10:30 AM, with a password expiration interval of 30 days, the password expires on August 31, 2007 at 10:30 AM.

For details on the syntax and rules for using the new parameter, see `sp_addlogin` in the *Reference Manual: System Procedures*.

**Creating a password expiration interval for a new role**

Use `create role` to set the password expiration interval for a new role.

This example creates the new role `intern_role` with the password “temp244”, and sets the password expiration interval for `intern_role` to 7 days:

```sql
create role intern_role with passwd "temp244", passwd expiration 7
```

The password for `intern_role` expires after 7 days from the time of day you created the role, or 2 days from when the password was last changed.

*password expiration interval* specifies the password expiration interval in days. It can be any value between 0 and 32767, inclusive. For example, if you create a new login on August 1, 2007 at 10:30 AM, with a password expiration interval of 30 days, the password expires on August 31, 2007 at 10:30 AM.

For details on the syntax and rules for using passwd expiration, see `create role` in the *Reference Manual: Commands*. 
Establishing a password and login policy

Creation date added for passwords

Passwords are stamped with a “creation date” equal to the upgrade date of a given server. The creation date for login passwords is stored in the pwdate column of syslogins. The creation date for role passwords is stored in the pwdate column of syssrvroles.

Changing or removing password expiration interval for login or role

Use sp_modifylogin to change the password expiration interval for an existing login, add a password expiration interval to a login that did not have one, or remove a password expiration interval. sp_modifylogin only effects login passwords, not role passwords.

Example 1  Changes the password expiration interval for the login “joe” to 5 days:

sp_modifylogin "joe", @option="passwd expiration", @value="5"

The password expires 5 days from the time of day you ran password expiration.

Example 2  Changes the value of the overrides for the password expiration for all logins to 3 days:

sp_modifylogin "all overrides", @option="passwd expiration", @value="3"

Example 3  Removes the value of the overrides for the password expiration for all logins:

sp_modifylogin "all overrides", @option="passwd expiration", @value="-1"

Securing login passwords on the network

Adaptive Server allows the use of asymmetric encryption to securely transmit passwords from client to server using the RSA public key encryption algorithm. Adaptive Server generates the asymmetric key pair and sends the public key to clients that use the new login protocol. The client encrypts the user’s login password with the public key before sending it to the server. The server decrypts the password with the private key to begin the authentication of the client connecting.

You can configure Adaptive Server to require clients to use this protocol. Set the Adaptive Server configuration parameter `net password encryption reqd` to require all username- and password-based authentication requests to use RSA asymmetric encryption. See “`net password encryption reqd`” on page 169 for more information.

Generating an asymmetric key pair

Adaptive Server generates a new key pair:

- At each server startup,
- Automatically at 24-hour intervals using the Adaptive Server housekeeper mechanism, and
- When an administrator with `sso_role` requests key pair regeneration.

The key pair is kept in memory. A message is recorded in the errorlog and in the audit trail when the key pair is regenerated.

To generate the key pair on demand, use:

```
sp_passwordpolicy "regenerate keypair"
```

**Note** Depending on the system load, there may be a delay between the time this command is executed and the time the key pair is actually generated. This is because the housekeeper task runs at a low priority and may be delayed by higher priority tasks.

To generate the key pair at a specific time, use:

```
sp_passwordpolicy "regenerate keypair", "datetime string"
```

For example, a datetime string of “Jan 16, 2007 11:00PM” generates the key pair at the specified time. The datetime string can also just be a time of day, such as “4:07AM”. When only time of day is specified, key pair regeneration is scheduled for that time of day in the next 24 hour period.
Establishing a password and login policy

Server option "net password encryption"

Adaptive Server also acts as a client when establishing a remote procedure call (RPC).

When connecting to remote servers, Adaptive Server uses the "net password encryption" server option to determine whether it will use password encryption.

Adaptive Server uses either RSA or Sybase proprietary algorithms when this server option is set to "true." The command to enable "net password encryption" is:

```
sp_serveroption server, "net password encryption", "true"
```

The setting is stored in master..sysservers and the value of server options are displayed using `sp_helpserver` stored procedure.

In this release, the default value for `net password encryption` is "true" for any new server added using `sp_addserver`. During upgrade, Adaptive Server sets `net password encryption` to "true" for `sysservers` entries with an ASEnterprise class value. No other server classes are modified. This is done to improve password security between two communicating Adaptive Servers.

Note The administrator may reset `net password encryption` to false if you encounter problems establishing a connection to a server. However, if the option is set to false, your password is transmitted in clear text on the network.

Backward compatibility

- Sybase recommends that you use the RSA algorithm to protect passwords on the network.
- To use the RSA algorithm, you must have Adaptive Server version 15.0.2 and new Connectivity SDK clients (version 15.0 ESD#7 and later.) Sybase provides the `net password encryption reqd` configuration parameter and the `net password encryption server option` to allow settings equivalent to pre-15.0.2 release and maintain backward compatibility with older clients and older servers.
- Older clients that do not support RSA algorithm can set the property to encrypt passwords using the Sybase proprietary algorithm, available since pre-12.0 releases. Adaptive Server then uses the Sybase proprietary algorithm.
New clients that support both RSA and Sybase proprietary algorithms can set properties for both algorithms. When communicating with such clients, Adaptive Server 15.0.2 uses RSA encryption. A pre-15.0.2 Adaptive Server uses Sybase’s proprietary algorithm.

Securing login passwords stored on disk and in memory

Login passwords used by Adaptive Server to authenticate client connections are stored securely on disk as SHA-256 hash digest. The SHA-256 algorithm is a one-way encryption algorithm. The digest it produces can not be decrypted, making its storage on disk secure. To authenticate the user connection, the SHA-256 algorithm is applied to the password sent by the client and the result compared with the value stored on disk.

To prevent dictionary-based attacks on login passwords stored on disk, a salt is mixed with the password before the SHA-256 algorithm is applied. The salt is stored alongwith the SHA-256 hash and used during login authentication.

To ease the transition to the new on-disk encryption algorithm when migrating from pre-15.0.2 releases, Adaptive Server includes the password policy allow password downgrade. Upon upgrade from pre-15.0.2 releases, the policy has value “1” to indicate that passwords are stored in both the Sybase proprietary algorithm used in earlier releases and the new SHA-256 algorithm used in Adaptive Server 15.0.2.

As long as passwords are stored in both old and new forms, Adaptive Server can be downgraded to Adaptive Server 15.0 or 15.0.1 without resetting user passwords. When the policy allow password downgrade is set to 0, passwords will only be stored in new SHA-256 form, which is incompatible with older releases. When downgrading to previous releases, only passwords stored in SHA-256 are reset to random passwords and stored in the old form compatible with older releases. For more information, see “Backward compatibility” on page 460.

Due to the superior password security of the SHA-256 algorithm, Sybase recommends only using SHA-256 as soon the administrator is certain that downgrade to an earlier release will not be done. But consider the trade-offs when making this decision; should there be a need to downgrade to a pre-15.0.2 release, it requires administrator intervention to unlock user login passwords.
Establishing a password and login policy

Using only the SHA-256 algorithm

You can downgrade passwords when user password updates are retained in both the old and new encoding. Adaptive Server retains the old encoding to use after downgrade to an earlier release (for example, 15.0.1).

To end the period when password downgrade is allowed, execute:

```sql
sp_passwordpolicy set, "allow password downgrade", 0
```

Before executing this command, administrators should examine login accounts with `sp_displaylogin` to determine if the login account has been used, and whether the password is stored in the new SHA-256 encoding. If the password is not encoded with the new encoding, the account is locked by this action, and the password is reset with a generated password. To use the account again, administrator intervention is required to unlock the account and give the user a newly generated password.

The administrator may want to save the output from this command because it can contain information about locked login accounts and generated passwords for those accounts.

Adaptive Server takes the following actions when the password downgrade period ends:

- The datetime when the password downgrade period ended is recorded in `master.dbo.sysattributes`.
- The value of each `password` column in `syslogins` is rewritten to only use the new password on-disk structure.
- The logins that have not transitioned to the new algorithm have the password reset to a new server-generated password in the new SHA-256 format, and the login is locked. The generated password is displayed only to the administrator executing the `sp_passwordpolicy` procedure above. The lock reason is set to 3 ("Login or role not transitioned to SHA-256").

After the `sp_passwordpolicy` procedure completes, the following behavior is expected:

- The authentication of logins is done using the new algorithm only.
- Only the new password on disk structure for the `password` column is used.
Attempts to use the locked logins will fail authentication. In order to use the logins that were locked, the administrator must unlock the login with `sp_locklogin` and you must use the password generated by `sp_passwordpolicy`. Alternatively, the administrator may prefer to assign a new password instead of the generated password for locked login accounts.

Example 1

This example prepares an upgraded server to use only SHA-256. Examine login accounts to determine which encryption is used by the account using `sp_displaylogin`.

```
1> sp_displaylogin login993
2> go
Suid: 70
Loginame: login933
Fullname: 
Default Database: master
Default Language: 
Auto Login Script: 
Configured Authorization: 
Locked: NO
Date of Last Password Change: Apr 20 2007 2:55PM
Password expiration interval: 0
Password expired: NO
Minimum password length: 0
Maximum failed logins: 3
Current failed login attempts: 
Authenticate with: ANY
Login Password Encryption: SYB-PROP 
Last login date: 
(return status = 0)
```

The value “SYB-PROP” from the line `Login Password Encryption: SYB-PROP` indicates that only the old, Sybase proprietary password encryption is used for this account. This login has not been used since upgrade to Adaptive Server version 15.0.2 and would be locked and password reset if the procedure “`sp_passwordpolicy 'set', 'allow password downgrade', 0'`” is executed.

After the first login to the account after upgrading to Adaptive Server 15.0.2, the line would changed to show value “SYB-PROP, SHA-256” to show that both old and new encryption is used:

```
Login Password Encryption: SYB-PROP,SHA-256
```

This is the desired state for all active login accounts, so that executing “`sp_passwordpolicy 'set', 'allow password downgrade', 0'`” does not lock and reset the password for accounts.
Establishing a password and login policy

After the "sp_passwordpolicy 'set', 'allow password downgrade', 0" procedure is executed, only the new encryption is used, and the following line appears:

```
Login Password Encryption: SHA-256
```

Login accounts that show this value are fully transitioned to use the new, stronger, on-disk encryption algorithm.

**Example 2**

This example executes "sp_passwordpolicy 'set', 'allow password downgrade', 0" when all accounts have transitioned to new algorithm

When all passwords have been changed to use the new algorithm, executing the command shows no accounts reset or locked.

```
1> sp_passwordpolicy 'set', 'allow password downgrade', 0
2> go
Old password encryption algorithm usage eliminated from 0 login accounts, changes are committed.
(return status = 0)
```

**Example 3**

This example executes "sp_passwordpolicy 'set', 'allow password downgrade', 0" with accounts that have not transitioned to new algorithm

In this example, 990 out of 1000 login accounts have transitioned to using the new SHA-256 algorithm, but 10 accounts were still using SYB-PROP algorithm:

```
1> sp_passwordpolicy 'set', 'allow password downgrade', 0
2> go
Old password encryption algorithm found for login name login1000, suid 3, ver1 =5, ver2 = 0, resetting password to EcJxKmMvOrDsC4
Old password encryption algorithm found for login name login999, suid 4, ver1 =5, ver2 = 0, resetting password to MdZcUaFpXkFtM1
Old password encryption algorithm found for login name login998, suid 5, ver1 =5, ver2 = 0, resetting password to ZePiZdSeMqBdE6
Old password encryption algorithm found for login name login997, suid 6, ver1 =5, ver2 = 0, resetting password to IfWpXvGlBbDgW7
Old password encryption algorithm found for login name login996, suid 7, ver1 =5, ver2 = 0, resetting password to JhDjYnGcXwObI8
Old password encryption algorithm found for login name login995, suid 8, ver1 =5, ver2 = 0, resetting password to QaX1RuJ1CrPaE6
Old password encryption algorithm found for login name login994, suid 9, ver1 =5, ver2 = 0, resetting password to HlHcZdRrYcKyB2
Old password encryption algorithm found for login name login993, suid 10, ver1 =5, ver2 = 0, resetting password to UvMrXoVqKmZvU6
Old password encryption algorithm found for login name login992, suid 11, ver1 =5, ver2 = 0, resetting password to IxIwZqHxEePbX5
Old password encryption algorithm found for login name login991, suid 12,
```
ver1 =5, ver2 = 0, resetting password to HxYrPyQbLzPmJ3
Old password encryption algorithm usage eliminated from 10 login accounts, changes are committed.
(return status = 1)

Note: The login name, suid, and generated password are displayed to the administrator executing the procedure. The output of the command shows all 10 accounts that have not transitioned are reset (and locked).

Character set considerations for passwords

Passwords and other sensitive data that is encrypted must consider the character set of the clear text to accurately interpret the result when it is decrypted or when hash values are compared during authentication.

For example, a client connects to Adaptive Server using isql and establishes a new password. Regardless of the character set used in the client, characters are always converted to the server’s default character set for processing within Adaptive Server. Assuming the Adaptive Server default character set is “iso_1,” consider the procedure call:

```
sp_password old_passwd, new_passwd
```

The parameters are of type varchar and are expressed as a quoted string and stored with “iso_1” encoding before encryption. If the Adaptive Server default character set changes later, the encrypted password remains an encrypted string of characters encoded with the old default character set. This could result in authentication failure due to mismatched character mapping. Although changing default character set is a rare occurrence, it becomes more important when migration occurs between platforms.

To address this limitation with character sets, Adaptive Server converts the clear text password to canonical form before encryption so that it can be used across platforms, chip architectures, and character sets.

To use canonical form for storage in syslogins:

1. Convert clear text password string to UTF-16.
2. Convert the UTF-16 string to network byte order.
3. Append a small buffer (the salt) with random bytes to the password.
4. Apply SHA-256 hash algorithm.
Establishing a password and login policy

5 Store digest, salt, and version in the password column.

At authentication time, the steps are:
1 Convert clear text password string to UTF-16.
2 Convert the UTF-16 string to network byte order.
3 Append the salt from the password column in syslogins to the password.
4 Apply the hash algorithm.
5 Compare results with password column in syslogins, if they match then authentication is successful.

Use of the new SHA-256 algorithm with the above steps allow passwords to be transported across platforms, chip architectures, and character sets.

Upgrade and and downgrade behavior

Behavior changes on upgraded master database

In an upgraded Adaptive Server master database, the server maintains encrypted passwords in syslogins catalogs using both old and new algorithms in the password column.

Users can call sp_displaylogin to determine which “Login password encryption” a login uses.

On first authentication of a login after an upgrade:
- The user authenticates using the contents of the password column and the old algorithm.
- Adaptive Server then updates the password column with the old encryption algorithm followed by the new encryption algorithm.

On subsequent authentication of a login after upgrade, before “allow password downgrade” is set to 0, the user authenticates using the new algorithm.
Behavior changes on new master database

In a new Adaptive Server 15.0.2 master database, or an upgraded master
database after "allow password downgrade" is set to 0, the server maintains
encrypted passwords in syslogins catalog using only the new algorithm in
password column. Only the SHA-256 algorithm is used to authenticate the
connection requests and for storing the password on disk.

The administrator can distinguish a server that has been upgraded and
maintaining passwords with both old and new algorithms from a server with a
new master database that only uses the new algorithm by executing:

```
sp_passwordpolicy "list", "allow password downgrade"
```

Retaining password encryption after upgrade and downgrade

You can upgrade to Adaptive Server release 15.0.2 and later to use new
password encryption and retain the ability to downgrade to an earlier release.
This password downgrade period is introduced only for upgrade in the event
that problems arise after upgrade and the administrator chooses to downgrade
back to Adaptive Server 15.0.1, for example. The ability to downgrade
passwords is enabled by default upon upgrade and remains in effect until an
administrator chooses to end this by disabling it. While enabled, only the new
functionality is lost by downgrade to Adaptive Server 15.0.1, the password
changes made by users are not lost. All changes take place in master database;
no user databases are affected.

Note If you run sp_downgrade then shutdown the server and for some reason
reboot the 15.0.2 server all the downgrade changes are undone. In this case you
must run sp_downgrade again.

Installation

The master database contains the syslogins and syssrvroles tables. When the
new encryption algorithms and password policies are enabled on Adaptive
Server, additional disk space in master database and transaction log is
needed. The database administrator should use the alter database command to
add sufficient space to the master database and transaction log to handle the
additional disk requirements for the user population of the site.
The space for `syslogins` in master database is estimated to increase by about 30% for the same number of users. The maximum row length is increased 135 bytes per login account (row in `syslogins`). The ratio of rows per page has decreased from about 16 rows per 2k page to 12 rows per 2k page between Adaptive Server versions 15.0.1 and 15.0.2. During the period when the value for "allow password downgrade" is 1, when both old and new password encryption algorithms are used, the ratio further decreases to about 10 rows per 2K page.

For example, if a server has 1,000 login accounts in Adaptive Server 15.0.1 and data pages fit into 59 pages, it may require about 19 additional pages in Adaptive Server 15.0.2 on a new master database or 33 pages additional pages if upgraded from 15.0.1 and the value of "allow password downgrade" is 1.

Additional space in transaction log is required for the updated `password` column at first login (829 2K pages per 1,000 logins), for password changes made by users and during upgrade and downgrade actions (about 343 pages per 1,000 logins). To ensure that sufficient log space is available, verify that there is about one page (2K page) of free log space per login available before password upgrade or downgrade actions are taken, and while users make their first login to Adaptive Server 15.0.2.

On bootstrap of a new master database, the value for “allow password downgrade” is NULL.

Note  This value is different for a bootstrap of master databases than for an upgraded master database.

### Upgrade

The following actions are taken during an upgrade from an earlier Adaptive Server release (not in a new master database):

- Schema changes to `syslogins` and `syssrvroles`.
- Add a new row to `sysattributes` for “allow password downgrade,” set with value 1 (default). The ability to downgrade passwords is kept on upgraded master databases until the stored procedure “sp_passwordpolicy set, 'allow password downgrade', 0” is called to disable it. The system attribute “allow password downgrade” is enabled by default upon upgrade.
Add a new row to `sysattributes` for “enable last login updates,” set with value 1 (default). The update of `syslogins` column `lastlogindate` is controlled by this attribute value. The administrator can disable this behavior by calling stored procedure `“sp_passwordpolicy set, 'enable last login updates', 0.”

### Downgrade

Downgrade to Adaptive Server release 15.0 or 15.0.1 is supported, and can happen without the administrator taking action for each login account.

If “allow password downgrade” is 0 or NULL, or a password has otherwise been stored in `syslogins` with only the new SHA-256 algorithm, then the password is reset and the account is locked during downgrade. This will require administrator action before the account may be used again. Use `sp_displaylogin` on login accounts to determine what algorithm is used or `sp_downgrade "prepare"` to determine what accounts will be reset.

To know what actions will be taken and to verify that `sp_downgrade` may run successfully, you can perform a dry run using the “prepare” option, such as:

```
1> sp_downgrade 'prepare','15.0.1',1
2> go

Checking databases for downgrade readiness.

There are no errors which involve encrypted columns.

Allow password downgrade is set to 0. Login passwords may be reset, if old encryption version of password is not present.

Warning: New password encryption algorithm found for login name user103, suid 103.

Password will be reset during the downgrade phase.

sp_downgrade 'prepare' completed.
(return status = 0)
```

In the above example, the login “user103” was found to have only the new password format that is not used on previous releases. If downgrade occurs, the password will be reset to a random password and the account locked.

This procedure runs through actions and report readiness for downgrade to occur. The downgrade action does not occur if `prepare` does not succeed. For login passwords, it will report which passwords will be reset.
Establishing a password and login policy

Before executing `sp_downgrade`, Sybase recommends removing the login 'probe' from `syslogins`. To remove the login, connect to Adaptive Server as a system security officer or system administrator, and execute:

```sql
sp_droplogin 'probe'
```

If the login has user entries in databases, use the following command to drop users from databases and then drop the login.

```sql
use master
sp_dropuser 'probe'
```

The 'probe' login will be re-created when the `installmaster` script is run on the downgraded server.

Before executing `sp_downgrade`, Sybase recommends that you drop statistics for `syslogins` and `syssrvroles`. This avoids invalid column information, such as the length of password column, in `sysstatistics` after you perform a downgrade.

To drop statistics for `syslogins` and `syssrvroles`, enter:

1> delete statistics master..syslogins
2> delete statistics master..syssrvroles
3> go

The actions to downgrade password occur when stored procedure `sp_downgrade` is executed. For example:

```sql
1> sp_downgrade 'downgrade','15.0.1',1
2> go
```

Checking databases for downgrade readiness.
There are no errors which involve encrypted columns.

Allow password downgrade is set to 0. Login passwords may be reset, if old encryption version of password is not present.
Warning: New password encryption algorithm found for login name user103, suid 103 . Password will be reset during the downgrade phase.

Executing downgrade step 1 [sp_passwordpolicy 'downgrade'] for :
- Database: master (dbid: 1)

New password encryption algorithm found for login name user103, suid 103 . Resetting password to 'ZdSuPnPnBxAbW9'.

Total number of passwords reset during downgrade = 1
In the above example, the execution of \texttt{sp\_downgrade} resulted in the locking and password reset of login user103. The random password generated by Adaptive Server is shown only to the client executing \texttt{sp\_downgrade}. The administrator can redirect this output to a file so that these passwords are retained, or the administrator can reset them manually after rebooting on the downgraded master.

Additional messages appear in the errorlog to identify steps that occurred during \texttt{sp\_downgrade} and any system errors that may occur. Example errorlog output for the example downgrade procedure follows:

```
00:00000:00006:2007/05/21 05:34:07.81 server  Preparing ASE downgrade from 1502 to 1501.
00:00000:00006:2007/05/21 05:35:59.09 server  Preparing ASE downgrade from 1502 to 1501.
00:00000:00006:2007/05/21 05:35:59.19 server  Starting downgrading ASE.
00:00000:00006:2007/05/21 05:35:59.20 server  Downgrade : Downgrading login passwords.
00:00000:00006:2007/05/21 05:35:59.22 server  Downgrade : Starting password downgrade.
00:00000:00006:2007/05/21 05:35:59.23 server  Downgrade : Removed sysattributes rows.
00:00000:00006:2007/05/21 05:35:59.23 server  Downgrade : Updated 1 passwords.
00:00000:00006:2007/05/21 05:35:59.24 server  Downgrade : Removed columns in syslogins - lastlogindate, crdate, locksuid, lockreason, lockdate are removed.
00:00000:00006:2007/05/21 05:35:59.26 server  Downgrade : Truncated password lengths.
00:00000:00006:2007/05/21 05:35:59.28 server  Downgrade : Successfully completed password downgrade.
00:00000:00006:2007/05/21 05:35:59.28 server  Downgrade : Marking stored procedures to be recreated from text.
00:00000:00006:2007/05/21 05:36:03.69 server  Downgrade : Dropping Sysoptions system table.
00:00000:00006:2007/05/21 05:36:03.81 server  Downgrade : Setting master database minor upgrade version.
00:00000:00006:2007/05/21 05:36:03.83 server  Downgrade : Setting user databases minor upgrade version.
```
Establishing a password and login policy

This procedure makes the catalog changes and modifies password data to allow reverting to Adaptive Server version 15.0.1. The server must be in single-user mode in order to successfully execute `sp_downgrade`. A `dataserver` started with the "-m" command line option starts the server in single-user mode and allows only the `sa` to login.

After `sp_downgrade` is run, the only safe thing to do is shut down the server to avoid new logins or other actions that may modify data or system catalogs. If restarting Adaptive Server at version 15.0.2 after `sp_downgrade` has successfully executed and server is shutdown, internal upgrade actions are performed again and the changes to system catalogs are upgraded to version 15.0.2 level. If Adaptive Server 15.0.2 was restarted before booting the 15.0.x release to which you are reverting it is necessary to rerun `sp_downgrade`.

See details on other aspects of downgrade in the the “Downgrade” section of the Adaptive Server release bulletin.

Expiring passwords when `allow password downgrade` is set to 0

An administrator can choose to “expire” passwords in `syslogins` at the end of the password downgrade period to void copies of password in backups, transaction logs, and other persistent storage where they were kept with the old algorithm.

To configure login passwords to expire, use:

```
sp_passwordpolicy "expire login passwords", ["loginame | wildcard"]
```

To configure role passwords to expire, use:

```
sp_passwordpolicy "expire role passwords", ["rolename | wildcard"]
```

Expiration of passwords can be tuned to expire passwords of logins and roles with passwords that have not been changed after a datetime specified.

To configure stale login passwords to expire, use:

```
sp_passwordpolicy "expire stale login passwords", "datetime"
```

To configure stale role passwords to expire, use:

```
sp_passwordpolicy "expire stale role passwords", "datetime"
```
By executing `sp_passwordpolicy "expire stale login passwords"` and setting the `datetime` parameter with the value when the password downgrade period ended, then all login passwords that have not changed since that time will have their passwords expired. The benefit of expiring passwords after the password downgrade period ends is to require the user to change to a different password thereby making the old password useless. Additionally, this is done without requiring additional actions by the administrator.

A password that is encrypted with the old algorithm during the password downgrade period may remain in old pages, transaction logs, and other storage in the less securely encrypted form. An old password that is different than the current password prevents successful exploits on the old encryption algorithm by making the old password useless.

A stronger approach for the administrator to take is to lock stale logins or roles; however this would require the administrator to reset the password manually for legitimate users to access their login account again.

### Showing the current value of `allow password downgrade`

Use this to obtain the current value of attribute `allow password downgrade`:

```
sp_passwordpolicy list, "allow password downgrade"
```

The result set contains the current value and a message indicating its meaning.

For an upgraded `master` database that is still maintaining passwords in the old and new encodings, the result is:

```
sp_passwordpolicy list, "allow password downgrade"
go
value message
----------------- ---------------------------------------
1 Password downgrade is allowed.
(1 row affected)
```

For an upgraded `master` database that only uses new password encryption, the result is:

```
sp_passwordpolicy list, "allow password downgrade"
go
value message
----------------- ---------------------------------------
0 Last Password downgrade was allowed on <datetime>.
(1 row affected)
Establishing a password and login policy

For a new master database on Adaptive Server 15.0.2 that only uses new password encryption, the result is:

```
sp_passwordpolicy list, "allow password downgrade"
go
value message
-------- -----------------------------------------------------
NULL New master database.
(1 row affected)
```

Last login and locking inactive accounts

Adaptive Server provides security for user accounts by tracking the creation date, the last login time for an account, and the ability to determine which accounts are stale and may be locked due to inactivity. Additional information is kept to record the reason for an account to be locked and the identity of the user who locked the account.

Using syslogins to track if an account is locked

syslogins includes the lastlogindate, crdate, locksuid, lockreason, and lockdate columns to support last login and locking inactive accounts. These columns allow an account owner or administrator to know if an account is locked, when it was locked, who locked it, and the reason why it was locked.

At login creation, the new crdate column is set to current time.

At login time, if the "enable last login updates" password policy option is set to "1," the lastlogindate column is set to the current datetime and the previous value of the column is stored in the PSS of the login session. The update to syslogins and the PSS can occur at each login to Adaptive Server. The default value for this option in a new master database or an upgraded database is "1." The administrator can choose to disable this option by executing the procedure:

```
sp_passwordpolicy "set", "enable last login updates", 0
```

@@lastlogindate is available to each user login session and is specific to each login session and can be used by that session to determine the date and time of the previous login to the account. If the account has not been used previously or "enable last login updates" is 0, then the value of @@lastlogindate is NULL.

Updates to column lastlogindate in syslogins are not logged in the transaction log, because adding a log record for every login would cause master database to fill up quickly.
An administrator with sso_role can lock login accounts that have been inactive for a given number of days using the following sp_locklogin command:

```sql
sp_locklogin 'all', 'lock', [@except], 'number of inactive days'
```

This command has no effect if “enable last login updates” is set to “0” or the value of the lastlogindate column is NULL. The value of `number of inactive days` can be in the range 1 to 32767 (days).

When a login is locked using such a command, the `lockreason` column is updated with the reason. The value of the `lockdate` column is set to the current datetime.

When an account is unlocked, columns `lockreason`, `lockdate`, and `locksuid` are reset to NULL.

The `lockdate`, `locksuid` and `lockreason` columns are set internally by Adaptive Server. Table 14-14 describes the reasons and the value of `locksuid` in this release.

<table>
<thead>
<tr>
<th><code>lockreason</code></th>
<th><code>locksuid</code></th>
<th>Description of lockreason of account</th>
</tr>
</thead>
<tbody>
<tr>
<td>NULL</td>
<td>NULL</td>
<td>Account has not been locked.</td>
</tr>
<tr>
<td>0</td>
<td>suid of caller of <code>sp_locklogin</code></td>
<td>Account locked by <code>locksuid</code> by manually executing <code>sp_locklogin</code>.</td>
</tr>
<tr>
<td>1</td>
<td>suid of caller of <code>sp_locklogin</code></td>
<td>Account locked by <code>locksuid</code> due to account inactive, by manually executing <code>sp_locklogin 'all', 'lock', 'ndays'</code>.</td>
</tr>
<tr>
<td>2</td>
<td>suid of attempted login</td>
<td>Account locked by Adaptive Server due to failed login attempts reaching max failed logins.</td>
</tr>
<tr>
<td>3</td>
<td>suid of caller of <code>sp_passwordpolicy</code> set, &quot;allow password downgrade&quot;, 0</td>
<td>Account locked by <code>locksuid</code> as the password downgrade period has ended and login or role has not transitioned to SHA-256.</td>
</tr>
</tbody>
</table>

### Using passwords in a high availability environment

Password security impacts configuration of High Availability and the behavior of passwords in `syslogins` between primary and companion servers.
Establishing a password and login policy

High availability configuration

The primary and companion servers must have equivalent allow password downgrade values before they are configured for high availability. The allow password downgrade quorum attribute checks whether the value of allow password downgrade is same on both primary and secondary servers. This high availability advisory check succeeds when the value for the quorum attribute is the same, and fails when the values are different.

For example if value of allow password downgrade on primary server is set to 1 and on secondary server it is set to 0, then you see an output of `sp_companion` such as:

```
1> sp_companion "primary_server",configure
2> go

Step: Access verified from Server:'secondary_server' to Server:'primary_server'.
Step: Access verified from Server:'primary_server' to Server:'secondary_server'.
Msg 18836, Level 16, State 1:
Server 'secondary_server', Procedure 'sp_companion', Line 392: Configuration operation 'configure' can not proceed due to Quorum Advisory Check failure. Please run 'do_advisory' command to find the incompatible attribute and fix it.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Attrib Type</th>
<th>Local Value</th>
<th>Remote Value</th>
<th>Advisory</th>
</tr>
</thead>
<tbody>
<tr>
<td>allow password downg</td>
<td>allow password</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>
```

A value of 2 set in the advisory column of the output indicates that the user cannot proceed with the cluster operation unless the values on both the companions match.

`sp_companion do_advisory` also lists the difference in the value of allow password downgrade on both servers.

The administrator must execute the `sp_passwordpolicy 'allow password downgrade'` option independently on both the primary and secondary servers to get the value in sync and then configure using `sp_companion`, in order to ensure that the both the servers are in same state. At any given point of time the two servers should have same value set for allow password downgrade option in order to ensure proper behavior.
Passwords updated after upgrade

After upgrading and successfully configuring high availability, on the first connection to the primary server, the password of the user login is updated on both the primary and companion servers. This synchronizes the login password on primary and companion with the same on-disk encryption format. This is done to avoid password reset or locking when the allow password downgrade period ends as described in \texttt{sp\_passwordpolicy} and with password downgrade to earlier Adaptive Server 15.0 versions with \texttt{sp\_downgrade}. By synchronizing the password encryption format, the login passwords can continue to be used without being reset or locked by \texttt{sp\_passwordpolicy} or \texttt{sp\_downgrade}.

After successfully setting up the high availability environment, ending the allow password downgrade period must be done separately on the primary and companion servers. Similarly, if you need to downgrade to a previous version of Adaptive Server, the \texttt{sp\_downgrade} action must be executed separately on the primary and companion servers.

Monitoring license use

The License Use Monitor allows a system administrator to monitor the number of user licenses used in Adaptive Server, and to securely manage the license agreement data. That is, you can ensure that the number of licenses used on your Adaptive Server does not exceed the number specified in your license agreement.

The License Use Monitor tracks the number of licenses issued; it does not enforce the license agreement. If the License Use Monitor reports that you are using more user licenses than specified in your license agreement, see your Sybase sales representative.

You must have system administrator privileges to configure the License Use Monitor.

By default, the License Use Monitor is turned off when Adaptive Server is first installed or upgraded. The system administrator must configure the License Use Monitor to monitor license usage. See “Configuring the License Use Manager to monitor user licenses” on page 478 for configuration information.
Monitoring license use

How licenses are counted

A license is the combination of a host computer name and a user name. If a user logs in to Adaptive Server multiple times from the same host machine, it counts as one license. However, if the user logs in once from host A, and once from host B, it counts as two licenses. If multiple users log in to Adaptive Server from the same host, but with different user names, each distinct combination of user name and host name is counted.

Configuring the License Use Manager to monitor user licenses

Use `sp_configure` to specify the number of licenses in your license agreement, where `number` is the number of licenses:

```
sp_configure "license information", number
```

This example sets the maximum number of user licenses to 300, and reports an overuse for license number 301:

```
sp_configure "license information", 300
```

If you increase the number of user licenses, you must also change the `license information` configuration parameter.

The configuration parameter `housekeeper free write percent` must be set to 1 or more in for the License Use Manager to track license use.

Monitoring license use with the housekeeper task

After you configure the License Use Monitor, the housekeeper task determines how many user licenses are in use, based on the user ID and the host name of each user logged in to Adaptive Server. When the housekeeper task checks licenses, the License Use Monitor updates a variable that tracks the maximum number of user licenses in use:

- If the number of licenses in use is the same or has decreased since the previous housekeeper run, the License Use Monitor does nothing.
- If the number of licenses in use has increased since the previous housekeeper run, the License Use Monitor sets this number as the maximum number of licenses in use.
If the number of licenses in use is greater than the number allowed by the license agreement, the License Use Monitor issues message to the error log:

Exceeded license usage limit. Contact Sybase Sales for additional licenses.

The housekeeper chores task runs during Adaptive Server’s idle cycles. The housekeeper monitors the number of user licenses only if the license information configuration parameter is set to 1 or greater.

For more information about the housekeeper chores task, see Chapter 3, “Using Engines and CPUs,” in the Performance and Tuning Series: Basics.

Logging the number of user licenses

The syblicenseslog system table is created in the master database when you install or upgrade Adaptive Server. The License Use Monitor updates the columns in syblicenseslog at the end of each 24-hour period, as shown in Table 14-15.

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
</table>
| status | -1 – housekeeper cannot monitor licenses.  
0 – number of licenses not exceeded.  
1 – number of licensees exceeded. |
| logtime | Date and time the log information was inserted. |
| maxlicenses | Maximum number of licenses used during the previous 24 hours. |

The syblicenseslog looks similar to this:

<table>
<thead>
<tr>
<th>status</th>
<th>logdate</th>
<th>maxlicenses</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Jul 17 1998 11:43AM</td>
<td>123</td>
</tr>
<tr>
<td>0</td>
<td>Jul 18 1998 11:47AM</td>
<td>147</td>
</tr>
<tr>
<td>1</td>
<td>Jul 19 1998 11:51AM</td>
<td>154</td>
</tr>
<tr>
<td>0</td>
<td>Jul 20 1998 11:55AM</td>
<td>142</td>
</tr>
<tr>
<td>0</td>
<td>Jul 21 1998 11:58AM</td>
<td>138</td>
</tr>
<tr>
<td>0</td>
<td>Jul 21 1998 3:14PM</td>
<td>133</td>
</tr>
</tbody>
</table>

In this example, the number of user licenses used exceeded the limit on July 19, 1998.
Getting information about usage: chargeback accounting

If Adaptive Server is shut down, License Manager updates `syblicenseslog` with the current maximum number of licenses used. Adaptive Server starts a new 24-hour monitoring period when it is restarted.

The second row for July 21, 1998 was caused by a shutdown and restart of the server.

Getting information about usage: chargeback accounting

When a user logs in to Adaptive Server, the server begins accumulating CPU and I/O usage for that user. Adaptive Server can report total usage for an individual or for all users. Information for each user is kept in the `syslogins` system table in the `master` database.

Reporting current usage statistics

The system administrator can use `sp_reportstats` or `sp_clearstats` to get or clear current total usage data for individuals or for all users on Adaptive Server.

Displaying current accounting totals

`sp_reportstats` displays current accounting totals for Adaptive Server users. It reports total CPU and total I/O, as well as the percentage of those resources used. It does not record statistics for the “sa” login (processes with an `suid` of 1), checkpoint, network, and mirror handlers.

Initiating a new accounting interval

Adaptive Server accumulates CPU and I/O statistics until you clear the totals from `syslogins` by running `sp_clearstats`. `sp_clearstats` initiates a new accounting interval for Adaptive Server users and executes `sp_reportstats` to print out statistics for the previous period.

Choose the length of your accounting interval by deciding how you want to use the statistics at your site. For example, to do monthly cross-department charging for the percentage of Adaptive Server CPU and I/O usage, the system administrator would run `sp_clearstats` once a month.
For detailed information about these stored procedures, see the Reference Manual.

Specifying the interval for adding accounting statistics

A system administrator can use configuration parameters to decide how often accounting statistics are added to syslogins.

To specify how many machine clock ticks accumulate before accounting statistics are added to syslogins, use the cpu accounting flush interval configuration parameter. The default value is 200. For example:

\[
\text{sp\_configure "cpu accounting flush interval", 600}
\]

To find out how many microseconds a tick is on your system, run the following query in Adaptive Server:

\[
\text{select @@timeticks}
\]

To specify how many read or write I/Os accumulate before the information is added (flushed) to syslogins, use the i/o accounting flush interval configuration parameter. The default value is 1000. For example:

\[
\text{sp\_configure "i/o accounting flush interval", 2000}
\]

I/O and CPU statistics are flushed when a user accumulates more I/O or CPU usage than the specified value. The information is also flushed when the user exits an Adaptive Server session.

The minimum value allowed for either configuration parameter is 1. The maximum value allowed is 2,147,483,647.
Getting information about usage: chargeback accounting
This chapter discusses the steps the system administrator and system security officer of each Adaptive Server must execute to enable remote procedure calls (RPCs).

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>483</td>
</tr>
<tr>
<td>Managing remote servers</td>
<td>484</td>
</tr>
<tr>
<td>Adding remote logins</td>
<td>490</td>
</tr>
<tr>
<td>Password checking for remote users</td>
<td>494</td>
</tr>
<tr>
<td>Getting information about remote logins</td>
<td>495</td>
</tr>
<tr>
<td>Configuration parameters for remote logins</td>
<td>495</td>
</tr>
</tbody>
</table>

Overview

Users on a local Adaptive Server can execute stored procedures on a remote Adaptive Server. Executing an RPC sends the results of the remote process to the calling process—usually displayed on the user’s screen.

To enable RPCs, the system administrator and system security officer of each Adaptive Server 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:
Managing remote servers

- 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 `sp_addlogin`, 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.)

Figure 15-1 shows how to set up servers for remote access.

**Figure 15-1: Setting up servers to allow remote procedure calls**

The user “joe” on ROSE needs to access stored procedures on ZINNIA

```
sp_addserver ROSE, local
sp_addserver ZINNIA

interfaces files must have an entry for ZINNIA
```

```
sp_addserver ROSE
sp_addlogin joe
sp_addremotelogin ROSE, joe

sp_adduser joe (in the appropriate database)
grant execute on procedure_name to joe
```

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

Managing remote servers

Table 15-1 lists the tasks related to managing remote servers and the system procedures you use to perform the tasks.
### Adding a remote server

A system security officer uses `sp_addserver` to add entries to the `sysservers` table. On the server originating the call, you must add one entry for the local server, and one for each remote server that your server will call.

When you create entries for a remote server, you can either:

- Refer to them by the name listed in the `interfaces` file, or
- Provide a local name for the remote server. For example, if the name in the `interfaces` file is “MAIN_PRODUCTION,” you may want to call it simply “main.”

The syntax is:

```
sp_addserver lname [{, local | null}]
[, pname]
```

where:

- `lname` – provides the local “call name” for the remote server. If this name is not the same as the remote server’s name in the `interfaces` file, you must provide that name as the third parameter, `pname`.

The remote server must be listed in the `interfaces` file on the local machine. If it is not listed, copy the `interfaces` file entry from the remote server and append it to your existing `interfaces` file. Keep the same port numbers.
Managing remote servers

• local – identifies the server being added as a local server. The local value is used only after start-up, or after a restart, to identify the local server name so that it can appear in messages printed out by Adaptive Server. null specifies that this server is a remote server.

Note For users to be able to run RPCs successfully from the local server, the local server must be added with the local option and restarted. The restarting is required to set the global variable @@servername.

• pname – is the remote server listed in the interfaces file for the server named lname. This optional argument permits you to establish local aliases for any other Adaptive Server, Open Server™, or Backup Server that you may need to communicate with. If you do not specify pname, it defaults to lname.

Examples of adding remote servers

This example creates an entry for the local server named DOCS:

sp_addserver DOCS, local

This example creates an entry for a remote server named GATEWAY:

sp_addserver GATEWAY

To run a remote procedure such as sp_who on the GATEWAY server, execute either:

GATEWAY.sybsystemprocs.dbo.sp_who

or:

GATEWAY...sp_who

This example gives a remote server called MAIN_PRODUCTION the local alias “main:”

sp_addserver main, null, MAIN_PRODUCTION

The user can then enter:

main...sp_who

Managing remote server names

The master.dbo.sysservers table has two name columns:
- **srvname** is the unique server name that users must supply when executing remote procedure calls.
- **srvnetname** is the server’s network name, which must match the name in the *interfaces* file.

To add or drop servers from your network, you can use `sp_addserver` to update the server’s network name in `srvnetname`.

For example, to remove the server MAIN from the network, and move your remote applications to TEMP, you can use the following statement to change the network name, while keeping the local alias:

```
sp_addserver MAIN, null, TEMP
```

`sp_addserver` displays a message telling you that it is changing the network name of an existing server entry.

### Setting server connection options

`sp_serveroption` sets the server options `timeouts`, `net password encryption`, `rpc security model A`, and `rpc security model B`, which affect connections with remote servers. Additionally, if you have set the remote procedure security model to `rpc security model B`, you can use `sp_serveroption` to set these additional options: security mechanism, mutual authentication, use message confidentiality, and use message integrity.

The options you specify for `sp_serveroption` do not affect the communication between Adaptive Server and Backup Server.

The following sections describe `timeouts`, `net password encryption`, `rpc security model A`, and `rpc security model B`. For information about the additional options you can specify when `rpc security model B` is on, see “Establishing security for remote procedures” on page 516.

### Using the `timeouts` option

A system administrator can use the `timeouts` option to disable and enable the normal timeout code used by the local server.

By default, `timeouts` is set to `true`, and the site handler process that manages remote logins times out if there has been no remote user activity for one minute. By setting `timeouts` to `false` on both of the servers involved in remote procedure calls, the automatic timeout is disabled. This example changes `timeouts` to `false`:
Managing remote servers

```
sp_serveroption GATEWAY, "timeouts", false
```

After you set `timeouts` to `false` on both servers, when a user executes an RPC in either direction, the site handler on each machine runs until one of the servers is shut down. When the server is brought up again, the option remains `false`, and the site handler is reestablished the next time a user executes an RPC. If users execute RPCs frequently, it is probably efficient in terms of system resources to set this option to `false`, since there is some system overhead involved in setting up the physical connection.

Using the `net password encryption` option

A system security officer can use `net password encryption` to specify whether connections with a remote server are to be initiated with a client-side password encryption handshake or with the usual unencrypted password handshake sequence. The default is `false`.

If `net password encryption` is set to `true`:

1. The initial login packet is sent without passwords.
2. The client indicates to the remote server that encryption is desired.
3. The remote server sends back an encryption key, which the client uses to encrypt its plain text passwords.
4. The client then encrypts its own passwords, and the remote server uses the key to authenticate them when they arrive.

This example sets `net password encryption` to `true`:

```
sp_serveroption GATEWAY, "net password encryption", true
```

This option does not affect Adaptive Server’s interaction with Backup Server.

Using the `rpc security model` options

The `rpc security model A` and `rpc security model B` options determine what kind of security is available for RPCs. If you use model A, which is the default, Adaptive Server does not support security services such as message confidentiality via encryption between the two servers.
For security model B, the local Adaptive Server gets a credential from the security mechanism and uses the credential to establish a secure physical connection with the remote Adaptive Server. With this model, you can choose one or more of these security services: mutual authentication, message confidentiality via encryption, and message integrity.

To set security model A for the server GATEWAY, execute:

```
sp_serveroption GATEWAY, "rpc security model A", true
```

For information about how to set up servers for security model B, see “Establishing security for remote procedures” on page 516.

### Getting information about servers

`sp_helpserver` reports on servers. Without an argument, it provides information about all the servers listed in `sysservers`. When you include a server name, it provides information about that server only. The syntax is:

```
sp_helpserver [server]
```

`sp_helpserver` checks for both `srvname` and `srvnetname` in the `master..sysremotelogs` table.

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

### Dropping remote servers

A system security officer can use `sp_dropserver` to drop servers from `sysservers`. The syntax is:

```
sp_dropserver server[, droplogins]
```

where:

- `server` – is the name of the server you want to drop.
- `droplogins` – allows you to drop a remote server and all of that server’s remote login information in one step. If you do not use `droplogins`, you cannot drop a server that has remote logins associated with it.

The following statement drops the GATEWAY server and all of the remote logins associated with it:

```
sp_dropserver GATEWAY, droplogins
```
Adding remote logins

You do not have to use droplogins to drop the local server; that entry does not have remote login information associated with it.

Adding remote logins

The system security officer and system administrator of any Adaptive Server share control over which remote users can access the server, and what identity the 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.

Mapping users’ server IDs

Logins from a remote server can be mapped to a local server in three ways:

- A particular remote login can be mapped 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 can be mapped 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 can 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.

Use sp_dropremotelogin to remove the old mapping.

Use sp_addremotelogin to add remote logins. The syntax is:

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

If the local names are not listed in master.syslogins, add them as Adaptive Server logins with sp_addlogin before adding the remote logins.

Only a system administrator can execute sp_addremotelogin. For more information, see the Reference Manual.
CHAPTER 15    Managing Remote Servers

Mapping remote logins to particular local names

The following example maps the login named “pogo” from a remote system to
the local login name “bob”. The user logs in to the remote system as “pogo”. When that user
executes remote procedure calls from GATEWAY, the local system maps the remote login
name to “bob”.

```
sp_addlogin bob
sp_addremotelogin GATEWAY, bob, pogo
```

Mapping all remote logins to one local name

The following 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, as described in the previous section. For example, if you
mapped “pogo” to “bob”, and then the rest of the logins to “albert”, “pogo” still
maps to “bob”.

```
sp_addlogin albert
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 the logins as trusted, the logins are not allowed to execute
RPCs on the local server unless they specify passwords for the local server
when they log in to the remote server. Users, when they use Open Client
Client-Library, can use the routine `ct_remote_pwd` to specify a password for
server-to-server connections. `isql` and `bcp` do not permit users to specify a
password for RPC connections. See “Password checking for remote users” on
page 494 for more information about `sp_remoteoption`.

Warning! 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.
Adding remote logins

If you are using network based security

If users are logged in to the remote server using “unified login,” the logins must also be trusted on the local server, or they must specify passwords for the server when they log into the remote server. For more information, see “Using unified login” on page 511.

Warning! Using the trusted mode of sp_remoteoption reduces the security of your server, as passwords from such “trusted” users are not verified.

Keeping remote login names for local servers

To enable remote users to keep their remote login names while using a local server:

1 Use sp_addlogin to create a login for each login from the remote server.

2 Use sp_addremotelogin for the server as a whole 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

Example of remote user login mapping

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
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`:

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

<table>
<thead>
<tr>
<th>suid</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>sa</td>
</tr>
<tr>
<td>2</td>
<td>vp</td>
</tr>
<tr>
<td>3</td>
<td>admin</td>
</tr>
<tr>
<td>4</td>
<td>writer</td>
</tr>
</tbody>
</table>

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.
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.

Note: In many cases, the passwords for users on the remote server must match passwords on the local server.

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 (“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. The syntax is:

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

The following example sets trusted mode for the user “bob”:

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

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. Open Client applications can be written 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 changing it on your local server. This is because of the password checking. If you change your password on the local server first, when you issue the remote procedure call to execute sp_password on the remote server, your passwords no longer match.

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

```
remote_server...sp_password caller_passwd, new_passwd
```

On the local server, the syntax is:

```
sp_password caller_passwd, new_passwd
```

See “Changing passwords” on page 421 for more information about changing your password.

### Getting information about remote logins

sp_helpremotelogin prints information about the remote logins on a server. The following example shows the remote login “pogo” mapped locally to login name “bob”, with all other remote logins keeping their remote names:

```
sp_helpremotelogin

<table>
<thead>
<tr>
<th>server</th>
<th>remote_user_name</th>
<th>local_user_name</th>
<th>options</th>
</tr>
</thead>
<tbody>
<tr>
<td>GATEWAY</td>
<td><strong>mapped locally</strong></td>
<td><strong>use local name</strong></td>
<td>untrusted</td>
</tr>
<tr>
<td>GATEWAY</td>
<td>pogo</td>
<td>bob</td>
<td>untrusted</td>
</tr>
</tbody>
</table>
```

### Configuration parameters for remote logins

Table 15-2 shows the configuration parameters that affect RPCs. All these configuration parameters are set using sp_configure and do not take effect until Adaptive Server is restarted.

---

CHAPTER 15  Managing Remote Servers
Configuration parameters for remote logins

Table 15-2: Configuration parameters that affect RPCs

<table>
<thead>
<tr>
<th>Configuration parameter</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>allow remote access</td>
<td>1</td>
</tr>
<tr>
<td>number of remote logins</td>
<td>20</td>
</tr>
<tr>
<td>number of remote sites</td>
<td>10</td>
</tr>
<tr>
<td>number of remote connections</td>
<td>20</td>
</tr>
<tr>
<td>remote server pre-read packets</td>
<td>3</td>
</tr>
</tbody>
</table>

Allowing remote access

To allow remote access to or from a server, including Backup Server, set allow remote access to 1:

    sp_configure "allow remote access", 1

To disallow remote access at any time, set allow remote access to 0:

    sp_configure "allow remote access", 0

Only a system security officer can set the allow remote access parameter.

Note You cannot perform database or transaction log dumps while the allow remote access parameter is set to 0.

Controlling the number of active user connections

To set the number of active user connections from this site to remote servers, use number of remote logins. This command sets number of remote logins to 50:

    sp_configure "number of remote logins", 50

Only a system administrator can set the number of remote logins parameter.
Controlling the number of remote sites

To control the number of remote sites that can access a server simultaneously, use number of remote sites. All accesses from an individual site are managed by one site handler. This parameter controls the number of site handlers, not the number of individual, simultaneous procedure calls. For example, if you set number of remote sites to 5, and each site initiates three remote procedure calls, sp_who shows 5 site handler processes for the 15 processes. Only a system administrator can set the number of remote sites.

Controlling the number of active remote connections

To control the limit on active remote connections that are initiated to and from a server, use the number of remote connections parameter. This parameter controls connections initiated from the server and connections initiated from remote sites to the server. Only a system administrator can set number of remote connections.

Controlling number of preread packets

To reduce the needed number of connections, all communication between two servers is handled through one site handler. This site handler can preread and keep track of data packets for each user before the user process that needs them is ready.

To control how many packets a site handler prereads, use remote server pre-read packets. The default value, 3, is adequate in all cases; higher values can use too much memory. Only a system administrator can set remote server pre-read packets. For more information, see "remote server pre-read packets" on page 215.
Configuration parameters for remote logins
CHAPTER 16

External Authentication

This chapter describes the Adaptive Server features that enable you to authenticate users with authentication data stored in repositories external to Adaptive Server.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>499</td>
</tr>
<tr>
<td>Configuring Adaptive Server for Network-Based Security</td>
<td>500</td>
</tr>
<tr>
<td>Configuring Adaptive Server for LDAP User Authentication</td>
<td>539</td>
</tr>
<tr>
<td>Configuring Adaptive Server for authentication using PAM</td>
<td>556</td>
</tr>
<tr>
<td>Enhanced login controls</td>
<td>559</td>
</tr>
</tbody>
</table>

Overview

You can enhance the security for large, heterogeneous applications by authenticating logins with a central repository. Adaptive Server supports a variety of external authentication methods:

- Kerberos – A security mechanism available with Network-Based Security. Kerberos provides a centralized and secure authentication mechanism in enterprise environments that employ the Kerberos infrastructure. Authentication occurs with a trusted, third-party server called a Key Distribution Center (KDC) that verifies both the client and the server.

- LDAP User Authentication – Lightweight Directory Access Protocol (LDAP) provides a centralized authentication mechanism based on a user’s login name and password.

- PAM User Authentication – Pluggable Authentication Module (PAM) provides a centralized authentication mechanism that uses interfaces provided by the operating system for administration and runtime application interfaces.
Configuring Adaptive Server for Network-Based Security

In a distributed client/server computing environment, intruders can view or tamper with confidential data. Adaptive Server works with third-party providers to provide security services that:

- Authenticate users, clients, and servers – make sure they are who they say they are.
- Provide data confidentiality with encryption – ensure that data cannot be read by an intruder.
- Provide data integrity – prevent data tampering, and detect when it has occurred.

How applications use security services

The following illustration shows a client application using a security mechanism to ensure a secure connection with Adaptive Server.

*Figure 16-1: Establishing secure connections between a client and Adaptive Server*

The secure connection between a client and a server can be used for:

- Login authentication
- Message protection
Login authentication

If a client requests authentication services:

1. The client validates the login with the security mechanism. The security mechanism returns a credential, which contains security-relevant information.
2. The client sends the credential to Adaptive Server.
3. Adaptive Server authenticates the client’s credential with the security mechanism. If the credential is valid, a secure connection is established between the client and Adaptive Server.

Message protection

If the client requests message protection services:

1. The client uses the security mechanism to prepare the data packet it sends to Adaptive Server. Depending upon which security services are requested, the security mechanism might encrypt the data or create a cryptographic signature associated with the data.
2. The client sends the data packet to Adaptive Server.
3. When Adaptive Server receives the data packet, it uses the security mechanism to perform any required decryption and validation.
4. Adaptive Server returns results to the client, using the security mechanism to perform the security functions that were requested; for example, Adaptive Server may return the results in encrypted form.

Security services and Adaptive Server

Depending upon the security mechanism you choose, Adaptive Server allows you to use one or more of these security services:

- Unified login – authenticates users once without requiring them to supply a name and password every time they log in to an Adaptive Server.
- Message confidentiality – encrypts data over the network.
- Mutual authentication – verifies the identity of the client and the server. This must be requested by the client and cannot be required by Adaptive Server.
Configuring Adaptive Server for Network-Based Security

- Message integrity – verifies that data communications have not been modified.
- Replay detection – verifies that data has not been intercepted by an intruder.
- Out-of-sequence check – verifies the order of data communications.
- Message origin checks – verifies the origin of the message.
- Remote procedure security – establishes mutual authentication, message confidentiality, and message integrity for remote procedure communications.

Note The security mechanism you are using may not employ all of these services. For information about the services available to you, see “Getting information about available security services” on page 526.

Administering network-based security

Table 16-1 provides an overall process for using the network-based security functions provided by Adaptive Server. You must install Adaptive Server before you can complete the steps in Table 16-1.

**Table 16-1: Process for administering network-based security**

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Set up the configuration files:</td>
<td>• “Setting up configuration files for security” on page 503</td>
</tr>
<tr>
<td></td>
<td>* libtcl.cfg</td>
<td>• The Open Client/Server Configuration Guide for your platform</td>
</tr>
<tr>
<td></td>
<td>* objectid.dat</td>
<td></td>
</tr>
<tr>
<td></td>
<td>* interfaces (or Directory Service)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Edit the libtcl.cfg file.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Edit the objectid.dat file.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Edit the interfaces file or Directory Service.</td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>Make sure the security administrator for the security mechanism has created logins for each user and for the Adaptive Server and Backup Server.</td>
<td>• The documentation supplied with your security mechanism</td>
</tr>
<tr>
<td></td>
<td>The security administrator must add names and passwords for users and servers in the security mechanism.</td>
<td>• “Identifying users and servers to the security mechanism” on page 509</td>
</tr>
<tr>
<td></td>
<td>For DCE, the security administrator must create a keytab file for server entries.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Use sp_configure.</td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>Restart Adaptive Server.</td>
<td>“Restarting the server to activate security services” on page 514</td>
</tr>
<tr>
<td></td>
<td>Activates the use security services parameter.</td>
<td></td>
</tr>
</tbody>
</table>
Setting up configuration files for security

Configuration files are created during installation at a default location in the Sybase directory structure. Table 16-2 provides an overview of the configuration files required for using network-based security.

Table 16-2: Names and locations for configuration files

<table>
<thead>
<tr>
<th>File name</th>
<th>Description</th>
<th>Location</th>
</tr>
</thead>
</table>
| libtcl.cfg  | The driver configuration file contains information regarding directory, security, and network drivers and any required initialization information. | UNIX platforms: $SYBASE/$SYBASE_OCS/config  
|             |                                                                           | Windows platforms: %SYBASE%\%SYBASE_OCS%\ini |
| objectid.dat | The object identifiers file maps global object identifiers to local names for character set, collating sequence, and security mechanisms. | UNIX platforms: $SYBASE/config  
|             |                                                                           | Windows platforms: %SYBASE%\ini |
Configuring Adaptive Server for Network-Based Security

### Entries for network drivers

The syntax for a network driver entry is:

```
driver=protocol description
```

where:

- `driver` – is the name of the network driver.
- `protocol` – is the name of the network protocol.

---

For a detailed description of the configuration files, see the *Open Client/Server Configuration Guide* for your platform.

---

### Preparing `libtcl.cfg` to use network-based security

`libtcl.cfg` and `libtcl64.cfg` (for 64bit applications) contains information about three types of drivers:

- Network (Net-Library)
- Directory Services
- Security

A **driver** is a Sybase library that provides an interface to an external service provider. Drivers are dynamically loaded so that you can change the driver used by an application without relinking the application.

---

<table>
<thead>
<tr>
<th>File name</th>
<th>Description</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNIX: interfaces</td>
<td>The <code>interfaces</code> file contains connection and security information for each server listed in the file.</td>
<td><em>UNIX platforms: $SYBASE</em></td>
</tr>
<tr>
<td>Desktop platforms:</td>
<td></td>
<td><em>Desktop platforms: SYBASE_home\ini</em></td>
</tr>
<tr>
<td>sql.ini</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Note* As of Adaptive Server version 12.5.1, you can use a Directory Service instead of the `interfaces` file.
• *description* – is a description of the entry. This element is optional.

**Note** If you do not specify a network driver, an appropriate driver for your application and platform is automatically used. For example, for UNIX platforms, a driver that can handle threads is automatically chosen when security services are being used.

**Entries for Directory Services**

Entries for Directory Services apply if you want to use a Directory Service instead of the *interfaces* file. For information about directory entries, see the configuration documentation for your platform, and the *Open Client/Server Configuration Guide* for your platform.

**Entries for security drivers**

The syntax for a security driver entry is:

```
provider=driver init-string
```

where:

• *provider* – is the local name for the security mechanism. The mapping of the local name to a global object identifier is defined in *objectid.dat*.

The default local names are:

• “dce” – for the DCE security mechanism.
• “csfkrb5” – for the CyberSAFE or MIT Kerberos security mechanism.
• “LIBSMSSP” – for Windows LAN Manager on Windows NT or Windows 95 (clients only).

If you use a local mechanism name other than the default, you must change the local name in the *objectid.dat* file (see “The objectid.dat file” on page 507 for an example).

• *driver* – is the name of the security driver. The default location of all drivers for Unix platforms is `SYBASE/SYBASE_OCS/lib`. The default location for Windows platform is `%SYBASE%\%SYBASE_OCS%\dll`.

• *init-string* – is an initialization string for the driver. This element is optional. The value for *init-string* varies by driver:

  • DCE driver – the following is the syntax for *init-string*, where *cell_name* is the name of your DCE cell:
secbase=/.cell_name

- Kerberos driver – the following is the syntax for *init-string*, where *realm* is the default Kerberos realm name:
  ```
  secbase=@realm
  ```

- Windows NT LAN Manager – *init-string* is not applicable.

**UNIX platform information**

This section contains information specific to UNIX platforms. For more information, see the *Open Client/Server Configuration Guide for UNIX*.

No special tools for editing the *libtcl.cfg* file are available. Use your favorite editor to comment and uncomment the entries that are already in place after you install Adaptive Server.

The *libtcl.cfg* file, after installation of Adaptive Server on a UNIX platform, already contains entries for the three sections of the file:

- [DRIVERS]
- [DIRECTORY]
- [SECURITY]

The sections do not have to be in a specific order.

Make sure that the entries you do not want to use are commented (begin with “;”) and the entries you want are uncommented (do not begin with “;”).

**Sample *libtcl.cfg* for Sun Solaris**

[DRIVERS]
```
;libtli.so=tcp unused ; This is the non-threaded tli driver.
;libtli_r.so=tcp unused ; This is the threaded tli driver.
```

[DIRECTORY]
```
;dce=libsybddce.so ditbase=./subsys/sybase/dataservers
;dce=libsybddce.so ditbase=./users/cfrank
```

[SECURITY]
```
dce=libsybsdce.so secbase=./svrsole4_cell
```

This *libtcl.cfg* file is set up to use the DCE security service. This file does not use Directory Services because all [DIRECTORY] section entries are commented.
CHAPTER 16  External Authentication

Because all entries in the [DRIVERS] section for network drivers are also commented, appropriate drivers are chosen automatically by the system. A threaded driver is chosen automatically when security services are being used, and a nonthreaded driver is chosen automatically for applications that cannot work with threaded drivers. For example, Backup Server does not support security services and does not work with a threaded driver.

Desktop platform information

This section contains information specific to desktop platforms. For more information, see the Open Client/Server Configuration Guide for Desktop Platforms.

Use the ocscfg utility to edit the libtcl.cfg file. See the Open Client/Server Configuration Guide for Desktop Platforms for instructions for using ocscfg.

The ocscfg utility creates section headings automatically for the libtcl.cfg file.

Sample libtcl.cfg file for desktop platforms

```
[NT_DIRECTORY]
nreg_dsa=LIBDREG  ditbase=software\sybase\serverdsa

[DRIVERS]
NLWNSCK=TCP  Winsock TCP/IP Net-Lib driver
NLMSNMP=NAMEPIPE  Named Pipe Net-Lib driver
NLNWLINK=SPX  NT NWLINK SPX/IPX Net-Lib driver
NLDECNET=DECNET  DecNET Net-Lib driver

[SECURITY]
NTLM=LIBSMSSP
```

The objectid.dat file

The objectid.dat file maps global object identifiers, such as the one for the DCE service (“1.3.6.1.4.1.897.4.6.1”) to local names, such as “dce”. The file contains sections such as [CHARSET] for character sets and [SECURITY] for security services. Of interest here is the security section. Following is a sample objectid.dat file:

```
[secmech]
  1.3.6.1.4.1.897.4.6.1  = dce
  1.3.6.1.4.1.897.4.6.3  = NTLM
  1.3.6.1.4.1.897.4.6.6  = csfkrb5
```

To change this file only if you have changed the local name of a security service in the libtcl.cfg file. Use a text editor to edit the file.
For example, if you changed:

    [SECURITY]
    dce=libsysbsdce.so secbase=/.../svrsole4_cell

to:

    [SECURITY]
    dce_group=libsysbsdce.so secbase=/.../svrsole4_cell

in libtcl.cfg, change the objectid.dat file to reflect the change. Simply change the local name in the line for DCE in objectid.dat:

    1.3.6.1.4.1.897.4.6.1 = dce_group

Note  You can specify only one local name per security mechanism.

Specifying security information for the server

You can choose to use an interfaces file or a Directory Service to provide information about the servers in your installation.

The interfaces file contains network and security information for servers. To use security services, the interfaces file must include a “secmech” line, which gives the global identifier or identifiers of the security services you plan to use.

Instead of using the interfaces file, Adaptive Server supports Directory Services to keep track of information about servers. A Directory Service manages the creation, modification, and retrieval of information about network servers. The advantage of using a Directory Service is that you do not need to update multiple interfaces files when a new server is added to your network or when a server moves to a new address. To use security services with a Directory Service, the secmech security attribute must be defined. It must point to one or more global identifiers of the security services you plan to use.

UNIX tools for specifying the security mechanism

To specify the security mechanism or mechanisms:

- If you are using the interfaces file, use the dscp utility.
CHAPTER 16  External Authentication

- If you are using a Directory Service, use the `dscp_r` utility.

  **Note** The `dsedit` tool, which helps you create entries for either the `interfaces` file or a Directory Service, is available on UNIX platforms. However, it does not support the creation of `secmech` entries for security mechanisms.

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

**Desktop tools for specifying server attributes**

To provide information about the servers for your installation in the `sql.ini` file or a Directory Service, use the `dsedit` utility. This utility provides a graphical user interface for specifying server attributes such as the server version, name, and security mechanism. For the security mechanism attribute, you can specify one or more object identifiers for the security mechanisms you plan to use. For information about using `dsedit`, see the *Open Client/Server Configuration Guide for Desktop Platforms*.

**Identifying users and servers to the security mechanism**

The security administrator for the security mechanism must define principals, which include both users and servers, to the security mechanism. Table 16-3 lists tools you can use to add users and servers.

<table>
<thead>
<tr>
<th>Security mechanism</th>
<th>Command or tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCE</td>
<td>Use the DCE <code>dcecp</code> tool’s <code>user create</code> command to create a new principal (user or server). In addition, use the <code>keytab create</code> command to create a DCE keytab file, which contains a principal’s password in encrypted form. When you are defining a server to DCE, use command options that specify that the new principal can act as a server.</td>
</tr>
<tr>
<td>Kerberos</td>
<td>See your Kerberos vendor-specific tools for information about defining users and servers. See “Using Kerberos” on page 527 for more information about Kerberos and Adaptive Server.</td>
</tr>
<tr>
<td>Windows NT LAN Manager</td>
<td>Run the User Manager tool to define users to the Windows NT LAN Manager. Be sure to define the Adaptive Server name as a user to Windows NT LAN Manager and bring up Adaptive Server as that user name.</td>
</tr>
</tbody>
</table>

Table 16-3: Defining users and servers to the security mechanism
In a production environment, you must control the access to files that contain the keys of the servers and users. If users can access the keys, they can create a server that impersonates your server.

Refer to the documentation available from the third-party provider of the security mechanism for detailed information about how to perform required administrative tasks.

Configuring Adaptive Server for security

Adaptive Server includes several configuration parameters for administering network-based security. To set these parameters, you must be a system security officer. All parameters for network-based security are part of the “Security-Related” configuration parameter group.

Configuration parameters are used to:

- Enable network-based security
- Require unified login
- Require message confidentiality with data encryption
- Require one or more message integrity security services

Enabling network-based security

To enable or disable network-based security, use `sp_configure` to set the `use security services` configuration parameter. Set this parameter to 1 to enable network-based security. If this parameter is 0 (the default), network-based security services are not available. The syntax is:

```
sp_configure "use security services", [0|1]
```

For example, to enable security services, execute:

```
sp_configure "use security services", 1
```

Note This configuration parameter is static; you must restart Adaptive Server for it to take effect. See “Restarting the server to activate security services” on page 514.
CHAPTER 16   External Authentication

Using unified login

Configuration parameters are available to:

- Require unified login
- Establish a default secure login

All the parameters for unified login take effect immediately. You must be a
system security officer to set the parameters.

Requiring unified login

To require all users, other than the user with system security officer (sso) role,
to be authenticated by a security mechanism, set the unified login required
configuration parameter to 1. Only the user with the sso_role can log in to the
server with a user name and password when this configuration parameter is set:

```
sp_configure "unified login required", [0|1]
```

For example, to require all logins to be authenticated by a security mechanism,
execute:

```
sp_configure "unified login required", 1
```

Establishing a secure default login

When a user with a valid credential from a security mechanism logs in to
Adaptive Server, the server checks whether the user name exists in
master..syslogins. If it does, that user name is used by Adaptive Server. For
example, if a user logs in to the DCE security mechanism as “ralph,” and
“ralph” is a name in master..syslogins, Adaptive Server uses all roles and
authorizations defined for “ralph” in the server.

However, if a user with a valid credential logs in to Adaptive Server, but is
unknown to the server, the login is accepted only if a secure default login is
defined with sp_configure. Adaptive Server uses the default login for any user
who is not defined in master..syslogins, but who is preauthenticated by a
security mechanism. The syntax is:

```
sp_configure "secure default login", 0, login_name
```

The default value for secure default login is “guest.”

This login must be a valid login in master..syslogins. For example, to set the
login “gen_auth” to be the default login:

1  Use sp_addlogin to add the login as a valid user in Adaptive Server:

```
sp_addlogin gen_auth, pwgenau
```
This procedure sets the initial password to “pwgenau”.

2 Use sp_configure to designate the login as the security default.

```
sp_configure "secure default login", 0, gen_auth
```

Adaptive Server uses this login for a user who is preauthenticated by a security mechanism but is unknown to Adaptive Server.

---

**Note** More than one user can assume the *suid* associated with the secure default login. Therefore, you might want to activate auditing for all activities of the default login. You may also want to consider using `sp_addlogin` to add all users to the server.

---

For more information about adding logins, see “Adding logins to support unified login” on page 515 and “Adding logins to Adaptive Server” on page 394.

---

**Mapping security mechanism login names to server names**

Some security mechanisms may allow login names that are not valid in Adaptive Server. For example, login names that are longer than 30 characters, or login names containing special characters such as !, %, *, and & are invalid names in Adaptive Server. All login names in Adaptive Server must be valid identifiers. For information about what identifiers are valid, see Chapter 3, “Expressions, Identifiers, and Wildcard Characters,” in the Reference Manual.

Table 16-4 shows how Adaptive Server converts invalid characters in login names:

<table>
<thead>
<tr>
<th>Invalid characters</th>
<th>Converts to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ampersand &amp;</td>
<td>Underscore _</td>
</tr>
<tr>
<td>Apostrophe '</td>
<td></td>
</tr>
<tr>
<td>Backslash \</td>
<td></td>
</tr>
<tr>
<td>Colon :</td>
<td></td>
</tr>
<tr>
<td>Comma ,</td>
<td></td>
</tr>
<tr>
<td>Equals sign =</td>
<td></td>
</tr>
<tr>
<td>Left quote '</td>
<td></td>
</tr>
<tr>
<td>Percent %</td>
<td></td>
</tr>
<tr>
<td>Right angle bracket &gt;</td>
<td></td>
</tr>
<tr>
<td>Right quote '</td>
<td></td>
</tr>
<tr>
<td>Tilde ~</td>
<td></td>
</tr>
</tbody>
</table>
To require all messages into and out of Adaptive Server to be encrypted, set the `msg confidentiality reqd` configuration parameter to 1. If this parameter is 0 (the default), message confidentiality is not required but may be established by the client.

The syntax for setting this parameter is:

```
sp_configure "configuration parameter", [0 | 1]
```

For example, to require that all messages be encrypted, execute:

```
sp_configure "msg confidentiality reqd", 1
```

### Requiring message confidentiality with encryption

Adaptive Server allows you to use the `msg integrity reqd` configuration parameters to require that one or more types of data integrity be checked for all messages. `msg integrity reqd` sets this parameter to 1 to require that all messages be checked for general tampering. If this parameter is 0 (the default), message integrity is not required but may be established by the client if the security mechanism supports it.

<table>
<thead>
<tr>
<th>Invalid characters</th>
<th>Converts to</th>
</tr>
</thead>
<tbody>
<tr>
<td>caret ^</td>
<td>dollar sign $</td>
</tr>
<tr>
<td>curly braces { }</td>
<td></td>
</tr>
<tr>
<td>exclamation point !</td>
<td></td>
</tr>
<tr>
<td>left angle bracket &lt;</td>
<td></td>
</tr>
<tr>
<td>parenthesis ( )</td>
<td></td>
</tr>
<tr>
<td>period .</td>
<td></td>
</tr>
<tr>
<td>question mark ?</td>
<td></td>
</tr>
<tr>
<td>asterisk *</td>
<td>pound sign #</td>
</tr>
<tr>
<td>minus sign -</td>
<td></td>
</tr>
<tr>
<td>pipe</td>
<td></td>
</tr>
<tr>
<td>plus sign +</td>
<td></td>
</tr>
<tr>
<td>quotation marks &quot;</td>
<td></td>
</tr>
<tr>
<td>semicolon ;</td>
<td></td>
</tr>
<tr>
<td>slash /</td>
<td></td>
</tr>
<tr>
<td>square brackets [ ]</td>
<td></td>
</tr>
</tbody>
</table>

---

**Invalid characters Converts to**

- caret `^`
- dollar sign `$`
- curly braces `{ }`
- exclamation point `!`
- left angle bracket `<`
- parenthesis `(`
- period `.`
- question mark `?`
- asterisk `*`
- pound sign `#`
- minus sign `-`
- pipe `|`
- plus sign `+`
- quotation marks `"`
- semicolon `;`
- slash `/`
- square brackets `[ ]`

**Requiring message confidentiality with encryption**

To require all messages into and out of Adaptive Server to be encrypted, set the `msg confidentiality reqd` configuration parameter to 1. If this parameter is 0 (the default), message confidentiality is not required but may be established by the client.

The syntax for setting this parameter is:

```
sp_configure configuration parameter, [0 | 1]
```

For example, to require that all messages be encrypted, execute:

```
sp_configure "msg confidentiality reqd", 1
```

**Requiring data integrity**

Adaptive Server allows you to use the `msg integrity reqd` configuration parameters to require that one or more types of data integrity be checked for all messages. `msg integrity reqd` sets this parameter to 1 to require that all messages be checked for general tampering. If this parameter is 0 (the default), message integrity is not required but may be established by the client if the security mechanism supports it.
Memory requirements for network-based security

Allocate approximately 2K additional memory per secure connection. The value of the `max total_memory` configuration parameter specifies the amount of memory that Adaptive Server requires at start-up. For example, if your server uses 2K logical pages, and if you expect the maximum number of secure connections occurring at the same time to be 150, increase the `max total_memory` parameter by 150, which increases memory allocation by 150 2K blocks.

The syntax is:

```
sp_configure "max total_memory", value
```

For example, if Adaptive Server requires 75,000 2K blocks of memory, including the increased memory for network-based security, execute:

```
sp_configure "max total_memory", 75000
```

For information about estimating and specifying memory requirements, see the Chapter 3, “Configuring Memory,” in System Administration Guide: Volume 2.

Restarting the server to activate security services

Once you have configured security services, you must restart Adaptive Server.

Determining security mechanisms to support

If `use security services` is set to 0, Adaptive Server supports no security mechanisms.

If `use security services` is set to 1, Adaptive Server supports a security mechanism when both of the following circumstances are true:

- The security mechanism’s global identifier is listed in the `interfaces` file or Directory Service.
- The global identifier is mapped in `objectid.dat` to a local name that is listed in `libtcl.cfg`.

For information about how Adaptive Server determines which security mechanism to use for a particular client, see “Using security mechanisms for the client” on page 525.
Adding logins to support unified login

When users log in to Adaptive Server with a preauthenticated credential, Adaptive Server:

1. Checks whether the user is a valid user in `master..syslogins`. If the user is listed in `master..syslogins`, Adaptive Server accepts the login without requiring a password.

2. If the user name is not in `master..syslogins`, Adaptive Server checks whether a default secure login is defined. If the default login is defined, the user is logged in successfully as that login. If a default login is not defined, Adaptive Server rejects the login.

Therefore, consider whether you want to allow only those users who are defined as valid logins to use Adaptive Server, or whether you want users to be able to log in with the default login. You must add the default login in `master..syslogins` and use `sp_configure` to define the default. For details, see “Establishing a secure default login” on page 511.

General procedure for adding logins

Follow the general procedure described in Table 16-5 to add logins to the server and, optionally, to add users to one or more databases with appropriate roles and authorizations to one or more databases.

<table>
<thead>
<tr>
<th>Task</th>
<th>Required role</th>
<th>Command or procedure</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Add a login for the user.</td>
<td>System security officer</td>
<td><code>sp_addlogin</code></td>
<td>“Adding logins to Adaptive Server” on page 394</td>
</tr>
<tr>
<td>2. Add the user to one or more databases.</td>
<td>System administrator or Database owner</td>
<td><code>sp_adduser</code> – execute this procedure from within the database.</td>
<td>“Adding users to databases” on page 396</td>
</tr>
</tbody>
</table>
| 3. Add the user to a group in a database. | System administrator or Database owner | `sp_changegroup` – execute this procedure from within the database. | • “Changing a user’s group membership” on page 423  
• `sp_changegroup` in the Reference Manual |
| 4. Grant system roles to the user. | System administrator or system security officer | `grant role` | • “Creating and assigning roles to users” on page 403  
• `grant` in the Reference Manual |
Establishing security for remote procedures

Adaptive Server acts as the client when it connects to another server to execute a remote procedure call (RPC) as shown in Figure 16-2.

Figure 16-2: Adaptive Server acting as client to execute an RPC

One physical connection is established between the two servers. The servers use the physical connection to establish one or more logical connections—one logical connection for each RPC.

Adaptive Server 11.5 and later supports two security models for RPCs: security model A and security model B.

Security model A

For security model A, Adaptive Server does not support security services such as message confidentiality via encryption between the two servers. Security model A is the default.
Security model B

For security model B, the local Adaptive Server gets a credential from the security mechanism and uses the credential to establish a secure physical connection with the remote Adaptive Server. With this model, you can use one or more of these security services:

- Mutual authentication – the local server authenticates the remote server by retrieving the credential of the remote server and verifying it with the security mechanism. With this service, the credentials of both servers are authenticated and verified.
- Message confidentiality via encryption – messages are encrypted when sent to the remote server, and results from the remote server are encrypted.
- Message integrity – messages between the servers are checked for tampering.

Unified login and the remote procedure models

If the local server and remote server are set up to use security services, you can use unified login on both servers with either model, using one of these two methods:

- The system security officer defines a user as “trusted” with `sp_remoteoption` on the remote server. With this method, a security mechanism such as DCE authenticates the user and password. The user gains access to the local server via “unified login” and executes an RPC on the remote server. The user is trusted on the remote server and does not need to supply a password.
- A user specifies a password for the remote server when he or she connects to the local server. The facility to specify a remote server password is provided by the `ct_remote_pwd` routine available with Open Client Client-Library/C. For more information about this routine, see the *Open Client Client-Library/C Reference Manual*.

Establishing the security model for RPCs

To establish the security model for RPCs, use `sp_serveroption`. The syntax is:

```
sp_serveroption server, optname, [true | false]
```

To establish the security model, set `optname` to `rpc security model A` or `rpc security model B`. `server` names the remote server.

For example, to set security model B for remote server TEST3, execute:
Configuring Adaptive Server for Network-Based Security

The default model is “A,” that is, remote procedure calls are handled the same as in versions earlier than 11.5. No server options need to be set for model A.

Setting server options for RPC security model B

For RPC security model B, you can set options with `sp_serveroption`. The syntax is:

```
sp_serveroption server, optname, optvalue
```

where:

- `server` – is the name of the remote server.
- `optname` – is the name of the option. Values can be:
  - `security mechanism` – the name of the security mechanism to use when running an RPC on a remote server.
  - `mutual authentication` – set this option to 1 for the local Adaptive Server to authenticate and verify the remote server. If this parameter is 0 (the default), the remote server still verifies the local server when it sends an RPC, but the local server does not check the validity of the remote server.
  - `use message confidentiality` – set this option to 1 for all messages for the RPCs to be encrypted when they are sent to the remote server and received from the remote server. If this parameter is 0 (the default), data for the RPCs are not encrypted.
  - `use message integrity` – set this option to 1 to require that all RPC messages be checked for tampering. If this parameter is 0 (the default), RPC data will not be checked for tampering.

- `optvalue` – must be equal to “true” or “false” for all values of `optname`, except `security mechanism`. If the option you are setting is `security mechanism`, specify the name of the security mechanism. To find the list of security mechanisms, execute:

```
select * from syssecmechs
```

For information about the `syssecmechs` system table, see “Determining enabled security services” on page 527.

For example, to set up the local server to execute RPCs on a remote server, `TEST3`, which uses the “dce” security mechanism, and to use mutual authentication for all RPCs between the two servers, execute:

```
sp_serveroption test3, "rpc security model B", true
```
sp_serveroption TEST3, "security mechanism", dce
sp_serveroption TEST3, "mutual authentication", true

Rules for setting up security model B for RPCs

Follow these rules when setting up security model B for RPCs:

- Both servers must be using security model B.
- Both servers must be using the same security mechanism, and that security mechanism must support the security services set with sp_serveroption.
- The system security officer of the local server must specify any security services that are required by the remote server. For example, if the remote server requires that all messages use the message confidentiality security service, the system security officer must use sp_serveroption to activate use message confidentiality.
- Logins that are authenticated by a security mechanism and log in to Adaptive Server using “unified login” are not permitted to execute RPCs on the remote procedure unless the logins are specified as “trusted” on the remote server or the login specifies the password for the remote server. Users, when they use Open Client Client-Library can use the routine ct_remote_pwd to specify a password for server-to-server connections. A system administrator on Adaptive Server can use sp_remoteoption to specify that a user is trusted to use the remote server without specifying a password.

Preparing to use security model B for RPCs

Table 16-6 provides steps for using security model B to establish security for RPCs.

<table>
<thead>
<tr>
<th>Task, who performs it, and where</th>
<th>Command, system procedure, or tool</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>System administrator from the operating system: 1. Make sure the interfaces file or the Directory Service contains an entry for both servers and a secmech line listing the security mechanism.</td>
<td>UNIX: dscp Desktop: dsedit</td>
<td>“Specifying security information for the server” on page 508 dscp in the Open Client/Server Configuration Guide for UNIX dsedit in the Open Client/Server Configuration Guide for Desktop Platforms</td>
</tr>
</tbody>
</table>
### Configuring Adaptive Server for Network-Based Security

<table>
<thead>
<tr>
<th>Task, who performs it, and where</th>
<th>Command, system procedure, or tool</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>System security officer on remote server:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Add the local server to master..sys servers.</td>
<td><code>sp_addserver</code></td>
<td>“Adding a remote server” on page 485</td>
</tr>
<tr>
<td><strong>System security officer on remote server:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Add logins to master..syslogins.</td>
<td><code>sp_addlogin</code></td>
<td>“Adding logins to Adaptive Server” on page 394</td>
</tr>
<tr>
<td></td>
<td><code>sp_addlogin user1, &quot;pwuser1&quot;</code></td>
<td><code>sp_addlogin</code> in the Reference Manual</td>
</tr>
<tr>
<td><strong>System security officer on local server:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Set use security services on, and set the rpc security model B as the model for connections with the local server.</td>
<td><code>sp_configure</code> – to set use security services <code>sp_serveroption</code> – to set the RPC security model.</td>
<td>“Establishing the security model for RPCs” on page 517</td>
</tr>
<tr>
<td></td>
<td><code>sp_configure &quot;use security services&quot;, 1</code></td>
<td>“Enabling network-based security” on page 510</td>
</tr>
<tr>
<td></td>
<td><code>sp_serveroption lcl_server, &quot;rpc security model B&quot;, true</code></td>
<td>use security services in Chapter 5, “Setting Configuration Parameters”</td>
</tr>
<tr>
<td><strong>System administrator on remote server:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. Optionally, specify certain users as “trusted” to log in to the remote server from the local server without supplying a password.</td>
<td><code>sp_remoteoption</code></td>
<td>“Password checking for remote users” on page 494</td>
</tr>
<tr>
<td></td>
<td><code>sp_remoteoption lcl_server, user1, user1, trusted, true</code></td>
<td><code>sp_remoteoption</code> in the Reference Manual</td>
</tr>
<tr>
<td><strong>System security officer on local server:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6. Add both the local server and the remote server to master..sys servers.</td>
<td><code>sp_addserver</code></td>
<td>“Adding a remote server” on page 485</td>
</tr>
<tr>
<td></td>
<td><code>sp_addserver rem_server</code></td>
<td></td>
</tr>
<tr>
<td><strong>System security officer on local server:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7. Add logins to master..syslogins.</td>
<td><code>sp_addlogin</code></td>
<td>“Adding logins to Adaptive Server” on page 394</td>
</tr>
<tr>
<td></td>
<td><code>sp_addlogin user1, &quot;pwuser1&quot;</code></td>
<td><code>sp_addlogin</code> in the Reference Manual</td>
</tr>
</tbody>
</table>
### Example of setting up security model B for RPCs

Assume that:

- A local server, `lcl_serv`, will run RPCs on a remote server, `rem_serv`.
- Both servers will use security model B and the DCE security service.
- These RPC security services will be in effect: mutual authentication and message integrity.
- Users “user1” and “user2” will use unified login to log in to the local server, `lcl_serv`, and run RPCs on `rem_serv`. These users will be “trusted” on `rem_serv` and will not need to specify a password for the remote server.
- User “user3” will not use unified login, will not be trusted, and must supply a password to Adaptive Server when logging in.

Use the following sequence of commands to set up security for RPCs between the servers:

**System security officer on remote server (rem_serv):**

```plaintext
sp_addserver 'lcl_serv'
sp_addlogin user1, "eracgl2"
sp_addlogin user2, "esirpret"
sp_addlogin user3, "drabmok"
sp_configure "use security services", 1
```

### Task, who performs it, and where | Command, system procedure, or tool | See
--- | --- | ---
**System security officer on local server:** | | |
8. Set use security services on, and set the rpc security model B as the model for connections with the remote server. | `sp_configure` – to set use security services. | “Establishing the security model for RPCs” on page 517
| | `sp_serveroption` – to set the RPC security model. | “Enabling network-based security” on page 510
| | Example: `sp_configure "use security services", 1` | use security services in Chapter 5. “Setting Configuration Parameters”
| | `sp_serveroption rem_server, "rpc security model B", true` | `sp_configure` and `sp_serveroption` in the Reference Manual

**System security officer on local server:** | | |
9. Specify the security mechanism and the security services to use for connections with the remote server. | `sp_serveroption` | “Setting server connection options” on page 487
| | `sp_serveroption rem_server, "use message integrity", true` | |
sp_serveroption lcl_serv, "rpc security model B", true
sp_serveroption lcl_serv, "security mechanism", dce

System administrator on remote server (rem_serv):
sp_remoteoption lcl_serv, user1, user1, trusted, true
sp_remoteoption lcl_serv, user2, user2, trusted, true

System security officer on local server (lcl_serv):
sp_addserver lcl_serv, local
sp_addserver rem_serv
sp_addlogin user1, "eracgl2"
sp_addlogin user2, "esirpret"
sp_addlogin user3, "drabmo1"
sp_configure "use security services", 1
sp_configure rem_serv, "rpc security model B", true
sp_serveroption rem_serv, "security mechanism", dce
sp_configure rem_serv, "mutual authentication" true
sp_serveroption rem_serv, "use message integrity" true

In addition, the interfaces file or Directory Service must have entries for rem_serv and lcl_serv. Each entry should specify the “dce” security service. For example, you might have these interfaces entries, as created by the dscp utility:

```plaintext
## lcl_serv (3201)
lcl_serv
master tli tcp /dev/tcp \x0002000c182d6551100000000000000000
query tli tcp /dev/tcp \x0002000c182d6551100000000000000000
secmech 1.3.6.1.4.1.897.4.6.1
## rem_serv (3519)
rem_serv
master tli tcp /dev/tcp \x000214ad82d655110000000000000000000
query tli tcp /dev/tcp \x000214ad82d655110000000000000000000
secmech 1.3.6.1.4.1.897.4.6.1
```

**Note** To actually use the security services on either server, you must restart the server so that the static parameter use security services takes effect.

For detailed information about setting up servers for remote procedure calls, see Chapter 15, “Managing Remote Servers.”
CHAPTER 16  
External Authentication

Getting information about remote servers

sp_helpserver displays information about servers. When it is used without an argument, it provides information about all the servers listed in sysservers. You can specify a particular server to receive information about that server. The syntax is:

```
sp_helpserver [server]
```

For example, to display information about the GATEWAY server, execute:

```
sp_helpserver GATEWAY
```

Connecting to the server and using the security services

The isql and bcp utilities include the following commandline options to enable network-based security services on the connection:

- `-K keytab_file`
- `-R remote_server_principal`
- `-V security_options`
- `-Z security_mechanism`

These options are described in the following paragraphs.

- `-K keytab_file` — can be used only with DCE security, and specifies a DCE keytab file that contains the security key for the user logging in to the server. You can create keytab files with the DCE dcecp utility—see your DCE documentation for more information.

  If the `-K` option is not supplied, the user of isql must be logged in to DCE. If the user specifies the `-U` option, the name specified with `-U` must match the name defined for the user in DCE.

- `-R remote_server_principal` — specifies the principal name for the server as defined to the security mechanism. By default, a server’s principal name matches the server’s network name (which is specified with the `-S` option or the DSQUERY environment variable). The `-R` option must be used when the server’s principal name and network name are not the same.
Configuring Adaptive Server for Network-Based Security

- `-V security_options` – specifies network-based user authentication. With this option, the user must log in to the network’s security system before running the utility. In this case, if a user specifies the `-U` option, the user must supply the network user name known to the security mechanism; any password supplied with the `-P` option is ignored. `-V` – can be followed by a `security_options` string of key-letter options to enable additional security services. These key letters are:
  - `c` – enable data confidentiality service.
  - `i` – enable data integrity service.
  - `m` – enable mutual authentication for connection establishment.
  - `o` – enable data origin stamping service.
  - `r` – enable data replay detection.
  - `q` – enable out-of-sequence detection.

- `-Z security_mechanism` – specifies the name of a security mechanism to use on the connection.

Security mechanism names are defined in the `libtcl.cfg` configuration file. If no `security_mechanism` name is supplied, the default mechanism is used. For more information about security mechanism names, see the Open Client/Server Configuration Guide for your platform.

If you log in to the security mechanism and then log in to Adaptive Server, you do not need to specify the `-U` option on the utility because Adaptive Server gets the user name from the security mechanism. For example, consider the following session:

```
svrsole4% dce_login user2
Enter Password:
svrsole4% $SYBASE/bin/isql_r -V
1> select suser_name()
2> go
-------------------------------
user2
```

For this example, “user2” logs in to DCE with `dce_login` and then logs in to Adaptive Server without specifying the `-U` option. The `-V` option without parameters implicitly specifies one security service: unified login.

For more information about Adaptive Server utilities, see the Utility Guide.
If you are using Client-Library to connect to Adaptive Server, you can define security properties before connecting to the server. For example, to check message sequencing, set the CS_SEC_DETECTSEQ property. For information about using security services with Client-Library, see the Open Client Client-Library/C Reference Manual.

Example of using security services

Assume that your login is “mary” and you want to use the DCE security mechanism with unified login (always in effect when you specify the -V option of isql or bcp), message confidentiality, and mutual authentication for remote procedures. You want to connect to server WOND and run remote procedures on GATEWAY with mutual authentication. Assuming that a system security officer has set up both WOND and GATEWAY for rpc Model B, added you as a user on both servers, and defined you as a remote, “trusted” user on GATEWAY, you can use the following process:

1. Log in to the DCE security mechanism and receive a credential:

   dce_login mary

2. Log in to the Adaptive Server with isql:

   isql -SWOND -Vcm

3. Run:

   GATEWAY...sp_who
   GATEWAY...mary_prcl
   GATEWAY...mary_prc2

Now, all messages that Mary sends to the server and receives from the server are encrypted (message confidentiality), and when she runs remote procedures, both the WOND and GATEWAY servers are authenticated.

Using security mechanisms for the client

Adaptive Server, when it is started, determines the set of security mechanisms it supports. (See “Determining security mechanisms to support” on page 514. From the list of security mechanisms that Adaptive Server supports, it must choose the one to be used for a particular client.

If the client specifies a security mechanism (for example with the -Z option of isql), Adaptive Server uses that security mechanism. Otherwise, it uses the first security mechanism listed in the libtcl.cfg file.
Getting information about available security services

Adaptive Server enables you to:

- Determine what security mechanisms and services are supported by Adaptive Server
- Determine what security services are active for the current session
- Determine whether a particular security service is enabled for the session

Determining supported security services and mechanisms

A system table, `syssecmechs`, provides information about the security mechanisms and security services supported by Adaptive Server. The table, which is dynamically built when you query it, contains these columns:

- `sec_mech_name`—is the name of the security mechanism; for example, the security mechanism might be “dce” or “NT LANMANAGER.”
- `available_service`—is the name of a security service supported by the security mechanism; for example, the security service might be “unified login.”

The table may have several rows for a single security mechanism: one row for each security service supported by the mechanism.

To list all the security mechanisms and services supported by Adaptive Server, run this query:

```sql
select * from syssecmechs
```

The result might look something like this:

<table>
<thead>
<tr>
<th>sec_mech_name</th>
<th>available_service</th>
</tr>
</thead>
<tbody>
<tr>
<td>dce</td>
<td>unifiedlogin</td>
</tr>
<tr>
<td>dce</td>
<td>mutualauth</td>
</tr>
<tr>
<td>dce</td>
<td>delegation</td>
</tr>
<tr>
<td>dce</td>
<td>integrity</td>
</tr>
<tr>
<td>dce</td>
<td>confidentiality</td>
</tr>
<tr>
<td>dce</td>
<td>detectreplay</td>
</tr>
<tr>
<td>dce</td>
<td>detectseq</td>
</tr>
</tbody>
</table>
Determining enabled security services

To determine which security services are enabled for the current session, use the function `show_sec_services`. For example:

```sql
select show_sec_services()
--------------------------------------------------
unifiedlogin mutualauth confidentiality
(1 row affected)
```

Determining whether a security service is enabled

To determine whether a particular security service, such as “mutualauth” is enabled, use the function `is_sec_service_on`. The following is the syntax, where `security_service_nm` is a security service that is available:

```sql
is_sec_service_on(security_service_nm)
```

Use the name that is displayed when you query `syssecmechs`.

For example, to determine whether “mutualauth” is enabled, execute:

```sql
select is_sec_service_on("mutualauth")
-----------
1
(1 row affected)
```

A result of 1 indicates the security service is enabled for the session. A result of 0 indicates the service is not in use.

Using Kerberos

Kerberos is a network authentication protocol that uses secret key cryptography so that a client can prove its identity to a server across a network connection. User credentials are obtained when the user logs in to the operating system, or by executing an authentication program. These credentials are then used by each application to perform authentication. Users only have to log in once, instead of having to log in to each application.

Kerberos assumes the KDC is running and properly configured for your realm, and the client libraries are installed under or on each client host in your realm. For configuration information, consult the documentation and the reference pages that come with the Kerberos software.

Adaptive Server supports Kerberos through:
Configuring Adaptive Server for Network-Based Security

- CyberSafe Kerberos libraries
- MIT Kerberos libraries, version 1.3.1
- Native libraries

Note To enable Kerberos security options, you must have ASE_SECDIR, the “Security and directory services” package.

Kerberos compatibility

Table 16-7 shows which variation of Kerberos is supported on which platforms.

<table>
<thead>
<tr>
<th>Hardware platforms</th>
<th>KDC server</th>
<th>GSS client</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solaris 32</td>
<td>CSF, AD, MIT</td>
<td>CSF, MIT, Native</td>
</tr>
<tr>
<td>Solaris 64</td>
<td>CSF, AD, MIT</td>
<td>CSF, MIT, Native</td>
</tr>
<tr>
<td>Linux 32</td>
<td>CSF, AD, MIT</td>
<td>MIT, Native</td>
</tr>
<tr>
<td>Windows 32</td>
<td>CSF, AD</td>
<td>CSF</td>
</tr>
<tr>
<td>AIX 32</td>
<td>CSF</td>
<td>CSF</td>
</tr>
</tbody>
</table>

Use the following keys to read the interoperability matrix:

- CSF – CyberSafe Ltd.
- AD – Microsoft Active Directory
- MIT – MIT version 1.3.1

For the latest Adaptive Server feature matrix, see http://www.sybase.com/detail?id=1034492.

Starting Adaptive Server under Kerberos

To start Adaptive Server under Kerberos, add the Adaptive Server name to the KDC and extract the service key to a key table file. For example:

```
/krb5/bin/admin admin/ASE -k -t /krb5/v5srvtab -R
addrn my_ase; mod
my_ase attr nopwchg; ext -n my_ase eytabfile.krb5
Connecting as: admin/ASE
Connected to csfA5v01 in realm ASE.
Principal added.
Principal modified.
```
Key extracted.
Disconnected.

**Note** The administrator can also be authenticated using a password on the command line. In this example, the `-k` option is used, which tells the administrator to search the `/krb5/v5srvtab` file (specified using the `-t` option) for the administrator and the Adaptive Server key, instead of prompting for a password, which is useful for writing shell scripts.

**Configuring Kerberos**

The configuration process is similar, regardless of which variety of Kerberos is used. To configure Kerberos:

1. Set up Kerberos third-party software and create a Kerberos administrative user. To do this, you must:
   - Install Kerberos client software on machines where Open Client Server clients or Adaptive Server will run. The following client packages have been verified to be working:
     - CyberSafe TrustBroker 4.0
     - MIT Kerberos version 1.3.1
   - Install the Kerberos KDC server on a separate, dedicated machine.

   **Note** KDCs from CyberSafe TrustBroker 4.0, MIT Kerberos v.1.3.1, and Microsoft Windows Active Directory have been verified for use with Adaptive Server.

   - Create an administrator account on the Kerberos server with administration privileges. This account is used for subsequent client actions such as creating principals from the client machines.

   **Note** Execute the remainder of these steps on the Kerberos client machine.

2. Add Kerberos principal for Adaptive Server `ase120srv` or `ase120srv@MYREALM`.

3. Extract the `keytab` file for principal `ase120srv@MYREALM` and store it as a file:
/krb5/v5srvtab

The following UNIX examples use the command line tool kadmin, available with CyberSafe or MIT Kerberos. There are also GUI tools available to aid in administration of Kerberos and users:

CyberSafe Kadmin:
% kadmin aseadmin
Principal - aseadmin@MYREALM
Enter password:
Connected to csfA5v01 in realm ASE.
Command: add ase120srv
Enter password:
Re-enter password for verification:
Principal added.
Command: ext -n ase120srv
Service Key Table File Name (/krb5/v5srvtab):
Key extracted.
Command: quit
Disconnected.

In a production environment, you must control the access to the keytab file. If a user can read the keytab file, he or she can create a server that impersonates your server.

Use chmod and chgrp so that /krb5/v5srvtab is:
-rw-r----- 1 root sybase 45 Feb 27 15:42 /krb5/v5srvtab

When using Active Directory as the KDC, log in to the Domain Controller to add users and Adaptive Server principals. Use the Active Directory Users and Computers wizard to guide you through the creation of users and principals.

Extracting the keytab file for use with Adaptive Server requires an optional tool called ktpass, which is included in the Microsoft Support Tools package.

With Active Directory, extracting the keytab with ktpass is done as a separate step from creating the principal. The keytab file on Windows for Adaptive Server is located with the CyberSafe program files. For example, c:\Program Files\CyberSafe\v5srvtab is the expected location of Adaptive Server’s keytab file when CyberSafe software is installed on the C: drive.

Add a Kerberos principal for the user “sybuser1” as “sybuser1@MYREALM”.
5 Start Adaptive Server and use isql to log in as “sa”. The following steps configure Adaptive Server parameters to use Kerberos security services, and create the user login account. These are the same on both Windows or UNIX machines:

- Change configuration parameter use security services to 1:
  
  `1> sp_configure 'use security services', 1`

- Add new login for user, “sybuser1” and then add the user:
  
  `1> sp_addlogin sybuser1, password`

6 Shut down Adaptive Server and modify administrative files and connectivity configuration files.

- On UNIX platforms, the `interfaces` file is under `$SYBASE/` and has an entry that looks similar to:
  
  ```
  ase120srv
  master tli tcp myhost 2524
  query tli tcp myhost 2524
  secmech 1.3.6.1.4.1.897.4.6.6
  ```

  On Windows platforms, the `sql.ini` file is in `%SYBASE%\ini`, and has an equivalent server entry that looks like:

  ```
  [ase120srv]
  master=TCP,myhost,2524
  query=TCP,myhost,2524
  secmech=1.3.6.1.4.1.897.4.6.6
  ```

- The `libtcl.cfg` or `libtcl64.cfg` file is located in `$SYBASE/$SYBASE_OCS/config/` on UNIX platforms. The SECURITY section should have an entry that looks similar to the following for CyberSafe Kerberos client libraries:

  ```
  [SECURITY]
  csfkrb5=libsybskrb.so secbase=@MYREALM
  libgss=/krb5/lib/libgss.so
  ```

  A 64-bit CyberSafe Kerberos client library entry follows:

  ```
  [SECURITY]
  csfkrb5=libsybskrb64.so secbase=@MYREALM libgss=\/
  /krb5/appsec-rt/lib/64/libgss.so
  ```

  For a machine that uses MIT Kerberos client libraries, the entry looks something like:
For a machine that uses Native OS provided libraries, such as Linux, it looks similar to:

```
[SECURITY]
csfkrb5=libsybskrb.so
secbase=@MYREALM
libgss=/opt/mitkrb5/lib/libgssapi_krb5.so
```

On Windows NT, the `%SYBASE%\%SYBASE_OCS%\ini\libtcl.cfg` file contains an entry like:

```
[SECURITY]
csfkrb5=libskrb  secbase=@MYREALM
libgss=C:\WinNT\System32\gssapi32.dll
```

Note: Note the `libgss=<gss shared object path>` that specifies the GSS API library to be used. It is important that you distinctly locate the Kerberos Client libraries being used, especially when multiple versions are installed on a machine.

- Also check the `objectid.dat` under `$SYBASE/$SYBASE_OCS/config/` and make sure the `[secmech]` section has an entry for `csfkrb5`:

```
[secmech]
1.3.6.1.4.1.897.4.6.6 = csfkrb5
```

You can use environment variables to override default locations of `keytab` files, Kerberos configuration, and realm configuration files. This is Kerberos-specific behavior and may not work consistently on all platforms.

For example, the `CSFC5KTNAME` environment variable can be used on CyberSafe UNIX platforms to specify the `keytab` file:

```
% setenv CSFC5KTNAME /krb5/v5srvtab
```

For MIT Kerberos, the equivalent environment variable is `KRB5_KTNAME`.

See the vendor documentation for information about these environment variables.
Your application may also need to modify the environment variable for dynamic library search paths. On UNIX, the most commonly used environment variable is LD_LIBRARY_PATH; on Windows, PATH is typically set to include DLL locations. You may need to modify these environment variables to enable applications to load the third-party objects correctly. For example this command adds the location of CyberSafe 32-bit libgss.so shared object to the search path in a C-Shell environment:

```
% set path = ( /krb5/lib $path )
```

8 Restart Adaptive Server. You should see the following log message during start-up:

```
00:0000:0000:0000:2001/07/25 11:43:09.91 server
Successfully initialized the security mechanism 'csfkrb5'. The SQL Server will support use of this security mechanism.
```

9 Use `isql` as UNIX user “sybuser1” (without the `-U` and `-P` arguments) to connect:

```
% $SYBASE/$SYBASE_OCS/bin/isql -Sase120srv -V
```

You can also use the encryption option:

```
$SYBASE/$SYBASE_OCS/bin/isql -Sase120srv –Vc
```

Using principal names

The principal name is the name the server uses to authenticate with the Kerberos Key Distribution Center (KDC). When you have multiple instances of Adaptive Server running, you must have different principal names for each Adaptive Server.

Specifying the Adaptive Server principal name

Use a `dataserver` option and an environment variable to specify a principal name different from the Adaptive Server name. Adaptive Server name is specified by environment variables DSLISTEN and DSQUERY, or the `dataserver` command-line option `-s servername`.

You can set the principal name either the `setenv` command or the `-k dataserver` option.
By default, the principal name is the name of Adaptive Server. To specify a different name, set SYBASE_PRINCIPAL before starting Adaptive Server to use Kerberos:

```
setenv SYBASE_PRINCIPAL <name of principal>
```

Once you have set an Adaptive Server principal name, Adaptive Server uses the value of this variable to authenticate itself with Kerberos.

You can use the following command-line option to specify an Adaptive Server principal name, when starting Adaptive Server:

```
-k <server principal name>
```

When you start an Adaptive Server with the Kerberos security mechanism enabled, Adaptive Server first uses the principal name specified with the `-k` option for Kerberos authentication. If the `-k` option is not specified, Adaptive Server looks for the principal name in the environment variable SYBASE_PRINCIPAL. If neither is specified, Adaptive Server uses the server name for authentication.

**Example**

In this example, the Adaptive Server name is “secure_ase” and realm name is “MYREALM.COM,” the Adaptive Server name is specified on the command line with `-s` parameter to the `dataserver` command. The current realm is specified in `libtcl.cfg` by a seccbase attribute value:

```
[SECURITY]
csfkrb5=libskrb.so libgss=/krb5/lib/libgss.so 
seccbase=@MYREALM.COM
```

The default Adaptive Server principal name is “secure_ase@MYREALM.COM.” If the principal name defined in the Adaptive Server `keytab` file is “aseprincipal@MYREALM.COM,” you can override the default Adaptive Server principal name by setting a server principal name using options 1 or 2 below:

- **Option 1:** `-k` is specified:

  ```
  %
  $SYBASE/$SYBASE_ASE/bin/dataserver -dmaster.dat 
  -s secure_ase  -k aseprincipal@MYREALM.COM
  ```

  The Adaptive Server principal name used to authenticate with Kerberos is “aseprincipal@MYREALM.COM.”

- **Option 2:** `-k` is not specified but SYBASE_PRINCIPAL is set:

  ```
  setenv SYBASE_PRINCIPAL aseprincipal@MYREALM.COM
  $SYBASE/$SYBASE_ASE/bin/dataserver -dmaster.dat 
  -s secure_ase
  ```
The Adaptive Server principal name used to authenticate with Kerberos is “aseprincipal@MYREALM.COM,” the value of $SYBASE_PRINCIPAL.

- Option 3: Neither -k nor SYBASE_PRINCIPAL is set

  % $SYBASE/$SYBASE_ASE/bin/dataserver -dmaster.dat -s secure_ase

  The Adaptive Server principal name used to authenticate with Kerberos is “secure_ase@MYREALM.COM.”

For more information about Kerberos, see the Security section of the System Administration Guide, Volume One.

Using sybmapname to handle user principal names

sybmapname converts external user principal names used in Kerberos environment to the namespace of Adaptive Server user logins. sybmapname is a customizable shared object that can map names given on its input buffer to a name suitable for Adaptive Server login on its output buffer.

You can use sybmapname shared object to perform the custom mapping between the user principal name and the Adaptive Server login name. This shared object is optionally loaded at server start-up, and the function syb_map_name contained in the shared object is called after a successful Kerberos authentication and just before the user principal is mapped to a login in the syslogins table. This function is useful when the user principal name and the login name to be mapped are not identical.

The customizable logic is the function:

  syb_map_name(NAMEMAPTYPE *protocol, char *orig, int origlen, char *mapped, int *mappedlen)

Where:

- NAMEMAPTYPE *protocol – refers to a structure reserved for usage of this function.
- char *orig – is an input buffer that is not null-terminated.
- int origlen – is the input buffer length. It should be less than or equal to 255 characters.
- char *mapped – is an output buffer that should not be null-terminated.
- int *mappedlen – is an output buffer length. It should be less than or equal to 30.
The function returns a value greater than 0 if the mapping succeeds, it returns a value of 0 if no mapping occurred, and it returns a value less than 0 when an error occurs in \texttt{syb\_map\_name()}. When an error occurs, a message displays in the Adaptive Server error log to report the mapping failure.

For example, to authenticate a Kerberos user on Adaptive Server:


A sample \texttt{sybmapname.c} file is located in 
$\texttt{SYBASE}/\$\texttt{SYBASE\_ASE}/\texttt{sample/server/sybmapname.c}.

2. Modify \texttt{sybmapname.c} to implement your logic. Take precautions while coding as it may interfere with the proper running of Adaptive Server. See “Precautions when using sybmapname” on page 538.

3. Build the shared object or DLL using the generic platform specific \texttt{makefile} supplied. The \texttt{makefile} may need to be modified to suit your platform specific settings.

4. Place the resulting shared object generated in a location specified in your $\texttt{LD\_LIBRARY\_PATH}$ on UNIX machines, and $\texttt{PATH}$ variable on Windows machines. The file should have read and execute permissions for the sybase user.

\textbf{Note}  Sybase recommends that only the “sybase” user is allowed read and execute permissions, and that all other access should be denied.

\textbf{Verifying your login to Adaptive Server using Kerberos authentication}

To verify your login to Adaptive Server using Kerberos authentication, assume that:

- \$\texttt{SYBASE}$ refers to your release and installation directory.
- \$\texttt{SYBASE\_ASE}$ refers to the Adaptive Server version directory that contains your server binary.
- \$\texttt{SYBASE\_OCS}$ refers to the Open Client/Server version directory.
Example 1  If a client’s principal name is user@REALM, and the
corresponding entry in syslogins table is user_REALM, then sybmapname can
be coded to accept the input string user@realm and to convert the input string
to the output string user_REALM.

Example 2  If the client principal name is user, and the corresponding entry
in syslogins table is USER, then sybmapname can be coded to accept the input
string user and convert this string to uppercase string USER.

sybmapname is loaded by Adaptive Server at runtime and uses its logic to do
the necessary mapping.

The following actions and output further illustrate the sybmapname function
described in Example 2. The sybmapname.c file containing the customized
definition for syb__map_name() should be compiled and built as a shared
object (or DLL), and finally placed in the appropriate path location. Start
Adaptive Server with the Kerberos security mechanism enabled.

To initialize the Ticket Granted Ticket (TGT):

\$ /krb5/bin/kinit johnd@public
Password for johnd@public:
\$

To list the TGT:

\$ /krb5/bin/klist
  Cache Type: Kerberos V5 credentials cache
  Cache Name: /krb5/tmp/cc/krb5cc_9781
  Default principal: johnd@public

Log in as “sa” and verify the user login for “johnd”:

\$ $SYBASE/$SYBASE_OCS/bin/isql -Usa -P
      -Ipwd`/interfaces
1>

1> sp_displaylogin johnd
2> go
No login with the specified name exists.
(return status = 1)

1> sp_displaylogin JOHND
2> go
  Suid: 4
  Loginame: JOHND
  Fullname:
  Default Database: master
  Default Language:
Auto Login Script:  
Configured Authorization:  
Locked: NO  
Password expiration interval: 0  
Password expired: NO  
Minimum password length: 6  
Maximum failed logins: 0  
Current failed login attempts:  
Authenticate with: ANY  
(return status = 0)

Successful Kerberos authentication, which maps lower case johnd to uppercase JOHND using the sybmapname utility and allows user johnd to log in to Adaptive Server:

```bash
$ $SYBASE/$SYBASE_OCS/bin/isql -V -I'pwd'/interfaces 1>
```

Precautions when using sybmapname

Be aware of the following issues when coding for sybmapname:

- Take care with the sample sybmapname.c program and any modifications to it. Avoid using code that may create a segmentation fault, that may call exit(), that may call system calls(), that may change UNIX signals, or that makes any blocking calls. Improper coding or calls may interfere with the Adaptive Server engine.

  **Note** Sybase bears no responsibility for coding errors in sybmapname.

- Code defensively, check all pointers before dereferencing them, and avoid system calls. The functions you write must be quick name-filtering functions.

- Do not use goto statements since, depending on the platform, they may cause unexpected side effects.

- If you use multiple realms, take care to map the user principal names to a suitable login name to reflect the realm information. For example, if you have two users whose user principal names are userA@REALMONE and userB@REALMTWO, respectively, map them to the login names userA_REALMONE and userB_REALMTWO, instead of userA or userB. This distinguishes the two users who belong to different realms.
Configuring Adaptive Server for LDAP User Authentication

The LDAP user authentication allows client applications to send user name and password information to Adaptive Server for authentication by the LDAP server instead of syslogins. Authentication using the LDAP server allows you to use server-wide passwords instead of Adaptive Server or application-specific passwords.

LDAP user authentication is ideal for organizations with an existing computing environment who want to simplify and centralize user administration, or for users in a new computing environment who want to avoid unnecessary complexities for administering users.

LDAP user authentication works with directory servers that meet Version 3 of the LDAP protocol standard, including Active Directory, iPlanet, and OpenLDAP Directory Server.

You can use two authentication algorithms with LDAP user authentication, which differ in how they obtain a user’s Distinguished Name (DN). The algorithms use either:

- Composed DN for authentication, available for Adaptive Server version 12.5.1 or later, or,
- Searched DN for authentication, available for Adaptive Server version 12.5.2 and later.

The primary data structure used with the LDAP protocol is the LDAP URL. An LDAP URL specifies a set of objects or values on an LDAP server. Adaptive Server uses LDAP URLs to specify an LDAP server and search criteria to use to authenticate login requests.

The LDAP URL uses this syntax:

```
ldapurl::=ldap://host:port/node/?attributes?base | one | sub?filter
```

where:

- `host` – is the host name of the LDAP server.
- `port` – is the port number of the LDAP server.
- `node` – specifies the node in the object hierarchy at which to start the search.
- `attributes` – is a list of attributes to return in the result set. Each LDAP server may support a different list of attributes.
base | one | sub – qualifies the search criteria. base specifies a search of the base node; one specifies a search of the base node and one sublevel below the base node; sub specifies a search of the base node and all node sublevels.

filter – specifies the attribute or attributes to be authenticated. The filter can be simple, such as uid=*, or compound, such as (uid=*)(ou=group).

Composed DN algorithm

The following steps describe the login sequence when you use the composed DN algorithm:
1. Open Client connects to an Adaptive Server listener port.
2. The Adaptive Server listener accepts the connection.
3. Open Client sends an internal login record.
5. Adaptive Server binds to the LDAP server with a DN composed from the primary URL and the login name from the login record. This bind also uses the password from the login record.
6. The LDAP server authenticates the user, returning either a success or failure message.
7. If the Primary URL specifies a search, then Adaptive Server sends the search request to the LDAP server.
8. The LDAP server returns the results of the search.
9. Adaptive Server accepts or rejects the login, based on the search results.

Searched DN algorithm

The following steps describe the login sequence when you use the searched DN algorithm:
1. Open Client connects to an Adaptive Server listener port.
2. The Adaptive Server listener accepts the connection.
3. Open Client sends an internal login record.
CHAPTER 16  External Authentication

5 Adaptive Server binds to the LDAP Server with a directory server access account.
   The connection established in steps 5 and 6 may persist between authentication attempts from Adaptive Server to reuse connections to DN searches.

6 The LDAP server authenticates the user, returning either a success or failure message.

7 Adaptive Server sends search requests to LDAP server based on the login name from the login record and the DN lookup URL.

8 The LDAP server returns the results of the search.

9 Adaptive Server reads the results to obtain an a value of attribute from the DN lookup URL.

10 Adaptive Server uses the value of attribute as the DN and the password from the login record to bind to the LDAP server.

11 The LDAP server authenticates the user, returning either a success or failure message.

12 If the primary URL specifies a search, Adaptive Server sends the search request to the LDAP server.

13 The LDAP Server returns the results of the search.

14 Adaptive Server accepts or rejects the login, based on the search results.

Adaptive Server reports a generic login failure to the client if any of these authentication criteria are not met.

You may skip steps 12 and 13 by not specifying search criteria in the primary or secondary URL strings. When you do not specify criteria in the primary or secondary URL strings, the authentication completes, displaying the success or failure returned by step 11.

Configuring LDAP

These are the steps for configuring Adaptive Server for LDAP authentication.

1 Specify the Adaptive Server LDAP URL search strings and access account values.

2 Set enable ldap user auth to 2.
3 Add users in the LDAP directory server using LDAP vendor-supplied tools.

4 Add users to Adaptive Server using `sp_addlogin`. You can also use `sp_maplogin` to automatically create login accounts upon authentication or apply other login controls.

To avoid disruption of service in existing server installations, migrate Adaptive Server to LDAP:

- Specify an LDAP URL search string to Adaptive Server.
- Set the configuration parameter `enable ldap user auth` to 1.
- Add users in the LDAP directory server.
- When all users are added to the LDAP server, set `enable ldap user auth` to 2 to require all authentications to be performed with LDAP, or use `sp_maplogin` to override configuration parameters with login controls.

### LDAP user authentication administration

Use `sp_ldapadmin` to create or list an LDAP URL search string, verify an LDAP URL search string or login, and specify the access accounts and tunable LDAPUA related parameters. You must have the SSO role to execute `sp_ldapadmin`.

See the Reference Manual: Commands for more information about `sp_ldapadmin`.

### `sp_ldapadmin` examples

You can use a composed DN algorithm for user authentication if you use a simple LDAP server topology and schema. If you use commercially available schemas (for example, iPlanet Directory Servers or OpenLDAP Directory Servers), users are created as objects in the same container in the LDAP server tree, and Adaptive Server determines the user’s DN from the object’s location. However, there are restrictions on the LDAP server’s schema:

- You must specify the filter with the attribute name that uniquely identifies the user to be authenticated.
- You must specify the filter with the attribute `name=*`. The asterisk is a wildcard character. The appropriate attribute name to use in the filter depends on the schema used by the LDAP server.
CHAPTER 16  External Authentication

- The Adaptive Server login name is the same as the short user name for example, a UNIX user name.

- The DN uses the short user name rather than a full name with embedded spaces or punctuation. For example, jquplic public meets the restriction for a DN, but “John Q. Public” as the DN does not.

**iPlanet example**

LDAP vendors may use different object names, schema, and attributes than those used in these examples. There are many possible LDAP URL search strings, and valid sites may also extend schemas locally or use them in ways different from each other:

- This example uses the *uid=* filter. To compose the DN, Adaptive Server replaces the wildcard with the Adaptive Server login name to be authenticated, and appends the resulting filter to the node parameter in the LDAP URL. The resulting DN is:

  \[ \text{uid=myloginname,ou=People,dc=mycomany,dc=com} \]

- After a successful bind operation, Adaptive Server uses the connection to search for attribute names such as \( \text{uid} \), that are equal to the login name:

```bash
sp_ldapadmin set_primary_url,
'ldap://myhost:389/ou=People,dc=mycomany,dc=com??sub?uid=*
```

- This example uses the schema defined in OpenLDAP 2.0.25, with an attribute name of \( \text{cn} \).

  The composed DN is \( \text{cn=myloginname,dc=mycompany,dc=com} \):

```bash
sp_ldapadmin set_primary_url,
'ldap://myhost:389/dc=mycompany,dc=com??sub?cn=*
```

**Searched DN examples**

Use the searched DN to use an Active Directory server or other LDAP server environment that does not meet the restrictions to use the composed DN algorithm.

- Perform these steps for an Active Directory server using commercially available user schema from a Windows 2000 Server.

  a  Set the access account information:

  ```bash
  sp_ldapadmin set_access_acct,
  'cn=Admin Account, cn=Users, dc=mycompany, dc=com',
  'Admin Account secret password'
  ``

  b  Set the primary URL:

  ```bash
  sp_ldapadmin set_primary_url, 'ldap://hostname:389/
  ``

  c  Set the DN lookup URL search string:
sp_ldapadmin set_dn_lookup_url,
'ldap://hostname:389/cn=Users,dc=mycompany,dc=com?distinguishedName
?one?samaccountname=*'

On Windows 2000, the short name is typically referred to as the “User Logon
Name” and is given the attribute name samaccountname in the default schema.
This is the attribute name used to match the Adaptive Server login name. The
DN for a user contains a full name with punctuation and embedded spaces (for
example, cn=John Q. Public, cn=Users, dc=mycomany, dc=com. The
DN on Windows does not use the short name, so the searched DN algorithm is
appropriate for sites using the Active Directory schema (the default) for their
LDAP server. The primary URL does not specify a search. Instead, it relies on
the bind operation for the authentication.

Examples using
search filters to
restrict Adaptive
Server access

You can use LDAP URL search strings to restrict access to groups of users on
LDAP servers. For example, to restrict logins to users in an accounting group,
use a compound filter to restrict access to the group of users where attribute
group=accounting.

- The following LDAP URL string uses the composed DN algorithm for an
  iPlanet server:

  sp_ldapadmin set_primary_url,
  'ldap://myhost:389/ou=People,dc=mycompany,
  dc=com??sub?(&(uid=*)(group=accounting))'

  Adaptive Server binds with DN
  uid=mylogin,ou=People,dc=mycompany,dc=com. After successfully
  binding with this identity, it searches for:

  "ou=People,dc=mycompany,dc=com??sub?(&{uid=mylogin}(group=accounting))"

  Authentication succeeds if this search returns any objects.

  These examples use LDAP URL strings with compound filters:

  sp_ldapadmin set_primary_url,
  'ldap://myhost:389/ou=people,dc=mycompany,dc=com??sub?(&{uid=*)(ou=accounting) (l=Santa Clara))'

  sp_ldapadmin, set_primary_url,
  'ldap://myhost:389/ou=people,dc=mycompany,dc=com??sub?(&{uid=*)(ou=Human%20Resources))'

LDAP user authentication password information changes

There are two LDAP user authentication-related informational messages that
Adaptive Server obtains from the LDAP server and passes to the client:

544 Adaptive Server Enterprise
• If you log in to an Adaptive Server using an LDAP authentication mechanism with an LDAP user authentication password that is about to expire, the following message displays:

    Your password will expire in <number> days.

• If you attempt to log in to Adaptive Server using an LDAP authentication mechanism after the LDAP server administrator resets your password or after your LDAP server password has expired, you see a message 4002:

    Login failed

    If auditing is enabled and the errors auditing option is turned on, a 4099 message is sent to the audit log:

    sp_audit 'errors', 'all', 'all', 'on'

    The 4099 message reads:

    Your LDAP password has expired.

**Note** Configure your LDAP server to give this additional information. Additionally, Adaptive Server must support the transmission of LDAP password controls to an LDAP client.

**Failover support**

When a major failure occurs in the LDAP directory server specified by the primary URL and the server no longer responds to network requests, Adaptive Server attempts to connect to the secondary LDAP directory server specified by the secondary URL. Adaptive Server uses the LDAP function `ldap_init` to determine if it can open a connection to the LDAP directory server. A NULL or invalid primary URL string causes Adaptive Server to attempt failover to a secondary URL. Failures returned by LDAP bind or search operations do not cause Adaptive Server to fail over to the secondary URL.
Adaptive Server logins and LDAP user accounts

Once you enable LDAP user authentication, choose and set an authentication algorithm and URL strings, you must configure the user accounts. The LDAP administrator creates and maintain accounts in the LDAP server, and the database administrator creates and maintains accounts in Adaptive Server. Alternatively, the database administrator can choose administration options that allow flexibility with login accounts when integrating Adaptive Server with external authentication mechanisms such as LDAP server. The database administrator continues to administer the Adaptive Server account roles, default database, default language, and other login-specific attributes using traditional commands and procedures.

Table 16-8 describes the updates to syslogins table Adaptive Server makes at login time. These updates assume that LDAP user authentication is configured, the login is not restricted from using LDAP, and you have not set the create login mapping.

<table>
<thead>
<tr>
<th>Does the row exist in syslogins?</th>
<th>LDAP server authentication succeeds?</th>
<th>Changes in syslogins</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>Yes</td>
<td>No change, login fails</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>No change, login fails</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Update row if password has changed</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>No change</td>
</tr>
</tbody>
</table>

Secondary lookup server support

Adaptive Server provides uninterrupted support to Adaptive Server clients authenticated by an LDAP server. You can specify a secondary LDAP lookup server to fail over from a primary LDAP server in the event of the LDAP server failure or planned downtime.

The health of the URL set is monitored through the following states:

- INITIAL – indicates that LDAP user authentication is not configured.
- RESET – indicates that the URL has been entered with Adaptive Server administrative commands.
- READY – indicates that the URL is ready to accept connections.
CHAPTER 16  External Authentication

- ACTIVE – indicates that the URL has performed a successful LDAP user authentication.
- FAILED – indicates that there is a problem connecting to the LDAP server.
- SUSPENDED – indicates that the URL is in maintenance mode, and will not be used.

The following sequence of events describe the failover and manual failback:

1. The primary and secondary URL sets are configured and in a READY state.
2. The connections are authenticated using the primary server infrastructure.
3. The primary server fails, and its state is changed to FAILED.
4. Connections automatically begin authentication through the secondary server infrastructure.
5. The primary server is repaired and brought back online by an LDAP administrator. The primary LDAP server state is changed by an Adaptive Server administrator to READY.
6. New connections are authenticated using the primary server.

Note  Once Adaptive Server has failed over to the secondary LDAP server, a database administrator must manually activate the primary LDAP server before it can be used again.

When Adaptive Server encounters errors connecting to an LDAP server, it retries the authentication three times. If the errors persist the LDAP server is marked as FAILED. See “Troubleshooting LDAP user authentication errors” on page 554 for information on the LDAP errors which force Adaptive Server to get into a retry loop.

Use sp_ldapadmin to configure secondary lookup LDAP servers:

- To set the secondary DN lookup URL, enter:
  
  sp_ldapadmin set_secondary_dn_lookup_url, <URL>
- To set the administrative access account for the secondary DN lookup URL, enter:
  
  sp_ldapadmin set_secondary_access_acct, <DN>, <password>
To suspend the use of a primary or secondary URL for authentication, enter:

```
sp_ldapadmin suspend, {primary | secondary}
```

To activate the set of primary or secondary URLs for authentication, enter:

```
sp_ldapadmin activate, {primary | secondary}
```

To display details about the primary and secondary LDAP Server settings and status, enter:

```
sp_ldapadmin list
```

`sp_ldapadmin list` combines previous outputs from `list_access_acct` and `list_urls`. It has the following expected output for the primary and secondary servers:

- Search URL
- Distinguished Name Lookup URL
- Access Account DN
- Active [True | False]
- Status [Ready | Active | Failed | Suspended | Reset]

Adaptive Server version 12.5.4 includes the following `sp_ldapadmin` option changes to support secondary servers.

- To display DN lookup URLs for the secondary server, enter:
  
  ```
  sp_ldapadmin list_urls
  ```

- To display the administrative account for the secondary DN lookup URL, enter:
  
  ```
  sp_ldapadmin list_access_acct
  ```

- To display new subcommands, enter:
  
  ```
  sp_ldapadmin [help | invalid sub-command]
  ```

### LDAP server state transitions

Tables Table 16-9—Table 16-14 list LDAP server state transitions when each of the `sp_ldapadmin` commands is executed.

Table 16-9 shows the state transitions when you execute `sp_ldapadmin set_URL`, where `set_URL` represents either of the following commands:
set_dn_lookup_url
set_primary_url
set_secondary_dn_lookup_url
set_secondary_url

Table 16-9: State transitions when sp_ldapadmin set_URL is executed

<table>
<thead>
<tr>
<th>Initial state</th>
<th>Final state</th>
</tr>
</thead>
<tbody>
<tr>
<td>INITIAL</td>
<td>RESET</td>
</tr>
<tr>
<td>RESET</td>
<td>RESET</td>
</tr>
<tr>
<td>READY</td>
<td>READY</td>
</tr>
<tr>
<td>ACTIVE</td>
<td>RESET</td>
</tr>
<tr>
<td>FAILED</td>
<td>RESET</td>
</tr>
<tr>
<td>SUSPENDED</td>
<td>RESET</td>
</tr>
</tbody>
</table>

Table 16-10 shows the state transitions when you execute sp_ldapadmin suspend.

Table 16-10: State transitions when sp_ldapadmin suspend is executed

<table>
<thead>
<tr>
<th>Initial state</th>
<th>Final state</th>
</tr>
</thead>
<tbody>
<tr>
<td>INITIAL</td>
<td>Error</td>
</tr>
<tr>
<td>RESET</td>
<td>SUSPENDED</td>
</tr>
<tr>
<td>READY</td>
<td>SUSPENDED</td>
</tr>
<tr>
<td>ACTIVE</td>
<td>SUSPENDED</td>
</tr>
<tr>
<td>FAILED</td>
<td>SUSPENDED</td>
</tr>
<tr>
<td>SUSPENDED</td>
<td>SUSPENDED</td>
</tr>
</tbody>
</table>

Table 16-11 shows the state transitions when you execute sp_ldapadmin activate.

Table 16-11: State transitions when sp_ldapadmin activate is executed

<table>
<thead>
<tr>
<th>Initial state</th>
<th>Final state</th>
</tr>
</thead>
<tbody>
<tr>
<td>INITIAL</td>
<td>Error</td>
</tr>
<tr>
<td>RESET</td>
<td>READY</td>
</tr>
<tr>
<td>READY</td>
<td>READY</td>
</tr>
<tr>
<td>ACTIVE</td>
<td>ACTIVE</td>
</tr>
<tr>
<td>FAILED</td>
<td>READY</td>
</tr>
<tr>
<td>SUSPENDED</td>
<td>READY</td>
</tr>
</tbody>
</table>

The following tables show the LDAP server state transitions carried out implicitly by Adaptive Server.
Table 16-12 shows the state transitions when Adaptive Server is restarted:

**Table 16-12: State transitions when Adaptive Server is restarted**

<table>
<thead>
<tr>
<th>Initial state</th>
<th>Final state</th>
</tr>
</thead>
<tbody>
<tr>
<td>INITIAL</td>
<td>INITIAL</td>
</tr>
<tr>
<td>RESET</td>
<td>RESET</td>
</tr>
<tr>
<td>READY</td>
<td>READY</td>
</tr>
<tr>
<td>ACTIVE</td>
<td>READY</td>
</tr>
<tr>
<td>FAILED</td>
<td>FAILED</td>
</tr>
<tr>
<td>SUSPENDED</td>
<td>SUSPENDED</td>
</tr>
</tbody>
</table>

Adaptive Server only attempts an LDAP login if the LDAP server is in a READY or ACTIVE state. Table 16-13 shows the state transitions:

**Table 16-13: State transitions when an LDAP login succeeds**

<table>
<thead>
<tr>
<th>Initial state</th>
<th>Final state</th>
</tr>
</thead>
<tbody>
<tr>
<td>READY</td>
<td>ACTIVE</td>
</tr>
<tr>
<td>ACTIVE</td>
<td>ACTIVE</td>
</tr>
</tbody>
</table>

Table 16-14 shows the state transitions when an LDAP login fails:

**Table 16-14: State transitions when an LDAP login fails**

<table>
<thead>
<tr>
<th>Initial state</th>
<th>Final state</th>
</tr>
</thead>
<tbody>
<tr>
<td>READY</td>
<td>FAILED</td>
</tr>
<tr>
<td>ACTIVE</td>
<td>FAILED</td>
</tr>
</tbody>
</table>

**LDAP user authentication tuning**

You can configure and tune Adaptive Server options based on load of incoming connections and the Adaptive Server-LDAP server infrastructure. You can configure the following two options based on the number of simultaneous incoming requests:

- Use `sp_configure` to set `max native threads`, which indicates the number of native threads per engine.

- Use `sp_ldapadmin` to configure `max_ldapua_native_threads`, which indicates the number of LDAP user authentication native threads per engine.

Configure the following option based on the network and the health of the Adaptive Server/LDAP server infrastructure:
• Use `sp_ldapadmin` to configure `set_timeout` which indicates the LDAP server bind and search timeouts.

Configure the following option to specify Adaptive Server behavior when incoming connections have consumed `max ldapua native threads`:

• Use `sp_ldapadmin` to configure `set_abandon ldapua when full`.

You can use `sp_ldapadmin` to configure the LDAP server for better performance with these options:

• `set_max ldapua desc` – manages the concurrency of the LDAPUA connection requests. If you are using a distinguished name algorithm, setting `set_max ldapua desc` to a larger number expedites the LDAPUA connections Adaptive Server is processing.

• `set_num_retries` – sets the number of attempts. Tune this number according to the number of transient errors between Adaptive Server and the LDAP server. You can nullify the transient errors by configuring the number of retries.

• `set_log_interval` – controls the number of messages sent to the Adaptive Server errorlog for diagnostic purposes. Using a low number clutters the errorlog but is helpful for pinpointing specific errors. Using a large number sends fewer messages to the errorlog, but does not have the same investigative value. Tune `set_log_interval` according to your errorlog size.

Adding tighter controls on login mapping

Use `sp_maplogin` to map users that are authenticated with LDAP or PAM to the local Adaptive Server login.

**Note** To map a user authenticated with Kerberos, use `sybmapname` instead of `sp_maplogin`.

Only users with `sso role` can create or modify login mappings using `sp_maplogin`.

Adaptive Server avoids conflicts between an authentication mechanism setting for a login and a mapping that uses the login. Potential mapping conflicts are detected by the stored procedures `sp_maplogin`, `sp_modifylogin`, or `sp_addlogin`.

The tighter controls no longer permit a map:

• From one Adaptive Server login name to another login name
Configuring Adaptive Server for LDAP User Authentication

- From an external name that already exists as a local login
- To a nonexistent login name

Additionally, when the authentication mechanism is specified with a mapping, the mechanism is checked with the authentication mechanism set in the target login.

If a target login’s authentication mechanism restricts the login to use a particular authentication mechanism, then the mechanism specified with the mapping must match either that specified for the login or match the “ANY” authentication mechanism.

When `sp_maplogin` detects that a conflict exists, `sp_maplogin` fails and reports an error to identify the conflict.

Similarly, `sp_modifylogin` and `sp_addlogin` check for an existing mapping that may conflict with the `authenticate with` option for the user login.

When `sp_modifylogin` or `sp_addlogin` detect a conflict, an error is reported to identify any conflicts with a login mapping.

**Examples**

**Example 1** Maps an LDAP user to Adaptive Server “sa” login. A company has adopted LDAP as their repository for all user accounts and has a security policy that requires LDAP authentication of all users including database administrators, “adminA” and “adminB”, who may manage hundreds of Adaptive Servers. Auditing is enabled and login events are recorded in the audit trail.

To map these administrator accounts to “sa”, enter:

```sql
sp_maplogin LDAP, 'adminA', 'sa'
go
sp_maplogin LDAP, 'adminB', 'sa'
go
```

Use `enable ldap user auth` to require all users to authenticate using LDAP authentication:

```sql
sp_configure 'enable ldap user auth', 2
```

When “adminA” authenticates during login to Adaptive Server, the distinguished name associated with “adminA” rather than only “sa” is recorded in the login audit event. This allows each individual performing an action to be identified in the audit trail.

Because “adminA” and “adminB” password is set in the LDAP server, there is no need to maintain the “sa” password on all Adaptive Servers being managed.
This example also allows different external identities and passwords to be used for authentication, while their actions within Adaptive Server still require the special privileges associated with “sa” account.

**Example 2**  Uses both PAM and LDAP to map users to application logins. A company has adopted both PAM and LDAP authentication but for different purposes. The company security policy defines LDAP as the authentication mechanism for general user accounts, and PAM for special users such as a middle-tier application. A middle-tier application may establish a pool of connections to Adaptive Server to handle requests on behalf of users of the middle-tier application.

Configure Adaptive Server for both LDAP and PAM user authentication:

```sql
sp_configure 'enable ldap user auth', 2
sp_configure 'enable pam user auth', 2
```

Establish an Adaptive Server login `appX` locally with permissions that are appropriate for the middle-tier application:

```sql
sp_addlogin 'appX', password
sp_modifylogin appX, 'authenticate with', PAM
```

Instead of hard-coding a simple password in “appX” and maintaining the password consistently in several different Adaptive Servers, develop a custom PAM module to authenticate the application in a centralized repository using additional facts to verify the middle-tier application.

Client application login “appY” requires LDAP authentication of the user with its LDAP identity and password. Use `sp_maplogin` to map all LDAP authenticated users to login “appY”:

```sql
sp_addlogin 'appY', password
sp_maplogin LDAP, NULL, 'appY'
```

Users of “appY” are authenticated with their company identity and password, then mapped to a local Adaptive Server login “appY” to execute database actions. Authentication has occurred with the identity of the LDAP user, which is recorded in the audit trail, and executes with permissions appropriate to the application login “appY”.

```sql
sp_addlogin 'appY', password
sp_maplogin LDAP, NULL, 'appY'
```
Troubleshooting LDAP user authentication errors

Adaptive Server may experience the following transient errors when communicating with the LDAP server. These errors are generally resolved by retrying the connection. If the errors persist after three retry attempts, Adaptive Server marks the LDAP server as FAILED.

- LDAP_BUSY – server is busy
- LDAP_CONNECT_ERROR – error during a connect
- LDAP_LOCAL_ERROR – error on the client side
- LDAP_NO_MEMORY – cannot allocate memory on the client side
- LDAP_OPERATIONS_ERROR – error on the server side
- LDAP_OTHER – unknown error code
- LDAP_ADMINLIMIT_EXCEEDED – a search has exceeded a limit
- LDAP_UNAVAILABLE – server cannot process the request
- LDAP_UNWILLING_TO_PERFORM – server is not going to process the request
- LDAP_LOOP_DETECT – a loop has been detected during a referral
- LDAP_SERVER_DOWN – server is not reachable (connection fails)
- LDAP_TIMEOUT – LDAP API fails because operation does not complete in the user-specified amount of time

Transient errors and a large number of simultaneous login requests could lead to a large number of repeated error messages in the error log. To increase the readability of the log, the following error message logging algorithm is used:

1. If a message is being logged for the first time, log it.
2. If the last time the message was logged was greater than 3 minutes:
   - Log the error message.
   - Log the number of times the message was repeated since the message was last printed.
   - Log the time elapsed, in minutes, since the message was printed.

Authentication failures arising from the following are not considered LDAP errors and are not conditions for retrying the authentication request.

- Bind failure due to bad password or an invalid distinguished name.
• A search after a successful bind that returns a result set of 0 or no attribute value.

Syntax errors found while parsing the URL are caught when an LDAP URL is set, and therefore do not fall into any of the above categories.

Configuring LDAPS


❖ Configure an LDAPS connection

1 Make sure that all trusted root certificates are located in the same file.

   After you define the trusted servers, Adaptive Server configures a secure connection, where servername is the name of the current Adaptive Server. If you:

   • Have defined $SYBASE_CERTDIR, Adaptive Server loads certificates from $SYBASE_CERTDIR/servername.txt (for UNIX) or %SYBASE_CERTDIR%servername.txt (for Windows).

   • Have not defined $SYBASE_CERTDIR Adaptive Server loads certificates from $SYBASE/$SYBASE_ASE/certificates/servername.txt (for UNIX) or %SYBASE%$SYBASE_ASE%certificates\servername.txt (for Windows).

2 Restart Adaptive Server to change the trusted root certificate file.

3 Use sp_ladadmin, specifying ldaps:// URLs instead of ldap:// URLs, to establish a secure connection to a secure port of the LDAPS server.

4 Establish a TLS session over a plain TCP connection:
   
   sp_ladadmin 'starttls_on_primary', {true | false}

   or

   sp_ladadmin 'starttls_on_secondary', {true | false}

Note LDAPS connections do not have a connect timeout option; if the LDAP server stops responding, all login connections also stop responding.
Configuring Adaptive Server for authentication using PAM

The pluggable authentication modules (PAM) support allows multiple authentication service modules to be stacked and made available without modifying the applications that require the authentication.

PAM integrates Adaptive Server with Solaris and Linux operating systems and simplifies the management and administration of user accounts and authentication mechanisms, thus reducing the total cost of ownership. An additional benefit is that users can customize or write their own authentication and authorization modules.

Note: PAM support is currently available on Linux and on Solaris platforms. For more information on PAM user authentication, see your operating system documentation.

Figure 16-3: PAM architecture

Adaptive Server passes the login name and credentials obtained from the login packet to the PAM API. PAM loads a service provider module as specified in the operating system configuration files and calls appropriate functions to complete the authentication process.
Enabling PAM in Adaptive Server

Perform the tasks in this section to enable PAM on Adaptive Server.

Determining which PAM module to use

Both Linux and Solaris have predefined PAM modules. You can choose to either use one of these modules or to create one of your own. When creating your own modules, follow the guidelines in your operating system documentation on creating a PAM module.

Note PAM modules you create should comply with RFC 86.0 “Unified Login With Pluggable Authentication Modules (PAM).” Adaptive Server supports the authentication management module of the RFC. It does not support the account management, session management, or password management modules.

Configuring operating system files

To enable PAM support, configure your operating system as follows:

- For Solaris, add the following line to /etc/pam.conf:
  ```
  ase auth required /user/lib/security/$ISA/pam_unix.so.1
  ```
- For Linux, create a new file called /etc/pam.d/ase, and add:
  ```
  auth required /lib/security/pam_unix.so
  ```

For more information on how to create these entries, see your operating system documentation.

Running a 32- and 64-bit server on the same machine

$ISA is an environment variable that stands for Instruction Set Architecture. It allows 32- and 64-bit libraries to run together.

On Solaris 32-bit machines, $ISA is replaced by an empty string, while on 64-bit machines, it is replaced by the string “sparcv9”.

To use both 32- and 64-bit servers, place the 32-bit PAM module in a directory, and to place the 64-bit version in a subdirectory of this directory.

The entry in pam.conf should look similar to:

```
$ ls /usr/lib/security/pam_whatever.so.1
```
Configuring Adaptive Server for authentication using PAM

```
pam_whatever.so.1 ->
/wherever/pam_whatever_32bits.so.1

$ ls /usr/lib/security/sparcv9/pam_whatever.so.1
pam_whatever.so.1 ->
/wherever/pam_whatever_64bits.so.1

ase auth required
/usr/lib/security/$ISA/pam_whatever.so.1
```

**Note** $ISA is the only variable allowed in *pam.conf*.

---

**Configuring Adaptive Server for PAM user authentication**

*enable pam user auth* is a new configuration parameter that enables PAM user authentication support. It can be set as follows:

```
sp_configure "enable pam user auth", 0 | 1 | 2
```

where:

- 0 – disables PAM authentication. This is the default.
- 1 – indicates Adaptive Server will try PAM authentication first, and then fall back to *syslogins* authentication if PAM authentication fails.
- 2 – indicates only PAM authentication may be used.

**Note** When PAM is enabled, password management is delegated to the PAM service providers.
Adaptive Server logins and PAM user accounts

After you have set enable PAM user authentication and completed the PAM configuration for both Adaptive Server and the operating system, you must configure the user accounts. The operating system or network security administrator creates and maintains user accounts in the PAM service provider, and the database administrator creates and maintains accounts in Adaptive Server. Alternatively, the database administrator can choose administration options that allow flexibility with login accounts when integrating Adaptive Server with external authentication mechanisms such as PAM. The database administrator continues to administer the Adaptive Server account roles, default database, default language, and other login-specific attributes using traditional commands and procedures.

Table 16-15 describes updates to syslogins made at login time. It assumes that PAM user authentication is configured, the login is not restricted from using PAM, and you have not set the create login mapping.

<table>
<thead>
<tr>
<th>Table 16-15: Updates to syslogins from PAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Does the row exist in syslogins?</td>
</tr>
<tr>
<td>No</td>
</tr>
<tr>
<td>No</td>
</tr>
<tr>
<td>Yes</td>
</tr>
<tr>
<td>Yes</td>
</tr>
</tbody>
</table>

Enhanced login controls

You can configure Adaptive Server to allow the server-wide authentication mechanism according to the methods discussed in the LDAP and PAM sections earlier. You can also configure Adaptive Server to specify the authentication mechanism for each individual login on the server using its Enhanced login controls.

Login specific controls are useful when a server is transitioning between the authentication mechanisms or for server-specific logins that local server administration may require and are not associated with a centrally managed user login.
Forcing authentication

You can force a login to use a specific authentication process by using these parameters for `sp_modifylogin` and `sp_addlogin`:

- **ASE** – use Adaptive Server internal authentication using passwords from `syslogins` table.
- **LDAP** – use external authentication with an LDAP server.
- **PAM** – use external authentication with PAM.
- **ANY** – by default, users are authenticated using this authentication method. A user with ANY authentication means that Adaptive Server checks if there is any external authentication mechanism defined, and if there is, it is used. Otherwise, it uses Adaptive Server’s authentication.

Adaptive Server checks for external authentication mechanisms in the following order:

1. LDAP.
2. Pluggable Authentication Modules (PAM). If both LDAP and PAM are enabled, PAM authentication is never attempted for a user.
3. If neither PAM nor LDAP is enabled, Adaptive Server uses `syslogins` to authenticate the login.

Login accounts such as “sa” continue to be validated using the `syslogins` catalog. Only the SSO role can set authenticate for a login.

For example, the following authenticates the login with `sp_modifylogin`:

```sql
sp_modifylogin "nightlyjob", "authenticate with", "ASE"
sp_displaylogin "nightlyjob"
```

Displays output similar to:

```
Suid: 1234
Loginname: nightlyjob
Fullname: Batch Login
Default Database: master
. . .
Date of Last Password Change: Oct 2 2003 7:38 PM
Password expiration interval: 0
Password expired: N
Minimum password length: 6
Maximum failed logins: 0
Current failed login attempts:
Authenticate with: ASE
```
Mapping logins using **sp_maplogin**

Use **sp_maplogin** to map logins:

```
sp_maplogin (authentication_mech | null),
(client_username | null), (action | login_name | null)
```

Where:

- `authentication_mech` – is one of the valid values specified for authenticate with option in `sp_modifylogin`.
- `client_username` – is an external user name, which can be an operating system name, a user name for an LDAP server, or anything else the PAM library understands. A null value indicates that any login name is valid.
- `action` – indicates create login or drop. When you use create login the login is created as soon as it is authenticated. Use drop to remove logins.
- `login_name` is an Adaptive Server login that already exists in `syslogins`.

This example maps external user “jsmith” to the Adaptive Server user “guest.” Once authenticated, “jsmith” has the privileges of “guest.” The audit login record shows both the client username and the Adaptive Server user name:

```
sp_maplogin NULL, "jsmith", "guest"
```

This example tells Adaptive Server to create a new login for all external users authenticated with LDAP, if a login does not already exist:

```
sp_maplogin LDAP, NULL, "create login"
```

Displaying mapping information

**sp_helpmaplogin** displays mapping information:

```
sp_helpmaplogin [ (authentication_mech | null), (client_username | null) ]
```

Where `authentication_mech` is one of the valid values specified for authenticate with option in `sp_modifylogin`, and `client_username` is an external user name.

If you do not include any parameters, **sp_helpmaplogin** displays login information about all users currently logged in to Adaptive Server. You can restrict the output to specific sets of client user names or authentication mechanisms by using the parameters listed above.

This displays information about all logins:

```
sp_helpmaplogin
authentication    client name    login name
----------------    ----------    --------------
```
Enhanced login controls

<table>
<thead>
<tr>
<th></th>
<th>NULL</th>
<th>jsmith</th>
<th>guest</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP</td>
<td>NULL</td>
<td>create login</td>
<td></td>
</tr>
</tbody>
</table>

Determining the authentication mechanism

Use the `@@authmech` global variable to determine the authentication mechanism Adaptive Server uses.

For example, if Adaptive Server is enabled for LDAP user authentication with failover (enable ldap user auth = 2) and user “Joe” is an external user with authentication set to ANY, when Joe logs in, Adaptive Server attempts to authenticate Joe, using LDAP user authentication. If Joe fails authentication as a user in LDAP, Adaptive Server authenticates Joe using Adaptive Server authentication, and if that succeeds, he logs in successfully.

`@@authmech` global has this value:

```sql
select @@authmech
-------------------
ase
```

If Adaptive Server is configured for strict LDAP user authentication (enable ldap user auth = 2) and Joe is added as a valid user in LDAP, when Joe logs in, the value for `@@authmech` is:

```sql
select @@authmech
-------------------
ldap
```
Managing User Permissions

This chapter describes the use and implementation of user permissions.

### Overview

Discretionary access controls (DACs) allow you to restrict access to objects and commands based on a user’s identity, group membership and active roles. The controls are “discretionary” because a user with a certain access permission, such as an object owner, can choose whether to pass that access permission on to other users.

Adaptive Server’s discretionary access control system recognizes the following types of users:

- Users possessing one or more system defined roles: system administrator, system security officer, Operator, and other roles
- Database owners
- Database object owners
- Other users
System administrators operate outside the DAC system and have access permissions on all database objects at all times except encryption keys (see *User Guide for Encrypted Columns*). System security officers can always access the audit trail tables in the `syssecurity` database.

Database owners do not automatically receive permissions on objects owned by other users; however, they can:

- Temporarily acquire all permissions of a user in the database by using the `setuser` command to assume the identity of that user.
- Permanently acquire permission on a specific object by using the `setuser` command to assume the identity of the object owner, and then using `grant` commands to grant the permissions.

For details on assuming another user’s identity to acquire permissions on a database or object, see “Acquiring the permissions of another user” on page 590.

Object owners can grant access to those objects to other users and can also grant other users the ability to pass the access permission to other users. You can give various permissions to users, groups, and roles with the `grant` command, and rescind them with the `revoke` command. Use `grant` and `revoke` to give users permission to:

- Create databases
- Create objects within a database
- Execute certain commands such as `dbcc` and `set proxy`
- Access specified tables, views, stored procedures, encryption keys, and columns

`grant` and `revoke` can also be used to set permissions on system tables.

For permissions that default to “public,” no `grant` or `revoke` statements are needed.

Some commands can be used at any time by any user, with no permission required. Others can be used only by users of a particular status and they are not transferable.

The ability to assign permissions for the commands that can be granted and revoked is determined by each user’s role or status (as system administrator, database owner, system security officer, or database object owner), and by whether the user was granted a role with permission that includes the option to grant that permission to other users.
You can also use views and stored procedures as security mechanisms. See “Using views and stored procedures as security mechanisms” on page 600.

Permissions for creating databases

Only a system administrator can grant permission to use the `create database` command. The user that receives `create database` permission must also be a valid user of the `master` database because all databases are created while using `master`.

In many installations, the system administrator maintains a monopoly on `create database` permission to centralize control of database placement and database device space allocation. In these situations, a system administrator creates new databases on behalf of other users, and then transfers ownership to the appropriate user.

To create a database that is to be owned by another user:

1. Issue the `create database` command in the `master` database.
2. Switch to the new database with the `use` command.
3. Execute `sp_changedbowner`.

Changing database ownership

Use `sp_changedbowner` to change the ownership of a database. Often, system administrators create the user databases, then give ownership to another user after some of the initial work is complete. Only the system administrator can execute `sp_changedbowner`.

Sybase suggests that you transfer ownership before the user has been added to the database, and before the user has begun creating objects in the database. The new owner must already have a login name on Adaptive Server, but cannot be a user of the database, or have an alias in the database. You may have to use `sp_dropuser` or `sp_dropalias` before you can change a database’s ownership, and you may have to drop objects before you can drop the user.

Issue `sp_changedbowner` in the database whose ownership is to be changed. The syntax is:

```
sp_changedbowner loginame [, true ]
```
Database owner privileges

This example makes “albert” the owner of the current database and drops aliases of users who could act as the old “dbo:”

```
sp_changedbowner albert
```

Include the `true` parameter to transfer aliases and their permissions to the new “dbo:”

**Note** You cannot change the ownership of the `master`, `model`, `tempdb`, or `sybsystemprocs` databases and should not change the ownership of any other system databases.

Database owner privileges

Database owners and system administrators are the only users who can grant object creation permissions to other users (except for `create encryption key` and `create trigger` permission which can only be granted by the system security officer). The database owner has full privileges to do anything inside that database, and must explicitly grant permissions to other users with the `grant` command.

Permission to use the following commands is automatically granted to the database owner and cannot be transferred to other users:

- `checkpoint`
- `dbcc`
- `alter database`
- `online database`
- `drop database`
- `dump database`
- `dump transaction`
- `grant (object creation permissions)`
- `load database`
- `load transaction`
- `revoke (object creation permissions)`
Database object owner privileges

A user who creates a database object (a table, view, encryption key, or stored procedure) owns the object and is automatically granted all object access permissions on it. Users other than the object owner, including the owner of the database, are automatically denied all permissions on that object, unless they are explicitly granted by either the owner or a user who has grant permission on that object.

As an example, suppose that Mary is the owner of the pubs2 database, and has granted Joe permission to create tables in it. Now Joe creates the table new_authors; he is the owner of this database object.

Initially, object access permissions on new_authors belong only to Joe. Joe can grant or revoke object access permissions for this table to other users.

The following object altering permissions default to the owner of a table and cannot be transferred to other users:

- alter table
- drop table
Other database user privileges

- create index

Permission to use the grant and revoke commands to grant specific users select, insert, update, delete, references, decrypt, truncate table, update statistics, delete statistics, and execute permissions on specific database objects can be transferred, using the grant with grant option command.

Permission to drop an object—a table, view, index, stored procedure, rule, encryption key, trigger, or default—defaults to the object owner and cannot be transferred.

Other database user privileges

At the bottom of the hierarchy are other database users. Permissions are granted to or revoked from them by object owners, database owners, users who were granted permissions, system administrator or a system security officer. These users are specified by user name, group name, or the keyword public.

Permissions on system procedures

Set permissions on system procedures in the sybsystemprocs database, where the system procedures are stored.

Security-related system procedures can be run only by system security officers. Certain other system procedures can be run only by system administrators.

Some of the system procedures can be run only by database owners. These procedures make sure that the user executing the procedure is the owner of the database from which they are being executed.

Other system procedures can be executed by any user who has been granted permission. A user must have permission to execute a system procedure in all databases, or in none of them.
Users who are not listed in sybsystemprocs..sysusers are treated as “guest” in sybsystemprocs, and are automatically granted permission on many of the system procedures. To deny a user permission on a system procedure, the system administrator must add him or her to sybsystemprocs..sysusers and issue a revoke statement that applies to that procedure. The owner of a user database cannot directly control permissions on the system procedures from within his or her own database.

Granting and revoking permissions

You can control the following types of permissions with grant and revoke:

- Object access permissions
- Permission to select from functions
- Permission to execute commands
- Permission to execute dbcc commands
- Permission to execute some set commands
- Default permissions on system tables

Each database has its own independent protection system. Having permission to use a certain command in one database does not give you permission to use that command in other databases.

Object access permissions

Object access permissions regulate the use of certain commands that access certain database objects. For example, you must explicitly be granted permission to use the select command on the authors table. Object access permissions are granted and revoked by the object owner (and system administrators or system security officers), who can grant them to other users.

Table 17-1 lists the types of object access permissions and the objects to which they apply.
Granting and revoking permissions

<table>
<thead>
<tr>
<th>Permission</th>
<th>Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>select</td>
<td>Table, view, column</td>
</tr>
<tr>
<td>update</td>
<td>Table, view, column</td>
</tr>
<tr>
<td>insert</td>
<td>Table, view</td>
</tr>
<tr>
<td>delete</td>
<td>Table, view</td>
</tr>
<tr>
<td>references</td>
<td>Table, column</td>
</tr>
<tr>
<td>execute</td>
<td>Stored procedure</td>
</tr>
<tr>
<td>truncate table</td>
<td>Table</td>
</tr>
<tr>
<td>update statistics</td>
<td>Table</td>
</tr>
<tr>
<td>decrypt</td>
<td>Table, view, column</td>
</tr>
<tr>
<td>select</td>
<td>Encryption key</td>
</tr>
</tbody>
</table>

The references permission refers to referential integrity constraints that you can specify in an alter table or create table command. The decrypt permission refers to the permission required to decrypt an encrypted column. An encryption key’s select permission refers to the permissions required to use encryption keys in create table, alter table or select into command to encrypt columns. The other permissions refer to SQL commands. Object access permissions default to the object’s owner, or system administrators or system security officers for decrypt on an encrypted column and select on an encryption key, and can be granted to other users.

Use the grant command to grant object access permissions. The syntax is:

```sql
grant \{all \[privileges\]|permission_list\}
on \{table_name \{(column_list)\}
| view_name \{(column_list)\}
| stored_procedure_name\}
to \{public | name_list | role_name\}
[with grant option]
```

Use the revoke command to revoke object access permissions. The syntax is:

```sql
revoke \{grant option for\}
\{all \[privileges\]|permission_list\}
on \{table_name \{(column_list)\}
| view_name \{(column_list)\}
| stored_procedure_name\}
from \{public | name_list | role_name\}
[cascade]
```
• all or all privileges specifies all permissions applicable to the specified object, except decrypt permission. All object owners can use all with an object name to grant or revoke permissions on their own objects. If you are granting or revoking permissions on a stored procedure, all is the same as execute.

Note insert, update statistics, delete statistics, truncate table, and delete permissions do not apply to columns, so you cannot include them in a permission list (or use the keyword all) if you specify a column list.

• permission_list is the list of permissions that you are granting. If you name more than one permission, separate them with commas. Table 17-2 illustrates the access permissions that can be granted on each type of object:

Table 17-2: Object access permissions

<table>
<thead>
<tr>
<th>Object</th>
<th>permission_list can include</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table or view</td>
<td>select, insert, delete, update, references, truncate table, update statistics, decrypt, delete statistics references applies to tables but not views; the other permissions apply to both tables and views. update statistics, delete statistics, and truncate table apply to tables on, not views.</td>
</tr>
<tr>
<td>Column</td>
<td>select, update, delete statistics, and truncate table</td>
</tr>
<tr>
<td>Stored procedure</td>
<td>execute</td>
</tr>
<tr>
<td>Encryption key</td>
<td>select</td>
</tr>
</tbody>
</table>

You can specify columns in the permission_list or the column_list, but not both.

• on specifies the object for which the permission is being granted or revoked. You can grant or revoke permissions for only one table, view, encryption key, or stored procedure object at a time. You can grant or revoke permissions for more than one column at a time, but all the columns must be in the same table or view. You can grant or revoke permissions only on objects in your current database.

• public refers to the group “public,” which includes all Adaptive Server users. public means slightly different things for grant and revoke:

  • For grant, public includes the object owner. Therefore, if you have revoked permissions from yourself on your object, and later you grant permissions to public, you regain the permissions along with the rest of “public.”

  • For revoke, public excludes the owner.
Granting and revoking permissions

- **name_list** includes:
  - Group names
  - User names
  - A combination of user and group names, each separated from the next by a comma

- **role_name** is an Adaptive Server system-defined or user-defined role. You can create and define a hierarchy of user-defined roles and grant them privileges based on the specific role granted. System-defined roles include `sa_role` (system administrator), `sso_role` (system security officer), and `oper_role` (Operator). You cannot create or modify system-defined roles.

- with grant option in a grant statement allows the users specified in **name_list** to grant the specified object access permissions to other users. If a user has with grant option permission on an object, that permission is not revoked when permissions on the object are revoked from public or a group of which the user is a member.

- grant option for revokes with grant option permissions, so that the users specified in **name_list** can no longer grant the specified permissions to other users. If those other users have granted permissions to other users, you must use the cascade option to revoke permissions from them as well. The user specified in **name_list** retains permission to access the object, but can no longer grant access to other users. **grant option for** applies only to object access permissions, not to object creation permissions.

- The cascade option in a revoke statement removes the specified object access permissions from the user(s) specified in **name_list**, and also from any users they granted those permissions to.

You may grant and revoke permissions only on objects in the current database.

If several users grant access to an object to a particular user, the user’s access remains until access is revoked by all those who granted access or until a system administrator revokes the access. That is, if a system administrator revokes access, the user is denied access even though other users have granted access.

Only a system security officer can grant or revoke permission to create encryption keys. The database owner can create triggers on any user table. Users can create triggers only on tables that they own.

Permission to issue the `create trigger` command is granted to users by default.
When the system security officer revokes permission for a user to create triggers, a revoke row is added in the `sysprotects` table for that user. To grant permission to that user to issue `create trigger`, issue two `grant` commands: the first command removes the revoke row from `sysprotects`; the second inserts a grant row. The system security officer must grant permission to create triggers. If permission to create triggers is revoked, the user cannot create triggers even on tables that the user owns. Revoking permission to create triggers from a user affects only the database where the `revoke` command was issued.

Concrete identification

Adaptive Server identifies users during a session by login name. This identification applies to all databases in the server. When the user creates an object, the server associates both the owner’s database user ID (`uid`) and the creator’s login name with the object in the `sysobjects` table. This information concretely identifies the object as belonging to that user, which allows the server to recognize when permissions on the object can be granted implicitly.

If an Adaptive Server user creates a table and then creates a procedure that accesses the table, any user who is granted permission to execute the procedure does not need permission to access the object directly. For example, by giving user “mary” permission on `proc1`, she can see the `id` and `descr` columns from `table1`, though she does not have explicit select permission on the table:

```sql
create table table1 (id int,
  amount money,
  descr varchar(100))

create procedure proc1 as select id, descr from table1

grant execute on proc1 to mary
```

There are, however, some cases where implicit permissions are only useful if the objects can be concretely identified. One case is where aliases and cross-database object access are both involved.

Special requirements for SQL92 standard compliance

When you have used the `set` command to turn `ansi_permissions` on, additional permissions are required for `update` and `delete` statements. Table 17-3 summarizes the required permissions.
Granting and revoking permissions

### Table 17-3: ANSI permissions for update and delete

<table>
<thead>
<tr>
<th>Permissions required: set ansi_permissions off</th>
<th>Permissions required: set ansi_permissions on</th>
</tr>
</thead>
<tbody>
<tr>
<td>update update permission on columns where values are being set</td>
<td>update permission on columns where values are being set and select permission on all columns appearing in the where clause</td>
</tr>
<tr>
<td>delete delete permission on the table</td>
<td>delete permission on the table from which rows are being deleted and select permission on all columns appearing in the where clause</td>
</tr>
</tbody>
</table>

If ansi_permissions is on and you attempt to update or delete without having all the additional select permissions, the transaction is rolled back and you receive an error message. If this occurs, the object owner must grant you select permission on all relevant columns.

### Examples of granting object access permissions

This statement gives Mary and the “sales” group permission to insert into and delete from the titles table:

```sql
grant insert, delete on titles to mary, sales
```

This statement gives Harold permission to use the stored procedure makelist:

```sql
grant execute on makelist to harold
```

This statement grants permission to execute the custom stored procedure sa_only_proc to users who have been granted the system administrator role:

```sql
grant execute on sa_only_proc to sa_role
```

This statement gives Aubrey permission to select, update, and delete from the authors table and to grant the same permissions to other users:

```sql
grant select, update, delete on authors to aubrey with grant option
```
Examples of revoking object access permissions

These two statements both revoke permission for all users except the table owner to update the `price` and `total_sales` columns of the `titles` table:

```
revoke update
on titles (price, total_sales)
from public
```

This statement revokes permission from Clare to update the `authors` table, and simultaneously revokes that permission from all users to whom she had granted that permission:

```
revoke update
on authors
from clare
cascade
```

This statement revokes permission from operators to execute the custom stored procedure `new_sproc`:

```
revoke execute
on new_sproc
from oper_role
```

Granting and revoking permissions for `update statistics`, `delete statistics`, and `truncate table`

Adaptive Server allows you to grant and revoke permissions for users, roles, and groups for the `update statistics`, `delete statistics`, and `truncate table` commands. Table owners can also provide permissions through an implicit grant by adding `update statistics`, `delete statistics`, and `truncate table` to a stored procedure and then granting execute permissions on that procedure to a user or role.

You cannot grant or revoke permissions for `update statistics` at the column level. You must have the `sso_role` to run `update statistics` or `delete statistics` on `sysroles`, `syssrvroles`, and `sysloginroles` security tables.

By default, users with the `sa_role` have permission to run `update statistics` and `delete statistics` on system tables other than `sysroles`, `syssrvroles` and `sysloginroles`, and can transfer this privilege to other users.

The partial syntax for `grant` and `revoke` is:

```
grant [truncate table | update statistics | delete statistics] on table_name
to {user_name | role_name | group_name}
```
Granting and revoking permissions

revoke [truncate table | update statistics | delete statistics] on
table_name from (user_name | role_name | group_name)

You can also issue grant all to grant permissions on update statistics, delete
statistics, and truncate table.

For example, the following allows user “harry” to use truncate table and
updates statistics on the authors table:

grant truncate table on authors to harry
grant update statistics on authors to harry

The following revokes truncate table and update statistics privileges from
“harry” on the authors table:

revoke truncate table on authors from harry
revoke update statistics on authors from harry

The following allows user “billy” to use the delete statistics command on the
authors table:

grant delete statistics on authors to billy

The following revokes the delete statistics privileges from user “billy” on the
authors table:

revoke delete statistics on authors from billy

The following grants truncate table and update and delete statistics privileges
to all users with the oper_role (if users “billy” and “harry” possess the oper_role,
they can now run these commands on authors):

grant truncate table on authors to oper_role
grant update statistics on authors to oper_role
grant delete statistics on authors to oper_role

The following revokes truncate table and update and delete statistics privileges
from all users with the oper_role:

revoke truncate table on authors from oper_role
revoke update statistics on authors from oper_role
revoke delete statistics on authors from oper_role

Users “billy” and “harry” can no longer run these commands on authors.

You can also implicitly grant permissions for truncate table, delete statistics,
and update statistics through a stored procedure. For example, assuming “billy”
owns the authors table, he can run the following to grant “harry” privileges to
run truncate table and update statistics on authors:

create procedure sprocl
as
You can also implicitly grant permissions at the column level for update statistics and delete statistics through stored procedures.

**Note** Once you grant permission to execute `update statistics` to a user, they also have permission to execute variations of this command, such as `update all statistics`, `update partition statistics`, `update index statistics`, `update statistics table`, and so on. For example, the following grants “billy” permission to run all variations of `update statistics` on the authors table:

```sql
grant update statistics on authors to billy
```

If you revoke a user’s permission to execute `update statistics`, you also revoke their ability to execute the variations of this command.

You cannot grant variants of `update statistics` (for example, `update index statistics`) separately. That is, you cannot issue:

```sql
grant update all statistics to harry
```

However, you can write stored procedures that control who executes these commands. For example, the following grants “billy” execute permission for `update index statistics` on the authors table:

```sql
create proc sp_ups as
update index statistics on authors
go
revoke update statistics on authors from billy
go
grant execute on sp_ups to billy
```

You cannot grant and revoke `delete statistics` permissions at the column level.

Although Adaptive Server audits `truncate table` as a global, miscellaneous audit, it does not audit `update statistics`. To retain clear audit trails for both `truncate table` and `update statistics`, Sybase recommends that you include both commands in a stored procedure to which you grant users execute permission, as described above.

The command fails and generates error number 10330 if a user issues `update statistics`, `delete statistics` or `truncate table` and they:

- Do not own the table.
Granting and revoking permissions

- Do not have the sa_role.
- Are not a database owner who has successfully used setuser to become the user who is the owner of the table.
- Have not been granted update statistics, delete statistics, or truncate table privileges.

Granting permissions on functions

Use grant select on builtin function_name to grant a user permission to use the functions set_appcontext, get_appcontext, list_appcontext, and rm_appcontext.

The syntax is:

```
grant select on [builtin] function_name
to {name_list | role_list}
```

Where:

- **builtin** – Used to distinguish between a table and a grantable function with the same name.
- **function_name** – Name of the function for which you are granting permission. Functions for which select permission can be granted are set_appcontext, get_appcontext, list_appcontext, and rm_appcontext.
- **name_list** – List of users’ database names and group names.
- **role_list** – List of the names of system or user-defined roles to which permission is being granted, and cannot be a variable.

This grants select permission on the get_appcontext function to public:

```
grant select on builtin get_appcontext to public
```

Granting and revoking permissions to execute commands

This section describes how to grant and revoke permissions for users to execute specific commands.
Granting permissions to execute commands

Object creation permissions regulate the use of commands that create objects. Other than commands for creating objects, other commands like `connect` and `set session authorization` can be granted. These permissions can be granted only by a system administrator or a database owner (unless otherwise noted).

The commands are:

- `connect`
- `create database`
- `create default`
- `create procedure`
- `create rule`
- `create table`
- `create view`
- `set session authorization`
- `create encryption key` (only grantable by system security officer)
- `create trigger` (only grantable by system security officer)

The syntax for command permissions differs slightly from the syntax for object access permissions. The syntax for `grant` is:

```
grant {all [privileges] | command_list} to {public | name_list | role_name}
```

The syntax for `revoke` is:

```
revoke {all [privileges] | command_list} from {public | name_list | role_name}
```

where:

- `all` or `all privileges` – can be used only by a system administrator or the database owner. When used by a system administrator in the master database, `grant all assigns all create permissions, including create database (except create encryption key and create trigger). If the system administrator executes `grant all from another database, all create permissions are granted except create database, create trigger and create encryption key. When the database owner uses `grant all, Adaptive Server grants all create permissions except create database, create trigger, and create encryption key, and prints an informational message.`
Granting and revoking permissions

- command_list – is the object creation and other command permissions that you are granting or revoking. Separate commands with commas. The list can include create database, create default, create procedure, create rule, create table, connect, create encryption key, set session authorization, create view, and create trigger. create database permission can be granted only by a system administrator, and only from within the master database. You must have system security officer privileges to grant create encryption key, set session authorization, and create trigger permissions.

- public – is all users except the database owner (who “owns” object creation permissions within the database).

- name_list – is a list of user or group names, separated by commas.

- role_name – is the name of an Adaptive Server system or user-defined role. You can create and define a hierarchy of user-defined roles and grant them privileges based on the specific role granted.

Granting command permission examples

The first example grants Mary and John permission to use create database and create table. Because create database permission is being granted, this command can be executed only by a system administrator within the master database. Mary and John’s create table permission applies only to the master database.

grant create table, create database
to mary, john

This command grants permission to create tables and views in the current database to all users:

grant create table, create view
to public

Revoking command permission example

This example revokes permission to create tables and rules from “mary:”

revoke create table, create rule
from mary
Granting proxy authorization

System security officers use the `grant set proxy` or `grant set session authorization` command to give a user permission to impersonate another user within the server. The user with this permission can then execute either `set proxy` or `set session authorization` to become another user.

To grant proxy authorization permission, you must be a system security officer and execute the `grant` command from the `master` database. The syntax is:

```
grant set proxy to user | role
                             [restricted role role_list | all | system]
```

where:

- `role_list` – list of roles you are restricting for the target login. If the grantees do not yet have the roles in the `role_list` granted to them, `set proxy` to the target login fails if the target login contains roles in the `role_list` granted.
- `all` – when used to `grant set proxy to role_list`, restricts granting the grantee any new roles when switching identities.
- `system` – ensures the grantee has the same set of system roles as the target login.

**Example 1**

Example 1: This example grants `set proxy` to user “joe” but restricts him from switching identities to any user with the `sa_role`, `sso_role`, or `admin_role` roles (however, if he already has these roles, he can `set proxy` for any user with these roles):

```
grant set proxy to joe
       restricted role sa_role, sso_role, admin_role
```

When “joe” tries to switch his identity to a user with `admin_role` (in this example, `Our_admin_role`), the command fails unless he already has `admin_role`:

```
set proxy Our_admin_role
Msg 10368, Level 14, State 1:
Server 's', Line 2:Set session authorization permission denied because the target login has a role that you do not have and you have been restricted from using.
```

After “joe” is granted the `admin_role` and retries the command, it succeeds:

```
grant role admin_role to joe
set proxy Our_admin_role
```

**Example 2**

Example 2: Restricts “joe” from being granted any new roles when switching identities:
**Granting and revoking permissions**

```
grant set proxy to joe  
    restricted role all
```

“joe” can grant `set proxy` only to users who have the same (or a subset of) roles that he has.

**Example 3**

Example 3: Restricts Joe from acquiring any new system roles when using `set proxy`:

```
grant set proxy to joe  
    restricted role system
```

`set proxy` fails if the target login has system roles that Joe lacks.

**Granting permissions on dbcc commands**

System administrators can grant the permission to execute `dbcc` commands to users and roles that do not have system administrator-level privileges in Adaptive Server. This **discretionary access control** allows system administrators to control access to database objects or to certain database- and server-level actions.

See the *Adaptive Server Reference Manual: Commands* for the complete `dbcc` syntax.

**Server-wide and database-specific dbcc commands**

`dbcc` commands are either:

- **Database-specific** – `dbcc` commands that execute on a particular target database (for example, `checkalloc`, `checktable`, `checkindex`, `checkstorage`, `checkdb`, `checkcatalog`, `checkverify`, `fix_text`, `indexalloc`, `reindex`, `tablealloc`, and `textalloc`). Although these commands are database-specific, only system administrators can grant or revoke them.

- **Server-wide** – `dbcc` commands such as `tune` that are effective server-wide and are not associated with any particular database. These commands are granted server-wide by default and are not associated with any database.

System administrators can allow users to execute the `dbcc` command in all databases by making them valid users in those databases. However, it may be more convenient to grant `dbcc` to roles instead of individual users, since this allows users to use databases as a “guest” user instead of requiring that they each be added manually to the database.
From a security administration perspective, system administrators may prefer to grant permission to execute database-specific dbcc commands server-wide. For example, you can execute `grant dbcc checkstorage` on all databases to a user-defined role called `storage_admin_role`, thereby eliminating the need to execute `grant dbcc checkstorage` to `storage_admin_role` in every database.

The following commands are effective server-wide, but are not database-specific:

- Server-wide dbcc commands such as `tune`.
- Database-specific dbcc commands that are granted server-wide, such as `grant dbcc checkstorage` granted to `storage_admin_role`.

### dbcc grantees and users in databases

grant dbcc and revoke dbcc work on users in databases.

Since roles are automatically added as users in a database on their first grant in a database, there are no additional requirements when roles are granted dbcc privileges. Logins must be valid users in the database where permissions are granted. Valid users include “guest.”

For server-wide dbcc commands, the login must be a valid user in `master`, and the system administrator must be in `master` when granting the permission.

For database-specific dbcc commands the login should be a valid user in the target database.

### Permissions on system tables

Permissions for use of the system tables can be controlled by the database owner, just like permissions on any other tables. When a database is created, `select` permission on some system tables is granted to `public`, and `select` permission on some system tables is restricted to administrators. For some other tables, a few columns have restricted `select` permissions for `public`.

To determine the current permissions for a particular system table, execute:

```sql
sp_helpprotect system_table_name
```

For example, to check the permissions of `syssrvroles` in the master database, execute:

```sql
use master
go
```
Granting and revoking permissions

```
sp_helprotect syssrvroles
go
```

The default situation is that no users—including database owners—can modify the system tables directly. Instead, the T-SQL commands and the system procedures supplied with Adaptive Server modify the system tables. This helps guarantee integrity.

Warning! Although Adaptive Server provides a mechanism that allows you to modify system tables, Sybase strongly recommends that you do not do so.

Granting default permissions to system tables and stored procedures

The `grant` and `revoke` commands include the `default permissions` parameter. `installmodel` or `installmaster` do not grant default permissions on any system tables (see the table below). Instead, the default permissions on the system tables are assigned when Adaptive Server builds a new database. The partial syntax is:

```
grant default permissions on system tables
revoke default permissions on system tables
```

where `default permissions on system tables` specifies that you grant or revoke the default permissions for the following system tables when you issue it from any database:

<table>
<thead>
<tr>
<th>Tables</th>
<th>Tables</th>
<th>Tables</th>
<th>Tables</th>
</tr>
</thead>
<tbody>
<tr>
<td>sysattributes</td>
<td>syskeys</td>
<td>sysreferences</td>
<td>sysusermessages</td>
</tr>
<tr>
<td>syscolumns</td>
<td>syslogs</td>
<td>sysroles</td>
<td>sysusers</td>
</tr>
<tr>
<td>syscomments</td>
<td>sysobjects</td>
<td>syssegments</td>
<td>sysxtypes</td>
</tr>
<tr>
<td>sysconstraints</td>
<td>syspartitions</td>
<td>syscallstatistics</td>
<td></td>
</tr>
<tr>
<td>sysdepends</td>
<td>sysprocedures</td>
<td>systabstats</td>
<td></td>
</tr>
<tr>
<td>sysindexes</td>
<td>sysprotects</td>
<td>systhresholds</td>
<td></td>
</tr>
</tbody>
</table>

`default permissions on system tables` also makes the following changes:

- Revokes `select` on `syscolumns(encrkeyid)` from public
- Revokes `select` on `syscolumns(encrkeydb)` from public
- Grants `select` on `syscolumns` to `sso_role`
- Revokes `sysobjects(audflags)` permissions from public
- Grants permissions for `sysobjects` to `sso_role`
CHAPTER 17  Managing User Permissions

- Revokes select on all columns of sysencryptkeys from public
- Grants select on all columns of sysencryptkeys to sso_role

If you run this command from the master database, default permissions for the following system tables are granted or revoked:

<table>
<thead>
<tr>
<th>syscharsets</th>
<th>syslanguages</th>
<th>sysremotelogins</th>
<th>systransactions</th>
</tr>
</thead>
<tbody>
<tr>
<td>sysconfigures</td>
<td>syslocks</td>
<td>sysresourcelimits</td>
<td>sysusages</td>
</tr>
<tr>
<td>syscurconfigs</td>
<td>syslogins</td>
<td>sysservers</td>
<td></td>
</tr>
<tr>
<td>sysdatabases</td>
<td>sysmessages</td>
<td>syssessions</td>
<td></td>
</tr>
<tr>
<td>sysdevices</td>
<td>sysprocesses</td>
<td>systimeranges</td>
<td></td>
</tr>
</tbody>
</table>

The command also makes the following changes:
- Revokes select on sysdatabases(audflags) from public
- Revokes select on syscolumns(encrkeyid) from public
- Revokes select on syscolumns(encrkeydb) from public
- Grants select on syscolumns to sso_role
- Revokes select on sysdatabases(deftabaud) from public
- Revokes select on sysdatabases(defvwaud) from public
- Revokes select on sysdatabases(defpraud) from public
- Revokes select on sysdatabases(audflags2) from public
- Grants select on sysdatabases to sso_role.
- Revokes select on syslogins(password) to public
- Revokes select on syslogins(audflags) from public
- Grants select on syslogins to sso_role
- Revokes select on syslisteners(net_type) from public
- Revokes select on syslisteners(address_info) from public
- grant select on syslisteners to sso_role
- Revokes select on syssrvroles(srid) from public
- Revokes select on syssrvroles(name) from public
- Revokes select on syssrvroles(password) from public
- Revokes select on syssrvroles(pwdate) from public
Granting and revoking permissions

- Revokes `select on syssrvroles(status)` from public
- Revokes `select on syssrvroles(logincount)` from public
- `grant select on syssrvroles to sso_role`
- Revokes `select on sysloginroles(suid)` from public
- Revokes `select on sysloginroles(srid)` from public
- Revokes `select on sysloginroles(status)` from public
- Revokes `select on sysloginroles` to `sso_role`

Combining grant and revoke statements

You can assign specific permissions to specific users, or, if most users are going to be granted most privileges, it may be easier to assign all permissions to all users, and then revoke specific permissions from specific users.

For example, a database owner can grant all permissions on the `titles` table to all users by issuing:

```
grant all
   on titles
   to public
```

The database owner can then issue a series of `revoke` statements, for example:

```
revoke update
   on titles (price, advance)
   from public
revoke delete
   on titles
   from mary, sales, john
```

Grant and revoke statements are order-sensitive: in case of a conflict, the most recently issued statement supersedes all others.

**Note** Under SQL rules, you must use the `grant` command before using the `revoke` command, but the two commands cannot be used within the same transaction. Therefore, when you grant “public” access to objects, and then revoke that access from an individual, there is a short period of time during which the individual has access to the objects in question. To prevent this situation, use the `create schema` command to include the `grant` and `revoke` clauses within one transaction.
Understanding permission order and hierarchy

grant and revoke statements are sensitive to the order in which they are issued. For example, if Jose’s group has been granted select permission on the titles table and then Jose’s permission to select the advance column has been revoked, Jose can select all the columns except advance, while the other users in his group can still select all the columns.

A grant or revoke statement that applies to a group or role changes any conflicting permissions that have been assigned to any member of that group or role. For example, if the owner of the titles table has granted different permissions to various members of the sales group, and wants to standardize, he or she might issue the following statements:

```sql
revoke all on titles from sales
grant select on titles(title, title_id, type, pub_id) to sales
```

Similarly, a grant or revoke statement issued to public changes, for all users, all previously issued permissions that conflict with the new regime.

The same grant and revoke statements issued in different orders can create entirely different situations. For example, the following set of statements leaves Jose, who belongs to the public group, without any select permission on titles:

```sql
grant select on titles(title_id, title) to jose
revoke select on titles from public
```

In contrast, the same statements issued in the opposite order result in only Jose having select permission and only on the title_id and title columns:

```sql
revoke select on titles from public
grant select on titles(title_id, title) to jose
```

When you use the keyword public with grant, you are including yourself. With revoke on object creation permissions, you are included in public unless you are the database owner. With revoke on object access permissions, you are included in public unless you are the object owner. You may want to deny yourself permission to use your own table, while giving yourself permission to access a view built on it. To do this, you must issue grant and revoke statements explicitly setting your permissions. You can reinstitute the permission with a grant statement.
Grant dbcc and set proxy issue warning for fipsflagger

Grant dbcc and set proxy issue the following warning when they are issued while set fipsflagger option is enabled:

SQL statement on line number 1 contains Non-ANSI text.
The error is caused due to the use of DBCC.

Granting and revoking roles

After a role is defined, it can be granted to any login account or role in the server, provided that it does not violate the rules of mutual exclusivity and hierarchy. Table 17-4 lists the tasks related to roles, the role required to perform the task, and the command to use.

<table>
<thead>
<tr>
<th>Task</th>
<th>Required role</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grant the sa_role role</td>
<td>System administrator</td>
<td>grant role</td>
</tr>
<tr>
<td>Grant the sso_role role</td>
<td>System security officer</td>
<td>grant role</td>
</tr>
<tr>
<td>Grant the oper_role role</td>
<td>System security officer</td>
<td>grant role</td>
</tr>
<tr>
<td>Grant user-defined roles</td>
<td>System security officer</td>
<td>grant role</td>
</tr>
<tr>
<td>Create role hierarchies</td>
<td>System security officer</td>
<td>grant role</td>
</tr>
<tr>
<td>Modify role hierarchies</td>
<td>System security officer</td>
<td>revoke role</td>
</tr>
<tr>
<td>Revoke system roles</td>
<td>System security officer</td>
<td>revoke role</td>
</tr>
<tr>
<td>Revoke user-defined roles</td>
<td>System security officer</td>
<td>revoke role</td>
</tr>
</tbody>
</table>

Granting roles

To grant roles to users or other roles, use:

```
grant role role_granted [, role_granted]...
to grantee [, grantee]...
```

where:

- `role_granted` – is the role being granted. You can specify any number of roles to be granted.
- `grantee` – is the name of the user or role. You can specify any number of grantees.
All roles listed in the `grant` statement are granted to all grantees. If you grant one role to another, it creates a role hierarchy.

For example, to grant Susan, Mary, and John the “financial_analyst” and the “payroll_specialist” roles, enter:

```
grant role financial_analyst, payroll_specialist
to susan, mary, john
```

**Understanding `grant` and roles**

You can use the `grant` command to grant permission on objects to all users who have been granted a specified role, whether system or user-defined. This allows you to restrict use of an object to users who have been granted any of these roles:

- Any system-defined role
- Any user-defined role

A role can be granted only to a login account or another role. However, `grant` permission does not prevent users who do not have the specified role from being granted execute permission on a stored procedure. To ensure, for example, that only system administrators can successfully execute a stored procedure, use the `proc_role` system function within the stored procedure itself. See “Displaying information about roles” on page 433 for more information.

Permissions granted to roles override permissions granted to users or groups. For example, assume John has been granted the system security officer role, and `sso_role` has been granted permission on the `sales` table. If John’s individual permission on `sales` is revoked, he can still access `sales` when he has `sso_role` active because his role permissions override his individual permissions.

In granting permissions, a system administrator is treated as the object owner. If a system administrator grants permission on another user’s object, the owner’s name appears as the grantor in `sysprotects` and in `sp_helprotect` output.

If several users grant access to an object to a particular user, the user’s access remains until access is revoked by all those who granted access. If a system administrator revokes access, the user is denied access, even though other users have granted access.
Revoking roles

Use `revoke role` to revoke roles from users and other roles:

```
revoke role role_name [{, role_name}] from grantee [{, grantee}]
```

where:

- `role_name` – is the role being revoked. You can specify any number of roles to be revoked.
- `grantee` – is the name of the user or role. You can specify any number of grantees.

All roles listed in the `revoke` statement are revoked from all grantees.

You cannot revoke a role from a user while the user is logged in.

Acquiring the permissions of another user

Adaptive Server provides two ways to acquire another user’s identity and permissions status:

- A database owner can use the `setuser` command to “impersonate” another user’s identity and permissions status in the current database. See “Using setuser” on page 590.
- `proxy authorization` allows one user to assume the identity of another user on a server-wide basis. See “Using proxy authorization” on page 591.

Using setuser

A database owner may use `setuser` to:

- Access an object owned by another user
- Grant permissions on an object owned by another user
- Create an object that will be owned by another user
- Temporarily assume the DAC permissions of another user for some other reason
While the `setuser` command enables the database owner to automatically acquire another user’s DAC permissions, the command does not affect the roles that have been granted.

`setuser` permission defaults to the database owner and cannot be transferred. The user being impersonated must be an authorized user of the database. Adaptive Server checks the permissions of the user being impersonated.

System administrators can use `setuser` to create objects that will be owned by another user. However, system administrators operate outside the DAC permissions system; therefore, they need not use `setuser` to acquire another user’s permissions. The `setuser` command remains in effect until another `setuser` command is given, the current database is changed, or the user logs off.

The syntax is:

```
setuser ["user_name"]
```

where `user_name` is a valid user in the database that is to be impersonated.

To reestablish your original identity, use `setuser` with no value for `user_name`.

This example shows how the database owner would grant Joe permission to read the `authors` table, which is owned by Mary:

```
setuser "mary"

grant select on authors to joe

setuser /*reestablishes original identity*/
```

### Using proxy authorization

With the proxy authorization capability of Adaptive Server, system security officers can grant selected logins the ability to assume the security context of another user, and an application can perform tasks in a controlled manner on behalf of different users. If a login has permission to use proxy authorization, the login can impersonate any other login in Adaptive Server.

**Warning!** The ability to assume another user’s identity is extremely powerful and should be limited to trusted administrators and applications. `grant set proxy ... restrict role` can be used to restrict which roles users cannot acquire when switching identities.
**Acquiring the permissions of another user**

A user executing `set proxy` or `set session authorization` operates with both the login name and server user ID of the user being impersonated. The login name is stored in the `name` column of `master..syslogins` and the server user ID is stored in the `suid` column of `master..syslogins`. These values are active across the entire server in all databases.

**Note** `set proxy` and `set session authorization` are identical in function and can be used interchangeably. The only difference between them is that `set session authorization` is ANSI-SQL92-compatible, and `set proxy` is a Transact-SQL extension.

**Using set proxy to restrict roles**

You can grant `set proxy...restrict role`, which allows you to restrict which roles cannot be acquired when switching identities.

The syntax for `set proxy` is:

```
grant set proxy to user | role
    [restrict role role_list | all | system]
```

where:

- `role_list` – list of roles you are restricting for the target login. The grantee must have all roles on this list, or the `set proxy` command fails.
- `all` – ensures the grantee can run `set proxy` only for those users who have the same roles, or a subset of the roles, as the grantee.
- `system` – ensures the grantee has the same set of system roles as the target login.

For example, this grants `set proxy` to user “joe” but restricts him from switching identities to any user with the `sa`, `sso`, or `admin` roles (however, if he already has these roles, he can `set proxy` for any user with these roles):

```
grant set proxy to joe
    restrict role sa_role, sso_role, admin_role
```

When “joe” tries to switch his identity to a user with `admin` role (in this example, `Our_admin_role`), the command fails unless he already has `admin` role:

```
set proxy Our_admin_role
Msg 10368, Level 14, State 1:
Server 's', Line 2: Set session authorization permission denied because the target login has a role that you do
not have and you have been restricted from using.

After “joe” is granted the admin_role and retries the command, it succeeds:

grant role admin_role to joe
set proxy Our_admin_role

For more information about the set proxy command, see the Reference Manual: Commands.

Executing proxy authorization

Follow these rules when you execute set proxy or set session authorization:

• You cannot execute set proxy or set session authorization from within a transaction.

• You cannot use a locked login for the proxy of another user. For example, if “joseph” is a locked login, the following command is not allowed:

set proxy "joseph"

• You can execute set proxy or set session authorization from any database you are allowed to use. However, the login_name you specify must be a valid user in the database, or the database must have a “guest” user defined for it.

• Only one level is permitted; to impersonate more than one user, you must return to your original identity between impersonations.

• If you execute set proxy or set session authorization from within a procedure, your original identity is automatically resumed when you exit the procedure.

If you have a login that has been granted permission to use set proxy or set session authorization, you can set proxy to impersonate another user. The following is the syntax, where login_name is the name of a valid login in master.syslogins:

set proxy login_name

or

set session authorization login_name

Enclose the login name in quotation marks.

For example, to set proxy to “mary,” execute:

set proxy "mary"
After setting proxy, check your login name in the server and your user name in the database. For example, assume that your login is “ralph” and that you have been granted set proxy authorization. You want to execute some commands as “sallyn” and as “rudolph” in pubs2 database. “sallyn” has a valid name (“sally”) in the database, but Ralph and Rudolph do not. However, pubs2 has a “guest” user defined. You can execute:

```sql
set proxy "sallyn"
go
use pubs2
go
select suser_name(), user_name()
go
-------------------------------------------
sallyn  sally
```

To change to Rudolph, you must first change back to your own identity. To do so, execute:

```sql
set proxy "ralph"
select suser_name(), user_name()
go
-------------------------------------------
ralph  guest
```

Notice that Ralph is a “guest” in the database.

Then execute:

```sql
set proxy "rudolph"
go
select suser_name(), user_name()
go
-------------------------------------------
rudolph  guest
```

Rudolph is also a guest in the database because Rudolph is not a valid user in the database.

Now, impersonate the “sa” account. Execute:

```sql
set proxy "ralph"
go
set proxy "sa"
go
select suser_name(), user_name()
go
-------------------------------------------
sa  dbo
```
**Proxy authorization for applications**

Figure 17-1 shows an application server logging in to Adaptive Server with the generic login “appl” to execute procedures and commands for several users. While “appl” impersonates Tom, the application has Tom’s permissions. Likewise, when “appl” impersonates Sue and John, the application has only Sue’s and John’s permissions, respectively.

*Figure 17-1: Applications and proxy authorization*

Tom, Sue, and John establish sessions with the Application Server:

<table>
<thead>
<tr>
<th>Tom</th>
<th>Sue</th>
<th>John</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Server logs in as “appl” with set proxy permission.</td>
<td>Adaptive Server executes:</td>
<td></td>
</tr>
<tr>
<td>set proxy &quot;tom&quot; (SQL command for Tom)</td>
<td>set proxy &quot;sue&quot; (SQL command for Sue)</td>
<td>set proxy &quot;John&quot; (SQL command for John)</td>
</tr>
</tbody>
</table>

**Reporting on permissions**

Table 17-5 lists the system procedures for reporting information about proxies, object creation, and object access permissions:
Reporting on permissions

<table>
<thead>
<tr>
<th>Table 17-5: System procedures for reporting on permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>To report information on</td>
</tr>
<tr>
<td>--------------------------</td>
</tr>
<tr>
<td>Proxies</td>
</tr>
<tr>
<td>Users and processes</td>
</tr>
<tr>
<td>Permissions on database objects or users</td>
</tr>
<tr>
<td>Permissions on specific tables</td>
</tr>
<tr>
<td>Permissions on specific columns in a table</td>
</tr>
</tbody>
</table>

Querying the sysprotects table for proxy authorization

To display information about permissions that have been granted to—or revoked from—users, groups, and roles, query the sysprotects table. The action column specifies the permission. For example, the action value for set proxy or set session authorization is equal to 167.

You might execute this query:

```sql
select * from sysprotects where action = 167
```

The results provide the user ID of the user who granted or revoked the permission (column grantor), the user ID of the user who has the permission (column uid), and the type of protection (column protecttype). The protecttype column can contain these values:

- 0 for grant with grant
- 1 for grant
- 2 for revoke

For more information about the sysprotects table, see the Reference Manual.

Displaying information about users and processes

`sp_who` displays information about all current Adaptive Server users and processes or about a particular user or process. The results of `sp_who` include the loginame and origname. If a user is operating under a proxy, origname contains the name of the original login. For example, assume that “ralph” executes the following, then executes some SQL commands:

```sql
set proxy susie
sp_who
```

`sp_who` returns “susie” for loginame and “ralph” for origname.
sp_who queries the master.sysprocesses system table, which contains columns for the server user ID (suid) and the original server user ID (oridsuid).

For more information, see sp_who in the Reference Manual.

**Reporting permissions on database objects or users**

Use sp_helprotect to report on permissions by database object or by user, and (optionally) by user for a specified object. Any user can execute this procedure. The syntax is:

```
sp_helprotect [name[, username [, "grant" [, "none"|"granted"|"enabled"|role_name]]]]
```

where:

- `name` – is either the name of the table, view, or stored procedure, or the name of a user, group, or role in the current database. If you do not provide a name, sp_helprotect reports on all permissions in the database.
- `username` – is a user’s name in the current database.

If you specify `username`, only that user’s permissions on the specified object are reported. If `name` is not an object, sp_helprotect checks whether `name` is a user, group, or role and if it is, lists the permissions for the user, group, or role. If you specify the keyword `grant`, and `name` is not an object, sp_helprotect displays all permissions granted by with grant option.

- `grant` – displays the permissions granted to `name` with grant option.
- `none` – ignores roles granted to the user.
- `granted` – includes information on all roles granted to the user.
- `enabled` – includes information on all roles activated by the user.
- `role_name` – displays permission information for the specified role only, regardless of whether this role has been granted to the user.

For example, suppose you issue the following series of `grant` and `revoke` statements:

```
grant select on titles to judy
grant update on titles to judy
revoke update on titles(contract) from judy
grant select on publishers to judy
    with grant option
```
To determine the permissions Judy now has on each column in the `titles` table, enter:

```
sp_helprotect titles, judy
```

<table>
<thead>
<tr>
<th>grantor</th>
<th>grantee</th>
<th>type</th>
<th>action</th>
<th>object</th>
<th>column</th>
<th>grantable</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbo</td>
<td>judy</td>
<td>Grant</td>
<td>Select</td>
<td>titles</td>
<td>All</td>
<td>FALSE</td>
</tr>
<tr>
<td>dbo</td>
<td>judy</td>
<td>Grant</td>
<td>Update</td>
<td>titles</td>
<td>advance</td>
<td>FALSE</td>
</tr>
<tr>
<td>dbo</td>
<td>judy</td>
<td>Grant</td>
<td>Update</td>
<td>titles</td>
<td>notes</td>
<td>FALSE</td>
</tr>
<tr>
<td>dbo</td>
<td>judy</td>
<td>Grant</td>
<td>Update</td>
<td>titles</td>
<td>price</td>
<td>FALSE</td>
</tr>
<tr>
<td>dbo</td>
<td>judy</td>
<td>Grant</td>
<td>Update</td>
<td>titles</td>
<td>pub_id</td>
<td>FALSE</td>
</tr>
<tr>
<td>dbo</td>
<td>judy</td>
<td>Grant</td>
<td>Update</td>
<td>titles</td>
<td>pubdate</td>
<td>FALSE</td>
</tr>
<tr>
<td>dbo</td>
<td>judy</td>
<td>Grant</td>
<td>Update</td>
<td>titles</td>
<td>title</td>
<td>FALSE</td>
</tr>
<tr>
<td>dbo</td>
<td>judy</td>
<td>Grant</td>
<td>Update</td>
<td>titles</td>
<td>title_id</td>
<td>FALSE</td>
</tr>
<tr>
<td>dbo</td>
<td>judy</td>
<td>Grant</td>
<td>Update</td>
<td>titles</td>
<td>total_sales</td>
<td>FALSE</td>
</tr>
<tr>
<td>dbo</td>
<td>judy</td>
<td>Grant</td>
<td>Update</td>
<td>titles</td>
<td>type</td>
<td>FALSE</td>
</tr>
</tbody>
</table>

The first row shows that the database owner (“dbo”) gave Judy permission to select all columns of the `titles` table. The rest of the lines indicate that she can update only the columns listed in the display. Judy cannot give select or update permissions to any other user.

To see Judy’s permissions on the `publishers` table, enter:

```
sp_helprotect publishers, judy
```

In this display, the `grantable` column indicates TRUE, meaning that Judy can grant the permission to other users.

<table>
<thead>
<tr>
<th>grantor</th>
<th>grantee</th>
<th>type</th>
<th>action</th>
<th>object</th>
<th>column</th>
<th>grantable</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbo</td>
<td>judy</td>
<td>Grant</td>
<td>Select</td>
<td>publishers</td>
<td>all</td>
<td>TRUE</td>
</tr>
</tbody>
</table>

**Reporting permissions on specific tables**

Use `sp_table_privileges` to return permissions information about a specified table. The syntax is:

```
sp_table_privileges table_name [ , table_owner
[ , table_qualifier]]
```

where:

- `table_name` – is the name of the table, and is required.
• *table_owner* – can be used to specify the name of the table owner, if it is not “dbo” or the user executing sp_table_privileges.
• *table_qualifier* – is the name of the current database.

Use null for parameters that you want to skip.

For example, this statement returns information about all permissions granted on the *titles* table:

```
sp_table_privileges titles
```

For more information about the output of sp_table_privileges, see the Reference Manual.

### Reporting permissions on specific columns

Use sp_column_privileges to return information about permissions on columns in a table. The syntax is:

```
sp_column_privileges table_name [, table_owner
[ , table_qualifier [, column_name]]]
```

where:

• *table_name* – is the name of the table.
• *table_owner* – can be used to specify the name of the table owner, if it is not “dbo” or the user executing sp_column_privileges.
• *table_qualifier* – is the name of the current database.
• *column_name* – is the name of the column on which you want to see permissions information.

Use null for parameters that you want to skip.

For example, this statement returns information about the *pub_id* column of the *publishers* table:

```
sp_column_privileges publishers, null, null, pub_id
```

For more information about the output of sp_column_privileges, see the Reference Manual.
Using views and stored procedures as security mechanisms

Views and stored procedures can serve as security mechanisms. You can give users controlled access to database objects via a view or stored procedure without granting them direct access to the data. For example, you might give a clerk execute permission on a procedure that updates cost information in a projects table without letting the user see confidential data in the table. To use this feature, you must own the procedure or view as well as its underlying objects. If you do not own the underlying objects, users must have permission to access the objects. For more information about when permissions are required, see “Understanding ownership chains” on page 603.

Adaptive Server makes permission checks, as required, when the view or procedure is used. When you create the view or procedure, Adaptive Server makes no permission checks on the underlying objects.

Using views as security mechanisms

Through a view, users can query and modify only the data they can see. The rest of the database is neither visible nor accessible.

Permission to access the view must be explicitly granted or revoked, regardless of the permissions on the view’s underlying tables. If the view and underlying tables are owned by the same owner, no permissions need to be given on the underlying tables. Data in an underlying table that is not included in the view is hidden from users who are authorized to access the view but not the underlying table.

By defining different views and selectively granting permissions on them, a user (or any combination of users) can be restricted to different subsets of data. Access can be restricted to:

- A subset of the rows of a base table (a value-dependent subset). For example, you might define a view that contains only the rows for business and psychology books to keep information about other types of books hidden from some users.
- A subset of the columns of a base table (a value-independent subset). For example, you might define a view that contains all the rows of the titles table, but omits the price and advance columns, since this information is sensitive.
- A row-and-column subset of a base table.
• The rows that qualify for a join of more than one base table. For example, you might define a view that joins the titles, authors, and titleauthor tables. This view hides personal data about authors and financial information about the books.

• A statistical summary of data in a base table. For example, you might define a view that contains only the average price of each type of book.

• A subset of another view, or of some combination of views and base tables.

Let’s say you want to prevent some users from accessing the columns in the titles table that display money and sales amounts. You can create a view of the titles table that omits those columns, and then give all users permission on the view but only the Sales Department permission on the table:

grant all on bookview to public
grant all on titles to sales

An equivalent way of setting up these privilege conditions, without using a view, is to use the following statements:

grant all on titles to public
revoke select, update on titles (price, advance, total_sales) from public
grant select, update on titles (price, advance, total_sales) to sales

One possible problem with the second solution is that users not in the sales group who enter the select * from titles command might be surprised to see the message that includes the phrase:

permission denied

Adaptive Server expands the asterisk into a list of all the columns in the titles table, and since permission on some of these columns has been revoked from nonsales users, access to these columns is denied. The error message lists the columns for which the user does not have access.

To see all the columns for which they do have permission, the nonsales users must name them explicitly. For this reason, creating a view and granting the appropriate permissions on it is a better solution.

You can also use views for context-sensitive protection. For example, you can create a view that gives a data entry clerk permission to access only those rows that he or she has added or updated. To do so, add a column to a table in which the user ID of the user entering each row is automatically recorded with a default. You can define this default in the create table statement, like this:
**Using views and stored procedures as security mechanisms**

```sql
create table testtable
    (empid           int,
     startdate      datetime,
     username       varchar(30) default user)
```

Next, define a view that includes all the rows of the table where `uid` is the current user:

```sql
create view context_view
as
    select *
    from testtable
    where username = user_name()
    with check option
```

The rows retrievable through this view depend on the identity of the person who issues the `select` command against the view. By adding `with check option` to the view definition, you make it impossible for any data entry clerk to falsify the information in the `username` column.

**Using stored procedures as security mechanisms**

If a stored procedure and all underlying objects are owned by the same user, that owner can grant users permission to use the procedure without granting permissions on the underlying objects. For example, you might give a user permission to execute a stored procedure that updates a row-and-column subset of a specified table, even though that user does not have any other permissions on that table.

**Roles and stored procedures**

Use the `grant execute` command to grant execute permission on a stored procedure to all users who have been granted a specified role. `revoke execute` removes this permission. But `grant execute` permission does not prevent users who do not have the specified role from being granted execute permission on the stored procedure.
For further security, you can restrict the use of a stored procedure by using the `proc_role` system function within the procedure to guarantee that a procedure can be executed only by users who have a given role. `proc_role` returns 1 if the user has a specific role (`sa_role`, `sso_role`, `oper_role`, or any user-defined role) and returns 0 if the user does not have that role. For example, here is a procedure that uses `proc_role` to see if the user has the system administrator role:

```sql
create proc test_proc
as
if (proc_role("sa_role") = 0)
begin
    print "You don't have the right role"
    return -1
end
else
    print "You have SA role"
    return 0
```

See “System Functions” in the *Reference Manual* for more information about `proc_role`.

**Understanding ownership chains**

Views can depend on other views or tables. Procedures can depend on other procedures, views, or tables. These dependencies can be thought of as an *ownership chain*.

Typically, the owner of a view also owns its underlying objects (other views and tables), and the owner of a stored procedure owns all the procedures, tables, and views referenced by the procedure.

A view and its underlying objects are usually all in the same database, as are a stored procedure and all the objects it references; however, this is not required. If objects are in different databases, a user wanting to use the view or stored procedure must be a valid user or guest user in all of the databases containing the objects. This prevents users from accessing a database unless the database owner has authorized it.

When a user who has been granted `execute` permission on a procedure or view uses it, Adaptive Server does not check permissions on any of the underlying objects if:

- These objects and the view or procedure are owned by the same user, and
Using views and stored procedures as security mechanisms

- The user accessing the view or procedure is a valid user or guest user in each of the databases containing the underlying objects.

However, if all objects are not owned by the same user, Adaptive Server checks object permissions when the ownership chain is broken. That is, if object A references object B, and B is not owned by the user who owns object A, Adaptive Server checks the permissions for object B. In this way, Adaptive Server allows the owner of the original data to retain control over who is authorized to access it.

Ordinarily, a user who creates a view needs to worry only about granting permissions on that view. For example, say Mary has created a view called auview1 on the authors table, which she also owns. If Mary grants select permission to Sue on auview1, Adaptive Server allows Sue to access it without checking permissions on authors.

However, a user who creates a view or stored procedure that depends on an object owned by another user must be aware that any permissions he or she grants depend on the permissions allowed by those other owners.

Example of views and ownership chains

Say Joe creates a view called auview2, which depends on Mary’s view auview1. Joe grants Sue select permission on auview2.

<table>
<thead>
<tr>
<th>Sue’s permission</th>
<th>Objects</th>
<th>Ownership</th>
<th>Checks</th>
</tr>
</thead>
<tbody>
<tr>
<td>select</td>
<td>auview2</td>
<td>Joe</td>
<td>Sue not owner</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Check permissions</td>
</tr>
<tr>
<td>select</td>
<td>auview1</td>
<td>Mary</td>
<td>Different owner</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Check permissions</td>
</tr>
<tr>
<td>none</td>
<td>authors</td>
<td>Mary</td>
<td>Same owner</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>No permission check</td>
</tr>
</tbody>
</table>

Adaptive Server checks the permissions on auview2 and auview1, and finds that Sue can use them. Adaptive Server checks ownership on auview1 and authors and finds that they have the same owner. Therefore, Sue can use auview2.
Taking this example a step further, suppose that Joe’s view, auview2, depends on auview1, which depends on authors. Mary decides she likes Joe’s auview2 and creates auview3 on top of it. Both auview1 and authors are owned by Mary.

The ownership chain looks like this:

*Figure 17-3: Ownership chains and permission checking for views, case 2*

<table>
<thead>
<tr>
<th>Sue’s permission</th>
<th>Objects</th>
<th>Ownership</th>
<th>Checks</th>
</tr>
</thead>
<tbody>
<tr>
<td>select</td>
<td>auview3</td>
<td>Mary</td>
<td>Sue not owner</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Check permissions</td>
</tr>
<tr>
<td>select</td>
<td>auview2</td>
<td>Joe</td>
<td>Different owner</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Check permissions</td>
</tr>
<tr>
<td>select</td>
<td>auview1</td>
<td>Mary</td>
<td>Different owner</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Check permissions</td>
</tr>
<tr>
<td>none</td>
<td>authors</td>
<td>Mary</td>
<td>Same owner</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>No permission check</td>
</tr>
</tbody>
</table>

When Sue tries to access auview3, Adaptive Server checks permissions on auview3, auview2, and auview1. If Joe has granted permission to Sue on auview2, and Mary has granted her permission on auview3 and auview1, Adaptive Server allows the access. Adaptive Server checks permissions only if the object immediately before it in the chain has a different owner (or if it is the first object in the chain). For example, it checks auview2 because the object before it—auview3—is owned by a different user. It does not check permission on authors, because the object that immediately depends on it, auview1, is owned by the same user.

### Example of procedures and ownership chains

Procedures follow the same rules as views. For example, suppose the ownership chain looks like this:
To execute procedure `proc4`, Sue must have permission to execute `proc4`, `proc2`, and `proc1`. Permission to execute `proc3` is not necessary because `proc3` and `proc4` have the same owner.

Adaptive Server checks Sue’s permissions on `proc4` and all objects it references each time she executes `proc4`. Adaptive Server knows which referenced objects to check: it determined this the first time Sue executed `proc4`, and it saved the information with the procedure’s execution plan. Unless one of the objects referenced by the procedure is dropped or redefined, Adaptive Server does not change its initial decision about which objects to check.

This protection hierarchy allows every object’s owner to fully control access to the object. Owners can control access to views and stored procedures, as well as to tables.
Permissions on triggers

A trigger is a special kind of stored procedure used to enforce integrity, especially referential integrity. Triggers are never executed directly, but only as a side effect of modifying a table. You cannot grant or revoke permissions for triggers.

Only an object owner can create a trigger. However, the ownership chain can be broken if a trigger on a table references objects owned by different users. The protection hierarchy rules that apply to procedures also apply to triggers.

While the objects that a trigger affects are usually owned by the user who owns the trigger, you can write a trigger that modifies an object owned by another user. If this is the case, any users modifying your object in a way that activates the trigger must have permission on the other object as well.

If Adaptive Server denies permission on a data modification command because a trigger affects an object for which the user does not have permission, the entire data modification transaction is rolled back.

For more information on triggers, see the Transact-SQL User’s Guide or the Reference Manual.

Using row-level access control

Row-level access control enables the database owner or table owner to create a secure data access environment automatically, by providing:

- More granular data security: you can set permissions for individual rows, not just tables and columns
- Automatic data filtering according to group, role, and application
- Data-level security encoded in the server

Row-level access control restricts access to data in a table’s individual rows, through three features:

- Access rules that the database owner defines and binds to the table
- Application Context Facility, which provides built-in functions that define, store, and retrieve user-defined contexts
- Login triggers that the database owner, sa_role, or the user can create
Using row-level access control

Adaptive Server enforces row-level access control for all data manipulation languages (DMLs), preventing users from bypassing the access control to get to the data.

The syntax for configuring your system for row-level access control is:

```
sp_configure "enable row level access", 1
```

This option slightly increases the amount of memory Adaptive Server uses, and you need an ASE_RLAC license option. Row-level access control is a dynamic option, so you need not restart Adaptive Server.

Access rules

To use the row-level access control feature, add the `access` option to the existing `create rule` syntax. Access rules restrict any rows that can be viewed or modified.

Access rules are similar to domain rules, which allow table owners to control the values users can insert or update on a column. The domain rule applies restrictions to added data, functioning on `update` and `insert` commands.

Access rules apply restrictions to retrieved data, enforced on `select`, `update`, and `delete` operations. Adaptive Server enforces the access rules on all columns that are read by a query, even if the columns are not included in the select list. In other words, in a given query, Adaptive Server enforces the domain rule on the table that is updated, and the access rule on all tables that are read.

For example:

```
insert into orders_table
select * from old_orders_table
```

In this query, if there are domain rules on the `orders_table` and access rules on the `old_orders_table`, Adaptive Server enforces the domain rule on the `orders_table`, because it is updated, and the access rule on the `old_orders_table`, because it is read.

Using access rules is similar to using views, or using an ad hoc query with `where` clauses. The query is compiled and optimized after the access rules are attached, so it does not cause performance degradation. Access rules provide a virtual view of the table data, the view depending on the specific access rules bound to the columns.
Access rules can be bound to user-defined datatypes, defined with `sp_addtype`. Adaptive Server enforces the access rule on user tables, which frees the table owner or database owner from the maintenance task of binding access rules to columns in the normalized schema. For instance, you can create a user-defined type, whose base type is `varchar(30)`, call it `username`, and bind an access rule to it. Adaptive Server enforces the access rule on any tables in your application that have columns of type `username`.

Application developers can write flexible access rules using Java and application contexts, described in “Access rules as user-defined Java functions” on page 614, and “Using the Application Context Facility” on page 617.

Syntax for access rules

Use the `access` parameter in the `create rule` syntax to create access rules.

```
create [or|and] access rule (access_rule_name)
    as (condition)
```

Creating a sample table with access rules

This section shows the process of creating a table and binding an access rule to it.

Creating a table

A table owner creates and populates table T (`username char(30), title char(30), classified_data char(1024)`):

```
AA, "Administrative Assistant","Memo to President"
AA, "Administrative Assistant","Tracking Stock Movements"
VP1, "Vice President", "Meeting Schedule"
VP2, "Vice President", "Meeting Schedule"
```

Creating and binding access rules

The table owner creates access rule `uname_acc_rule` and binds it to the `username` column on table T.

```
create access rule uname_acc_rule
    as @username = suser_name()

sp_bindrule uname_acc_rule, "T.username"
```

Querying the table

When you issue the following query:

```
select * from T
```
Using row-level access control

Adaptive Server processes the access rule that is bound to the username column on table T and attaches it to the query tree. The tree is then optimized and an execution plan is generated and executed, as though the user had executed the query with the filter clause given in the access rule. In other words, Adaptive Server attaches the access rule and executes the query as:

```sql
select * from T where T.username = suser_name().
```

The condition `where T.username = suser_name()` is enforced by the server. The user cannot bypass the access rule.

The result of an Administrative Assistant executing the select query is:

```
AA, "Administrative Assistant","Memo to President"
AA, "Administrative Assistant","Tracking Stock Movements"
```

Dropping an access rule

Before you drop an access rule, you must unbind it from any columns or datatypes, using `sp_unbindrule`, as in the following example:

```sql
sp_unbindrule "T.username",
NULL, "all"
```

`sp_unbindrule` unbinds any domain rules attached to the column by default.

After you unbind the rule, you can drop it:

```sql
drop rule "rule_name"
```

For example:

```sql
drop rule "T.username"
```

Syntax for extended access rule

Each access rule is bound to one column, but you can have multiple access rules in a table. `create rule` provides AND and OR parameters to handle evaluating multiple access rules. To create AND access rules and OR access rules, use extended access rule syntax:

- **AND access rule**:
  ```sql
  create and access rule rule_name
  ```

- **OR access rule**
  ```sql
  create or access rule rule_name as
  ```
You can bind AND access rules and OR access rules to a column or user-defined datatype. With the extended access rule syntax, you can bind multiple access rules to the table, although you can bind only one per column. When the table is accessed, the access rules go into effect, the AND rules bound first by default, and then the OR access rules.

If you bind multiple access rules to a table without defining AND or OR access, the default access rule is AND.

If there is only one access rule on a row of the table and it is defined as an OR access rule, it behaves as an AND access rule.

Using access and extended access rules

Create access rules

The following steps create access rules:

create access rule empid1_access
as @empid = 1

create access rule deptno1_access
as @deptid = 2

The following steps create OR access rules:

create or access rule name1_access
as @name = "smith"

create or access rule phone_access
as @phone = "9999"

Create table

This step creates a test table:

create table testtab1 (empno int, deptno int,name char(10), phone char(4))

Bind rules to table

The following steps bind access rules to the test table columns:

sp_bindrule empid1_access, "testtab1.empno"
/*Rule bound to table column.*/
(return status = 0)

sp_bindrule deptno1_access,"testtab1.deptno"
/*Rule bound to table column.*/
(return status = 0)

sp_bindrule name1_access,"testtab1.name"
/*Rule bound to table column.*/
(return status = 0)
Using row-level access control

sp_bindrule phone_access,"testtab1.phone"
/*Rule bound to table column.*/
(return status = 0)

Insert data into table

The following steps insert values into the test table:

insert testtab1 values (1,1,"smith","3245")
(1 row affected)

insert testtab1 values(2,1,"jones","0283")
(1 row affected)

insert testtab1 values(1,2,"smith","8282")
(1 row affected)

insert testtab1 values(2,2,"smith","9999")
(1 row affected)

Access rule examples

The following examples show how access rules return specific rows containing information limited by access rules.

Example 1

This example returns information from two rows:

/* return rows when empno = 1 and deptno = 2 and ( name = "smith" or phone = "9999" ) */

select * from testtab1

<table>
<thead>
<tr>
<th>empno</th>
<th>deptno</th>
<th>name</th>
<th>phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>smith</td>
<td>8282</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>jones</td>
<td>9999</td>
</tr>
</tbody>
</table>

(2 rows affected)

/* unbind access rule from specific column */

sp_unbindrule "testtab1.empno",NULL,"accessrule"

/*Rule unbound from table column.*/

(return status = 0)

Example 2

This example returns information from four rows:
/* return rows when deptno = 2 and ( name = "smith" or phone = "9999" ) */

```sql
select * from testtabl
```

<table>
<thead>
<tr>
<th>empno</th>
<th>deptno</th>
<th>name</th>
<th>phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>smith</td>
<td>8282</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>smith</td>
<td>9999</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>smith</td>
<td>8888</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>jones</td>
<td>9999</td>
</tr>
</tbody>
</table>

(4 rows affected)

/* unbind all deptno rules from specific column */

```sql
sp_unbindrule "testtabl.deptno", NULL,"all"
```

/* Rule unbound from table column. */

(return status = 0)

**Example 3**

This example returns information from six rows:

/* return the rows when name = "smith" or phone = "9999" */

```sql
select * from testtabl
```

<table>
<thead>
<tr>
<th>empno</th>
<th>deptno</th>
<th>name</th>
<th>phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>smith</td>
<td>3245</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>smith</td>
<td>8282</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>smith</td>
<td>9999</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>smith</td>
<td>8888</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>jones</td>
<td>9999</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>jones</td>
<td>9999</td>
</tr>
</tbody>
</table>
Using row-level access control

Access rules and alter table command

When the table owner uses the `alter table` command, Adaptive Server disables access rules during the execution of the command and enables them upon completion of the command. The access rules are disabled to avoid filtering the table data during the `alter table` command.

Access rules and `bcp`

Adaptive Server enforces access rules when data is copied out of a table using the bulk copy utility (`bcp`). Adaptive Server cannot disable access rules, as it does with `alter table`, because `bcp` can be used by any user who has select permission on the table.

For security purposes, the database owner should lock the table exclusively and disable access rules during bulk copy out. The lock disables access to other users while the access rules are disabled. The database owner should bind the access rules and unlock the table after the data has been copied.

Access rules as user-defined Java functions

Access rules can use user-defined Java functions. For example, you can use Java functions to write sophisticated rules using the profile of the application, the user logged in to the application, and the roles that the user is currently assigned for the application.

The following Java class uses the method `GetSecVal` to demonstrate how you can use Java methods that use JDBC as user-defined functions inside access rules:

```java
import java.sql.*;
import java.util.*;

public class sec_class {
  static String _url = "jdbc:sybase:asejdbc";
  public static int GetSecVal(int c1) {
    try {
      PreparedStatement pstmt;
      ResultSet rs = null;
      Connection con = null;
      int pno_val;

      pstmt = null;
```
Class.forName("sybase.asejdbc.ASEDriver"); con = DriverManager.getConnection(_url);

if (con == null)
{
    return (-1);
}

pstmt = con.prepareStatement("select classification
from sec_tab where id = ?");

if (pstmt == null)
{
    return (-1);
}

pstmt.setInt(1, c1);
rs = pstmt.executeQuery();
rs.next();
pno_val = rs.getInt(1);
rs.close();
pstmt.close();
con.close();
return (pno_val);
}
catch (SQLException sqe)
{
    return(sqe.getErrorCode());
}
catch (ClassNotFoundException e)
{
    System.out.println("Unexpected exception : " + e.toString());
    System.out.println("\nThis error usually indicates that " + "your Java CLASSPATH environment has not been set properly.");
Using row-level access control

e.printStackTrace();
return (-1);
}
catch (Exception e)
{
System.out.println("Unexpected exception : " +
e.toString());
e.printStackTrace();
return (-1);
}

After compiling the Java code, you can run the same program from isql, as follows.

For example:

javac sec_class.java
jar cufo sec_class. jar sec_class.class
installjava -Usa -Password -
f/work/work/FGAC/sec_class.jar -
-D testdb

From isql:

/*to create new user datatype class_level*/
sp_addtype class_level, int
/*to create the sample secure data table*/
create table sec_data (c1 varchar(30),
c2 varchar(30),
c3 varchar(30),
clevel class_level)
/*to create the classification table for each user*/
create table sec_tab (userid int, clevel class-level
int)

insert into sec_tab values (1,10)
insert into sec_tab values (2,9)
insert into sec_tab values (3,7)
insert into sec_tab values (4,7)
insert into sec_tab values (5,4)
insert into sec_tab values (6,4)
insert into sec_tab values (7,4)

declare @v1 int
select @v1 = 5
while @v1 > 0
Using the Application Context Facility

Applications on a database server must limit access to the data. Applications are carefully coded to consider the profile of the user. For example, a Human Resources application is coded to know which users are allowed to update salary information.
Using row-level access control

The attributes that enable this coding comprise an application context. The Application Context Facility (ACF) consists of three built-in functions that provide a secure environment for data access, by allowing access rules to compare against the intrinsic values assigned to users in a session.

An application context consists of context_name, attribute_name, and attribute_value. Users define the context name, the attributes, and the values for each context. You can use the default read-only application context that Sybase provides, SYS_SESSION, to access some session-specific information. This application context is shown as Table 17-6 on page 625. You can also create your own application contexts, as described in “Creating and using application contexts” on page 620.

The user profile, combined with the application profile, which is defined in a table created by the system administrator, permits cumulative and overlapping security schemes.

ACF allows users to define, store, and retrieve:

- User profiles (the roles authorized to a user and the groups to which the user belongs)
- Application profiles currently in use

Any number of application contexts per session are possible, and any context can define any number of attribute/value pairs. ACF context rows are specific to a session, and not persistent across sessions; however, unlike local variables, they are available across nested levels of statement execution. ACF provides built-in functions that set, get, list, and remove these context rows.

Setting permissions for using application context functions

You execute an application context function in a select statement. The owner of the function is the system administrator of the server. You can create, set, retrieve, and remove application contexts using built-in functions.

The data used in the built-in functions is defined in a table that contains all logins for all tables, which created by the system administrator. For more information about this table, see “Using login triggers” on page 627.

- set_appcontext() stores:
  
  ```sql
  select set_appcontext ("titles", "rlac", "1")
  ```

- get_appcontext() supplies two parts of a context in a session, and retrieves the third:
  
  ```sql
  select get_appcontext ("titles", "rlac")
  ```
Granting and revoking

You can grant and revoke privileges to users, roles, and groups in a given database to access objects in that database. The only exceptions are create database, set session authorization, and connect. A user granted these privileges should be a valid user in the master database. To use other privileges, the user must be a valid user in the database where the object is located.

The use of built-in functions means that unless special arrangements are made, any logged-in user can reset the profiles of the session. Although Adaptive Server audits built-in functions, security may be compromised before the problem is noticed. To restrict access to these built-in functions, use grant and revoke privileges. Only users with the sa_role can grant or revoke privileges on the built-in functions. Only the select privilege is checked as part of the server-enforced data access control checks performed by the functions.

Valid users

Built-in functions do not have an object ID and they do not have a home database. Therefore, each database owner must grant the select privilege for the functions to the appropriate user. Adaptive Server finds the user’s default database and checks the permissions against this database. With this approach, only the owner of the users’ default database needs to grant the select privilege. If other databases should be restricted, the owner of those databases must explicitly revoke permission from the user in those databases.

Only the application context built-in functions perform data access control checks on the user when you grant and revoke privileges on them. Granting or revoking privileges for other functions has no effect in Adaptive Server.

Privileges granted to public affect only users named in the table created by the system administrator. For information about the table, see “Using login triggers” on page 627. Guest users have privileges only if the sa_role specifically grants it by adding them to the table.

A system administrator can execute the following commands to grant or revoke select privileges on specific application context functions:

grant select on set_appcontext to user_role
grant select on set_appcontext to joe_user
revoke select on set_appcontext from joe_user
Creating and using application contexts

The following built-in functions are available for creating and maintaining application contexts. For more information, see the Reference Manual.

- set_appcontext
- get_appcontext
- list_appcontext
- rm_appcontext

set_appcontext

Sets an application context name, attribute name, and attribute value, defined by the attributes of an application, for a specified user session.

set_appcontext ("context_name", "attribute_name", "attribute_value")

- context_name – a row that specifies an application context name, saved as the datatype char(30).
- attribute_name – a row that specifies an application context name, saved as the datatype char(30)
- attribute_value – a row that specifies an application attribute value, saved as the datatype char(255).

Examples

This example creates an application context called CONTEXT1, with an attribute ATTR1 that has the value VALUE1:

```sql
select set_appcontext ("CONTEXT1", "ATTR1", "VALUE1")
```

```
0
```

This example shows an attempt to override the existing application context. The attempt fails, returning -1:

```sql
select set_appcontext("CONTEXT1", "ATTR1", "VALUE1")
```

```
-1
```

This example shows how set_appcontext can include a datatype conversion in the value:

```sql
declare @val numeric
select @val = 20
```
CHAPTER 17  Managing User Permissions

```
select set_appcontext("CONTEXT1", "ATTR2",
convert(char(20), @val))
-----------
0

This example shows the result when a user without appropriate permissions
attempts to set the application context. The attempt fails, returning -1:

select set_appcontext("CONTEXT1", "ATTR2", "VALUE1")
-----------
-1
```

Usage

- `set_appcontext` returns 0 for success and -1 for failure.
- If you set values that already exist in the current session, `set_appcontext`
  returns -1.
- `set_appcontext` cannot override the values of an existing application
  context. To assign new values to a context, remove the context and re-
  create it using the new values.
- `set_appcontext` saves attributes as char datatypes. If you create an access
  rule that must compare the attribute value to another datatype, the rule
  should convert the char data to the appropriate datatype.
- All arguments in this function are required.

get_appcontext

Returns the value of the attribute in a specified context.

```
get_appcontext("context_name", "attribute_name")
```

- `context_name` – a row specifying an application context name, saved as
datatype char(30).
- `attribute_name` – a row specifying an application context attribute name,
saved as datatype char(30).

Examples

This example shows VALUE1 returned for ATTR1:

```
select get_appcontext("CONTEXT1", "ATTR1")
-----------
VALUE1
```
Using row-level access control

ATTR1 does not exist in CONTEXT2:

```sql
select get_appcontext("CONTEXT2", "ATTR1")
------------
NULL
```

This example shows the result when a user without appropriate permissions attempts to get the application context:

```sql
select get_appcontext("CONTEXT1", "ATTR2")
select permission denied on built-in get_appcontext, database dbid
------------
-1
```

Usage

- `get_appcontext` returns 0 for success and -1 for failure.
- If the attribute you require does not exist in the application context, `get_appcontext` returns “null.”
- `get_appcontext` saves attributes as `char` datatypes. If you create an access rule that compares the attribute value to other datatypes, the rule should convert the `char` data to the appropriate datatype.
- All arguments in this function are required.

list_appcontext

Lists all the attributes of all the contexts in the current session.

```sql
list_appcontext ("context_name")
```

- `context_name` – names all the application context attributes in the session. `list_appcontext` has a datatype of `char(30)`.

Examples

To use `list_appcontext`, the user must have appropriate permissions. For more information, see “Setting permissions for using application context functions” on page 618.

This example shows the results of a user with appropriate permissions listing the application contexts:

```sql
select list_appcontext ("*", "*")
Context Name: (CONTEXT1)
Attribute Name: (ATTR1) Value: (VALUE2)
```
This example shows a user without appropriate permissions attempting to list the application contexts. The attempt fails, returning -1.

```sql
select list_appcontext()
Select permission denied on built-in list_appcontext, database DBID
---------
-1
```

**Usage**

- `list_appcontext` returns 0 for success and -1 for failure.
- Since built-in functions do not return multiple result sets, the client application receives `list_appcontext` returns as messages.

**rm_appcontext**

Removes a specific application context, or all application contexts.

```
rm_appcontext ("context_name", "attribute_name")
```
- `context_name` – a row specifying an application context name, saved as datatype `char(30)`.
- `attribute_name` – a row specifying an application context attribute name, saved as datatype `char(30)`.  

**Examples**

The following three examples show how to remove an application context by specifying some or all attributes. Use an asterisk ("*"), to remove all attributes in the specified context.

```
select rm_appcontext("CONTEXT1", "*")
---------
0
```

Use an asterisk ("*"), to remove all the contexts and attributes.

```
select rm_appcontext("**, "**")
---------
0
```
Using row-level access control

This example shows a user attempting to remove a nonexistent context. The attempt fails, returning -1.

```sql
select rm_appcontext("NON_EXISTING_CTX", "ATTR2")
---------
-1
```

This example shows the result of a user without appropriate permissions attempting to remove an application context.

```sql
select rm_appcontext("CONTEXT1", "ATTR2")
---------
-1
```

Usage

- `rm_appcontext` returns 0 for success, -1 for failure.
- All arguments in this function are required.

**SYS_SESSION system application context**

The `SYS_SESSION` context shows the default predefined application context, which provides session-specific pairs of attributes and values. The syntax for using the context is:

```sql
select list_appcontext ("SYS_SESSION", "*"
```

Then:

```sql
select get_appcontext ("SYS_SESSION", "<attribute>")
```
Table 17-6: SYS_SESSION attributes and values

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>username</td>
<td>Login name</td>
</tr>
<tr>
<td>hostname</td>
<td>Host name from which the client has connected</td>
</tr>
<tr>
<td>applname</td>
<td>Name of the application as set by the client</td>
</tr>
<tr>
<td>suserid</td>
<td>User ID of the user in the current database</td>
</tr>
<tr>
<td>groupid</td>
<td>Group ID of the user in the current database</td>
</tr>
<tr>
<td>dbid</td>
<td>ID of the user’s current database</td>
</tr>
<tr>
<td>dbname</td>
<td>Current database</td>
</tr>
<tr>
<td>spid</td>
<td>Server process ID</td>
</tr>
<tr>
<td>proxy_suserid</td>
<td>The server user ID of the proxy</td>
</tr>
<tr>
<td>client_name</td>
<td>Client name set by the middle-tier application, using the set client_name command</td>
</tr>
<tr>
<td>client_applname</td>
<td>Client application name set by the middle-tier application, using the set client_applname command</td>
</tr>
<tr>
<td>client_hostname</td>
<td>Client host name set by the middle-tier application, using the set client_hostname command</td>
</tr>
<tr>
<td>language</td>
<td>Current language the client is using by default or after using the set language command (@@language)</td>
</tr>
<tr>
<td>character_set</td>
<td>Character set the client is using (@@client_csname)</td>
</tr>
<tr>
<td>dateformat</td>
<td>Date expected by the client, set using the set dateformat command</td>
</tr>
<tr>
<td>is_showplan_on</td>
<td>Returns YES if set showplan is on, NO if it is off</td>
</tr>
<tr>
<td>is_noexec_on</td>
<td>Returns YES if set no exec is on, NO if it is off</td>
</tr>
</tbody>
</table>

Solving a problem using an access rule and ACF

This section shows the solution of a problem: each of five users, on different security levels, should see only rows with a value less than or equal to his or her security level. This solution uses access rules, with the Application Context Facility, to display only the rows that one of the users, Dave, sees.

There are five logins:

- Anne has security level 1.
- Bob has security level 1.
- Cassie has security level 2.
- Dave has security level 2.
Using row-level access control

- Ellie has security level 4.

Users should see only rows with a value in $rlac$ that is less than or equal to their own security level. To accomplish this, create an access rule and apply ACF.

The $rlac$ column is type integer, and appcontext arguments are type char.

```sql
create access rule rlac_rule as
    @value <= convert(int, get_appcontext("titles", "rlac"))

sp_bindrule rlac_rule, "titles.rlac"

/* log in as Dave and apply ACF value of 2*/
select set_appcontext("titles", "rlac", "2")

/*this value persists throughout the session*/
/*select all rows*/
select title_id, rlac from titles
```

<table>
<thead>
<tr>
<th>title_id</th>
<th>rlac</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC8888</td>
<td>1</td>
</tr>
<tr>
<td>BU1032</td>
<td>2</td>
</tr>
<tr>
<td>PS7777</td>
<td>1</td>
</tr>
<tr>
<td>PS3333</td>
<td>1</td>
</tr>
<tr>
<td>BU1111</td>
<td>2</td>
</tr>
<tr>
<td>FC1035</td>
<td>1</td>
</tr>
<tr>
<td>BU2075</td>
<td>2</td>
</tr>
<tr>
<td>PS2091</td>
<td>1</td>
</tr>
<tr>
<td>PS2106</td>
<td>1</td>
</tr>
<tr>
<td>BU7832</td>
<td>2</td>
</tr>
<tr>
<td>PS1372</td>
<td>1</td>
</tr>
</tbody>
</table>

(11 rows affected)
Using login triggers


Login triggers execute a specified stored procedure every time a user logs in. The login trigger is an ordinary stored procedure, except it executes in the background. It is the last step in a successful login process, and sets the application context for the user logging in.

Only the system security officer can register a login trigger to users in the server.

To provide a secure environment, the system administrator must:

1. Revoke select privilege on the set_appcontext function. The owner of a login trigger must have explicit permission to use set_appcontext, even if the owner has sa_role.
2. Configure a login trigger from a stored procedure for each user, and register the login trigger to the user.
3. Provide execute privilege to the login trigger that the user executes.

Creating login triggers

Create a login trigger as a stored procedure. Do not use the create trigger command. The following sample creates a login trigger stored procedure in the pubs2 database:

```sql
create loginproc as
    declare @appname varchar(20)
    declare @attr varchar(20)
    declare @value varchar(20)
    declare @retvalue int
    declare apctx cursor for
        select appname, attr, value from
        pubs2.dbo.lookup where login = suser_name()
    open apctx
    fetch apctx into @appname, @attr, @value
    While (@@sqlstatus = 0)
       begin
           select f@retval =
               set_appcontext (rtrim (@appname),
```
Using row-level access control

```sql
rtrim(@attr), rtrim(@value))
fetch apctx into @appname, @attr, @value
end
go

grant execute on loginproc to public

go
```

To associate a specific user with the login trigger, run `sp_modifylogin` in the user’s default database.

Configuring login triggers

You must have `sso_role` enabled to set, change, or drop a login trigger. The object ID of the login trigger is stored in the `syslogins.procid` column. Login triggers do not exist by default. They must be registered using `sp_modifylogin`. The syntax is:

```sql
sp_modifylogin <login_name>, "login script", <sproc_name>
```

- `login_name` – the user’s login name.
- "login script" – type in as shown; "login script" tells `sp_modifylogin` that the next parameter, "sproc_name", is a login trigger.
- `sproc_name` – the name of the stored procedure configured as a login trigger for this user.

Run this procedure from the user’s default database. The stored procedure you are registering as a login trigger must be available in the user’s default database, because Adaptive Server searches the `sysobjects` table in the user’s default database to find the login trigger object.

The following example configures the stored procedure `my_proc` (which must exist in the database you want to configure) as a login trigger for Adaptive Server login `my_login`:

```sql
sp_modifylogin my_login, "login script", my_proc
```

Again, you must execute the command from within the user’s default database. Adaptive Server checks to see whether the login has `execute` permissions on the stored procedure, but not until the user actually logs in and executes the login trigger.

Once you have configured a stored procedure as a login trigger, you cannot drop it. You must unconfigure it first, either by dropping the login trigger altogether, or by changing the login trigger to a different stored procedure. To drop the login trigger, enter:
Displaying the login trigger

To change the login trigger to a different stored procedure, enter:

```
sp_modifylogin my_login, "login script", NULL
```

To display the current login trigger, use `sp_displaylogin`:

```
sp_displaylogin my_login
```

go
(....)
Default Database: my_db
Default Language:
Auto Login Script: my_proc
....

Executing a login trigger

Login triggers are different from ordinary stored procedures in that once they are registered they execute in the background, without active user connections. Once you have configured a login trigger, Adaptive Server automatically executes it in the background as soon as the user logs in, but before the server executes any commands from the client application.

If one login makes multiple concurrent connections, the login trigger executes independently during each session. Similarly, multiple logins can configure the same stored procedure to be a login trigger.

Background execution means that you cannot use some standard features of stored procedures in a stored procedure configured as a login trigger. For instance, you cannot pass any parameters without default values to or from the procedure, nor does the procedure pass back any result values.

This special execution mode affects any stored procedures that are called by the login trigger stored procedure, as well as any output generated by the login trigger stored procedure itself.

You can also execute a login trigger stored procedure as a normal stored procedure, for example, from `isql`. The procedure executes and behaves normally, showing all output and error messages as usual.

Understanding login trigger output

The main effect of executing the stored procedure as a background task is that output from the login trigger is not written to the client application, but to the Adaptive Server error log file, as are some, but not all, error messages.
Using row-level access control

Output from print or raiserror messages is prefixed by the words background task message or background task error in the error log. For example, the statements print “Hello!” and raiserror 123456 in a login trigger appear in the Adaptive Server error log as:

(....) background task message: Hello!
(....) background task error 123456: This is test message 123456

However, not all output goes to the Adaptive Server error log:

- No result sets from select statements (which are normally sent to a client connection) appear anywhere, not even in the Adaptive Server error log. This information disappears.
- The following statements execute normally: insert...select and select...into statements, as well as other DML statements which do not ordinarily send a result set to the client application, and DDL statements ordinarily allowed in a stored procedure.

Using login triggers for other applications

Login triggers are part of the row-level access control feature in Adaptive Server. In this context, you can use a login trigger in combination with the features for access rules and application contexts to set up row-level access controls, once a session logs in to Adaptive Server. However, you can use login triggers for other purposes as well.

The following example limits the number of concurrent connections to Adaptive Server that a specific login can make. Each of the commands described in steps 1 and 2 in the example are executed in the default database of the user for whom the access needs to be restricted:

1. As system administrator, create the limit_user_sessions stored procedure:

```sql
create procedure limit_user_sessions
    as
    declare @cnt int,
        @limit int,
        @loginname varchar(32)

    select @limit = 2 -- max nr. of concurrent logins
    /* determine current #sessions */
    select @cnt = count(*)
    from master.dbo.sysprocesses
    where suid = suser_id()
```

Limiting the number of concurrent connections
/* check the limit */
if @cnt > @limit
begin
    select @loginname = suser_name()
    print "Aborting login [%1!]: exceeds session
         limit [%2!]",
        @loginname, @limit
    /* abort this session */
    select syb_quit()
end
end
grant exec on limit_user_sessions to public
gos

2 As system security officer, configure this stored procedure as a login
    trigger for user “bob”:
    sp_modifylogin "bob", "login script",
    "limit_user_sessions"
gos

3 Now, when user “bob” creates a third session for Adaptive Server, this
    session is terminated by the login trigger calling the syb_quit() function:
    % isql -SASE125 -Ubob -Pbobpassword
    1> select 1
    2> go
    CT-LIBRARY error:
    ct_results(): network packet layer: internal net
    library error: Net-Library operation terminated due
to disconnect

4 This message appears in the Adaptive Server error log file:
    (... background task message: Aborting login [my_login]: exceeds session
    limit [2]

Enforcing timed-based restrictions

This example describes how system administrators can create a login trigger to
    enforce time-based restrictions on user sessions. Each of the commands
    described in steps 1 – 4 are executed in the default database of the user for
    whom the access needs to be restricted:

1 As System Administrator, create this table:

    create table access_times (suid int not null,
Using row-level access control

```sql

dayofweek tinyint,
shiftstart time,
shiftend time)

2 As system administrator, insert the following rows in table access_times. These rows indicate that user “bob” is allowed to log into Adaptive Server on Mondays between 9:00am and 5:00pm, and user “mark” is allowed to login to Adaptive Server on Tuesdays between 9:00Am and 5:00PM

insert into access_times
select suser_id('bob'), 1, '9:00', '17:00'
go
insert into access_times
select suser_id('mark'), 2, '9:00', '17:00'
go

3 As system administrator, create the limit_access_time stored procedure, which references the access_time table to determine if login access should be granted:

create procedure limit_access_time as
declare @curdate date,
@curdow tinyint,
@curtime time,
@cnt int,
@loginname varchar(32)

-- setup variables for current day-of-week, time
select @curdate = current_date()
select @curdow = datepart(cdw,@curdate)
select @curtime = current_time()
select @cnt = 0

-- determine if current user is allowed access
select @cnt = count(*)
from access_times
where suid = suser_id()
and dayofweek = @curdow
and @curtime between shiftstart and shiftend

if @cnt = 0
    begin
    select @loginname = suser_name()
    print "Aborting login [%1!]: login attempt past normal working hours", @loginname

            -- abort this session
            return -4
```
end
go

grant exec on limit_access_time to public
go

4 As system security officer, configure the `limit_access_time` stored procedure as a login trigger for users “bob” and “mark”:

```sql
sp_modifylogin "bob", "login script", "limit_access_time"
go
sp_modifylogin "mark", "login script", "limit_access_time"
go
```

5 On Mondays, user “bob” can successfully create a session:

```sql
isql -Ubob -Ppassword
1> select 1
2> go
-------------
1
(1 row affected)
```

However, user “mark” is denied access to Adaptive Server:

```sql
isql -Umark -Ppassword
1> select 1
2> go
CT-LIBRARY error:
    ct_results(): network packet layer: internal net library error: Net-Library operation terminated due to disconnect
```

6 The following message is logged in the errorlog:

```
(...)
server back-ground task message: Aborting login [mark]: login attempt past normal working hours
```

The above examples show how you can limit the number of concurrent connections for a specific login and restrict access to specific times of day for that login, but it has one disadvantage: the client application cannot easily detect the reason the session was terminated. To display a message to the user, such as “Too many users right now—please try later,” use a different approach.

Instead of calling the built-in function `syb_quit()`, which causes the server to simply terminate the current session, you can deliberately cause an error in the stored procedure to abort the login trigger stored procedure.
For example, dividing by zero aborts the login trigger stored procedure, terminates the session, and causes a message to appear.

**Login trigger restrictions**

The following actions are restricted.

- You cannot use a login trigger to set session-specific options, such as `set nocount on`, `set rowcount on`, and so on. Setting session options in any stored procedure has an effect only inside that stored procedure.

- You cannot create `#temp` tables to use later in the session. Once the procedure completes, the `#temp` tables drop away automatically and the original session settings are restored, as in any other stored procedure.

- You should not use login triggers on the `sa` login; a failing login trigger can lock you out of Adaptive Server.

- Do not use a login trigger for anything that may take longer than a few seconds to process, or that risks processing problems.

**Issues and information**

- If you do not have access to the Adaptive Server error log, do not use login triggers. Always check the Adaptive Server error log for error messages.

- For Adaptive Server version 15.0.2 and later, any exportable option set or unset in a login trigger take effect in the login process when the server starts.

  To disable this behavior, execute `set export_options off` inside the login trigger.

  Adaptive Server versions 15.0.1, 12.5.4, and earlier required that you start Adaptive Server with trace flag 4073 to enable the options for a login trigger.

- A client application, like `isql`, is unaware of the existence or execution of a login trigger; it presents a command prompt immediately after the successful login, though Adaptive Server does not execute any commands before the login trigger successfully executes. This `isql` prompt displays even if the login trigger has terminated the user connection.
• The user logging in to Adaptive Server must have `execute` permission to use the login trigger stored procedure. If no `execute` permission has been granted, an error message appears in the Adaptive Server error log and the user connection closes immediately (though `isql` still shows a command prompt).

Adaptive Server error log shows a message similar to the following:

```
EXECUTE permission denied on object my_proc,
database my_db, owner dbo
```

• The login trigger stored procedure cannot contain parameters without specified default values. If parameters without default values appear in the stored procedure, the login trigger fails and an error similar to the following appears in the Adaptive Server error log:

```
Procedure my_proc expects parameter @param1, which was not supplied...
```

### Disabling execute privilege on login triggers

A database owner or administrator can disable `execute` privilege on the login trigger, or code the login trigger to permit access only at certain times. For example, you may want to prohibit regular users from using the server while the database owner or administrator is updating the table.

**Note** If the login trigger returns a minus number, the login fails.

### Exporting set options from a login trigger

Adaptive Server allows options for the `set` command that are inside login triggers to remain valid for the entire user session.

The following `set` options are automatically exported:

- `showplan`
- `arithabort [overflow | numeric_truncation]`
- `arithignore [overflow]`
- `colnames`
- `format`
Using row-level access control

- statistics io
- procid
- rowcount
- altnames
- nocount
- quoted_identifier
- forceplan
- fmtonly
- close on endtran
- fipsflagger
- self_recursion
- ansinull
- dup_in_subquery
- or_strategy
- flushmessage
- ansi_permissions
- string_rtruncation
- prefetch
- triggers
- replication
- sort_resources
- transactional_rpc
- cis_rpc_handling
- strict_dtm_enforcement
- raw_object_serialization
- texptr_parameters
- sort_merge
- remote_indexes
Setting global login triggers

Use `sp_logintrig` to set a global login trigger that is executed at each user login. To take user-specific actions, set a user specific login trigger using `sp_modifylogin` or `sp_addlogin`.

Note  You can activate this option by setting trace flag -T4073.
Using row-level access control
This chapter describes how to set up auditing for your installation.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction to auditing in Adaptive Server</td>
<td>639</td>
</tr>
<tr>
<td>Installing and setting up auditing</td>
<td>644</td>
</tr>
<tr>
<td>Setting global auditing options</td>
<td>661</td>
</tr>
<tr>
<td>Querying the audit trail</td>
<td>671</td>
</tr>
<tr>
<td>Understanding the audit tables</td>
<td>671</td>
</tr>
</tbody>
</table>

Introduction to auditing in Adaptive Server

A principal element of a secure system is accountability. One way to ensure accountability is to audit events on the system. Many events that occur in Adaptive Server can be recorded.

Auditing is an important part of security in a database management system. An audit trail can be used to detect penetration of the system and misuse of resources. By examining the audit trail, a system security officer can inspect patterns of access to objects in databases and can monitor the activity of specific users. Audit records are traceable to specific users, which may act as a deterrent to users who are misusing the system.

Each audit record can log the nature of the event, the date and time, the user responsible for it, and the success or failure of the event. Among the events that can be audited are logins and logouts, server boots, use of data access commands, attempts to access particular objects, and a particular user’s actions. The audit trail, or log of audit records, allows the system security officer to reconstruct events that have occurred on the system and evaluate their impact.

The system security officer is the only user who can start and stop auditing, set up auditing options, and process the audit data. As a system security officer, you can establish auditing for events such as:

- Server-wide, security-relevant events
Creating, deleting, and modifying database objects
- All actions by a particular user or all actions by users with a particular role active
- Granting or revoking database access
- Importing or exporting data
- Logins and logouts

Correlating Adaptive Server and operating system audit records
The easiest way to link Adaptive Server audit records with operating system records is to make Adaptive Server login names the same as operating system login names.
Alternatively, the system security officer can map users’ operating system login names to their Adaptive Server login names. However, this approach requires ongoing maintenance, as login names for new users must be recorded manually.

The audit system
The audit system consists of:
- The sybsecurity database, which contains global auditing options and the audit trail
- The in-memory audit queue, to which audit records are sent before they are written to the audit trail
- Configuration parameters for managing auditing
- System procedures for managing auditing

The sybsecurity database
The sybsecurity database is created during the auditing installation process. In addition to all the system tables found in the model database, it contains sysauditoptions, a system table for keeping track of server-wide auditing options, and system tables for the audit trail.
sysaudittable contains the current setting of global auditing options, such as whether auditing is enabled for disk commands, remote procedure calls, ad hoc user-defined auditing records, or all security-relevant events. These options affect the entire Adaptive Server.

The audit trail

Adaptive Server stores the audit trail in system tables named sysaudits_01 through sysaudits_08. When you install auditing, you determine the number of audit tables for your installation. For example, if you choose to have two audit tables, they are named sysaudits_01 and sysaudits_02. At any given time, only one audit table is current. Adaptive Server writes all audit data to the current audit table. A system security officer can use sp_configure to set, or change, which audit table is current.

The recommended number of tables is two or more with each table on a separate audit device. This allows you to set up a smoothly running auditing process in which audit tables are archived and processed with no loss of audit records and no manual intervention.

Warning! Sybase strongly recommends against using a single audit table on production systems. If you use only a single audit table, you may lose audit records. If you must use only a single audit table because of limited system resources, see “Single-table auditing” on page 657 for instructions.

Figure 18-1 shows how the auditing process works with multiple audit tables.
The auditing system writes audit records from the in-memory audit queue to the current audit table. When the current audit table is nearly full, a threshold procedure can automatically archive the table to another database. The archive database can be backed up and restored with the `dump` and `load` commands. Use archive database access for read-only access to archived audit tables from backup. See Chapter 14, "Archive Database Access," in the *System Administration Guide, Volume 2*. For more information about managing the audit trail, see “Setting up audit trail management” on page 648.
The audit queue

When an audited event occurs, an audit record first goes to the in-memory audit queue. The record remains in memory until the audit process writes it to the audit trail. You can configure the size of the audit queue with the `audit queue size` parameter of `sp_configure`.

Before you configure the size of the audit queue, consider the trade-off between the risk of losing records in the queue if the system crashes and the loss of performance when the queue is full. As long as an audit record is in the queue, it can be lost if the system crashes. However, if the queue repeatedly becomes full, overall system performance is affected. If the audit queue is full when a user process tries to generate an audit record, the process sleeps until space in the queue becomes available.

**Note**  Because audit records are not written directly to the audit trail, you cannot count on an audit record’s being stored immediately in the current audit table.

Auditing configuration parameters

Use these configuration parameters to manage the auditing process:

- **auditing** enables or disables auditing for the entire Adaptive Server. The parameter takes effect immediately upon execution of `sp_configure`. Auditing occurs only when this parameter is enabled.
- **audit queue size** establishes the size of the audit queue. Because the parameter affects memory allocation, the parameter does not take effect until Adaptive Server is restarted.
- **suspend audit when device full** controls the behavior of the audit process when an audit device becomes full. The parameter takes effect immediately upon execution of `sp_configure`.
- **current audit table** sets the current audit table. The parameter takes effect immediately upon execution of `sp_configure`.

System procedures for auditing

Use these system procedures to manage the auditing process:

- **sp_audit** enables and disables auditing options. This is the only system procedure required to establish the events to be audited.
Installing and setting up auditing

- `sp_displayaudit` displays the active auditing options.
- `sp_addauditrecord` adds user-defined audit records (comments) into the audit trail. Users can add these records only if a system security officer enables ad hoc auditing with `sp_audit`.

Installing and setting up auditing

Table 18-1 provides a general procedure for setting up auditing.

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Install auditing.</td>
<td>Set the number of audit tables and assign devices for the audit trail and the <code>syslogs</code> transaction log in the <code>sybsecurity</code> database.</td>
<td>“Installing the audit system” on page 645 and the Adaptive Server installation and configuration documentation</td>
</tr>
<tr>
<td>2. Set up audit trail management.</td>
<td>Write and establish a threshold procedure that receives control when the current audit table is nearly full. The procedure automatically switches to a new audit table and archives the contents of the current table. In addition, this step involves setting the audit queue size and the suspend audit when device full configuration parameters.</td>
<td>“Setting up audit trail management” on page 648 For single-table auditing, “Single-table auditing” on page 657</td>
</tr>
<tr>
<td>3. Set up transaction log management in the <code>sybsecurity</code> database.</td>
<td>Determine how to handle the <code>syslogs</code> transaction log in the <code>sybsecurity</code> database, how to set the <code>trunc log on chkpt</code> database option and establishing a last-chance threshold procedure for <code>syslogs</code> if <code>trunc log on chkpt</code> is off.</td>
<td>“Setting up transaction log management” on page 654</td>
</tr>
<tr>
<td>4. Set auditing options.</td>
<td>Use <code>sp_audit</code> to establish the events to be audited.</td>
<td>“Setting global auditing options” on page 661</td>
</tr>
<tr>
<td>5. Enable auditing.</td>
<td>Use <code>sp_configure</code> to turn on the auditing configuration parameter. Adaptive Server begins writing audit records to the current audit table.</td>
<td>“Enabling and disabling auditing” on page 656</td>
</tr>
<tr>
<td>6. Restarting auditing.</td>
<td>Use <code>sp_audit restart</code> to restart auditing if it fails.</td>
<td>“Restarting auditing” on page 660</td>
</tr>
</tbody>
</table>
Installing the audit system

The audit system is usually installed with auditinit, the Sybase installation program. Alternatively, you can install auditing without auditinit. For details, see “Installing auditing with installsecurity” on page 645. Installation and auditinit are discussed in the Adaptive Server installation and configuration documentation for your platform.

When you install auditing, you can establish the number of system tables you want to use for the audit trail, the device for each audit system table, and the device for the syslogs transaction log.

Tables and devices for the audit trail

You can specify up to eight system tables (sysaudits_01 through sysaudits_08). Plan to use at least two tables for the audit trail. Put each table on its own device separate from the master device. If you do this, you can use a threshold procedure to automatically archive the current audit table before it fills up and switch to a new empty table for the subsequent audit records.

Device for the syslogs transaction log table

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

Installing auditing with installsecurity

The $SYBASE/ASE-15_0/scripts directory contains installsecurity, a script for installing auditing.

Note This example assumes a server that uses a logical page size of 2K.

To use installsecurity to install auditing:

1 Create the auditing devices and auditing database with the disk init and create database commands. For example:

   disk init name = "auditdev",
   physname = "/dev/dsk/c2d0s4",
   size = "10M"
   disk init name = "auditlogdev",
Installing and setting up auditing

physname = "/dev/dsk/c2d0s5",
size = "2M"
create database sybsecurity on auditdev
log on auditlogdev

2 Use isql to execute the installsecurity script:

cd $SYBASE/ASE-12_5/scripts
setenv DSQUERY server_name
isql -Usa -Ppassword -Sserver_name < installsecurity

3 Shut down and restart Adaptive Server.

When you have completed these steps, the sybsecurity database has one audit table (sysaudits_01) created on its own segment. You can enable auditing at this time, but should add more auditing tables with sp_addaudittable. For information about disk init, create database, and sp_addaudittable, see the Reference Manual.

Moving the auditing database to multiple devices

Place the sybsecurity database on its own device, separate from the master database. If you have more than one audit table, place each table on its own device. It can also be helpful to put each table on a separate segment which points to a separate device. If you currently have sybsecurity on the same device as master, or if you want to move sybsecurity to another device, use one of the procedures described in the following sections. When you move the database, you can specify whether to save your existing global audit settings.

Moving sybsecurity without saving global audit settings

Note These steps include dropping the sybsecurity database, which destroys all audit records and global audit settings previously recorded in sybsecurity. Before you drop the sybsecurity database, make sure you archive existing records with a backup or by following instructions in “Archiving the audit table” on page 649 to avoid losing any historical data that remains in the sybsecurity tables.

To move the sybsecurity database without saving the global audit settings:

1 Execute the following to remove any information related to logins from the syslogins system table:

    sp_audit "all","all","all","off"
2 Drop the sybsecurity database.
3 Install sybsecurity again using the installation procedure described in either:
   • The configuration documentation for your platform, or
   • “Installing auditing with installsecurity” on page 645.
4 During the installation process, place the sybsecurity database on one or more devices, separate from the master device.

Moving sybsecurity and saving global audit settings

❖ To move the sybsecurity database and save the global audit settings
1 Dump the sybsecurity database:
   dump database sybsecurity to "/remote/sec_file"
2 Drop the sybsecurity database:
   drop database sybsecurity
3 Initialize the first device on which you want to place the sybsecurity database:
   disk init name = "auditdev",
   physname = "/dev/dsk/c2d0s4",
   size = "10M"
4 Initialize the device where you want to place the security log:
   disk init name = "auditlogdev",
   physname = "/dev/dsk/c2d0s5",
   size = "2M"
5 Create the new sybsecurity database:
   create database sybsecurity on auditdev
   log on auditlogdev
6 Load the contents of the old sybsecurity database into the new database.
   The global audit settings are preserved:
   load database sybsecurity from "/remote/sec_file"
7 Run online database, which upgrades sysaudits and sysauditoptions if necessary:
   online database sybsecurity
Installing and setting up auditing

8 Load the auditing system procedures using the configuration documentation for your platform.

❖ Creating more than one sysaudits table in sybsecurity

1 Initialize the device where you want to place the additional table:
   
   ```
   disk init name = "auditdev2",
   physname = "/dev/dsk/c2d0s6",
   size = "10M"
   ```

2 Extend the sybsecurity database to the device you initialized in step 1:
   
   ```
   alter database sybsecurity on auditdev2 = "2M"
   ```

3 Run `sp_addaudittable` to create the next sysaudits table on the device you initialized in step 1:
   
   ```
   sp_addaudittable auditdev2
   ```

4 Repeat steps 1 – 3 for each sysaudits table.

Setting up audit trail management

To effectively manage the audit trail:

1 Be sure that auditing is installed with two or more tables, each on a separate device. If not, consider adding additional audit tables and devices.

2 Write a threshold procedure and attach it to each audit table segment.

3 Set configuration parameters for the audit queue size and to indicate appropriate action should the current audit table become full.

The following sections assume that you have installed auditing with two or more tables, each on a separate device. If you have only one device for the audit tables, skip to “Single-table auditing” on page 657.

Setting up threshold procedures

Before enabling auditing, establish a threshold procedure to automatically switch auditing tables when the current table is full.

The threshold procedure for the audit device segments should:

- Make the next empty audit table current using `sp_configure` to set the current audit table configuration parameter.

- Archive the audit table that is almost full using the `insert...select` command.
Changing the current audit table

The current audit table configuration parameter establishes the table where Adaptive Server writes audit rows. As a system security officer, you can change the current audit table with `sp_configure`, using the following syntax, where `n` is an integer that determines the new current audit table:

```
sp_configure "current audit table", n
    [, "with truncate"]
```

The valid values for `n` are:

- 1 means `sysaudits_01`, 2 means `sysaudits_02`, and so forth.
- 0 tells Adaptive Server to automatically set the current audit table to the next table. For example, if your installation has three audit tables, `sysaudits_01`, `sysaudits_02`, and `sysaudits_03`, Adaptive Server sets the current audit table to:
  - 2 if the current audit table is `sysaudits_01`
  - 3 if the current audit table is `sysaudits_02`
  - 1 if the current audit table is `sysaudits_03`

The `with truncate` option specifies that Adaptive Server should truncate the new table if it is not already empty. If you do not specify this option and the table is not empty, `sp_configure` fails.

Note: If Adaptive Server truncates the current audit table and you have not archived the data, the table’s audit records are lost. Archive the audit data before you use the `with truncate` option.

To execute `sp_configure` to change the current audit table, you must have the `sso_role` active. You can write a threshold procedure to automatically change the current audit table.

Archiving the audit table

You can use `insert with select` to copy the audit data into an existing table having the same columns as the audit tables in `sybsecurity`.

Be sure that the threshold procedure can successfully copy data into the archive table in another database:

1. Create the archive database on a separate device from the one containing audit tables in `sybsecurity`. 
2. Create an archive table with columns identical to those in the `sybsecurity` audit tables. If such a table does not already exist, you can use `select into` to create an empty one by having a false condition in the `where` clause. For example:

```sql
use aud_db
go
select *
into audit_data
from sybsecurity.dbo.sysaudits_01
where 1 = 2
```

The `where` condition is always false, so an empty duplicate of `sysaudits_01` is created.

The `select into/bulk copy` database option must be turned on in the archive database (using `sp_dboption`) before you can use `select into`.

The threshold procedure, after using `sp_configure` to change the audit table, can use `insert` and `select` to copy data to the archive table in the archive database. The procedure can execute commands similar to these:

```sql
insert aud_db.sso_user.audit_data
select * from sybsecurity.dbo.sysaudits_01
```

**Example threshold procedure for audit segments**

This sample threshold procedure assumes that three tables are configured for auditing:

```sql
declare @audit_table_number int
/*
** Select the value of the current audit table
*/
select @audit_table_number = scc.value
from master.dbo.syscurconfigs scc, master.dbo.sysconfigures sc
where sc.config=scc.config and sc.name  = "current audit table"
/*
** Set the next audit table to be current.
** When the next audit table is specified as 0,
** the value is automatically set to the next one.
*/
exec sp_configure "current audit table", 0, "with truncate"
/*
** Copy the audit records from the audit table
** that became full into another table.
*/
if @audit_table_number = 1
```
begin
  insert aud_db.sso_user.sysaudits
  select * from sysaudits_01
  truncate table sysaudits_01
end
else if @audit_table_number = 2
begin
  insert aud_db.sso_user.sysaudits
  select * from sysaudits_02
  truncate table sysaudits_02
end
return(0)

Attaching the threshold procedure to each audit segment

To attach the threshold procedure to each audit table segment, use the sp_addthreshold.

Before executing sp_addthreshold:

- Determine the number of audit tables configured for your installation and the names of their device segments
- Have the permissions and roles you need for sp_addthreshold for all the commands in the threshold procedure

Warning! sp_addthreshold and sp_modifythreshold check to ensure that only a user with sa_role directly granted can add or modify a threshold. All system-defined roles that are active when you add or modify a threshold are inserted as valid roles for your login in the systhresholds table. However, only directly granted roles are activated when the threshold procedure fires.

Audit tables and their segments

When you install auditing, auditinit displays the name of each audit table and its segment. The segment names are “aud_seg1” for sysaudits_01, “aud_seg2” for sysaudits_02, and so forth. You can find information about the segments in the sybsecurity database if you execute sp_helpsegment with sybsecurity as your current database. One way to find the number of audit tables for your installation is to execute the following SQL commands:

use sybsecurity
go
select count(*) from sysobjects
Installing and setting up auditing

where name like "sysaudit%"

go

In addition, you can get information about the audit tables and the sybsecurity database by executing the following SQL commands:

sp_helpdb sybsecurity

go
use sybsecurity

go
sp_help sysaudits_01

go
sp_help sysaudits_02

go
...

Required roles and permissions

To execute sp_addthreshold, you must be either the database owner or a system administrator. A system security officer should be the owner of the sybsecurity database and, therefore, should be able to execute sp_addthreshold. In addition to being able to execute sp_addthreshold, you must have permission to execute all the commands in your threshold procedure. For example, to execute sp_configure for current audit table, the sso_role must be active. When the threshold procedure fires, Adaptive Server attempts to turn on all the roles and permissions that were in effect when you executed sp_addthreshold.

To attach the threshold procedure audit_thresh to three device segments:

use sybsecurity

go
sp_addthreshold sybsecurity, aud_seg_01, 250, audit_thresh
sp_addthreshold sybsecurity, aud_seg_02, 250, audit_thresh
sp_addthreshold sybsecurity, aud_seg_03, 250, audit_thresh

go

The sample threshold procedure audit_thresh receives control when fewer than 250 free pages remain in the current audit table.

For more information about adding threshold procedures, see Chapter 16, “Managing Free Space with Thresholds,” in System Administration Guide: Volume 2.
Auditing with the sample threshold procedure in place

After you enable auditing, Adaptive Server writes all audit data to the initial current audit table, sysaudits_01. When sysaudits_01 is within 250 pages of being full, the threshold procedure audit_thresh fires. The procedure switches the current audit table to sysaudits_02, and, immediately, Adaptive Server starts writing new audit records to sysaudits_02. The procedure also copies all audit data from sysaudits_01 to the audit_data archive table in the audit_db database. The rotation of the audit tables continues in this fashion without manual intervention.

Setting auditing configuration parameters

Set the following configuration parameters for your auditing installation:

- audit queue size sets the number of records in the audit queue in memory.
- suspend audit when device full determines what Adaptive Server does if the current audit table becomes completely full. The full condition occurs only if the threshold procedure attached to the current table segment is not functioning properly.

Setting the size of the audit queue

The default audit queue size is 100 bytes. The amount of memory consumed by the audit queue pool is defined by the audit queue size parameter, and includes data buffers and overhead for the memory pool. However, the amount of memory in the pool can vary between releases and chip architectures.

To set the length of the audit queue, use sp_configure. The syntax is:

```
sp_configure "audit queue size", [value]
```

value is the number of records that the audit queue can hold. The minimum value is 1, and the maximum is 65,535. For example, to set the audit queue size to 300, execute:

```
sp_configure "audit queue size", 300
```

For more information about setting the audit queue size and other configuration parameters, see Chapter 5, “Setting Configuration Parameters.”
Installing and setting up auditing

Suspending auditing if devices are full

If you have two or more audit tables, each on a separate device other than the master device, and have a threshold procedure for each audit table segment, the audit devices should never become full. Only if a threshold procedure is not functioning properly would the “full” condition occur. You can use sp_configure to set the suspend audit when device full parameter to determine what happens if the devices do become full. Choose one of these options:

• Suspend the auditing process and all user processes that cause an auditable event. Resume normal operation after a system security officer clears the current audit table.

• Truncate the next audit table and start using it. This allows normal operation to proceed without intervention from a system security officer.

To set this configuration parameter, use sp_configure. You must have the sso_role active. The syntax is:

```
sp_configure "suspend audit when device full", [0|1]
```

• 0 – truncates the next audit table and starts using it as the current audit table whenever the current audit table becomes full. If you set the parameter to 0, the audit process is never suspended; however, older audit records are lost if they have not been archived.

• 1 (the default value) – suspends the audit process and all user processes that cause an auditable event. To resume normal operation, the system security officer must log in and set up an empty table as the current audit table. During this period, the system security officer is exempt from normal auditing. If the system security officer’s actions would generate audit records under normal operation, Adaptive Server sends an error message and information about the event to the error log.

If you have a threshold procedure attached to the audit table segments, set suspend audit when device full to 1 (on). If it is set to 0 (off), Adaptive Server may truncate the audit table that is full before your threshold procedure has a chance to archive your audit records.

Setting up transaction log management

This section describes guidelines for managing the transaction log in sybsecurity.
If the `trunc log on chkpt` database option is active, Adaptive Server truncates syslogs every time it performs an automatic checkpoint. After auditing is installed, the value of `trunc log on chkpt` is on, but you can use `sp_dboption` to change its value.

**Truncating the transaction log**

If you enable the `trunc log on chkpt` option for the `sybsecurity` database, you do not need to worry about the transaction log becoming full. Adaptive Server truncates the log whenever it performs a checkpoint. With this option on, you cannot use `dump transaction` to dump the transaction log, but you can use `dump database` to dump the database.

If you follow the procedures in “Setting up threshold procedures” on page 648, audit tables are automatically archived to tables in another database. You can use standard backup and recovery procedures for this archive database.

If a crash occurs on the `sybsecurity` device, you can reload the database and resume auditing. At most, only the records in the in-memory audit queue and the current audit table are lost because the archive database contains all other audit data. After you reload the database, use `sp_configure` with `truncate` to set and truncate the current audit table.

If you have not changed server-wide auditing options since you dumped the database, all auditing options stored in `sysauditoptions` are automatically restored when you reload `sybsecurity`. If not, you can run a script to set the options prior to resuming auditing.

**Managing the transaction log with no truncation**

If you use `db_option` to turn the `trunc log on chkpt` off, the transaction log may fill up. Plan to attach a *last-chance threshold procedure* to the transaction log segment. This procedure gets control when the amount of space remaining on the segment is less than a threshold amount computed automatically by Adaptive Server. The threshold amount is an estimate of the number of free log pages that are required to back up the transaction log.
Installing and setting up auditing

The default name of the last-chance threshold procedure is sp_thresholdaction, but you can specify a different name with sp_modifythreshold, as long as you have the sa_role active.

Note sp_modifythreshold checks to ensure you have “sa_role” active. See “Attaching the threshold procedure to each audit segment” on page 651 for more information.

Adaptive Server does not supply a default procedure, but Chapter 16, “Managing Free Space with Thresholds” contains examples of last-chance threshold procedures. The procedure should execute the dump transaction command, which truncates the log. When the transaction log reaches the last-chance threshold point, any transaction that is running is suspended until space is available. The suspension occurs because the option abort xact when log is full is always set to false for the sybsecurity database. You cannot change this option.

With the trunc log on chkpt option off, you can use standard backup and recovery procedures for the sybsecurity database, but be aware that the audit tables in the restored database may not be in sync with their status during a device failure.

Enabling and disabling auditing

To enable or disable auditing, use sp_configure with the auditing configuration parameter. The syntax is:

\[
\text{sp_configure "auditing", [0 | 1 ]}
\]

- 1 – enables auditing.
- 0 – disables auditing.

For example, to enable auditing, enter:

\[
\text{sp_configure "auditing", 1}
\]

Note When you enable or disable auditing, Adaptive Server automatically generates an audit record. See event codes 73 and 74 in Table 18-5 on page 673.
Single-table auditing

Sybase strongly recommends that you *not* use single-device auditing for production systems. If you use only a single audit table, you create a window of time while you are archiving audit data and truncating the audit table during which incoming audit records are lost. There is no way to avoid this when using only a single audit table.

If you use only a single audit table, your audit table is likely to fill up. The consequences of this depend on how you have set `suspend audit when device full`. If you have `suspend audit when device full` set to on, the audit process is suspended, as are all user processes that cause auditable events. If `suspend audit when device full` is off, the audit table is truncated, and you lose all the audit records that were in the audit table.

For *non-production* systems, where the loss of a small number of audit records may be acceptable, you can use a single table for auditing, if you cannot spare the additional disk space for multiple audit tables, or you do not have additional devices to use.

The procedure for using a single audit table is similar to using multiple audit tables, with these exceptions:

- During installation, you specify only one system table to use for auditing.
- During installation, you specify only one device for the audit system table.
- The threshold procedure you create for archiving audit records is different from the one you would create if you were using multiple audit tables.

Figure 18-2 shows how the auditing process works with a single audit table.
Establishing and managing single-table auditing

The steps to configure for single-table auditing is the same as for multiple-table auditing. See Table 18-1 for more information.

Threshold procedure for single-table auditing

For single-table auditing, the threshold procedure should:

- Archive the almost-full audit table to another table, using the `insert` and `select` commands.
- Truncate the audit table to create space for new audit records, using the `truncate table` command.
Before you can archive your audit records, create an archive table that has the same columns as your audit table. After you have done this, your threshold procedure can use `insert with select` to copy the audit records into the archive table.

Here is a sample threshold procedure for use with a single audit table:

```sql
create procedure audit_thresh as

/*
** copy the audit records from the audit table to
** the archive table
*/
insert aud_db.sso_user.audit_data
    select * from sysaudits_01
return(0)
go
/*
** truncate the audit table to make room for new
** audit records
*/
truncate table "sysaudits_01"
go
```

After you have created your threshold procedure, you will need to attach the procedure to the audit table segment. For instructions, see “Attaching the threshold procedure to each audit segment” on page 651.

---

**Warning!** On a multiprocessor, the audit table may fill up even if you have a threshold procedure that triggers before the audit table is full. For example, if the threshold procedure is running on a heavily loaded CPU, and a user process performing auditable events is running on a less heavily loaded CPU, the audit table may fill up before the threshold procedure triggers. The configuration parameter `suspend audit when device full` determines what happens when the audit table fills up. For information about setting this parameter, see “Suspending auditing if devices are full” on page 654.

---

**What happens when the current audit table is full?**

When the current audit table is full:

1. The audit process attempts to insert the next audit record into the table. This fails, so the audit process terminates. An error message is written to the error log.
Installing and setting up auditing

2 When a user attempts to perform an auditable event, the event cannot be completed because auditing cannot proceed. The user process terminates. Users who do not attempt to perform an auditable event are unaffected.

3 If you have login auditing enabled, no one can log in to the server except a system security officer.

4 If you are auditing commands executed with the sso_role active, the system security officer cannot execute commands.

Recovering when the current audit table is full

If the current audit device and the audit queue become full, the system security officer becomes exempt from auditing. Every auditable event performed by a system security officer after this point sends a warning message to the error log file. The message states the date and time and a warning that an audit has been missed, as well as the login name, event code, and other information that would normally be stored in the extrainfo column of the audit table.

When the current audit table is full, the system security officer can archive and truncate the audit table as described in “Archiving the audit table” on page 649. A system administrator can execute shutdown to stop the server and then restart the server to reestablish auditing.

If the audit system terminates abnormally, the system security officer can shut down the server after the current audit table has been archived and truncated. Normally, only the system administrator can execute shutdown.

Restarting auditing

If the audit process is forced to terminate due to an error, sp_audit can be manually restarted by entering:

```
sp_audit restart
```

The audit process can be restarted provided that no audit was currently running, but the audit process must be enabled with sp_configure “auditing” 1.
CHAPTER 18  Auditing

Setting global auditing options

After you have installed auditing, you can use sp_audit to set auditing options. The syntax for sp_audit is:

```
sp_audit option, login_name, object_name [,setting]
```

If you run sp_audit with no parameters, it provides a complete list of the options. For details about sp_audit, see the Reference Manual.

Note  Auditing does not occur until you activate auditing for the server. For information on how to start auditing, see “Enabling and disabling auditing” on page 656.

Auditing options: types and requirements

The values you can specify for the login_name and object_name parameters to sp_audit depend on the type of auditing option you specify:

- Global options apply to commands that affect the entire server, such as booting the server, disk commands, and allowing ad hoc, user-defined audit records. Option settings for global events are stored in the sybsecurity..sysauditoptions system table.

- Database-specific options apply to a database. Examples include altering a database, bulk copy (bcp in) of data into a database, granting or revoking access to objects in a database, and creating objects in a database. Option settings for database-specific events are stored in the master..sysdatabases system table.

- Object-specific options apply to a specific object. Examples include selecting, inserting, updating, or deleting rows of a particular table or view and the execution of a particular trigger or procedure. Option settings for object-specific events are stored in the sysobjects system table in the relevant database.

- User-specific options apply to a specific user or system role. Examples include accesses by a particular user to any table or view or all actions performed when a particular system role, such as sa_role, is active. Option settings for individual users are stored in master..syslogins. The settings for system roles are stored in master..sysauditoptions.

Table 18-2 shows:
Setting global auditing options

- Valid values for the option and the type of each option – global, database-specific, object-specific, or user-specific
- Valid values for the login_name and object_name parameters for each option
- The database to be in when you set the auditing option
- The command or access that is audited when you set the option
- An example for each option

The default value for all options is off.

<table>
<thead>
<tr>
<th>Option (option type)</th>
<th>login_name</th>
<th>object_name</th>
<th>Database to be in to set the option</th>
<th>Command or access being audited</th>
</tr>
</thead>
<tbody>
<tr>
<td>adhoc (user-specific)</td>
<td>all</td>
<td>all</td>
<td>Any</td>
<td>Allows users to use sp_addauditrecord</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(Enables ad hoc user-defined auditing records.)</td>
</tr>
<tr>
<td>all (user-specific)</td>
<td>A login name or role</td>
<td>all</td>
<td>Any</td>
<td>All actions of a particular user or by users with a particular role active</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(Turns auditing on for all actions in which the sa_role is active.)</td>
</tr>
<tr>
<td>alter (database-specific)</td>
<td>all</td>
<td></td>
<td>Database to be audited</td>
<td>alter database, alter table</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(Turns auditing on for all executions of alter database and alter table in the master database.)</td>
</tr>
<tr>
<td>bcp (database-specific)</td>
<td>all</td>
<td></td>
<td>Database to be audited</td>
<td>bcp in</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(Returns the status of bcp auditing in the pubs2 database. If you do not specify a value for setting, Adaptive Server returns the status of auditing for the option you specify)</td>
</tr>
<tr>
<td>bind (database-specific)</td>
<td>all</td>
<td></td>
<td>Database to be audited</td>
<td>sp_bindefault, sp_bindmsg, sp_bindrule</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(Turns bind auditing off for the planning database.)</td>
</tr>
<tr>
<td>Option (option type)</td>
<td>login_name</td>
<td>object_name</td>
<td>Database to be in to set the option</td>
<td>Command or access being audited</td>
</tr>
<tr>
<td>----------------------</td>
<td>------------</td>
<td>-------------</td>
<td>------------------------------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>cmdtext (user-specific)</td>
<td>all</td>
<td>Any</td>
<td>SQL text entered by a user. (Does not reflect whether or not the text in question passed permission checks or not. eventmod always has a value of 1.)</td>
<td></td>
</tr>
<tr>
<td>create (database-specific)</td>
<td>all</td>
<td>Database to be audited</td>
<td>Any</td>
<td>create database, create table, create procedure, create trigger, create rule, create default, sp_addmessage, create view, create index, create function</td>
</tr>
<tr>
<td>dbaccess (database-specific)</td>
<td>all</td>
<td>Database to be audited</td>
<td>Any</td>
<td>Any access to the database from another database</td>
</tr>
<tr>
<td>dbcc (global)</td>
<td>all</td>
<td>all</td>
<td>Any</td>
<td>All dbcc commands that require permissions</td>
</tr>
<tr>
<td>delete (object-specific)</td>
<td>all</td>
<td>Name of the table or view to be audited, or default view or default table</td>
<td>The database of the table or view (except tempdb)</td>
<td>delete from a table, delete from a view</td>
</tr>
<tr>
<td>disk (global)</td>
<td>all</td>
<td>all</td>
<td>Any</td>
<td>disk init, disk refit, disk reinit, disk mirror, disk unmirror, disk remirror, disk resize</td>
</tr>
</tbody>
</table>

**Note** Specify master for object_name to audit create database. You are also auditing the creation of other objects in master.

**Example** `sp_audit "cmdtext", "sa", "all", "off"`
(Turns text auditing off for database owners.)

**Example** `sp_audit "create", "all", "planning", "pass"`
(Turns on auditing of successful object creations in the planning database. The current status of auditing create database is not affected because you did not specify the master database.)

**Example** `sp_audit "dbaccess", "all", "project", "on"`
(Audits all external accesses to the project database.)

**Example** `sp_audit "dbcc", "all", "all", "on"`
(Audits all dbcc commands that require permissions)

**Example** `sp_audit "delete", "all", "default table", "on"`
(Audits all delete actions for all future tables in the current database.)

**Example** `sp_audit "disk", "all", "all", "on"`
(Audits all disk actions for the server.)
### Setting global auditing options

<table>
<thead>
<tr>
<th>Option (option type)</th>
<th>login_name</th>
<th>object_name</th>
<th>Database to be in to set the option</th>
<th>Command or access being audited</th>
</tr>
</thead>
<tbody>
<tr>
<td>drop (database-specific)</td>
<td>all</td>
<td>Database to be audited</td>
<td>Any</td>
<td>drop database, drop table, drop procedure, drop index, drop trigger, drop rule, drop default, sp_dropmessage, drop view, drop function</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Example: <code>sp_audit &quot;drop&quot;, &quot;all&quot;, &quot;financial&quot;, &quot;fail&quot;</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(Audits all drop commands in the financial database that fail permission checks.)</td>
</tr>
<tr>
<td>dump (database-specific)</td>
<td>all</td>
<td>Database to be audited</td>
<td>Any</td>
<td>dump database, dump transaction</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Example: <code>sp_audit &quot;dump&quot;, &quot;all&quot;, &quot;pubs2&quot;, &quot;on&quot;</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(Audits dump commands in the pubs2 database.)</td>
</tr>
<tr>
<td>encryption_key (database-specific)</td>
<td>all</td>
<td>Database to be audited</td>
<td>Any</td>
<td>alter encryption key, create encryption key, drop encryption key, sp_encryption</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Example: Audits all the above commands in the pubs2 database: <code>sp_audit &quot;encryption_key&quot;, &quot;all&quot;, &quot;pubs2&quot;, &quot;on&quot;</code></td>
</tr>
<tr>
<td>errors (global)</td>
<td>all</td>
<td>all</td>
<td>Any</td>
<td>Fatal error, non-fatal error</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Example: <code>sp_audit &quot;errors&quot;, &quot;all&quot;, &quot;all&quot;, &quot;on&quot;</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(Audits errors throughout the server.)</td>
</tr>
<tr>
<td>errorlog</td>
<td>all</td>
<td>all</td>
<td>Any</td>
<td>sp_errorlog or the errorlog_admin function</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Example: <code>sp_audit &quot;errorlog&quot;, &quot;all&quot;, &quot;all&quot;, &quot;on&quot;</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(Audits attempts to &quot;change log&quot; to move to a new Adaptive Server error log file.)</td>
</tr>
<tr>
<td>exec_procedure (object-specific)</td>
<td>all</td>
<td>Name of the procedure to be audited or default procedure</td>
<td>The database of the procedure (except tempdb)</td>
<td>execute</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Example: <code>sp_audit &quot;exec_procedure&quot;, &quot;all&quot;, &quot;default procedure&quot;, &quot;off&quot;</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(Turns automatic auditing off for new procedures in the current database.)</td>
</tr>
<tr>
<td>exec_trigger (object-specific)</td>
<td>all</td>
<td>Name of the trigger to be audited or default trigger</td>
<td>The database of the trigger (except tempdb)</td>
<td>Any command that fires the trigger</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Example: <code>sp_audit &quot;exec_trigger&quot;, &quot;all&quot;, &quot;trig_fix_plan&quot;, &quot;fail&quot;</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(Audits all failed executions of the trig_fix_plan trigger in the current database.)</td>
</tr>
</tbody>
</table>
### Option (option type) | login_name | object_name | Database to be in to set the option | Command or access being audited
--- | --- | --- | --- | ---
func_dbaccess (database-specific) | all | Name of the database you are auditing | Any | Access to the database using the following functions: curunreserved_pgs, db_name, db_id, lct_admin, settbrepstat, setrepstatus, setrepdefmode, is_repagent_enabled, rep_agent_config, rep_agent_admin

**Example** `sp_audit @option="func_dbaccess", @login_name="all", @object_name = "strategy", @setting = "on"
(Audits accesses to the strategy database via built-in functions.)`

func_obj_access (object-specific) | all | Name of any object that has an entry in sysobjects | Any | Access to an object using the following functions: schema_inc, col_length, col_name, data_pgs, index_col, object_id, object_name, reserved_pgs, rowcnt, used_pgs, has_subquery

**Example** `sp_audit @option="func_obj_access", @login_name="all", @object_name = "customer", @setting = "on"
(Audits accesses to the customer table via built-in functions.)`

grant (database-specific) | all | Name of the database to be audited | Any | grant

**Example** `sp_audit @option="grant", @login_name="all", @object_name = "planning", @setting = "on"
(Audits all grants in the planning database.)`

insert (object-specific) | all | Name of the view or table to which you are inserting rows, or default view or default table | The database of the object (except tempdb) | insert into a table, insert into a view

**Example** `sp_audit "insert", "all", "dpt_101_view", "on"
(Audits all inserts into the dpt_101_view view in the current database.)`

install (database-specific) | all | Database to be audited | Any | install java

**Example** `sp_audit "install", "all", "planning", "on"
(Audits the installation of java classes in database planning)

load (database-specific) | all | Database to be audited | Any | load database, load transaction

**Example** `sp_audit "load", "all", "projects_db", "fail"
(Audits all failed executions of database and transaction loads in the projects_db database.)`
### Setting global auditing options

<table>
<thead>
<tr>
<th>Option (option type)</th>
<th>login_name</th>
<th>object_name</th>
<th>Database to be in to set the option</th>
<th>Command or access being audited</th>
</tr>
</thead>
<tbody>
<tr>
<td>login (global)</td>
<td>all</td>
<td>all</td>
<td>Any</td>
<td>Any login to Adaptive Server</td>
</tr>
<tr>
<td><strong>Example</strong></td>
<td>sp_audit &quot;login&quot;, &quot;all&quot;, &quot;all&quot;, &quot;fail&quot;</td>
<td>(Audits all failed attempts to log in to the server.)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>login_locked (global)</td>
<td>all</td>
<td>all</td>
<td>Any</td>
<td>login is locked because of exceeding the configured number of failed login attempts.</td>
</tr>
<tr>
<td><strong>Example</strong></td>
<td>sp_audit &quot;login_locked&quot;, &quot;all&quot;, &quot;all&quot;, &quot;on&quot;</td>
<td>(Login is locked because of exceeding the configured number of failed login attempts.)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>logout</td>
<td>all</td>
<td>all</td>
<td>Any</td>
<td>Any logout from Adaptive Server</td>
</tr>
<tr>
<td><strong>Example</strong></td>
<td>sp_audit &quot;logout&quot;, &quot;all&quot;, &quot;all&quot;, &quot;off&quot;</td>
<td>(Turns auditing off of logouts from the server.)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mount (global)</td>
<td>all</td>
<td>all</td>
<td>Any</td>
<td>mount database</td>
</tr>
<tr>
<td><strong>Example</strong></td>
<td>sp_audit &quot;mount&quot;, &quot;all&quot;, &quot;all&quot;, &quot;on&quot;</td>
<td>(Audits all mount database commands issued.)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>password</td>
<td>all</td>
<td>all</td>
<td>Any</td>
<td>Setting of global password and login policy options</td>
</tr>
<tr>
<td><strong>Example</strong></td>
<td>sp_audit &quot;password&quot;, &quot;all&quot;, &quot;all&quot;, &quot;on&quot;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>quiesce (global)</td>
<td>all</td>
<td>all</td>
<td>Any</td>
<td>quiesce database</td>
</tr>
<tr>
<td><strong>Example</strong></td>
<td>sp_audit &quot;quiesce&quot;, &quot;all&quot;, &quot;all&quot;, &quot;on&quot;</td>
<td>(Turns auditing on for quiesce database commands.)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>reference (object-specific)</td>
<td>all</td>
<td>Name of the view or table to which you are inserting rows, or default view or default table</td>
<td>Any</td>
<td>create table, alter table</td>
</tr>
<tr>
<td><strong>Example</strong></td>
<td>sp_audit &quot;reference&quot;, &quot;all&quot;, &quot;titles&quot;, &quot;off&quot;</td>
<td>(Turns off auditing of the creation of references to the titles table.)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>remove (database-specific)</td>
<td>all</td>
<td>all</td>
<td>Any</td>
<td>Audits the removal of Java classes</td>
</tr>
<tr>
<td><strong>Example</strong></td>
<td>sp_audit &quot;remove&quot;, &quot;all&quot;, &quot;planning&quot;, &quot;on&quot;</td>
<td>(Audits the removal of Java classes in the planning database.)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>revoke (database-specific)</td>
<td>all</td>
<td>Database to be audited</td>
<td>Any</td>
<td>revoke</td>
</tr>
<tr>
<td><strong>Example</strong></td>
<td>sp_audit &quot;revoke&quot;, &quot;all&quot;, &quot;payments_db&quot;, &quot;off&quot;</td>
<td>(Turns off auditing of the execution of revoke in the payments_db database.)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>rpc (global)</td>
<td>all</td>
<td>all</td>
<td>Any</td>
<td>Remote procedure calls (either in or out)</td>
</tr>
<tr>
<td><strong>Example</strong></td>
<td>sp_audit &quot;rpc&quot;, &quot;all&quot;, &quot;all&quot;, &quot;on&quot;</td>
<td>(Audits all remote procedure calls out of or into the server.)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Option (option type)</td>
<td>login_name</td>
<td>object_name</td>
<td>Database to be in to set the option</td>
<td>Command or access being audited</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------</td>
<td>-------------</td>
<td>--------------------------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>security (global)</td>
<td>all</td>
<td>all</td>
<td>Any</td>
<td>Server-wide security-relevant events. See the “security” option in Table 18-5.</td>
</tr>
<tr>
<td>Example</td>
<td>spAudit &quot;security&quot;, &quot;all&quot;, &quot;all&quot;, &quot;on&quot;</td>
<td>(Audits server-wide security-relevant events in the server.)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>select (object-specific)</td>
<td>all</td>
<td>Name of the view or table to which you are inserting rows, or default view or default table</td>
<td>The database of the object (except tempdb)</td>
<td>select from a table, select from a view</td>
</tr>
<tr>
<td>Example</td>
<td>spAudit &quot;select&quot;, &quot;all&quot;, &quot;customer&quot;, &quot;fail&quot;</td>
<td>(Audits all failed selects from the customer table in the current database.)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>setuser (database-specific)</td>
<td>all</td>
<td>all</td>
<td>Any</td>
<td>setuser</td>
</tr>
<tr>
<td>Example</td>
<td>spAudit &quot;setuser&quot;, &quot;all&quot;, &quot;projdb&quot;, &quot;on&quot;</td>
<td>(Audits all executions of setuser in the projdb database.)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>table_access (user-specific)</td>
<td>Login name</td>
<td>all</td>
<td>Any</td>
<td>select, delete, update, or insert access in a table</td>
</tr>
<tr>
<td>Example</td>
<td>spAudit &quot;table_access&quot;, &quot;smithson&quot;, &quot;all&quot;, &quot;on&quot;</td>
<td>(Audits all table accesses by the login named “smithson”.)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>truncate (database-specific)</td>
<td>all</td>
<td>Database to be audited</td>
<td>Any</td>
<td>truncate table</td>
</tr>
<tr>
<td>Example</td>
<td>spAudit &quot;truncate&quot;, &quot;all&quot;, &quot;customer&quot;, &quot;on&quot;</td>
<td>(Audits all table truncations in the customer database.)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>unbind (database-specific)</td>
<td>all</td>
<td>Database to be audited</td>
<td>Any</td>
<td>sp_unbindefault, sp_unbindrule, sp_unbindmsg</td>
</tr>
<tr>
<td>Example</td>
<td>spAudit &quot;unbind&quot;, &quot;all&quot;, &quot;master&quot;, &quot;fail&quot;</td>
<td>(Audits all failed attempts of unbinding in the master database.)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>unmount (global)</td>
<td>all</td>
<td>all</td>
<td>Any</td>
<td>unmount database</td>
</tr>
<tr>
<td>Example</td>
<td>spAudit &quot;unmount&quot;, &quot;all&quot;, &quot;all&quot;, &quot;on&quot;</td>
<td>(audits all attempts to unmount or create a manifest file with any database.)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Setting global auditing options

<table>
<thead>
<tr>
<th>Option (option type)</th>
<th>login_name</th>
<th>object_name</th>
<th>Database to be in to set the option</th>
<th>Command or access being audited</th>
</tr>
</thead>
<tbody>
<tr>
<td>update (object-specific)</td>
<td>all</td>
<td>Name specifying the object to be audited, default table or default view</td>
<td>The database of the object (except tempdb)</td>
<td>update to a table, update to a view</td>
</tr>
</tbody>
</table>

**Example**

```sql
sp_audit "update", "all", "projects", "fail"
```

(Audits all attempts by users to update the `projects` table in the current database.)

<table>
<thead>
<tr>
<th>view_access (user-specific)</th>
<th>login_name of the user to be audited</th>
<th>Any</th>
<th>select, delete, insert, or update to a view</th>
</tr>
</thead>
<tbody>
<tr>
<td>view_access</td>
<td>all</td>
<td>Any</td>
<td>select, delete, insert, or update to a view</td>
</tr>
</tbody>
</table>

**Example**

```sql
sp_audit "view_access", "joe", "all", "off"
```

(Turns off view auditing of user "joe").

### Examples of setting auditing options

Suppose you want to audit all failed deletions on the `projects` table in the `company_operations` database and for all new tables in the database. Use the object-specific delete option for the `projects` table and use default table for all future tables in the database. To set object-specific auditing options, you must be in the object's database before you execute `spAudit`:

```sql
sp_audit "security", "all", "all", "fail"
```

For this example, execute:

```sql
use company_operations

sp_audit "delete", "all", "projects", "fail"

sp_audit "delete", "all", "default table", "fail"
```

- `use company_operations`:
- `sp_audit "delete", "all", "projects", "fail"`:
- `sp_audit "delete", "all", "default table", "fail"`:
Hiding system stored procedure and command password parameters

When auditing is configured and enabled, and the sp_audit option 'cmdtext' is set, system stored procedure and command password parameters are replaced with a fixed length string of asterisks in the audit records contained in the audit logs.

For example, executing:

```
sp_password 'oldpassword', 'newpassword'
```

when auditing is enabled and sp_audit cmdtext is set, results in output similar to:

```
sp_password '******', '******'
```

This protects passwords from being seen by other with access to the audit log.

Determining current auditing settings

To determine the current auditing settings for a given option, use sp_displayaudit. The syntax is:

```
sp_displayaudit [procedure | object | login | database | global |
default_object | default_procedure [, name]]
```

For more information, see sp_displayaudit in the Reference Manual.

Adding user-specified records to the audit trail

sp_addauditrecord allows users to enter comments into the audit trail. The syntax is:

```
sp_addauditrecord [text] [, db_name] [, obj_name]
[ , owner_name] [, dbid] [, objid]
```

All the parameters are optional:

- **text** – is the text of the message that you want to add to the extrainfo audit table.
- **db_name** – is the name of the database referred to in the record, which is inserted into the dbname column of the current audit table.
- **obj_name** – is the name of the object referred to in the record, which is inserted into the objname column of the current audit table.
Setting global auditing options

- *owner_name* – is the owner of the object referred to in the record, which is inserted into the *objowner* column of the current audit table.

- *dbid* – is an integer value representing the database ID number of *db_name*, which is inserted into the *dbid* column of the current audit table. Do not place it in quotes.

- *objid* – is an integer value representing the object ID number of *obj_name*. Do not place it in quotes. *objid* is inserted into the *objid* column of the current audit table.

You can use *sp_addauditrecord* if:

- You have execute permission on *sp_addauditrecord*.
- The auditing configuration parameter was activated with *sp_configure*.
- The adhoc auditing option was enabled with *sp_audit*.

By default, only a system security officer and the database owner of *sybsecurity* can use *sp_addauditrecord*. Permission to execute it may be granted to other users.

Examples of adding user-defined audit records

The following example adds a record to the current audit table. The text portion is entered into the *extrainfo* column of the current audit table, “corporate” into the *dbname* column, “payroll” into the *objname* column, “dbo” into the *objowner* column, “10” into the *dbid* column, and “1004738270” into the *objid* column:

```sql
sp_addauditrecord "I gave A. Smith permission to view the payroll table in the corporate database. This permission was in effect from 3:10 to 3:30 pm on 9/22/92.", "corporate", "payroll", "dbo", 10, 1004738270
```

The following example inserts information only into the *extrainfo* and *dbname* columns of the current audit table:

```sql
sp_addauditrecord @text="I am disabling auditing briefly while we reconfigure the system", @db_name="corporate"
```
Querying the audit trail

To query the audit trail, use SQL to select and summarize the audit data. If you follow the procedures discussed in “Setting up audit trail management” on page 648, the audit data is automatically archived to one or more tables in another database. For example, assume that the audit data resides in a table called audit_data in the audit_db database. To select audit records for tasks performed by “bob” on July 5, 1993, execute:

```
use audit_db
go
select * from audit_data
    where loginname = "bob"
    and eventtime like "Jul 5% 93"
go
```

This command requests audit records for commands performed in the pubs2 database by users with the system security officer role active:

```
select * from audit_data
    where extrainfo like "%sso_role%"
    and dbname = "pubs2"
go
```

This command requests audit records for all table truncations (event 64):

```
select * from audit_data
    where event = 64
    go
```

To query the audit trail using the name of an audit event, use the audit_event_name function. For example, to request the the audit records for all database creation events, enter:

```
select * from audit_data where audit_event_name(event) = "Create Database"
go
```

Understanding the audit tables

The system audit tables can be accessed only by a system security officer, who can read the tables by executing SQL commands. The only commands that are allowed on the system audit tables are select and truncate.

Table 18-3 describes the columns in all audit tables.
Understanding the audit tables

Table 18-3: Columns in each audit table

<table>
<thead>
<tr>
<th>Column name</th>
<th>Datatype</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>event</td>
<td>smallint</td>
<td>Type of event being audited. See Table 18-5 on page 673.</td>
</tr>
<tr>
<td>eventmod</td>
<td>smallint</td>
<td>More information about the event being audited. Indicates whether or not the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>event in question passed permission checks. Possible values are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0 = no modifier for this event.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 1 = the event passed permission checking.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 2 = the event failed permission checking.</td>
</tr>
<tr>
<td>spid</td>
<td>smallint</td>
<td>ID of the process that caused the audit record to be written.</td>
</tr>
<tr>
<td>eventtime</td>
<td>datetime</td>
<td>Date and time that the audited event occurred.</td>
</tr>
<tr>
<td>sequence</td>
<td>smallint</td>
<td>Sequence number of the record within a single event. Some events require</td>
</tr>
<tr>
<td></td>
<td></td>
<td>more than one audit record.</td>
</tr>
<tr>
<td>suid</td>
<td>smallint</td>
<td>Server login ID of the user who performed the audited event.</td>
</tr>
<tr>
<td>dbid</td>
<td>int null</td>
<td>Database ID in which the audited event occurred, or in which the object,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>stored procedure, or trigger resides, depending on the type of event.</td>
</tr>
<tr>
<td>objid</td>
<td>int null</td>
<td>ID of the accessed object, stored procedure, or trigger.</td>
</tr>
<tr>
<td>xactid</td>
<td>binary(6)</td>
<td>ID of the transaction containing the audited event. For a multi-database</td>
</tr>
<tr>
<td></td>
<td>null</td>
<td>transaction, this is the transaction ID from the database where the transaction</td>
</tr>
<tr>
<td></td>
<td></td>
<td>originated.</td>
</tr>
<tr>
<td>loginname</td>
<td>varchar(30)</td>
<td>Login name corresponding to the suid.</td>
</tr>
<tr>
<td>dbname</td>
<td>varchar(30)</td>
<td>Database name corresponding to the dbid.</td>
</tr>
<tr>
<td>objname</td>
<td>varchar(30)</td>
<td>Object name corresponding to the objid.</td>
</tr>
<tr>
<td>objowner</td>
<td>varchar(30)</td>
<td>Name of the owner of objid.</td>
</tr>
<tr>
<td>extrainfo</td>
<td>varchar(255)</td>
<td>Additional information about the audited event. This column contains a</td>
</tr>
<tr>
<td></td>
<td>null</td>
<td>sequence of items separated by semicolons. For details, see “Reading the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>extrainfo column” on page 672.</td>
</tr>
<tr>
<td>nodeid</td>
<td>tinyint</td>
<td>Server nodeid in a cluster where the event occurred.</td>
</tr>
</tbody>
</table>

Reading the extrainfo column

The extrainfo column contains a sequence of data separated by semicolons. The data is organized in the following categories.

Table 18-4: Information in the extrainfo column

<table>
<thead>
<tr>
<th>Position</th>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Roles</td>
<td>A list of active roles, separated by blanks.</td>
</tr>
<tr>
<td>2</td>
<td>Keywords or Options</td>
<td>The name of the keyword or option that was used for the event. For example,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>for the alter table command, the add column or drop constraint options might</td>
</tr>
<tr>
<td></td>
<td></td>
<td>have been used. If multiple keywords or options are listed, they are separated by commas.</td>
</tr>
</tbody>
</table>
This example shows an `extrainfo` column entry for the event of changing an auditing configuration parameter.

```
  sso_role;suspend audit when device full;1;0;;ralph;
```

This entry indicates that a system security officer changed `suspend audit when device full` from 1 to 0. There is no “other information” for this entry. The sixth category indicates that the user “ralph” was operating with a proxy login. No principal name is provided.

The other fields in the audit record give other pertinent information. For example, the record contains the server user ID (`suid`) and the login name (`loginname`).

Table 18-5 lists the values that appear in the `event` column, arranged by `sp_audit` option. The “Information in extrainfo” column describes information that might appear in the `extrainfo` column of an audit table, based on the categories described in Table 18-4.

<table>
<thead>
<tr>
<th>Position</th>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Previous value</td>
<td>If the event resulted in the update of a value, this item contains the value prior to the update.</td>
</tr>
<tr>
<td>4</td>
<td>Current value</td>
<td>If the event resulted in the update of a value, this item contains the new value.</td>
</tr>
<tr>
<td>5</td>
<td>Other information</td>
<td>Additional security-relevant information that is recorded for the event.</td>
</tr>
<tr>
<td>6</td>
<td>Proxy information</td>
<td>The original login name if the event occurred while a <code>set proxy</code> was in effect.</td>
</tr>
<tr>
<td>7</td>
<td>Principal name</td>
<td>The principal name from the underlying security mechanism if the user’s login is the secure default login, and the user logged in to Adaptive Server via unified login. The value of this item is NULL if the secure default login is not being used.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Audit option</th>
<th>Command or access to be audited</th>
<th>event</th>
<th>Information in extrainfo</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Automatically audited event not controlled by an option)</td>
<td>Enabling auditing with <code>sp_configure auditing</code></td>
<td>73</td>
<td>—</td>
</tr>
<tr>
<td>(Automatically audited event not controlled by an option)</td>
<td>Disabling auditing with <code>sp_configure auditing</code></td>
<td>74</td>
<td>—</td>
</tr>
<tr>
<td>Unlocking Administrator’s account</td>
<td>Disabling auditing with <code>sp_configure auditing</code></td>
<td>74</td>
<td>—</td>
</tr>
<tr>
<td>adhoc</td>
<td>User-defined audit record</td>
<td>1</td>
<td>extrainfo is filled by the text parameter of <code>sp_addauditrecord</code></td>
</tr>
</tbody>
</table>
### Understanding the audit tables

<table>
<thead>
<tr>
<th>Audit option</th>
<th>Command or access to be audited</th>
<th>event</th>
<th>Information in extrainfo</th>
</tr>
</thead>
<tbody>
<tr>
<td>alter</td>
<td>alter database</td>
<td>2</td>
<td><strong>Subcommand keywords:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>alter maxhold</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>alter size</td>
</tr>
<tr>
<td></td>
<td>alter table</td>
<td>3</td>
<td><strong>Subcommand keywords:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>add/drop/modify column</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>replace columns</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>replace decrypt default</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>replace/add decrypt default</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>add constraint</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>drop constraint</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If one or more encrypted columns are added, extrainfo contains: add/drop/modify column column1/keyname1, [column2/keyname2] where keyname is the fully qualified name of the key.</td>
</tr>
<tr>
<td>bcp</td>
<td>bcp in</td>
<td>4</td>
<td>—</td>
</tr>
<tr>
<td>bind</td>
<td>sp_binddefault</td>
<td>6</td>
<td><strong>Other information:</strong> Name of the default</td>
</tr>
<tr>
<td></td>
<td>sp_bindmsg</td>
<td>7</td>
<td><strong>Other information:</strong> Message ID</td>
</tr>
<tr>
<td></td>
<td>sp_bindrule</td>
<td>8</td>
<td><strong>Other information:</strong> Name of the rule</td>
</tr>
<tr>
<td>cmdtext</td>
<td>All commands</td>
<td>92</td>
<td>Full text of command, as sent by the client</td>
</tr>
<tr>
<td>create</td>
<td>create database</td>
<td>9</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>create default</td>
<td>14</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>create procedure</td>
<td>11</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>create rule</td>
<td>13</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>create table</td>
<td>10</td>
<td>For encrypted columns, extrainfo contains column names and keynames. EK column1/keyname1, [column2/keyname2] where EK is a prefix indicating that subsequent information refers to encryption keys and keyname is the fully qualified name of the key.</td>
</tr>
<tr>
<td></td>
<td>create trigger</td>
<td>12</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>create view</td>
<td>16</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>create index</td>
<td>104</td>
<td><strong>Other information:</strong> Name of the index</td>
</tr>
<tr>
<td></td>
<td>create function</td>
<td>97</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>sp_addmessage</td>
<td>15</td>
<td><strong>Other information:</strong> Message number</td>
</tr>
<tr>
<td>Audit option</td>
<td>Command or access to be audited</td>
<td>event</td>
<td>Information in extra_info</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------------------</td>
<td>-------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>dbaccess</td>
<td>Any access to the database by any user</td>
<td>17</td>
<td><strong>Keywords or options:</strong> use cmd outside reference</td>
</tr>
<tr>
<td>dbcc</td>
<td>dbcc all keywords</td>
<td>81</td>
<td><strong>Keywords or options:</strong> Any of the dbcc keywords such as checkstorage and the options for that keyword.</td>
</tr>
<tr>
<td>delete</td>
<td>delete from a table</td>
<td>18</td>
<td><strong>Keywords or options:</strong> delete</td>
</tr>
<tr>
<td></td>
<td>delete from a view</td>
<td>19</td>
<td><strong>Keywords or options:</strong> delete</td>
</tr>
</tbody>
</table>
| disk         | disk init                        | 20    | **Keywords or options:** disk init  
|              |                                   |       | **Other information:** Name of the disk |
| disk mirror  |                                  | 23    | **Keywords or options:** disk mirror  
|              |                                   |       | **Other information:** Name of the disk |
| disk refit   |                                  | 21    | **Keywords or options:** disk refit  
|              |                                   |       | **Other information:** Name of the disk |
| disk reinit  |                                  | 22    | **Keywords or options:** disk reinit  
|              |                                   |       | **Other information:** Name of the disk |
| disk release |                                  | 87    | **Keywords or options:** disk release  
|              |                                   |       | **Other information:** Name of the disk |
| disk remirror|                                  | 25    | **Keywords or options:** disk remirror  
|              |                                   |       | **Other information:** Name of the disk |
| disk unmirror|                                  | 24    | **Keywords or options:** disk unmirror  
|              |                                   |       | **Other information:** Name of the disk |
| disk resize  |                                  | 100   | **Keywords or options:** disk resize  
|              |                                   |       | **Other information:** Name of the disk |
| drop         | drop database                    | 26    | —                          |
|              | drop default                     | 31    | —                          |
|              | drop procedure                   | 28    | —                          |
|              | drop table                       | 27    | —                          |
|              | drop trigger                     | 29    | —                          |
|              | drop rule                        | 30    | —                          |
|              | drop view                        | 33    | —                          |
|              | drop index                       | 105   | **Other information:** Index name |
|              | drop function                    | 98    | —                          |
|              | sp_dropmessage                   | 32    | **Other information:** Message number |
| dump         | dump database                    | 34    | —                          |
|              | dump transaction                 | 35    | —                          |
### Understanding the audit tables

<table>
<thead>
<tr>
<th>Audit option</th>
<th>Command or access to be audited</th>
<th>event</th>
<th>Information in extrainfo</th>
</tr>
</thead>
<tbody>
<tr>
<td>encryption_key</td>
<td>sp_encryption</td>
<td>106</td>
<td>If password is set the first time:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ENCR_ADMIN system_encr_passwd password **********</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If the password is subsequently changed:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ENCR_ADMIN system_encr_passwd password ********** **********</td>
</tr>
<tr>
<td>create encryption key</td>
<td></td>
<td>107</td>
<td>Keywords contain:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>algorithm name-bitlength/IV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[random</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>user/system</td>
</tr>
<tr>
<td>alter encryption key</td>
<td></td>
<td>108</td>
<td>default/not default</td>
</tr>
<tr>
<td>drop encryption key</td>
<td></td>
<td>109</td>
<td>AEK modify encryption</td>
</tr>
<tr>
<td>AEK modify encryption</td>
<td></td>
<td>118</td>
<td>modify encryption</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>with user passwd</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Note that keyvalue is displayed only for replication of alter encryption key modify encryption. For example, when user “stephen” modifies his key copy, the following information is saved:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>MODIFY ENCRYPTION for user stephen WITH USER PASSWD</td>
</tr>
<tr>
<td>AEK add encryption</td>
<td></td>
<td>119</td>
<td>add encryption for user user_name</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>for login association</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Note that keyvalue is displayed only for replication of alter encryption key add encryption.</td>
</tr>
<tr>
<td>alter encryption key drop encryption</td>
<td></td>
<td>120</td>
<td>drop encryption</td>
</tr>
<tr>
<td>alter encryption key modify owner</td>
<td></td>
<td>121</td>
<td>modify owner [new owner user_name]</td>
</tr>
</tbody>
</table>

**See the Encrypted Column User Guide.**
### Command or access to be audited

<table>
<thead>
<tr>
<th>Audit option</th>
<th>Event</th>
<th>Information in extrainfo</th>
</tr>
</thead>
<tbody>
<tr>
<td>alter encryption key recover key</td>
<td>122</td>
<td>recovery key [with key_value] with key_value is only used during replication of alter encryption key. See the Encrypted Column User Guide.</td>
</tr>
<tr>
<td>errorlog errorlog or errorlog_admin function</td>
<td>127</td>
<td>The parameters passed to errorlog_admin are logged to identify the subcommand: errorlog_admin (param1, param2,...).</td>
</tr>
<tr>
<td>errors Fatal error</td>
<td>36</td>
<td>Other information: Error number. Severity. State</td>
</tr>
<tr>
<td>errors Non-fatal error</td>
<td>37</td>
<td>Other information: Error number. Severity. State</td>
</tr>
<tr>
<td>exec_procedure Execution of a procedure</td>
<td>38</td>
<td>Other information: All input parameters</td>
</tr>
<tr>
<td>exec_trigger Execution of a trigger</td>
<td>39</td>
<td>—</td>
</tr>
<tr>
<td>func_obj_access, func_dbaccess Accesses to objects and databases via Transact-SQL functions. (Auditing must be enabled for the sa_role to audit functions.)</td>
<td>86</td>
<td>—</td>
</tr>
<tr>
<td>grant grant</td>
<td>40</td>
<td>—</td>
</tr>
</tbody>
</table>
| insert insert into a table | 41 | Keywords or option:  
  - If insert is used: insert  
  - If select into is used: insert into followed by the fully qualified object name |
| insert into a view | 42 | Keywords or options: insert |
| install install | 93 | — |
| load load database | 43 | — |
| load transaction | 44 | — |
| login Any login to the server | 45 | Other information:  
  - Host name and IP address of the machine from which the login was performed.  
| login_locked Login locked due to exceeding the configured number of failed login attempts | 112 | — |
| logout Any logouts from the server | 46 | Other information: Host name |
| mount mount database | 101 | — |
| password sp_passwordpolicy and all its actions except list. | 115 | Parameters for sp_passwordpolicy |
### Understanding the audit tables

<table>
<thead>
<tr>
<th>Audit option</th>
<th>Command or access to be audited</th>
<th>event</th>
<th>Information in extrainfo</th>
</tr>
</thead>
<tbody>
<tr>
<td>quiesce</td>
<td>quiesce database</td>
<td>96</td>
<td>—</td>
</tr>
</tbody>
</table>
| reference    | Creation of references to tables | 91    | Keywords or options: reference
Other information: Name of the referencing table |
| remove       | remove java                     | 94    | —                        |
| revoke       | revoke                          | 47    | —                        |
| rpc          | Remote procedure call from another server | 48    | Keywords or options: Name of client program
Other information: Server name, host name of the machine from which the RPC was executed. |
| security     | connect to (CIS only)           | 90    | Keywords or options: connect to |
| proc_role    | proc_role function (executed from within a system procedure) | 83    | —                        |
|             | Regeneration of a password by an sso | 80    | Other information: Required roles |
|             | Role toggling                   | 55    | Previous value: on or off
Current value: on or off
Other information: Name of the role being set |
|             | Server start                    | 50    | Other information: |
- dmastervideovname
- interfaces file path
- Sservername
- errorfilename |
|             | sp_webservices                  | 111   | Keywords or options: deploy if deploying a web service. deploy_all if deploying all web services |
|             | sp_webservices                  | 111   | Keywords or options: undeploy if undeploying a web service. undeploy_all if undeploying all web services |
|             | Server shutdown                 | 51    | Keywords or options: shutdown |
|             | set proxy or set session authorization | 88    | Previous value: Previous suid
Current value: New suid |
<table>
<thead>
<tr>
<th>Audit option</th>
<th>Command or access to be audited</th>
<th>event</th>
<th>Information in extrainfo</th>
</tr>
</thead>
</table>
| sp_configure |                                | 82    | Keywords or options: SETCONFIG  
Other information:  
• If a parameter is being set: number of configuration parameter  
• If a configuration file is being used to set parameters: name of the configuration file |
| sp_ssladmin enabled |                      | 99    | Keywords contains SSL_ADMIN addcert, if adding a certification. |
| Audit table access |                               | 61    | —                       |
| create login, drop login |                          | 103   | Keywords or options: create login, drop login |
| create, drop, alter, grant, or revoke role |                           | 85    | Keywords or options: create, drop, alter, grant, or revoke role |
| built-in functions |                               | 86    | Keywords or options: Name of function |
| Security command or access to be audited, specifically, starting Adaptive Server with -u option to unlock the administrator’s account | | 95    | Other information contains 'Unlocking admin account' |
| Changes to the LDAP state changes |                         | 123   | Keywords or options: Primary URL state and secondary URL state  
• Previous value  
• Current value  
Additional information indicates whether the state change happened automatically or because of a manually entered command. |
| The regeneration of asymmetric keypairs for network password encryption by the system or sp_passwordpolicy | | 117   | Information in extrainfo |
| select | select from a table | 62    | Keywords or options: select into select readtext |
| select from a view |                       | 63    | Keywords or options: select into select readtext |
| setuser | setuser             | 84    | Other information: Name of the user being set |
### Understanding the audit tables

<table>
<thead>
<tr>
<th>Audit option</th>
<th>Command or access to be audited</th>
<th>event</th>
<th>Information in extrainfo</th>
</tr>
</thead>
<tbody>
<tr>
<td>table_access</td>
<td>delete</td>
<td>18</td>
<td><strong>Keywords or options:</strong> delete</td>
</tr>
<tr>
<td></td>
<td>insert</td>
<td>41</td>
<td><strong>Keywords or options:</strong> insert</td>
</tr>
<tr>
<td></td>
<td>select</td>
<td>62</td>
<td><strong>Keywords or options:</strong> select into select readtext</td>
</tr>
<tr>
<td></td>
<td>update</td>
<td>70</td>
<td><strong>Keywords or options:</strong> update writetext</td>
</tr>
<tr>
<td>truncate</td>
<td>truncate table</td>
<td>64</td>
<td>—</td>
</tr>
<tr>
<td>unbind</td>
<td>sp_unbindefault</td>
<td>67</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>sp_unbindmsg</td>
<td>69</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>sp_unbindrule</td>
<td>68</td>
<td>—</td>
</tr>
<tr>
<td>unmount</td>
<td>unmount database</td>
<td>102</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>create manifest file</td>
<td>116</td>
<td><strong>Information in extrainfo</strong></td>
</tr>
<tr>
<td>update</td>
<td>update to a table</td>
<td>70</td>
<td><strong>Keywords or options:</strong> update writetext</td>
</tr>
<tr>
<td></td>
<td>update to a view</td>
<td>71</td>
<td><strong>Keywords or options:</strong> update writetext</td>
</tr>
<tr>
<td>view_access</td>
<td>delete</td>
<td>19</td>
<td><strong>Keywords or options:</strong> delete</td>
</tr>
<tr>
<td></td>
<td>insert</td>
<td>42</td>
<td><strong>Keywords or options:</strong> insert</td>
</tr>
<tr>
<td></td>
<td>select</td>
<td>63</td>
<td><strong>Keywords or options:</strong> select into select readtext</td>
</tr>
<tr>
<td></td>
<td>update</td>
<td>71</td>
<td><strong>Keywords or options:</strong> update writetext</td>
</tr>
</tbody>
</table>

Table 18-6 lists the values that appear in the event column, arranged by the audit event.

### Table 18-6: Audit event values

<table>
<thead>
<tr>
<th>Audit event ID</th>
<th>Command name</th>
<th>Audit event ID</th>
<th>Command name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ad hoc audit record</td>
<td>62</td>
<td>select table</td>
</tr>
<tr>
<td>2</td>
<td>alter database</td>
<td>63</td>
<td>select view</td>
</tr>
<tr>
<td>Audit event ID</td>
<td>Command name</td>
<td>Audit event ID</td>
<td>Command name</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------------</td>
<td>---------------</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td>3</td>
<td>alter table</td>
<td>64</td>
<td>truncate table</td>
</tr>
<tr>
<td>4</td>
<td>bcp in</td>
<td>65</td>
<td>Reserved</td>
</tr>
<tr>
<td>5</td>
<td>Reserved</td>
<td>66</td>
<td>Reserved</td>
</tr>
<tr>
<td>6</td>
<td>bind default</td>
<td>67</td>
<td>unbind default</td>
</tr>
<tr>
<td>7</td>
<td>bind message</td>
<td>68</td>
<td>unbind rule</td>
</tr>
<tr>
<td>8</td>
<td>bind rule</td>
<td>69</td>
<td>unbind message</td>
</tr>
<tr>
<td>9</td>
<td>create database</td>
<td>70</td>
<td>update table</td>
</tr>
<tr>
<td>10</td>
<td>create table</td>
<td>71</td>
<td>update view</td>
</tr>
<tr>
<td>11</td>
<td>create procedure</td>
<td>72</td>
<td>Reserved</td>
</tr>
<tr>
<td>12</td>
<td>create trigger</td>
<td>73</td>
<td>auditing enabled</td>
</tr>
<tr>
<td>13</td>
<td>create rule</td>
<td>74</td>
<td>auditing disabled</td>
</tr>
<tr>
<td>14</td>
<td>create default</td>
<td>75</td>
<td>Reserved</td>
</tr>
<tr>
<td>15</td>
<td>create message</td>
<td>76</td>
<td>SSO changed password</td>
</tr>
<tr>
<td>16</td>
<td>create view</td>
<td>77</td>
<td>Reserved</td>
</tr>
<tr>
<td>17</td>
<td>access to database</td>
<td>78</td>
<td>Reserved</td>
</tr>
<tr>
<td>18</td>
<td>delete table</td>
<td>79</td>
<td>Reserved</td>
</tr>
<tr>
<td>19</td>
<td>delete view</td>
<td>80</td>
<td>role check performed</td>
</tr>
<tr>
<td>20</td>
<td>disk init</td>
<td>81</td>
<td>dbcc</td>
</tr>
<tr>
<td>21</td>
<td>disk refit</td>
<td>82</td>
<td>config</td>
</tr>
<tr>
<td>22</td>
<td>disk reinit</td>
<td>83</td>
<td>online database</td>
</tr>
<tr>
<td>23</td>
<td>disk mirror</td>
<td>84</td>
<td>setuser command</td>
</tr>
<tr>
<td>24</td>
<td>disk unmirror</td>
<td>85</td>
<td>UDR command</td>
</tr>
<tr>
<td>25</td>
<td>disk remirror</td>
<td>86</td>
<td>built-in function</td>
</tr>
<tr>
<td>26</td>
<td>drop database</td>
<td>87</td>
<td>Disk release</td>
</tr>
<tr>
<td>27</td>
<td>drop table</td>
<td>88</td>
<td>set SSA command</td>
</tr>
<tr>
<td>28</td>
<td>drop procedure</td>
<td>89</td>
<td>kill or terminate command</td>
</tr>
<tr>
<td>29</td>
<td>drop trigger</td>
<td>90</td>
<td>connect</td>
</tr>
<tr>
<td>30</td>
<td>drop rule</td>
<td>91</td>
<td>reference</td>
</tr>
<tr>
<td>31</td>
<td>drop default</td>
<td>92</td>
<td>command text</td>
</tr>
<tr>
<td>32</td>
<td>drop message</td>
<td>93</td>
<td>JCS install command</td>
</tr>
<tr>
<td>33</td>
<td>drop view</td>
<td>94</td>
<td>JCS remove command</td>
</tr>
<tr>
<td>34</td>
<td>dump database</td>
<td>95</td>
<td>Unlock admin account</td>
</tr>
<tr>
<td>35</td>
<td>dump transaction</td>
<td>96</td>
<td>quiesce database</td>
</tr>
<tr>
<td>36</td>
<td>Fatal error</td>
<td>97</td>
<td>create SQLJ function</td>
</tr>
<tr>
<td>37</td>
<td>Non-fatal error</td>
<td>98</td>
<td>drop SQLJ function</td>
</tr>
<tr>
<td>38</td>
<td>execution of stored procedure</td>
<td>99</td>
<td>SSL administration</td>
</tr>
</tbody>
</table>
### Understanding the audit tables

<table>
<thead>
<tr>
<th>Audit event ID</th>
<th>Command name</th>
<th>Audit event ID</th>
<th>Command name</th>
</tr>
</thead>
<tbody>
<tr>
<td>39</td>
<td>Execution of trigger</td>
<td>100</td>
<td>disk resize</td>
</tr>
<tr>
<td>40</td>
<td>grant</td>
<td>101</td>
<td>mount database</td>
</tr>
<tr>
<td>41</td>
<td>insert table</td>
<td>102</td>
<td>unmount database</td>
</tr>
<tr>
<td>42</td>
<td>insert view</td>
<td>103</td>
<td>login command</td>
</tr>
<tr>
<td>43</td>
<td>load database</td>
<td>104</td>
<td>create index</td>
</tr>
<tr>
<td>44</td>
<td>load transaction</td>
<td>105</td>
<td>drop index</td>
</tr>
<tr>
<td>45</td>
<td>login</td>
<td>106</td>
<td>sp_encryption (encrypted column administration)</td>
</tr>
<tr>
<td>46</td>
<td>logout</td>
<td>107</td>
<td>create encryption key</td>
</tr>
<tr>
<td>47</td>
<td>revoke</td>
<td>108</td>
<td>Alter Encryption Key as/not default</td>
</tr>
<tr>
<td>48</td>
<td>rpc in</td>
<td>109</td>
<td>drop encryption key</td>
</tr>
<tr>
<td>49</td>
<td>rpc out</td>
<td>110</td>
<td>deploy user-defined web services</td>
</tr>
<tr>
<td></td>
<td></td>
<td>111</td>
<td>undeploy user defined web services</td>
</tr>
<tr>
<td>50</td>
<td>server boot</td>
<td>112</td>
<td>login has been locked</td>
</tr>
<tr>
<td>51</td>
<td>server shutdown</td>
<td>113</td>
<td>quiesce hold security</td>
</tr>
<tr>
<td>52</td>
<td>Reserved</td>
<td>114</td>
<td>quiesce release</td>
</tr>
<tr>
<td>53</td>
<td>Reserved</td>
<td>115</td>
<td>Password administration</td>
</tr>
<tr>
<td>54</td>
<td>Reserved</td>
<td>116</td>
<td>create manifest file</td>
</tr>
<tr>
<td>55</td>
<td>role toggling</td>
<td>117</td>
<td>regenerate keypair</td>
</tr>
<tr>
<td>56</td>
<td>Reserved</td>
<td>118</td>
<td>alter encryption key modify encryption</td>
</tr>
<tr>
<td>57</td>
<td>Reserved</td>
<td>119</td>
<td>alter encryption key add encryption</td>
</tr>
<tr>
<td>58</td>
<td>Reserved</td>
<td>120</td>
<td>alter encryption key drop encryption</td>
</tr>
<tr>
<td>59</td>
<td>Reserved</td>
<td>121</td>
<td>alter encryption key modify owner</td>
</tr>
<tr>
<td>60</td>
<td>Reserved</td>
<td>122</td>
<td>alter encryption key for key recovery</td>
</tr>
<tr>
<td>61</td>
<td>access to audit table</td>
<td>123</td>
<td>LDAP state changes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>127</td>
<td>Errorlog administration</td>
</tr>
</tbody>
</table>
Monitoring failed login attempts

The audit option `login_locked` and the event `Locked Login` (value 112) record when a login account is locked due to exceeding the configured number of failed login attempts. This event is enabled when audit option `login_locked` is set. To set `login_locked`, enter:

```
sp_audit "login_locked","all","all","ON"
```

If the audit tables are full and the event cannot be logged, a message with the information is sent to the errorlog.

The hostname and network IP address are included in the audit record. Monitoring the audit logs for the `Locked Login` event (112) helps to identify attacks on login accounts.

Auditing login failures

Although client applications may fail to login for many reasons, Adaptive Server does not provide them with any detailed information about the login failure. This is done to avoid giving information to malintentioned users attempting to crack passwords or otherwise breach Adaptive Server’s authentication mechanisms.

However, as a system administrator, detailed information is useful for diagnosing Adaptive Server administrative or configuration problems, and it is useful to security officers for investigating attempts to breach security.

This enables auditing for all login failures:

```
sp_audit "login", "all", "all", "fail"
```

In order to provide a barrier to inappropriate use of the information, only a user granted the SSO role can access the audit trail information containing this sensitive information.

Adaptive Server audits login failures for the following conditions:

- For Adaptive Server started as a Windows Service, if the Sybase SQLServer service is paused (for example, by the Microsoft Management Console for Services).
- If a remote server attempts to establish a site handler for server-to-server RPCs, but insufficient resources (or any of the other conditions listed here) cause the site handler initialization to fail.
Understanding the audit tables

- Using Adaptive Server for Windows with the Trusted Login or Unified Login configuration, but the specified user is not a trusted administrator (that is, an authentication failure).
- Adaptive Server does not support the SQL interface requested by the client.
- A user is attempting to log into Adaptive Server when it is in single-user mode. In single-user mode, exactly one user with the sa_role is allowed to log in to Adaptive Server. Additional logins are prevented, even if they have the sa_role.
- The syslogins table in the master database fails to open, indicating the master database has an internal error.
- A client attempts a remote login, but sysremotelogins cannot be opened, or there is no entry for the specified user account and no guest account exists.
- A client attempts a remote login and, although it finds an entry referring to a local account for the specified user in sysremotelogins, the referenced local account does not exist.
- A client program requests a security session (for example, a Kerberos authentication), but the security session could not be established because:
  - The Adaptive Server security subsystem was not initialized at startup.
  - Insufficient memory resources for allocated structures.
  - The authentication negotiation failed.
- An authentication mechanism is not found for the specified user.
- The specified password was not correct.
- syslogins does not contain the required entry for the specified login.
- The login account is locked.
- Adaptive Server has reached its limit for the number of user connections.
- The configuration parameter unified login required is set, but the login has not been authenticated by the appropriate security subsystem.
- Adaptive Server’s network buffers are unavailable, or the requested packet size is invalid.
- A client application requests a host-based communication socket connection, but memory resources for the host-based communication buffers are not available.
• A shutdown is in progress, but the specified user does not have the sa role.

• Adaptive Server could not open the default database for a login, and this login does not have access to the master database.

• A client makes a high availability login failover request, but the high availability subsystem is does not have a high availability session for this login, or the login is unable to wait for the failover to complete.

• A client requests a high availability login setup, but the high availability subsystem is unable to create the session or is unable to complete the TDS protocol negotiations for the high availability session.

• Adaptive Server fails to setup tempdb for a login.

• TDS Login Protocol errors are detected.
Understanding the audit tables
CHAPTER 19
Confidentiality of Data

This chapter describes how to configure Adaptive Server to ensure that all data is secure and confidential.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Secure Sockets Layer (SSL) in Adaptive Server</td>
<td>687</td>
</tr>
<tr>
<td>Kerberos confidentiality</td>
<td>707</td>
</tr>
<tr>
<td>Dumping and loading databases with password protection</td>
<td>707</td>
</tr>
</tbody>
</table>

Secure Sockets Layer (SSL) in Adaptive Server

Adaptive Server Enterprise security services now support Secure Sockets Layer (SSL) session-based security. SSL is the standard for securing the transmission of sensitive information, such as credit card numbers, stock trades, and banking transactions, over the Internet.

While a comprehensive discussion of public-key cryptography is beyond the scope of this document, the basics are worth describing so that you have an understanding of how SSL secures Internet communication channels. This document is not a comprehensive guide to public-key cryptography.

The implementation of Adaptive Server SSL features assume that there is a knowledgeable system security officer who is familiar with the security policies and needs of your site, and who has general understanding of SSL and public-key cryptography.
Internet communications overview

TCP/IP is the primary transport protocol used in client/server computing, and is the protocol that governs the transmission of data over the Internet. TCP/IP uses intermediate computers to transport data from sender to recipient. The intermediate computers introduce weak links to the communication system where data may be subjected to tampering, theft, eavesdropping, and impersonation.

Public-key cryptography

Several mechanisms, known collectively as public-key cryptography, have been developed and implemented to protect sensitive data during transmission over the Internet. Public-key cryptography consists of encryption, key exchange, digital signatures, and digital certificates.

Encryption

Encryption is a process wherein a cryptographic algorithm is used to encode information to safeguard it from anyone except the intended recipient. There are two types of keys used for encryption:

- **Symmetric-key encryption** – is where the same algorithm (key) is used to encrypt and decrypt the message. This form of encryption provides minimal security because the key is simple, and therefore easy to decipher. However, transfer of data that is encrypted with a symmetric key is fast because the computation required to encrypt and decrypt the message is minimal.

- **Public/private key encryption** – also known as asymmetric-key, is a pair of keys that are made up of public and private components to encrypt and decrypt messages. Typically, the message is encrypted by the sender with a private key, and decrypted by the recipient with the sender’s public key, although this may vary. You can use a recipient’s public key to encrypt a message, who then uses his private key to decrypt the message.

  The algorithms used to create public and private keys are more complex, and therefore harder to decipher. However, public/private key encryption requires more computation, sends more data over the connection, and noticeably slows data transfer.

Key exchange

The solution for reducing computation overhead and speeding transactions without sacrificing security is to use a combination of both symmetric key and public/private key encryption in what is known as a key exchange.
For large amounts of data, a symmetric key is used to encrypt the original message. The sender then uses either his private key or the recipient’s public key to encrypt the symmetric key. Both the encrypted message and the encrypted symmetric key are sent to the recipient. Depending on what key was used to encrypt the message (public or private) the recipient uses the opposite to decrypt the symmetric key. Once the key has been exchanged, the recipient uses the symmetric key to decrypt the message.

**Digital signatures** are used for tamper detection and non-repudiation. Digital signatures are created with a mathematical algorithm that generates a unique, fixed-length string of numbers from a text message; the result is called a hash or message digest.

To ensure message integrity, the message digest is encrypted by the signer’s private key, then sent to the recipient along with information about the hashing algorithm. The recipient decrypts the message with the signer’s public key. This process also regenerates the original message digest. If the digests match, the message proves to be intact and tamper free. If they do not match, the data has either been modified in transit, or the data was signed by an imposter.

Further, the digital signature provides **non-repudiation**—senders cannot deny, or repudiate, that they sent a message, because their private key encrypted the message. Obviously, if the private key has been compromised (stolen or deciphered), the digital signature is worthless for non-repudiation.

**Digital certificates** are like passports: once you have been assigned one, the authorities have all your identification information in the system. Like a passport, the certificate is used to verify the identity of one entity (server, router, Web sites, and so on) to another.

Adaptive Server uses two types of certificates:

- **Server certificates** – a server certificate authenticates the server that holds it. Certificates are issued by a trusted third-party Certificate Authority (CA). The CA validates the holder’s identity, and embeds the holder’s public key and other identification information into the digital certificate. Certificates also contain the digital signature of the issuing CA, verifying the integrity of the data contained therein and validating its use.

- **CA certificates** (also known as **trusted root certificates**) – is a list of trusted CAs loaded by the server at start-up. CA certificates are used by servers when they function as a client, such as during remote procedure calls (RPCs). Adaptive Server loads its CA trusted root certificate at start-up. When connecting to a remote server for RPCs, Adaptive Server verifies that the CA that signed the remote server’s certificate is a “trusted” CA listed in its own CA trusted roots file. If it is not, the connection fails.
Certificates are valid for a period of time and can be revoked by the CA for various reasons, such as when a security breach has occurred. If a certificate is revoked during a session, the session connection continues. Subsequent attempts to login fail. Likewise, when a certificate expires, login attempts fail.

The combination of these mechanisms protect data transmitted over the Internet from eavesdropping and tampering. These mechanisms also protect users from impersonation, where one entity pretends to be another (spoofing), or where a person or an organization says it is set up for a specific purpose when the real intent is to capture private information (misrepresentation).

SSL overview

SSL is an industry standard for sending wire- or socket-level encrypted data over secure network connections.

Before the SSL connection is established, the server and the client exchange a series of I/O round trips to negotiate and agree upon a secure encrypted session. This is called the SSL handshake.

SSL handshake

When a client requests a connection, the SSL-enabled server presents its certificate to prove its identity before data is transmitted. Essentially, the handshake consists of the following steps:

- The client sends a connection request to the server. The request includes the SSL (or Transport Layer Security, TLS) options that the client supports.
- The server returns its certificate and a list of supported cipher suites, which includes SSL/TLS support options, algorithms used for key exchange, and digital signatures.
- A secure, encrypted session is established when both client and server have agreed upon a CipherSuite.

For more specific information about the SSL handshake and the SSL/TLS protocol, see the Internet Engineering Task Force Web site at http://www.ietf.org.

For a list of cipher suites that Adaptive Server supports, see “Cipher Suites” on page 700.

SSL in Adaptive Server

Adaptive Server’s implementation of SSL provides several levels of security.
• The server authenticates itself—proves that it is the server you intended to contact—and an encrypted SSL session begins before any data is transmitted.

• Once the SSL session is established, the client requesting a connection can send his user name and password over the secure, encrypted connection.

• A comparison of the digital signature on the server certificate can determine whether the data received by the client was modified before reaching the intended recipient.

On most platforms, Adaptive Server uses SSL Plus(TM) library API from Certicom Corp. However, for Windows Opteron X64, Adaptive Server uses OpenSSL as the SSL provider.

SSL filter

The Adaptive Server directory service, such as the interfaces file, NT Registry, or LDAP service, defines the server address and port numbers, and determines the security protocols that are enforced for client connections. Adaptive Server implements the SSL protocol as a filter that is appended to the master and query lines of the directory services.

The addresses and port numbers on which Adaptive Server accepts connections are configurable, so you can enable multiple network and security protocols for a single server. Server connection attributes are specified with directory services, such as LDAP, or with the traditional Sybase interfaces file. See “Creating server directory entries” on page 696.

All connection attempts to a master or query entry in the interfaces file with an SSL filter must support the SSL protocol. A server can be configured to accept SSL connections and have other connections that accept clear text (unencrypted data), or use other security mechanisms.

For example, the interfaces file on UNIX that supports both SSL-based connections and clear-text connections looks like this:

```plaintext
SYBSRV1
  master tcp ether myhostname myport1 ssl
  query tcp ether myhostname myport1 ssl
  master tcp ether myhostname myport2
```

The SSL filter is different from other security mechanisms, such as DCE and Kerberos, which are defined with SECMECH (security mechanism) lines in the interfaces file (sql.ini on Windows).
Secure Sockets Layer (SSL) in Adaptive Server

Authentication via the certificate

The SSL protocol requires server authentication via a server certificate to enable an encrypted session. Likewise, when Adaptive Server is functioning as a client during RPCs, there must be a repository of trusted CAs that a client connection can access to validate the server certificate.

The server certificate

Each Adaptive Server must have its own server certificate file that is loaded at start-up. The following is the default location for the certificates file, where `servername` is the name of the Adaptive Server as specified on the command line during start-up with the `-s` flag, or from the environment variable `SDSLISTEN`:

- **UNIX**: `$SYBASE/$SYBASE_ASE/certificates/servername.crt`
- **NT**: `%SYBASE%\%SYBASE_ASE%\certificates\servername.crt`

The server certificate file consists of encoded data, including the server’s certificate and the encrypted private key for the server certificate.

Alternatively, you can specify the location of the server certificate file when using `sp_ssladmin`.

**Note** To make a successful client connection, the common name in the certificate must match the Adaptive Server name in the `interfaces` file.

The CA trusted roots certificate

The list of trusted CAs is loaded by Adaptive Server at start-up from the trusted roots file. The trusted roots file is similar in format to a certificate file, except that it contains certificates for CAs known to Adaptive Server. A trusted roots file is accessible by the local Adaptive Server in the following, where `servername` is the name of the server:

- **UNIX**: `$SYBASE/$SYBASE_ASE/certificates/servername.txt`
- **NT**: `%SYBASE%\%SYBASE_ASE%\certificates\servername.txt`

The trusted roots file is only used by Adaptive Server when it is functioning as a client, such as when performing RPC calls or Component Integration Services (CIS) connections.

The system security officer adds and deletes CAs that are to be accepted by Adaptive Server, using a standard ASCII-text editor.

**Warning!** Use the system security officer role (`sso_role`) within Adaptive Server to restrict access and execution on security-sensitive objects.
Adaptive Server provides tools to generate a certificate request and to authorize certificates. See “Using Adaptive Server tools to request and authorize certificates” on page 696.

Connection types

This section describes various client-to-server and server-to-server connections.

Client login to Adaptive Server

Open Client applications establish a socket connection to Adaptive Server similarly to the way that existing client connections are established. Before any user data is transmitted, an SSL handshake occurs on the socket when the network transport-level connect call completes on the client side and the accept call completes on the server side.

Server-to-server remote procedure calls

Adaptive Server establishes a socket connection to another server for RPCs in the same way that existing RPC connections are established. Before any user data is transmitted, an SSL handshake occurs on the socket when the network transport-level connect call completes. If the server-to-server socket connection has already been established, the existing socket connection and security context is reused.

When functioning as a client during RPCs, Adaptive Server requests the remote server’s certificate during connection. Adaptive Server then verifies that the CA that signed the remote server’s certificate is trusted; that is to say, on its own list of trusted CAs in the trusted roots file. It also verifies that the common name in the server certificate matches the common name used when establishing the connection.

Companion server and SSL

You can use a companion server to configure Adaptive Server for failover. You must configure both the primary and secondary servers with the same SSL and RPC configuration. When connections fail over or fail back, security sessions are reestablished with the connections.

Open Client connections

Component Integration Services, RepAgent, Distributed Transaction Management, and other modules in Adaptive Server use Client-Library to establish connections to servers other than Adaptive Server. The remote server is authenticated by its certificate. The remote server authenticates the Adaptive Server client connection for RPCs with user name and password.
Enabling SSL

Adaptive Server determines which security service it will use for a port based on the interface file (sql.ini on Windows).

❖ Enabling SSL

1. Generate a certificate for the server.
2. Create a trusted roots file.
3. Use sp_configure to enable SSL. From a command prompt, enter:
   ```sql
   sp_configure "enable ssl", 1
   ```
   - 1 – enables the SSL subsystem at start-up, allocates memory, and SSL performs wire-level encryption of data across the network.
   - 0 (the default) – disables SSL. This value is the default.
4. Add the SSL filter to the interfaces file. See “Creating server directory entries” on page 696.
5. Use sp_ssladmin to add a certificate to the certificates file. See “Administering certificates” on page 697.

Note To request, authorize, and convert third-party certificates, see the Utility Guide for information on the certauth, certreq, and certpk12 tools.

Unlike other security services, such as DCE, Kerberos, and NTLAN, SSL relies neither on the “Security” section of the Open Client/Open Server configuration file libtcl.cfg, nor on objects in objectid.dat.

The system administrator should consider memory use by SSL when planning for total physical memory. You need approximately 40K per connection (connections include user connections, remote servers, and network listeners) in Adaptive Server for SSL connections. The memory is reserved and preallocated within a memory pool and is used internally by Adaptive Server and SSL Plus libraries as requested.

Obtaining a certificate

The system security officer installs server certificates and private keys for Adaptive Server by:
• Using third-party tools provided with existing public-key infrastructure already deployed in the customer environment.

• Using the Adaptive Server certificate request tool in conjunction with a trusted third-party CA.

To obtain a certificate, you must request a certificate from a CA. If you request a certificate from a third party and that certificate is in PKCS #12 format, use the `certpk12` utility to convert the certificate into a format that is understood by Adaptive Server.

To test the Adaptive Server certificate request tool and to verify that the authentication methods are working on your server, Adaptive Server provides a tool, for testing purposes, that allows you to function as a CA and issue CA-signed certificate to yourself.

The main steps to creating a certificate for use with Adaptive Server are:

1. Generate the public and private key pair.
2. Securely store the private key.
3. Generate the certificate request.
4. Send the certificate request to the CA.
5. After the CA signs and returns the certificate, store it in a file and append the private key to the certificate.
6. Store the certificate in the Adaptive Server installation directory.

Most third-party PKI vendors and some browsers have utilities to generate certificates and private keys. These utilities are typically graphical wizards that prompt you through a series of questions to define a distinguished name and a common name for the certificate.

Follow the instructions provided by the wizard to create certificate requests. Once you receive the signed PKCS #12-format certificate, use `certpk12` to generate a certificate file and a private key file. Concatenate the two files into a `servername.crt` file, where `servername` is the name of the server, and place it in the `certificates` directory under `$SYBASE/$SYBASE_ASE`. See the `Utility Guide`. 
Adaptive Server provides two tools for requesting and authorizing certificates. `certreq` generates public and private key pairs and certificate requests. `certauth` converts a server certificate request to a CA-signed certificate.

Warning! Use `certauth` only for testing purposes. Sybase recommends that you use the services of a commercial CA because it provides protection for the integrity of the root certificate, and because a certificate that is signed by a widely accepted CA facilitates the migration to the use of client certificates for authentication.

Preparing the server’s trusted root certificate is a five-step process. Perform the first two steps to create a test trusted root certificate so you can verify that you are able to create server certificates. Once you have a test CA certificate (trusted roots certificate) repeat steps three through five to sign server certificates.

1. Use `certreq` to request a certificate.
2. Use `certauth` to convert the certificate request to a CA self-signed certificate (trusted root certificate).
3. Use `certreq` to request a server certificate and private key.
4. Use `certauth` to convert the certificate request to a CA-signed server certificate.
5. Append the private key text to the server certificate and store the certificate in the server’s installation directory.

For information about Sybase utilities, `certauth`, `certreq`, and `certpk12` for requesting, authorizing and converting third-party certificates, see the Utility Guide.

Note `certauth` and `certreq` are dependent on RSA and DSA algorithms. These tools only work with crypto modules that use RSA and DSA algorithms to construct the certificate request.

Creating server directory entries

Adaptive Server accepts client logins and server-to-server RPCs. The address and port numbers where Adaptive Server accepts connections are configurable so you can specify multiple networks, different protocols, and alternate ports.
In the *interfaces* file, SSL is specified as a filter on the master and query lines, whereas security mechanisms such as DCE or Kerberos are identified with a SECMECH line. The following example shows a TLI-based entry for an Adaptive Server using SSL in a UNIX environment:

An entry for an Adaptive Server with SSL and DCE security mechanisms on UNIX might look like:

```
SYBSRV1
  master tcp ether myhostname myport1 ssl
  query  tcp ether myhostname myport1 ssl
  master tcp ether myhostname myport2
  SECMECH 1.3.6.1.4.897.4.6.1
```

An entry for the server with SSL and Kerberos security mechanisms on NT might look like:

```
[SYBSRV2]
  query=nlwmsck, 18.52.86.120,2748,ssl
  master=nlwmsck 18.52.86.120,2748,ssl
  master=nlwmsck 18.52.86.120,2749
  secmech=1.3.6.1.4.897.4.6.6
```

The SECMECH lines for SYBSRV1 and SYBSRV2 in the examples contain an object identifier (OID) that refers to security mechanisms DCE and Kerberos, respectively. The OID values are defined in:

- UNIX – `$SYBASE/$SYBASE_OCS/config/objectid.dat`
- NT – `%SYBASE%\%SYBASE_OCS\ini\objectid.dat`

In these examples, the SSL security service is specified on port number 2748(0x0abc).

---

**Note** The use of SSL concurrently with a SECMECH security mechanism is intended to facilitate migration from SECMECHs to SSL security.

---

**Administering certificates**

To administer SSL and certificates in Adaptive Server, use `sp_ssladmin`. `sso_role` is required to execute the stored procedure.

`sp_ssladmin` is used to:

- Add local server certificates. You can add certificates and specify the password used to encrypt private keys, or require input of the password at the command line during start-up.
Secure Sockets Layer (SSL) in Adaptive Server

- Delete local server certificates.
- List server certificates.

The syntax for `sp_ssladmin` is:

```
sp_ssladmin {
    addcert, certificate_path [ , password|NULL ]
    [ , dropcert, certificate_path ]
    [ , lscert ]
    [ , help ]
    [ , lsciphers ]
    [ , setciphers, "FIPS" | "Strong" | "Weak" | "All"
        | quoted_list_of_ciphersuites ]
```

For example:

```
sp_ssladmin addcert, "/sybase/ASE-12_5/certificates/Server1.crt",
"mypassword"
```

This adds an entry for the local server, `Server1.crt`, in the certificates file in the absolute path to `/sybase/ASE-12_5/certificates` (\sybase\ASE-12_5\certificates on Windows). The private key is encrypted with the password "mypassword". The password should be the one specified when you created the private key.

Before accepting the certificate, `sp_ssladmin` verifies that:

- The private key can be decrypted using the provided password (except when NULL is specified).
- The private key and public key in the certificate match.
- The certificate chain, from root CA to the server certificate, is valid.
- The common name in the certificate matches the common name in the `interfaces` file.
If the common names do not match, `sp_ssladmin` issues a warning. If the other criteria fails, the certificate is not added to the certificates file.

---

**Warning!** Adaptive Server limits passwords to 64 characters. In addition, certain platforms restrict the length of valid passwords when creating server certificates. Select a password within these limits:

- Sun Solaris – both 32- and 64-bit platforms, 256 characters.
- Linux – 128 characters.
- IBM – both 32- and 64-bit platforms, 32 characters.
- HP – both 32- and 64-bit platforms, 8 characters.
- Windows NT – 256 characters.

The use of NULL as the password is intended to protect passwords during the initial configuration of SSL, before the SSL-encrypted session begins. Since you have not yet configured SSL, the password travels unencrypted over the connection. You can avoid this by specifying the password as NULL during the first login.

When NULL is the password, you must start `dataserver` with a `-y` flag, which prompts the administrator for the private-key password at the command line.

After restarting Adaptive Server with an SSL connection established, use `sp_ssladmin` again, this time using the actual password. The password is then encrypted and stored by Adaptive Server. Any subsequent starts of Adaptive Server from the command line use the encrypted password; you do not have to specify the password on the command line during start-up.

An alternative to using a NULL password during the first login is to avoid a remote connection to Adaptive Server via `isql`. You can specify “localhost” as the `hostname` in the `interfaces` file (`sql.ini` on Windows) to prevent clients from connecting remotely. Only a local connection can be established, and the password is never transmitted over a network connection.

---

**Note** Adaptive Server has sufficient memory in its network memory pool to allow `sp_ssladmin addcert` to set the certificate and private key password with its default memory allocations. However, if another network memory consumer has already allocated the default network memory, `sp_ssladmin` may fail and display this error to the client:

```
Msg 12823, Level 16, State 1:
Server 'servername', Procedure 'sp_ssladmin', Line 72:
```

---
Command 'addcert' failed to add certificate path
/work/REL125/ASE-12_5/certificates/servername.crt,
  system error: ErrMemory.
  (return status = 1)

Or the following message may appear in the error log:

    ... ssl_alloc: Cannot allocate using
    ubfalloc(rnetmempool, 131072)

As a workaround, you can increase the additional network memory
configuration parameter. Adaptive Server needs about 500K bytes of memory
for sp_ssladmin addcert to succeed, so increasing additional network memory
by this amount may allow it to succeed. This memory is reused by the network
memory pool when needed, or you can return additional network memory to its
previous value after sp_ssladmin has successfully completed.

Performance

There is additional overhead required to establish a secure session, because
data increases in size when it is encrypted, and it requires additional
computation to encrypt or decrypt information. The additional memory
requirements for SSL increases the overhead by 50-60 percent for network
throughput or for establishing a connection. You must have approximately 40K
more memory for each user connection.

Cipher Suites

During the SSL handshake, the client and server negotiate a common security
protocol via a CipherSuite. Cipher Suites are preferential lists of
key-exchange algorithms, hashing methods, and encryption methods used by
SSL-enabled applications. For a complete description of Cipher Suites, visit
the Internet Engineering Task Force (IETF) organization at

By default, the strongest CipherSuite supported by both the client and the
server is the CipherSuite that is used for the SSL-based session.
Adaptive Server supports the Cipher Suites that are available with the SSL Plus library API and the cryptographic engine, Security Builder™, both from Certicom Corp.

Note The Cipher Suites listed conform to the Transport Layer Specification (TLS). TLS is an enhanced version of SSL 3.0, and is an alias for the SSL version 3.0 Cipher Suites.

@@ssl_ciphersuite

The Transact-SQL® global variable @@ssl_ciphersuite allows users to know which cipher suite was chosen by the SSL handshake and verify that an SSL or a non-SSL connection was established.

Adaptive Server sets @@ssl_ciphersuite when the SSL handshake completes. The value is either NULL, indicating a non-SSL connection, or a string containing the name of the cipher suite chosen by the SSL handshake.

For example, an `isql` connection using SSL protocol displays the cipher suite chosen for it.

```
1> select @@ssl_ciphersuite
2> go
```

Output:

```
-----------------------------
TLS_RSA_WITH_AES_128_CBC_SHA
(1 row affected)
```

Setting SSL cipher suite preferences

In Adaptive Server, `sp_ssladmin` has two command options to display and set cipher suite preferences: `lsciphers` and `setciphers`. With these options, the set of cipher suites that Adaptive Server uses can be restricted, giving control to the system security officer over the kinds of encryption algorithms that may be used by client connections to the server or outbound connections from Adaptive Server. The default behavior for use of SSL cipher suites in Adaptive Server is the same as in earlier versions; it uses an internally defined set of preferences for cipher suites.

To display the values for any set cipher suite preferences, enter:
Secure Sockets Layer (SSL) in Adaptive Server

To set a specific cipher suite preference, enter:

```
sp_ssladmin setciphers, {"FIPS" | "Strong" | "Weak" | "All" | quoted_list_of_ciphersuites }
```

where:

- "FIPS" – is the set of encryptions, hash, and key exchange algorithms that are FIPS-compliant. The algorithms included in this list are AES, 3DES, DES, and SHA1.
- "Strong" – is the set of encryption algorithms using keys longer than 64 bits.
- "Weak" – is the set of encryption algorithms from the set of all supported cipher suites that are not included in the strong set.
- "All" – is the set of default cipher suites.
- quoted_list_of_ciphersuites – specifies a set of cipher suites as a comma-separated list, ordered by preference. Use quotes ("") to mark the beginning and end of the list. The quoted list can include any of the predefined sets as well as individual cipher suite names. Unknown cipher suite names cause an error to be reported, and no changes are made to preferences.

The detailed contents of the predefined sets are in Table 19-1 on page 703.

This restricts the available SSL cipher suites to the specified set of "FIPS", "Strong", "Weak", "All", or a quoted list of cipher suites. This takes effect on the next listener started, and requires that you restart Adaptive Server to ensure that all listeners use the new settings.

You can display any cipher suite preferences that have been set using `sp_ssladmin lsciphers`. If no preferences have been set, `sp_ssladmin lsciphers` returns 0 rows to indicate no preferences are set and Adaptive Server uses its default (internal) preferences.
Table 19-1: Predefined cipher suites in Adaptive Server

<table>
<thead>
<tr>
<th>Set name</th>
<th>Cipher suite names included in the set</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIPS</td>
<td>TLS_RSA_WITH_AES_256_CBC_SHA&lt;br&gt;</td>
</tr>
<tr>
<td></td>
<td>TLS_RSA_WITH_AES_128_CBC_SHA&lt;br&gt;</td>
</tr>
<tr>
<td></td>
<td>TLS_RSA_WITH_3DES_EDE_CBC_SHA&lt;br&gt;</td>
</tr>
<tr>
<td></td>
<td>TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA&lt;br&gt;</td>
</tr>
<tr>
<td></td>
<td>TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA&lt;br&gt;</td>
</tr>
<tr>
<td></td>
<td>TLS_RSA_WITH_DES_CBC_SHA&lt;br&gt;</td>
</tr>
<tr>
<td></td>
<td>TLS_DHE_DSS_WITH_DES_CBC_SHA&lt;br&gt;</td>
</tr>
<tr>
<td></td>
<td>TLS_DHE_RSA_WITH_DES_CBC_SHA&lt;br&gt;</td>
</tr>
<tr>
<td></td>
<td>TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA&lt;br&gt;</td>
</tr>
<tr>
<td></td>
<td>TLS_DHE_DSS_EXPORT1024_WITH_DES_CBC_SHA</td>
</tr>
<tr>
<td>Strong</td>
<td>TLS_RSA_WITH_AES_256_CBC_SHA&lt;br&gt;</td>
</tr>
<tr>
<td></td>
<td>TLS_RSA_WITH_AES_128_CBC_SHA&lt;br&gt;</td>
</tr>
<tr>
<td></td>
<td>TLS_RSA_WITH_3DES_EDE_CBC_SHA&lt;br&gt;</td>
</tr>
<tr>
<td></td>
<td>TLS_RSA_WITH_RC4_128_SHA&lt;br&gt;</td>
</tr>
<tr>
<td></td>
<td>TLS_RSA_WITH_RC4_128_MD5&lt;br&gt;</td>
</tr>
<tr>
<td></td>
<td>TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA&lt;br&gt;</td>
</tr>
<tr>
<td></td>
<td>TLS_DHE_DSS_WITH_RC4_128_SHA&lt;br&gt;</td>
</tr>
<tr>
<td></td>
<td>TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA&lt;br&gt;</td>
</tr>
<tr>
<td>Weak</td>
<td>TLS_RSA_WITH_DES_CBC_SHA&lt;br&gt;</td>
</tr>
<tr>
<td></td>
<td>TLS_DHE_DSS_WITH_DES_CBC_SHA&lt;br&gt;</td>
</tr>
<tr>
<td></td>
<td>TLS_DHE_RSA_WITH_DES_CBC_SHA&lt;br&gt;</td>
</tr>
<tr>
<td></td>
<td>TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA&lt;br&gt;</td>
</tr>
<tr>
<td></td>
<td>TLS_RSA_EXPORT1024_WITH_RC4_56_SHA&lt;br&gt;</td>
</tr>
<tr>
<td></td>
<td>TLS_DHE_DSS_EXPORT1024_WITH_RC4_56_SHA</td>
</tr>
<tr>
<td></td>
<td>TLS_DHE_DSS_EXPORT1024_WITH_DES_CBC_SHA</td>
</tr>
<tr>
<td></td>
<td>TLS_RSA_EXPORT_WITH_RC4_40_MD5&lt;br&gt;</td>
</tr>
<tr>
<td></td>
<td>TLS_RSA_EXPORT_WITH_DES40_CBC_SHA&lt;br&gt;</td>
</tr>
<tr>
<td></td>
<td>TLS_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA</td>
</tr>
<tr>
<td></td>
<td>TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA</td>
</tr>
</tbody>
</table>
Table 19-2 describes Cipher suites no longer supported for Adaptive Server 15.0 and later. 15.0. Attempts to use any dropped cipher suite results in an SSLHandshake failure and a failure to connect to Adaptive Server.

<table>
<thead>
<tr>
<th>Set name</th>
<th>Cipher suite names included in the set</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>TLS_RSA_WITH_AES_256_CBC_SHA</td>
</tr>
<tr>
<td></td>
<td>TLS_RSA_WITH_AES_128_CBC_SHA</td>
</tr>
<tr>
<td></td>
<td>TLS_RSA_WITH_3DES_EDE_CBC_SHA</td>
</tr>
<tr>
<td></td>
<td>TLS_RSA_WITH_RC4_128_SHA</td>
</tr>
<tr>
<td></td>
<td>TLS_RSA_WITH_RC4_128_MD5</td>
</tr>
<tr>
<td></td>
<td>TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA</td>
</tr>
<tr>
<td></td>
<td>TLS_DHE_DSS_WITH_RC4_128_SHA</td>
</tr>
<tr>
<td></td>
<td>TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA</td>
</tr>
<tr>
<td></td>
<td>TLS_RSA_WITH_DES_CBC_SHA</td>
</tr>
<tr>
<td></td>
<td>TLS_DHE_DSS_WITH_DES_CBC_SHA</td>
</tr>
<tr>
<td></td>
<td>TLS_DHE_RSA_WITH_DES_CBC_SHA</td>
</tr>
<tr>
<td></td>
<td>TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA</td>
</tr>
<tr>
<td></td>
<td>TLS_RSA_EXPORT1024_WITH_RC4_56_SHA</td>
</tr>
<tr>
<td></td>
<td>TLS_DHE_DSS_EXPORT1024_WITH_RC4_56_SHA</td>
</tr>
<tr>
<td></td>
<td>TLS_DHE_DSS_EXPORT1024_WITH_DES_CBC_SHA</td>
</tr>
<tr>
<td></td>
<td>TLS_RSA_EXPORT_WITH_RC4_40_MD5</td>
</tr>
<tr>
<td></td>
<td>TLS_RSA_EXPORT_WITH_DES40_CBC_SHA</td>
</tr>
<tr>
<td></td>
<td>TLS_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA</td>
</tr>
<tr>
<td></td>
<td>TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA</td>
</tr>
</tbody>
</table>
### Table 19-2: Dropped Cipher suites

<table>
<thead>
<tr>
<th>Set name</th>
<th>Cipher suite names dropped from the set</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIPS</td>
<td>TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA</td>
</tr>
<tr>
<td></td>
<td>TLS_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA</td>
</tr>
<tr>
<td></td>
<td>TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA</td>
</tr>
<tr>
<td>Strong</td>
<td>None dropped</td>
</tr>
<tr>
<td>Weak</td>
<td>TLS_RSA_EXPORT1024_WITH_RC4_56_SHA</td>
</tr>
<tr>
<td></td>
<td>TLS_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA</td>
</tr>
<tr>
<td></td>
<td>TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA</td>
</tr>
<tr>
<td>Others</td>
<td>TLS_DH_anon_EXPORT_WITH_DES40_CBC_SHA</td>
</tr>
<tr>
<td></td>
<td>TLS_DH_anon_EXPORT_WITH_RC4_40_MD5</td>
</tr>
<tr>
<td></td>
<td>TLS_DH_anon_WITH_3DES_EDE_CBC_SHA</td>
</tr>
<tr>
<td></td>
<td>TLS_DH_anon_WITH_DES_CBC_SHA</td>
</tr>
<tr>
<td></td>
<td>TLS_DH_anon_WITH_RC4_128_MD5</td>
</tr>
<tr>
<td></td>
<td>TLS_RSA_WITH_NULL_MD5</td>
</tr>
<tr>
<td></td>
<td>TLS_RSA_WITH_NULL_SHA</td>
</tr>
</tbody>
</table>

### Examples `sp_ssladmin`

On initial startup, before any cipher suite preferences have been set, no preferences are shown by `sp_ssladmin lscipher`.

```
1> sp_ssladmin lscipher
2> go
```

Output:

```
Cipher Suite Name   Preference
-------------------- -------------------
(0 rows affected)   (return status = 0)
```

The following example specifies the set of cipher suites that use FIPS algorithms.

```
1> sp_ssladmin setcipher, 'FIPS'
```

The following cipher suites and order of preference are set for SSL connections:

<table>
<thead>
<tr>
<th>Cipher Suite Name</th>
<th>Preference</th>
</tr>
</thead>
<tbody>
<tr>
<td>TLS_RSA_WITH_AES_256_CBC_SHA</td>
<td>1</td>
</tr>
<tr>
<td>TLS_RSA_WITH_AES_128_CBC_SHA</td>
<td>2</td>
</tr>
<tr>
<td>TLS_RSA_WITH_3DES_EDE_CBC_SHA</td>
<td>3</td>
</tr>
<tr>
<td>TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA</td>
<td>4</td>
</tr>
<tr>
<td>TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA</td>
<td>5</td>
</tr>
<tr>
<td>TLS_RSA_WITH_DES_CBC_SHA</td>
<td>6</td>
</tr>
<tr>
<td>TLS_DHE_DSS_WITH_DES_CBC_SHA</td>
<td>7</td>
</tr>
</tbody>
</table>
Secure Sockets Layer (SSL) in Adaptive Server

A preference of 0 (zero) sp_ssladmin output indicates a cipher suite is not used by Adaptive Server. The other, non-zero numbers, indicate the preference order that Adaptive Server uses the algorithm during the SSL handshake. The client side of the SSL handshake chooses one of these cipher suites that matches its list of accepted cipher suites.

This example uses a quoted list of cipher suites to set preferences in Adaptive Server:

1> sp_ssladmin setcipher, 'TLS_RSA_WITH_AES_128_CBC_SHA, TLS_RSA_WITH_AES_256_CBC_SHA'
2> go

The following cipher suites and order of preference are set for SSL connections:

<table>
<thead>
<tr>
<th>Cipher Suite Name</th>
<th>Preference</th>
</tr>
</thead>
<tbody>
<tr>
<td>TLS_RSA_WITH_AES_128_CBC_SHA</td>
<td>1</td>
</tr>
<tr>
<td>TLS_RSA_WITH_AES_256_CBC_SHA</td>
<td>2</td>
</tr>
</tbody>
</table>

Other considerations

When you upgrade to Adaptive Server version 12.5.3, the cipher suite preferences are the server defaults, and sp_ssladmin option lscipher displays no preferences. The server uses its default preferences, those defined by "All". The system security officer should consider the security policies employed at his or her site and the available SSL cipher suites to decide whether to restrict cipher suites and which cipher suites are appropriate for the security policies.

If you upgrade from Adaptive Server version 12.5.3 and have set cipher suite preferences, those preferences remain after upgrade. After the upgrade is complete, review your server's cipher suite preferences with current security policies and the lists of supported and unsupported cipher suites found in tables Table 19-1. Omit any cipher suites that are not supported.

If you have set SSL cipher suite preferences and want to remove all preferences from the server and use default preferences, delete the preferences from their storage location in system catalogs using the following commands:

1> sp_configure 'allow updates to system tables', 1
2> go

1> delete from master..sysattributes where class=24
2> go
1> sp_configure 'allow updates to system tables', 0
2> go

These commands can be executed only by the system security officer or system administrator.

Kerberos confidentiality

You can also ensure the confidentiality of all messages with Adaptive Server. To require all messages into and out of Adaptive Server to be encrypted, set the msg confidentiality reqd configuration parameter to 1. If this parameter is 0 (the default), message confidentiality is not required but may be established by the client.

For example, to require that all messages be encrypted, execute:

```
sp_configure "msg confidentiality reqd", 1
```

For more information about using Message Confidentiality with Kerberos and other Security Services supported, see “Administering network-based security” on page 502.

Dumping and loading databases with password protection

You can protect your database dump from unauthorized loads using the password parameter of the dump database command. If you include the password parameter when you make a database dump, you must also include this password when you load the database.

The partial syntax for the password-protected dump database and load database commands are:

```
dump database database_name to file_name [ with passwd = password ]
load database database_name from file_name [ with passwd = password ]
```

where:
- `database_name` – is the name of the database that is being dump or loaded.
Dumping and loading databases with password protection

- `file_name` – is the name of the dump file.
- `password` – is the password you provide to protect the dump file from unauthorized users.

Your password must be between 6 and 30 characters long. If you provide a password that is less than 6 or greater than 30 characters, Adaptive server issues an error message. If you issue an incorrect password when you attempt to load the database, Adaptive Server issues an error message and the command fails.

For example, the following uses the password “bluesky” to protect the database dump of the `pubs2` database:

```
dump database pubs2 to "/Syb_backup/mydb.db" with passwd = "bluesky"
```

The database dump must be loaded using the same password:

```
load database pubs2 from "/Syb_backup/mydb.db" with passwd = "bluesky"
```

**Passwords and earlier versions of Adaptive Server**

You can use the password-protected `dump` and `load` commands only with Adaptive Server version 12.5.2 and later. If you use the password parameter on a dump of a 12.5.2 version of Adaptive Server, the load fails if you try to load it on an earlier version of Adaptive Server.

**Passwords and character sets**

You can load the dump only to another server with the same character set. For example, if you attempt to load a dump from a server that uses an ASCII character set to a server that uses a non-ASCII character set, the load fails because the value of the ASCII password is different from the non-ASCII password.

Passwords entered by users are converted to Adaptive Server’s local character set. Because ASCII characters generally have the same value representation across character sets, if a user’s password is in an ASCII character set, the passwords for `dump` and `load` are recognized across all character sets.

Adaptive Server version 15.0.2 and later allows you to store portable passwords. See “Character set considerations for passwords” on page 465.
Symbols

& (ampersand)
    translated to underscore in login names 512
've (apostrophe) converted to underscore in login names 512
*(asterisk)
    converted to pound sign in login names 513
select and 601
\ (backslash)
    translated to underscore in login names 512
:= (BNF notation)
    in SQL statements xxiii
^ (caret)
    converted to dollar sign in login names 513
: (colon)
    converted to underscore in login names 512
, (comma)
    converted to underscore in login names 512
in SQL statements xxiii
{} (curly braces)
    converted to dollar sign in login names
    in SQL statements xxiii
%= (equals sign)
    converted to underscore in login names 512
! (exclamation point)
    converted to dollar sign in login names 513
< (left angle bracket)
    converted to dollar sign in login names 513
' (left quote), converted to underscore in login names 512
- (minus sign)
    converted to pound sign in login names 513
() (parentheses)
    converted to dollar sign in login names
    in SQL statements xxiii
% (percent sign)
    error message placeholder 345
    translated to underscore in login names 512
. (period)
    converted to dollar sign in login names 513
| (pipe)
    converted to pound sign in login names 513
+ (plus)
    converted to pound sign in login names 513
? (question mark)
    converted to dollar sign in login names 513
?? (question marks)
    for suspect characters 339
" " (quotation marks)
    converted to pound sign in login names
    enclosing parameter values 13
    enclosing punctuation 394
    enclosing values 393
> (right angle bracket)
    converted to underscore in login names 512
' (right quote), converted to underscore in login names 512
; (semicolon)
    converted to pound sign in login names 513
/ (slash)
    converted to pound sign in login names 513
[ ] (square brackets)
    converted to pound sign in login names
    in SQL statements xxiii
~ (tilde)
    converted to underscore in login names 512
$ISA 557
@@client_csexpansion global variable 328

Numerics

7-bit ASCII character data, character set conversion for 333

A

abort tran on log full database option 279
Index

abstract plan cache configuration parameter 79
abstract plan dump configuration parameter 80
abstract plan load configuration parameter 80
abstract plan replace configuration parameter 81
access 608
  restricting guest users 399
access control, row level 607
access permissions. See object access permissions
access protection. See permissions; security functions
access rules
  alter table command 614
  bcp 614
  creating 611
  creating and binding 609
  dropping 610
  examples 612
  extended 610
  sample table 609
accounting, chargeback 480
accounts, server
  See login:users
ACF (Application Context Facility), problem-solving with 625
activating roles 415
Adaptive Server principal name 533
adding
  comments to the audit trail 644
  database devices 194, 262–269
date strings 326
group to a database 396
guest users 398
logins to server 394–395
months of the year 326
remote logins 400, 490–492
remote servers 484–497
users to a database 194, 392
users to a group 397
additional network memory configuration parameter 81
address, server 16
administering security, getting started 379–383
generative garbage collection 361
  priority level 362
  aggressive housekeeper 361
  server 486
  aliases, user
See also login:users
  creating 425
  database ownership transfer and 566
dropping 427, 428
help on 428
all keyword
  grant 571, 579
  revoke 579
allocation pages 262
allocation units 262
See also size; space allocation
allow backward scans configuration parameter 84
allow nested triggers configuration parameter 85
allow nulls by default database option 279
allow procedure grouping configuration parameter 85
allow remote access configuration parameter 85, 496
allow resource limits configuration parameter 86
allow sendmsg configuration parameter 86
allow sql server async i/o configuration parameter 87
allow updates configuration parameter (now called allow updates to system tables) 14
allow updates to system tables configuration parameter 14, 88
alter database command
  omitting database device and 270, 272
  system tables and 258
alter role command 410, 440, 441
alternate identity. See alias, user
alternate languages. See languages, alternate
  and (&)
  translated to underscore in login names 512
ansi_permissions option, set
  permissions and 573
apostrophe converted to underscore in login names 512
Application Context Facility 617, 618
  granting and revoking privileges 619
  setting permissions 618
  valid users 619
application contexts
  built-in functions 620
  using 620
application design 194
applications
  proxy authorization and 595

710
Adaptive Server Enterprise
Arabic character set support 295
ASCII characters
  character set conversion and 333
assigning
  login names 382
asymmetric key pairs, generating 459
asterisk (*)
  converted to pound sign in login names 513
  select and 601
asynchronous I/O
  limiting Server requests for 148
asynchronous prefetch
  configuring 132
@@char_convert global variable 328
@@client_csid global variable 328
@@client_csname global variable 328
@@langid global variable 330
@@language global variable 330
@@max_connections global variable 193
@@maxcharlen global variable 328
@@ncharsize global variable 328
audit options
  displaying 644
  examples 662
  setting 661
audit queue 643, 653
  audit queue size configuration parameter 89, 643, 653
audit trail 31, 639, 671
  adding comments 644, 669
  changing current audit table 649
  illustration with multiple audit tables 641
  managing 648
  querying 671
  stacktrace of error messages 346
  threshold procedure for 648
auditing 388, 639, 639–671
  See also audit options
  adding comments to the audit trail 644
  configuration parameters 643
  devices for 645
  disabling 643
  displaying options for 644
  enabling 381, 643
  enabling and disabling 656
  installing 645
managing the audit trail 648
managing the transaction log 654
overview 639
queue, size of 89, 643
sysaudits_01...sysaudits_08 tables 671
system procedures for 643
threshold procedure for 648
turning on and off 656
auditing configuration parameter 90, 656
authentication 500, 501
  mutual 501
authorizations. See permissions
auto identity database option 280
automatic operations
  character conversions in logins 512
  primary and secondary database dumps 282
B
  Backing up
    master database 54
  backlash (\)
    translated to underscore in login names 512
  backtracing errors. See error logs
Backup Server
  error messages 355
  shutting down 369
  tape retention in days configuration parameter 239
  backups 42–45
  hints 42–45
Backus Naur Form (BNF) notation xxii, xxiii
Baltic character set support 295
base tables. See tables
bcp (bulk copy utility)
  character set conversion and 340, 341
  fast version 283
  security services and 523
  select into/bulkcopy/pllsort and 283
  sort order changes and 317
  with access rules 614
Big 5
  similarities to CP 950 295
binary sort order of character sets
Index

character set changes and database dumps 317
BNF notation in SQL statements xxii, xxiii
brackets. See square brackets [ ]
built-in functions
  security 527
bytes
  character 338

cache partitions
  configuring 131, 132
  cache, procedure 210
  caches, data
    database integrity errors and 354
calls, remote procedure 483–497
timeouts 487
cascade option, revoke 572
case sensitivity
  in SQL xxiv
certificates
  administration of 697
  authorizing 696
  CA certificates 689
  defined 689
  obtaining 694
  public-key cryptography 689
  requesting 696
  self-signed CA 696
  server certificates 689
chains, ownership 603
changing
  See also updating
  configuration parameters 76, 495
database options 277–286
  Database Owners 565
default database 423
passwords for login accounts 421
server logins 423
system tables, dangers of 12, 14
user information 420–424
user’s group 423

user’s identity 590
@@char_convert global variable 328
cascade option, revoke 572
charsets directory 328
chargeback accounting 480
character sets and password-protected dumps 708
characters
  disallowed in login names 512
  that cannot be converted 338
certificates
  administration of 697
  authorizing 696
  CA certificates 689
  defined 689
  obtaining 694
  public-key cryptography 689
  requesting 696
  self-signed CA 696
  server certificates 689
chains, ownership 603
changing
  See also updating
  configuration parameters 76, 495
database options 277–286
  Database Owners 565
default database 423
passwords for login accounts 421
server logins 423
system tables, dangers of 12, 14
user information 420–424
user’s group 423

CA certificates 689
  location of 692
  trusted root certificate 689
cascade option, revoke 572
case sensitivity
  in SQL xxiv
certificates
  administration of 697
  authorizing 696
  CA certificates 689
  defined 689
  obtaining 694
  public-key cryptography 689
  requesting 696
  self-signed CA 696
  server certificates 689
chains, ownership 603
changing
  See also updating
  configuration parameters 76, 495
database options 277–286
  Database Owners 565
default database 423
passwords for login accounts 421
server logins 423
system tables, dangers of 12, 14
user information 420–424
user’s group 423

user’s identity 590
@@char_convert global variable 328
cascade option, revoke 572
charsets directory 328
chargeback accounting 480
character sets and password-protected dumps 708
characters
  disallowed in login names 512
  that cannot be converted 338
checking passwords for at least one character 442
\textbf{checkpoint} command
  setting database options and 286
checkpoint process 213
  \textit{no chkpt on recovery} database option 282
  \textit{recovery interval} parameter and 214
  \textit{trunc log on chkpt} database option 213, 284–285
\textbf{checktable} option, \texttt{dbcc} 322
cipher suites
  defined 700
   supported 700
\textbf{cis bulk insert batch size} configuration parameter 91
\textbf{cis connect timeout} configuration parameter 92
\textbf{cis cursor rows} configuration parameter 92
\textbf{cis idle connectin timeout} configuration parameter 93
\textbf{cis packet size} configuration parameter 93
\textbf{cis rpc handling} configuration parameter 94
  \texttt{@@client_csid} global variable 328
  \texttt{@@client_csname} global variable 328
clients
   assigning client name, host name, and application name 424
   character set conversion 341
Closed Problem Reports 370
\textbf{cntritype} option
  \texttt{disk init} 269
colon (:) converted to underscore in login names 512
column name unqualified 351
columns permissions on 571, 599
comma (,) converted to underscore in login names 512
   in SQL statements xxiii
command
  \texttt{delete} 361
  \texttt{disk resize} 273–276
  \texttt{reorg reclaim_space} 362
comments adding to audit trail 644, 669
\texttt{common.loc} file 329
comparing values
  \textbf{cost of a cpu unit} configuration parameter 95
data type problems 350
cryptographic protection 315–320
database options and 214
  \textit{checkpoint} command
  recovering databases 286
  \textit{checkpoint process} 213
  \textit{no chkpt on recovery} database option 282
confidential data 500
configuration (server)
   character sets 315
   message language 315–320
   network-based security 503
   sort orders 315–323
configuration file
   default name and location 62
   specifying at start-up 67
   storage of configured value 62
configuration parameter
   \texttt{max native threads per engine} 152
   \texttt{rtm thread idle wait period} 218
configuration parameters 79–247
   audit-related 643
   changing 495
   chargeback accounting 481
   default settings of 61
   \texttt{dtm detach timeout period} 361
   help information on 64
   \texttt{housekeeper free write percent} 361
   listing values of 65
   remote logins and 85, 495–497
    configuring
   Kerberos 529
    conflicting permissions 587
   See also permissions
   connecting to Adaptive Server 16
   connections
      directory services 17
      interfaces files 16
      maximum user number 193
   consistency
      checking databases 44
      context-sensitive protection 601
    conventions
      See also syntax
      Transact-SQL syntax xxii
      used in the Reference Manual xxii
    copying selected data
      See also \texttt{insert} command; \texttt{select} command
    \textbf{cost of a cpu unit} configuration parameter 95
Index

cost of a logical io configuration parameter 95
cost of a physical io configuration parameter 95
CP 1252
similarities to ISO 8859-1 295
CP 950
similarities to Big 5 295
cp437 character set 102
cp850 character set 102
CPR files 370
cpu accounting flush interval configuration parameter 96, 481
cpu grace time configuration parameter 97
CPU usage
per user 480
create database command
default database size configuration parameter and 103
model database and 28
omitting database device and 270, 272
permission to use 565
system tables and 10
create index command 253, 259
create procedure command 14
create role command 409
create rule command, new functionality 608
create rule syntax 608
create rule, syntax 609
create table command 253
create trigger command 572
Creating
databases 53
guest users 54
users 55
creating
database objects 253
databases 565
groups 396
guest users 398
master database 255
model database 255
segments 255
stored procedures 14
sybsecurity database 645
system procedures 14
system tables 10
tempdb database 255
triggers 572
user aliases 425
user-defined error messages 348
credential, security mechanism and 501
cross-platform dump and load, handling suspect partitions 325
cs_connection command, number of user connections and 194
curly braces ()
converted to dollar sign in login names 513
curly braces (}) in SQL statements xxiii
current audit table configuration parameter 98, 649
current database 350
current usage statistics 480
current user
set proxy and 594
cursors
row count, setting 92
custom password checks 453
custom password complexity checks 445
cyrillic character set support 295

D
DAC. See discretionary access control (DAC)
data
See also permissions
confidentiality of 500
encryption 500
integrity of 500, 513
losing unlogged 284
packets 497
data caches
configuring partitions 131, 132
database integrity errors and 354
data dictionary. See system tables
database administration 3–7
database device space See segments; space allocation
database devices 261
See also disk mirroring; dump devices; master device adding 262–269
default 272–273
dropping 271
fragments 258
information about 270
initializing 261–269
names of 256, 263
number of server-usuable 174
placing objects on 255
database dumps
password-protected 707
database object owners 6
See also database owners
permissions 7, 564, 591
status not transferable 417
tasks of 6
database objects
See also individual object names
access permissions for 7, 570
assigning to devices 254
controlling user creation of 27
creating 27, 253, 567
dependent 604
dropping 567, 568
dropping users who own 417
errors affecting 353
finding 349
maximum number of open 186
ownership 6, 417, 567
permissions on 567
triggers on 607
database options 277–287
changing 286
listing 278
setting 279–285
showing settings 279
Database Owners 6
changing 565
error responsibilities of 349, 351
login name 4, 6
name inside database 417, 427
objects not transferred between 417
password forgotten by 405
permissions granted by 579
permissions of 6, 564, 566
See also database object owners 563
setuser command and 590–591
several users as same 425
tasks of 6
permissions
database size configuration variable 54

Databases
backing up 54
creating 53
guest users 54
databases
See also database objects; user databases
adding users 396–400
auditing 645
backing up 28, 42
creation permission 565
default 27, 394, 423
default storage for 25, 272
dropping users from 416
dumping 42
events affecting 353
integrity concerns 353
loading after character set change 317
loading after sort order change 317
new 28
number of open 182
options 277–286
ownership of 565
sequence numbers for recovery 283
size 28
system 23
database-specific dbcc, master and 583
datserver command
using to unlock logins and roles 441
date parts
alternate language 326
dates
adding date parts 326
alternate language 326
display formats 329
format in error messages 347
days
alternate language 326
dbcc and storage_admin_role command 583
dbcc (database consistency checker) 44
database damage and 349, 353
database-specific commands 582, 583
defined 582
described 582
discretionary access control 582
grant dbcc and roles 583
grant dbcc and users in databases 583
grant dbcc checkstorage command and 583
server-wide commands 582, 583
tune command and 583
when to use 353
DB-Library programs
number of user connections and 194
dbo use only database option 280
"dbo" user name 4, 6
dbprocess command, number of user connections and 194
DCE (Distributed Computing Environment) security mechanism 509
ddl in tran database option 280
deactivating roles 415
deadlock checking period configuration parameter 99
deadlock pipe active configuration parameter 100
deadlock pipe max messages configuration parameter 100
deadlock retries configuration parameter 101
deadlocks 350
descending scans and 84
dekanji character set 102
default character set id configuration parameter 102
default database changing user's 423
default database devices designating 272
default database size configuration parameter 103
default exp_row_size percent configuration parameter 103
default fill factor percent configuration parameter 104
default language id configuration parameter 105
default network packet size configuration parameter 105
default segment 255
default settings changing character set 315–324
changing sort order 317–323
character set ID number 102
configuration parameters 61
databases 27, 394
language 105
permissions 28
sort order 106, 107
system databases at installation 256
default sortorder id configuration parameter 107
default XML sortorder configuration parameter 107
defaulton | defaultoff option, sp_diskdefault 272
defaults
See also database objects
defn copy utility command
See also Utility Programs manual
character set conversion and 340, 341
delete command 361
delete statistics syntax 575
deleting files 271
users 55
denyng access to a user 418, 419
descending scans deadlocks and 84
detached transactions 109
development server 36
device fragments 258
device shrinkage, disk resize 274
devices 261
See also database devices; dump devices; master device
direct updates to system tables 88
directory drivers 504
example of entry in libtcl.cfg file 507
directory entries, creating 696
directory services in libtcl.cfg file 17, 505
directory structure character sets 328
internationalization files 328
localization files 330
*.loc files 330
dirty pages 213
disable character set conversions configuration parameter 108
disable disk mirroring configuration parameter 108

disabling auditing 643
disallowing simple passwords 445
discretionary access control (DAC) 563–607
See also permissions
  granting and revoking permissions 569
  of dbcc commands 582
  overview 386
  stored procedures and 602
  system administrators and 564
  user alias and 590
  views for 600
disk controllers 269
disk devices
  See also database devices; dump devices; space allocation
disk I/O
  configuration parameters for 176
  database loads and 151, 173, 179
disk I/O structures configuration parameter 109
disk init command 252, 257, 258, 262–269
disk mirror command 253
disk mirroring
  disabling 108
  enabling 108
  recovery and 254
  status in sysdevices table 271
disk reinit command
  See also disk init command
disk resize 252, 273–276
  device shrinkage 274
  insufficient disk space 274
  minimum size 274
  mirroring 274
  specifying device size 275
  syntax 274
  using 273
disks See database devices; devices; dump devices
Distributed Transaction Management (DTM) 31
Distributed Transaction Processing (DTP) 31
drop logins option, sp_dropserver 490
drop role command 417
dropping
  database devices 271
dump devices 271
groups 417
guest users of master 398
logins from servers 419
master device from default space pool 272
remote logins 489, 490
servers 489
user aliases 427, 428
user from a database 416
user-defined roles 417
users from servers 419
users who own database objects 417
dscp utility for specifying security mechanism 509
dsed utility for security services 509
dsync option
  disk init 271
dtm detach timeout period configuration parameter 109, 361
dtm lock timeout period configuration parameter 110
dump database command
  disk init and 262
  master database and 43
  model database and 28
  dump database syntax 707
  dump devices
    dropping 271
    information about 270
    sysdevices table and 257
  dump on conditions configuration parameter 111
dump transaction command
  trunc log on chkpt and 284–285
dump, database 42
dynamic allocation on demand configuration parameter 112
dynamic configuration parameters 62

E
Eastern Europe
  character set support 295
  empty pages, accumulating 362
enable cis configuration parameter 112, 115, 123
enable DTM configuration parameter 113
enable encrypted columns configuration parameter 113
Index

enable HA configuration parameter for high availability 116
enable housekeeper GC configuration parameter 116, 362
enable java configuration parameter 114, 118
enable job scheduler configuration parameter 118
enable ldap user auth configuration parameter 118
enable literal autoparam configuration parameter 119
enable logins during recovery configuration parameter 119, 120
enable merge join configuration parameter 120
enable metrics capture configuration parameter 120
enable monitoring configuration parameter 121
enable pam user auth configuration parameter 121
enable real time messaging configuration parameter 122
enable rep agent threads configuration parameter 122
enable row level access control configuration parameter 123
enable semantic partitioning configuration parameter 123
enable surrogate partitioning configuration parameter 124
enable unicode conversion configuration parameter 124
enable unicode conversions configuration parameter 124
enable unicode normalization configuration parameter 124
enable web services configuration parameter 125
enable xact coordination configuration parameter 125
enable xml configuration parameter 126
enabling
  auditing 381, 643
  SSL 694
encoding characters 331
encryption
  data 500
  key exchange 688
  public/private key 688
  public-key cryptography 688
  symmetric key 688
group
  identification numbers 347
  number of 156
environment variable
  $ISA 557
error logs 46, 352
creation and ownership 346
format 347
location 15
purging 347
error messages 345–354
  altering server-provided 329, 348
  character conversion 339
  creating user-defined 348
  for fatal errors 352–354
  numbering of 345
  severity levels of 348–354
  user-defined 348
errorlog pipe active configuration parameter 127
errorlog pipe max messages configuration parameter 127
errors
See also error logs; error messages
  character conversion 338
  fatal 352–354
  logging 346
  multiple 344
  reporting of 354
  server responses to 343–354
  state numbers 343
  types of information logged 15
  user 349, 349–352
esp execution priority configuration parameter 127
esp execution stack size configuration parameter 128
esp unload dll configuration parameter 128
eucjs character set 102
European currency symbol
  character sets 296
event buffers per engine configuration parameter 129
event log computer name configuration parameter 129
event logging configuration parameter 130
exclamation point (!)
  converted to dollar sign in login names 513
executable code size + overhead configuration parameter 131
execution
  ESPs and XP Server priority 127
expand down parameter
  sp activeroles 436
expiration interval for passwords 455
expiration of passwords 455
Index

expired passwords 238
exporting set options 635
**extended cache size** configuration parameter 131
extended stored procedures
  configuration parameters 127–250
extended UNIX character set 102

**F**
failures, media 354
fatal errors
  backtrace from kernel 346, 352
  error messages for 352–354
  severity levels 19 and up 352–354
file descriptors 193
  maximum per-process configured for your operating system 196
files
  character set translation (.xlt) 327
Closed Problem Reports (CPRs) 370
deleting 271
error log 16, 346
interfaces 16
internationalization 327
libtcl.cfg file 17
localization 329
System Problem Reports (SPRs) 370
fillfactor
  **default fill factor percent** configuration parameter 104
finding
  database objects 349
  user IDs 431
  user names 431
  users in a database 431
**fix_text option, dbcc** 323–324
For load 54
formats
  date, time, and money 329
  locale, unsupported 326–327
formulas
  user requirements and 194
forwarded rows
  reducing with **default exp_row_size** configuration parameter 103

**G**
garbage collection
  aggressive test 361
  lazy test 361
garbage collector
  configuring aggressive 362
  housekeeper utility 361
German
  character set support 295
get_appcontext 620, 621
**global async prefetch limit** configuration parameter 132
**global cache partition number** configuration parameter 132
global login triggers 637
grant command 564, 569–587
  **all** keyword 579
  public group and 571
  roles and 589
grant dbcc
  roles and 583
  users in databases and 583
grant option
  sp_helprotect 597
grant option for option, **revoke** 572
granting
  access permissions 6
  create trigger permission 572
  object creation permissions 6
  proxy authorization permission 581
  roles to roles 411
  roles with **grant role** 588
granting and revoking permissions for users and roles 575
granting default permissions on system tables 583–585
Greek
  character set support 295

System Administration Guide: Volume 1 719
Index

groups
See also public group
changing 423
conflicting permissions and 587
creating 396
dropping 417
grant and 574
naming 396
Public 55
revoke and 575
See also public group
changing 423
conflicting permissions and 587
creating 396
dropping 417
grant and 574
naming 396
Public 55
revoke and 575

groups, language 295
Guest users
creating 54
databases 54
guest users 569
adding 398
creating 398
permissions 398
sample databases and 32, 399
guidelines, security 381

H
Halloween problem
avoiding with unique auto_identity index option 285
handling suspect partitions 325
hardware
errors 354
hash
defined 689
message digest 689
hash buckets (lock) 145
heap memory per user configuration parameter 133
Hebrew
character set support 295
hierarchy of permissions. See permissions
hierarchy of roles. See role hierarchies
high availability
installhasvss script 116
insthasv script 116
setting enable HA 116
High Availability and passwords 475
histogram tuning factor configuration parameter 133
housekeeper chores 361
configuration parameter license information 361

housekeeper free write percent configuration parameter 135, 361
housekeeper garbage collector 361
housekeeper task
configuring 135
license use monitoring 478
space reclamation and 117
statistics flushing 135
housekeeper utility
functionality 360
housekeeper wash, housekeeper garbage collection,
housekeeper chores 360
three tasks 360
wash 361
wash task 135

I
I/O
usage statistics 480
i/o accounting flush interval configuration parameter 136, 481
i/o batch size configuration parameter 137
i/o polling process count configuration parameter 138
IBM character set 102
Icons 50
identification and authentication
See also logins
controls 384
identities
alternate 425
proxies and 591
session authorizations and 591
identity burning set factor configuration parameter 139
IDENTITY columns
automatic 280, 285
nonunique indexes 282
identity grab size configuration parameter 140
identity in nonunique index database option 282
identity of user. See aliases; logins; users
IDs, user 404, 431
system procedures and 14
impersonating a user. See setuser command

720

Adaptive Server Enterprise
Index descriptors
maximum number open 184
indexes
character set changes 323
classification-based 321
default fill factor percent percentage for 104
IDENTITY columns in nonunique 282
object allocation maps of 182
rebuilding 322
sort order changes 322
suspect 322, 353
individual accountability 381
information (server)
changing user 420–424
configuration parameters 65
database devices 270
database options 278–279
devices 270
dump devices 270
error messages 345–354
locked logs 419
logs 431
permissions 595–599
problems 346
remote server logs 495
remote servers 489
user aliases 428
users, database 428–481
information messages (server). See error messages;
severity levels
initializing
database devices 261–269
installation, server
audit system 645
establishing security after 381–383
interfaces file 17
status after 255
installhasvss script 116
installing
classification risks 32
classification script 116
insufficient disk space

    disk resize 274
insufficient permission 350
insufficient resource errors (Level 17) 351
interfaces file 16, 508

internal error, nonfatal 352
international language support. See character sets;
languages
internationalization
a sample system 291
advantages 290
definition 289
directory structure for character sets 328
files 327
is_sec_service_on security function 527
ISO 8859-1
similarities to CP 1252 295
iso_1 character set 102
isolation levels
level 0 reads 282
isql utility command
character set conversion and 340, 341
number of user connections and 194
passwords and 494
security services and 523
status and informational messages 349
system administration and 7

J
Japanese character sets 102

    sjis (Shift-JIS) 102
support 295
See also languages, alternate
Java configuration parameters 224
job scheduler interval configuration parameter 141
job scheduler tasks configuration parameter 141
joins
views and 601

K
-k option 534
kadmin 529
kanji. See Japanese character sets
Kerberos 527
compatibility 528
configuring 529
CyberSafe Kerberos libraries 527
Index

keytab file 529
licenses 528
MIT Kerberos libraries 527
Native libraries 527
Kerberos authentication 533
verifying 536
kernel
error messages 346, 352
key exchange
encryption 688
public/private key 688
symmetric key 688
key pairs, generating assymetric 459
keys, table
on system tables 11
keytab file
specifying for utility programs 523
kill command 356–359
kill command, changes 359
kill statusonly parameter 359
known problems 370
Korean
character set support 295

L
LAN Manager security mechanism 509
@@langid global variable 330
language defaults 105
changing user’s 321
us_english 105
@@language global variable 330
language groups 294, 295
languages
on server 294
supported by a character set 294
languages, alternate 327
See also character sets; charset.loc file; Japanese character sets
date formats in unsupported 326
localization files 311–330
supported languages 290
Latin alphabet 296
lazy garbage collection 361
LDAP

access restrictions 18
defined 18
multiple directory services 19
versus the interfaces file 20
LDAP server
state transitions 548
LDAP server support 546
LDAP user authentication 550
password changes 544
tighter controls on login mapping 551
troubleshooting 554
tuning 550
levels, severity.
See severity levels, error
libtcl.cfg file 17
example of 507
preparing for network-based security 504
tools for editing 506
license information configuration parameter 141, 478
license information, configuration parameter 361
license use
error log messages 479
monitoring 477
linkage, page
See also pages, data
linking users. See alias, user
list_appcontext 620, 622
listing
database options 278
load database syntax 707
load, database
number of large I/O buffers configuration parameter 108, 151, 173, 179
local and remote servers. See remote servers
local option, sp_addserver 486
local servers 486
locales directory 312
locales.dat file 329
localization 290
See also languages, alternate
tools for editing 329
lock address spinlock ratio configuration parameter 142
lock hash buckets 145
lock hash table
configuring size of 143
Index

lock hashtable size configuration parameter 143
lock promotion thresholds
setting with sp_configure 200–217
lock scheme
default 144
lock scheme configuration parameter 144
lock shared memory configuration parameter 144
lock spinlock ratio configuration parameter 145
lock table spinlock ratio configuration parameter 145
lock timeouts
configuring server-wide 146
lock wait period configuration parameter 146
locking
by dbcc commands 324
logins 418, 437
locking logins 55
locking scheme
server-wide default 144
locks
quantity of 180
log audit logon failure configuration parameter 147
log audit logon success configuration parameter 147
log file. See error logs
log on option
create database 258
logging
login failures 147
successful logins 147
Windows NT event log in 130, 131
logical
page sizes 35
login IDs, number of 400
login mapping
tighter controls 551
login names. See logins
login process
authentication 501
login triggers
and set options 635
configuring 627
disabling execute privilege 635
displaying 629
dropping and changing 628
executing 629
issues 634
issues and information 634
output 629
restrictions 634
restrictions on 634
syntax for configuring 628
syntax for creating 627
understanding output 629
using 627
using for other applications 629
logins
See also remote logins; users
adding to servers 394–395
alias 427, 428
assigning names for 382
database object owner 6
“dbo” user name 4, 6
displaying password information 441
dropping 419
finding 431
information on 431
invalid names 512
locking 55, 418, 437, 440
maximum attempts, changing 438
maximum attempts, setting 437
“sa” 381
unlocking 418, 440
logsegment log storage 255
lookup server
secondary 546
losing unlogged data 284

M
Macintosh character set 102, 338
mail session, starting 232
management, space. See space allocation; storage management
managing users. See users
mapping
device name to physical name 262
remote users 490–494
master database
backing up 54
Index

master database 9, 25–27, 42

See also disk mirroring; system tables
backing up 42
changing option settings 278
dropping guest users of 398
guest user in 398
keys for system tables in 11
ownership of 566
sysdevices table 270

master database, granting default permissions on system
tables 585
master database, revoking default permissions on system
tables 585
master device 24, 264, 270
See also database devices
removing from default space pool 271, 272
sp_diskdefault and 272
max async I/Os per engine configuration parameter 148
max async I/Os per server configuration parameter 149
max cis remote connections configuration parameter 150
max concurrently recovered db configuration parameter 151, 173
max memory configuration parameter 151
max native threads for LDAP user authentication 550
max native threads per engine configuration parameter 152
max network packet size configuration parameter 153
max number network listeners configuration parameter 155
max online engines configuration parameter 156
max online Q engines 156
max parallel degree configuration parameter 157
max repartition degree configuration parameter 158
max resource granularity configuration parameter 158
max roles enabled per user configuration parameter 168, 409
max scan parallel degree configuration parameter 159
max SQL text monitored configuration parameter 160
@@max_connections global variable 193
@@maxcharlen global variable 328
maximum buffers per lava operator configuration parameter 161
maximum dump conditions configuration parameter 160, 161
maximum failed logins 449
membership keyword, alter role 411
memory
See also space allocation
audit records 89, 653
freeing from XP Server 128
network-based security and 514
number of open databases and 183
memory alignment boundary configuration parameter 164
memory per worker process configuration parameter 164
message digest
defined 689
hash 689
messages
confidentiality 501, 513
error 15, 345–354
fatal error 15
integrity 502, 513
language setting for 290
origin checks 502
protection services for 501
start-up 15
system 345–354
user-defined 348
messaging memory configuration parameter 165
metadata caches
configuration parameters 72–199
metrics elap max 165
metrics elap max configuration parameter 165
metrics exec max 165
metrics exec max configuration parameter 165
metrics lio max 166
metrics lio max configuration parameter 166
metrics pio max 166
metrics pio max configuration parameter 166
Microsoft character set 102
min pages for parallel scan 166
minimum alphabetic characters in password 446
minimum digits in password 446
minimum number of lowercase letters in password 447
minimum number of uppercase letters in password 447
minimum pages for a parallel scan configuration parameter 166
minimum password length 448
minimum password length configuration parameter 167
minimum size, disk resize 274
minimum special characters in password 447
minus sign (-)
  converted to pound sign in login names 513
miscellaneous user error 351
mistakes, user See errors; severity levels, error
model database 54
model database 28
  changing database options 284
  changing options in 278
  creating 255
  keys for system tables in 11
  size 103, 265
modifying
  server logins 423
money
  local formats 329
monitoring
  spt_monitor table 14
  SQL text 160
  Windows NT Performance Monitor 226
monitoring tables
  configuration options 72
month values
  alternate language 326
MSDTC 113
msg confidentiality reqd configuration parameter 168
msg integrity reqd configuration parameter 168
multibyte character sets 323
  changing to 324
  default character set id configuration parameter 102
  incompatible 338
  multilingual character set 102
  multiple directory services
    LDAP 19
mut_excl_roles system function 435
mutual authentication server option 518
mutual exclusivity of roles 387, 435

name of device 263
sysdevices listing 258
names
  See also information (server); logins
  alias 427, 428, 590
  column, in commands 351
  finding user 431
  for logins 382
  group 572
  mapping remote user 491
  original identity 591
  partial, in option specification 286
  remote server 485
  remote user 491
  server 486
  system extended stored procedures 15
  system procedures 12
  user 397, 431, 568, 572
naming
  groups 396
  servers 486
  user-defined roles 408
Navigating
  to objects 50
@@nccharsize global variable 328
nested trigger configuration parameter (now called allow nested triggers) 85
net password encryption option 488
net password encryption reqd configuration parameter 169
network drivers 504
  example of entry in libtcl.cfg file 507
  syntax for in libtcl.cfg file 504
network-based security 499–527
  adding logins for unified login 515
  configuring server for 510
  connecting to server 523
  getting information about 523, 526
  identifying users and servers 509
  memory requirements 514
  overview 500
  process for administering 502
  rebooting server to activate 514
  remote procedure calls 516
  security mechanism 509
Index

setting up configuration files 503
using 523
networks
collections 16
directory services 17
interfaces files 16
software 38
no chkpt on recovery database option 282
no free space acctg database option 283
nonrepudiation, digital signature 689
nonstop recovery 254
NT LAN Manager security mechanism 509
null keyword
in sp_addlogin 394
null passwords 422
number (quantity of)
database devices 174
engines 156
locks 180
open databases on Server 182
open objects 186
remote sites 497
seconds for acquiring locks 146
user connections (@@max_connections) 193
number of alarms configuration parameter 170
number of aux scan descriptors configuration parameter
171
number of devices configuration parameter 174
number of dtx participants configuration parameter 174
number of histogram steps configuration parameter 177
number of index trips configuration parameter 178
number of large i/o buffers configuration parameter 179
number of locks configuration parameter 180
number of login IDs 400
number of mailboxes configuration parameter 181
number of messages configuration parameter 181
number of oam trips configuration parameter 182
number of open databases configuration parameter 182
number of open indexes configuration parameter 184
number of open objects configuration parameter 186
number of pre-allocated extents configuration parameter
189
number of Q engines at startup 189
number of remote connections configuration parameter
190, 497
number of remote logins configuration parameter
176, 191, 496
number of remote sites configuration parameter
191, 497
number of sort buffers configuration parameter 191
number of threads for memory dumps, determining
176
number of user connections configuration parameter
76, 192–194
number of users 400
number of worker processes configuration parameter
195
numbers
engine 347	error message 345
sort order 106, 107
status bit (sysdevices) 270
O
o/s file descriptors configuration parameter 196
object access permissions See permissions
object lockwait timing configuration parameter 196
object owners. See database object owners
object permissions
grant all 571, 579
objectid.dat file 507
location of 697
objects
icons 50
navigating to 50
See database objects
old and new password complexity checks 450
on keyword
grant 571
revoke 571
open index hash spinlock ratio configuration parameter
197
open index spinlock ratio configuration parameter
197
open object spinlock ratio configuration parameter
198
openVMS systems
foreign device 263
operating system commands
executing 15
operator role 5
permissions 406
optimization goals and configuration parameters 199
**optimization timeout limit** configuration parameter 200
options
database 277–287
remote logins 494
remote servers 487
server 487
unique string for 286
order of commands
grant and revoke statements 569–590
out-of-sequence checks 502
overflow errors
server stack 230
overflow stack (stack guard size configuration parameter) 228
overriding user permissions 55
owners. See database object owners 579
ownership chains 603

P
packets, network
pre-read 497
size, configuring 153–155
**page lock promotion HWM** configuration parameter 200
**page lock promotion LWM** configuration parameter 201, 216
**page lock promotion PCT** configuration parameter 202
pages, data 262
dirty 213
parameters, procedure 394
parentheses ()
converted to dollar sign in login names 513
parentheses ()
in SQL statements xxiii
**partition groups** configuration parameter 204
**partition spinlock ratio** configuration parameter 204
partitions
disk 263
partitions, suspect, fixing tables 325
partitions, suspect, handling 325
password changes for LDAP user authentication 544
password complexity
cross-checks 450
custom password checks 453
old and new 450
password complexity checks 445
custom password complexity checks 445
disallowing simple passwords 445
maximum failed logins 449
password expiration 448
password expiration warnings 449
resetting the password 449
specifying a minimum number of alphabetic characters 446
specifying a minimum number of digits 446
specifying a minimum number of lowercase characters in password 447
specifying a minimum number of special characters in password 447
specifying a minimum number of uppercase characters in password 447
specifying a minimum password length 448
password expiration 448
password expiration warnings 449
password security 436–477
generating an assymmetric key pair 459
generating key pairs using sp_passwordpolicy 459
securing login passwords on a network 459
password-protected database dumps 707
passwords 421
backward compatibility 460
changing 421
checking for at least one character 442
choosing 393
choosing secure 393
date of last change 430
displaying information 441
downgrading 461
encryption over network 488
expiration interval 455
expiration of 455
for roles 455
forgotten 405
Index

High Availability and 475
minimum length 443
null 422
protecting 393
protection against guessing 437
remote users 488, 494
roles and 415
rules for 393
sp_password 421
per object statistics active configuration parameter 205
per object statistics active configuration parameter 205
percent sign (%) 345
translated to underscore in login names 512
performance
audit queue size 89
default fill factor percent effect on 104
disk mirroring and 254
ESPs and XP Server priority 127
space allocation and 255
speed and 255
performance monitoring option configuration parameter 207
period (.) 513
converted to dollar sign in login names 513
permission cache entries configuration parameter 207
permissions
See also discretionary access control (DAC) 590
acquiring other users’ 590
aliases and 426
ansi_permissions option and 573
assigned by Database Owner 579
assigning 579
concrete identification 573
create database 565
database object owners 7
Database Owners 6, 564, 566
default 28
denying 350
disk init 269
for creating triggers 572
granting 569–587
group versus user 55
groups and 396
guest users 398
hierarchy of user 589
information on 595–599
insufficient (Level 14) 350
master database 27
model database 28
object 7, 567
object access 569, 569–575
object creation 579
operator 406
overriding 55
ownership chains and 603
proxy authorization 581
public group 568, 571, 587
remote users 494
revoking 569–587
selective assignment of 586
stored procedures 494, 567, 571
summary of 563
system administrator 564–565
system procedures 568
system tables 583
tables 567, 571
tables compared to views 600
tempdb database 30
transfers and 566
triggers and 607
using setuser 590
views 600–602
on views instead of columns 601
physical resources, managing.
See storage management
placeholders
error message percent sign (%) 345
plan text pipe active configuration parameter 208
plan text pipe max messages configuration parameter 208
Pluggable Authentication Module (PAM) 556
$ISA 557
32- and 64-bit servers on the same machine 557
configuring Adaptive Server for PAM 558
determining which module to use 557
enable pam user auth 558
password management 558
RFC 86.0 557
unified logins 557
plus (+)
Index

converted to pound sign in login names 513
preferences, user name 397
preventing garbage collection
accumulating empty pages 362
primary database 282
principal name
using sbmapname 535
with SYBASE_PRINCIPAL 534
with the -k option 534
principal name for Adaptive Server 533
print deadlock information configuration parameter
209
print recovery information configuration parameter
210
priority
XP Server 127
proc_role system function
stored procedures and 436, 603
procedure cache 210, 353
procedure calls.
See remote procedure calls
procedures. See stored procedures; system procedures
process ID, status of 359
process wait events configuration parameter 211
processes (server tasks) 356, 359
See also servers
administering Adaptive Server 379
current on server 429
information on 429
killing 356–359
prod-consumer overlap factor 211
production server 36
protection mechanisms. See security functions; stored
procedures; views
protection system
context-sensitive 601
hierarchy (ownership chains) 603
reports 595–599
summary 563
proxy authorization 590–599
executing 593
granting 581
granting permission for 581
how applications use it 595
how users use it 593
overview 591
using 591, 593
Public
membership 55
public group 396
See also groups
grant and 571, 580
guest user permissions and 398
permissions 568, 587
revoke and 571
sp_adduser and 397
sp_changegroup and 423
public keyword
grant 580
public/private key encryption 688
public-key cryptography
certificates 688
defined 688
digital signature 688
equivalence 688
pubs2 database
administering 32
image information in 33
pubs3 database
administering 32
Q
queries
conversion errors, preventing 339
question marks (?) for suspect characters 339
quotation marks ( " )
converted to pound sign in login names 513
R
read committed with lock configuration parameter
212
read only database option 283, 286, 321
reads
physical 255
rebooting the server 514
See restarts, server
reconfigure command 75
Index

- record keeping 46–48
  - configuration 47
  - contacts 46
  - maintenance 47
  - system 48
- records, audit 643
- recovery
  - configuration parameters for 212–214
  - loading databases 317
  - master database 42, 262
  - nonstop 254
  - planning backups for 28
  - after reconfiguration 317
  - sort order changes and 317
  - space allocation and 254
- recovery interval in minutes configuration parameter 212–214
- remote logins
  - adding 490–492
    - configuration parameters for 85, 495–497
  - dropping 489, 490
  - options for 494
  - timing out 487
  - trusted or untrusted mode 492
- remote procedure calls 483–497
  - configuration parameters for 495–497
  - example of setting security 521
  - network-based security 516
  - overall process for security model B 519
  - security models for 519
  - setting security options 518
  - unified login and 517
- remote server pre-read packets configuration parameter 215, 497
- remote server users. See remote logins
- remote servers 484–489
  - adding 484–497
  - dropping 489
  - information on 489
  - names of 485
  - options for 487
- remote users. See remote logins
- reorg command

- running manually 363
- reorg reclaim_space command 362
- replay detection 502
- reporting errors 349, 351, 354
- reporting usage statistics 480 reports
  - See also information (server)
  - server usage 480
- reset configuration.
  - See configuration parameters: reconfigure command
  - resetting passwords at first login 449
- resource limits
  - configuring 86
  - response time 241
- restarts, server
  - after reconfiguration 321
  - checkpoints and 283
  - reindexing after 321
  - from same directory 347
  - system tables and 321
  - temporary tables and 30
- retaindays option
  - dump database 239
  - dump transaction 239
- return status system procedures 13
- revoke command 564, 569–587
  - public group and 571
  - revoking
    - create trigger permission 572
    - roles with revoke role 590
  - revoking default permissions from system tables 585
  - revoking default permissions on master database system tables 585
- RFC 86.0 557
- rm_appcontext 620, 623
- role hierarchies 387
  - creating 589
  - displaying 435
  - displaying with role_contain 435
  - displaying with sp_displayroles 435
- role_contain system function 435
- roles
  - activating 415
  - configured for sa login 381
  - deactivating 415
in grant and revoke statements 572, 580
locking 437, 440
maximum login attempts, changing 439
maximum login attempts, setting 438
passwords for 455
permissions and 589
stored procedure permissions and 436
stored procedures and 589, 602
unlocking 440, 441
roles, system
  Operator 5
  system administrator 4
  System Security Officer 5
roles, user-defined
  planning 408
rolling back processes
  recovery interval and 213
  server stack capacity and 231
roman8 character set 102
row lock promotion HWM configuration parameter 215
row lock promotion LWM configuration parameter 216
row lock promotion PCT configuration parameter 217
row lock promotion thresholds
  setting with sp_configure 215, 217
rowlevel access control 607
rows, table
  sysindexes 259
RPCs. See remote procedure calls
rtm thread idle wait period configuration parameter 218
rules
  See also database objects
  protection hierarchy 606
runnable process search count configuration parameter 218
running out of space. See space
running reorg command manually 363
russian character set support 295

S
“sa” login 381
  changing password for 381
  configured with system administrator and system
  security officer roles 381
  security recommendations for using 381
savepoints
  error (Level 13) 350
scan descriptors 171–173
scripts 294
secmech specification 507
secondary database 282
secondary lookup server support 546
secondary lookup servers
  using sp_ldapadmin 547
secure default login 511
secure default login configuration parameter 220
securing login passwords on a network 459
security
  auditing 388
  discretionary access control 386
  establishing after installation 381–383
  identification and authentication controls 384
  Kerberos 527
  login features 436
  roles 387
security administration
  example of 382
  getting started 379–383
  guidelines 381
security drivers
  example of entry in libtcl.cfg file 507
  syntax for entries in libtcl.cfg file 505
security functions 527
security mechanism server option 518
security mechanisms 525
  how the server determines which to support 514
security models 516
  example of model B 521
  for RPCs 517
  model B 519
  setting up model B for RPCs 519
security services
  example 500–501
  overview of 500
  supported by Adaptive Server 501
Index

segmap column, sysusages table
   procedures that change 258
segments 259
   See also database devices; space allocation
creating 255
default 255
logsegment 255
syssegments table 259
system segment 255
select * command
   error message 601
select into/bulkcopy/pllsort database option
   model database and 28
   transaction log and 283
select on syscomments.text column configuration
   parameter 221
send doneinproc tokens 221
sensitive information, views of 600
separation of roles 387
sequence checks 502
server aliases 486
server authentication
   server certificates 692
server certificates 689
   location of 692
   server authentication 692
server information options. See information (server)
server user name and ID 431
server.loc file 329
server_name.cfg, default name of configuration file 62
servers
   See also processes (server tasks); remote servers
   adding new logins to 394–395
   adding users to 394–395
   connecting 16
   dropping logins from 419
   error message severity levels 348–354
   error messages 346
   fatal errors and 352–354
   installing 37, 255
   interfaces files 16
   local 486
   monitoring performance 76
   names of 486
   nonfatal internal errors 352
   passwords on 488, 494
   remote 485–491
   scheduler 241
   shutting down 368
   single-user mode 89, 284
   sort order consistency among 317
   stopping 368
   syntax errors 350
   unlocking logins or roles at startup 441
   user connections to 194
   user information 428–481
   values for configuration parameters 61
server-wide dbcc, master and 583
session authorization option, set 593
set command
   roles and 415
   set options
      exportable 635
set_appcontext 620
   setting timeout for LDAP user authentication 550
setuser command
   show_role and 434
   setuser, using 590
   7-bit ASCII character data, character set conversion for
      333
   severity levels, error 343, 348
      Backup Server 355
      levels 10-18 (user errors) 349
      levels 19-24 (fatal) 352
shared memory starting address configuration
   parameter 222
show_role system function 434
show_sec_services security function 527
shutdown command 368–370
shutting down servers 368
simplified Chinese
   character set support 295
single user database option 284
   single-user mode 89, 321
   site handlers 497
   sites, remote 497
   size
   See also space
      dbcc fix_text transaction 323
      error log 16
      model database 103, 265
      new database 28
tempdb database 29
transaction logs 284

size of auto identity column configuration parameter 223, 280
unique auto_identity index database option and 285
size of global fixed heap configuration parameter 223
size of process object fixed heap configuration parameter 223
size of shared class heap configuration parameter 224
size of unilib cache configuration parameter 225

sjis (Shift-JIS) character set. See Japanese character sets slash (/) converted to pound sign in login names 513

sort order changing 317–321
consistency among servers 317
default sortorder id 106, 107
default XML sortorder 107
definition files 327
installing new 328
numbers 106
rebuilding indexes after changing 322

sp_activeroles system procedure 435
sp_addalias system procedure 426
sp_addauditrecord system procedure 669
sp_addgroup system procedure 396
sp_addlanguage system procedure 326
sp_addlogin system procedure 394–395, 455, 457
sp_addremotelogin system procedure 490–492
sp_addsegment system procedure sysusages and 258
sp_addserver system procedure 485–487
sp_adduser system procedure 28, 396–399
sp_audit system procedure
setting options with 661
sp_changedbowner system procedure 565
sp_changegroup system procedure 396, 423
sp_column_privileges catalog stored procedure 599
sp_configure system procedure 65
See also individual configuration parameter names
configuring server for security services 510
remote logins and 495
sp_countmetadata system procedure 183, 185, 186, 188
sp_dboption system procedure 277–286
sp_deivceattr system procedure 252, 267
sp_diskdefault system procedure 252, 272–273
sp_displaylogin system procedure 430
sp_displayroles system procedure 435
sp_dropalias system procedure 427, 428
sp_dropdevice system procedure 271
sp_dropout system procedure 417
sp_droplogin system procedure 419
sp_dropremotelogin system procedure 490
sp_dropsegment system procedure sysusages and 258
sp_dropserver system procedure 489
sp_dropuser system procedure 416, 417
sp_extendsegment system procedure sysusages and 258
sp_helpconfig system procedure 183, 184, 186
sp_helpdb system procedure 14
database option information 279
sp_helpdevice system procedure 14, 269
sp_helpindex system procedure 14
sp_helpjoins system procedure 11
sp_helpkey system procedure 11
sp_helpremote system procedure 495
sp_helpremotelogin system procedure 495
sp_helpreset system procedure 597–598
sp_helpserver system procedure 489
sp_helptext system procedure 13
sp_helpuser system procedure 428
sp_insuspect system procedure 322
sp_ladadmin 547
sp_locklogin system procedure 418, 419
sp_locklogin trigger 637
sp_maplogin 551
sp_modifylogin system procedure 321, 423, 455, 458
sp_monitorconfig system procedure
configuring number of open databases and 183
configuring number of open indexes and 185
configuring number of open objects and 187, 188
sp_password system procedure 421
sp_passwordpolicy syntax 459
sp_remoteoption system procedure 494–495
sp_reportstats system procedure 480
sp_serveroption net password encryption description 460
sp_serveroption system procedure 487, 518
sp_showplan system procedure 367
sp_showpsexec system command, housekeeper output 360
sp_table_privileges catalog stored procedure 598
sp_who system procedure 429, 596
sp_who, housekeeper output 360
space
See also size; space allocation
running out of 284, 351
space allocation
See also database devices; segments; storage management
commands summary 252
recovery/performance and 254
sysusages table 258
space reclamation
enable housekeeper GC configuration parameter 116
Spanish
character set support 295
#spdevtab temporary table 14
specifying device size, disk resize 275
speed (server)
system performance and 255
#spindtab temporary table 14
spinlocks
lock hash table 145
splitting
tables across two disks 255
SPR files 370
spproc optimize timeout limit configuration parameter 225
spt_committab table 14
spt_monitor table 14
spt_values table 13
SQL batch capture configuration parameter 225
sql_server clock tick length configuration parameter 226
sql_text pipe active configuration parameter 227
sql_text pipe max messages configuration parameter 227, 228
square brackets [ ]
converted to pound sign in login names 513
in SQL statements xxiii
.srt files 327
srvname column, sysservers table 487
srvnetname column, sysservers table 487
SSL
defined 690
enabling SSL 694
filter, defined 691
handshake 690
SSL connections
for companion servers 693
for RPCs 693
Open Client 693
stack guard size configuration parameter 228
stack size configuration parameter 231
standalone utilities and character sets 340
start mail session configuration parameter 232
starting servers
Security Services and 514
startup delay configuration parameter 233
state transitions
LDAP server 548
statement pipe active configuration parameter 233, 234
statement pipe max messages configuration parameter 234
statement statistic active configuration parameter 234
statement statistics active configuration parameter 234
static configuration parameters 62
statistics
housekeeper flushing and 136
I/O usage 480
statistics, flushing with housekeeper task 135
status
information messages (Level 10) 349
status bits in sysservers 270
stem 582
steps
administering security 379
stopping
Backup Server 369
Servers 369
space allocation
storage management 251
commands summary 252
database device initialization 261–270
default database devices 272–273
Index

defaults at installation 255
issues 39–41, 254
See also space 251
system tables and stored procedures 256–259
stored procedure triggers. See triggers stored procedures
See also database objects; system procedures
checking for roles in 436
creating 14
granting execution permission to roles 436
ownership chains 603
permissions granted 571
permissions on 494, 567, 571
procedure cache and 210
remote user access to 494
roles and 602
as security mechanisms 602
system tables changes and 14
strict dtm enforcement configuration parameter 234
structure
internationalization files directory 328
localization files directory 330
suffix names, temporary table 30
suid (server user ID) 395
sun character set 102
suser_id system function 431–433
suser_name system function 431–433
suspect partitions, in cross-platform dump and load 325
suspend audit when device full configuration parameter 235, 654
syb__map_name 535
syb_sendmsg port number configuration parameter 236
Sybase Central, using for system administration tasks 8
SYBASE_PRINCIPAL 534
syblicenseslog table 479
sybmapname 535
sybsecurity database 31, 640
sybsystemdb database 31
sybsystemprocs database 12, 15, 29
See also databases
permissions and 569
symbols
in SQL statements xxii, xxiii
symmetric key encryption 688
syntax
  disk resize 274
dump database 707
erserts in 350
load database 707
syntax conventions, Transact-SQL xxii
sys_session application context table 624, 625
sysalternates table 427
See also syssusers table
sysconfigures table 78
syscurconfigs table 78
sysdevices table 257, 269
disk init and 258
sp_dropdevice and 271
sp_helpdevice and 269
status bits 270
sysindexes table 259, 321
syslogs table
sp_addlogin effect on 394
syslogs table
  modification of 12
syslogs transaction log for sybsecurity 654
sysmessages table 344, 345
sysobjects table 321
sysremotelogins table 492
syssegments table 259
sys.servers table 483, 484, 485, 489
sp_helpserver and 489, 523
srvname column 487
srvnetname column 487
system administration tasks
accomplishing with Sybase Central 8
System Administrator 3–7
error responsibilities of 349, 351–354
resolving system problems 349, 351
tasks for beginners 35–48
system administrator
permissions 564–565
system audit tables 671
system catalogs. See system tables
system databases 23–31
system extended stored procedures 15
system messages. See error messages 343
system problems
See also errors
<table>
<thead>
<tr>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server responses to severity levels 10 to 18</td>
</tr>
<tr>
<td>severity levels 19 to 24</td>
</tr>
<tr>
<td>System Problem Reports (SPRs)</td>
</tr>
<tr>
<td>system procedure tables</td>
</tr>
<tr>
<td>system procedures</td>
</tr>
<tr>
<td>See also information (server); stored procedures;</td>
</tr>
<tr>
<td>individual procedure names</td>
</tr>
<tr>
<td>for adding users</td>
</tr>
<tr>
<td>for changing user information</td>
</tr>
<tr>
<td>for dropping aliases</td>
</tr>
<tr>
<td>for managing remote servers</td>
</tr>
<tr>
<td>permissions</td>
</tr>
<tr>
<td>on temporary tables</td>
</tr>
<tr>
<td>using</td>
</tr>
<tr>
<td>system roles</td>
</tr>
<tr>
<td>activating</td>
</tr>
<tr>
<td>deactivating</td>
</tr>
<tr>
<td>granting with grant role</td>
</tr>
<tr>
<td>max_roles_enabled configuration parameter and show_role and 409</td>
</tr>
<tr>
<td>System Security Officer</td>
</tr>
<tr>
<td>system segment</td>
</tr>
<tr>
<td>system tables</td>
</tr>
<tr>
<td>See also individual table names</td>
</tr>
<tr>
<td>changes allowed to</td>
</tr>
<tr>
<td>changes dangerous to</td>
</tr>
<tr>
<td>corruption</td>
</tr>
<tr>
<td>create database and</td>
</tr>
<tr>
<td>creation of</td>
</tr>
<tr>
<td>dbcc reindex and</td>
</tr>
<tr>
<td>keys for</td>
</tr>
<tr>
<td>permissions on</td>
</tr>
<tr>
<td>querying</td>
</tr>
<tr>
<td>reindexing and</td>
</tr>
<tr>
<td>server restarts and</td>
</tr>
<tr>
<td>storage management relationships</td>
</tr>
<tr>
<td>stored procedures and</td>
</tr>
<tr>
<td>updating</td>
</tr>
<tr>
<td>for user databases</td>
</tr>
<tr>
<td>systemwide password expiration configuration parameter</td>
</tr>
<tr>
<td>sysusages table</td>
</tr>
<tr>
<td>corruption</td>
</tr>
<tr>
<td>sysusers table</td>
</tr>
<tr>
<td>permissions and</td>
</tr>
<tr>
<td>sysalterates table</td>
</tr>
</tbody>
</table>

| T                                                                       |
| Table editor                                                           | 56      |
| table owners. See database object owners                              |         |
| tables                                                                |         |
| See also databases                                                    |         |
| context-sensitive protection of dbcc checktable and 322               |         |
| integrity damage to 353                                               |         |
| object allocation maps of 182                                          |         |
| ownership chains for 603                                              |         |
| permissions information on 598                                        |         |
| permissions on, compared to views 600                                 |         |
| read-only 321                                                         |         |
| splitting across two disks 255                                        |         |
| system procedure 13                                                  |         |
| temporary 29                                                          |         |
| underlying 600                                                       |         |
| without indexes 323                                                  |         |
| tables with suspect partitions, fixing 325                            |         |
| tables, suspect, fixing 325                                           |         |
| tamper detection, digital signature 689                               |         |
| tape retention in days configuration parameter 239                   |         |
| tcp no delay configuration parameter                                  |         |
| tempdb database                                                      | 29–30   |
| See also databases                                                    |         |
| auto identity database option and 280                                 |         |
| creating 255                                                          |         |
| size of 29                                                            |         |
| unique auto_identity index database option and 285                   |         |
| temporary tables 29                                                  |         |
| select into/bulkcopy/plsort database option and 284                   |         |
| terminals                                                             |         |
| character set conversion for 341                                      |         |
| installing new definitions 328                                        |         |
| test servers 36–37                                                    |         |
| text datatype                                                         |         |
| changing character sets and 323                                      |         |
| multibyte character sets and 323                                     |         |
text prefetch size configuration parameter 240
text values, dbcc fix_text upgrade of 323
Thai character set support 295
three housekeepers 360
threshold procedures audit trail 648
time for acquiring locks 146
time slice configuration parameter 241
time values display format 329
timeouts option, sp_serveroption 487
total data cache size configuration parameter 241
traditional Chinese character set support 295
transaction logs alter database and 258
create database and 258
device placement 254, 258
primary and secondary database 282
purging 324
select into/bulkcopy/pllsort database option 283
size 284
trunc log on chkpt option and 213, 284–285
transactions error within 350
long-running 213
recovery and 213
two-phase commit 31
transferring ownership.
See database objects, ownership translation.
See character sets
triggers See also database objects; stored procedures
creating 572
nested 85
permissions and 607
troubleshooting LDAP user authentication 554
trunc log on chkpt database option 284–285
recovery interval in minutes and 213
truncate table syntax 575
trusted mode
remote logins and 494
trusted root certificate

Index

CA certificate 689
location of 692
tuning monitoring performance 76
tuning LDAP user authentication 550
turkish character set support 295
two-phase commit
transactions 31
txn to pss ratio configuration parameter 243

U underlying tables of views (base tables) 600
unichar datatype 296
Unicode 294, 296–300
character sets 295
unichar datatype 296
univarchar datatype 296
UTF-16 297
unified login 501, 511
mapping login names 512
remote procedure security models 517
requiring 511
secure default login 511
unified login required 244
unique auto_identity index database option 285
univarchar datatype 296
UNIX platforms, raw disk partition 263
unlocking
login accounts 441
roles 440, 441
unlocking login accounts 418
unlocking roles 441
unlogged operations 284
untrusted mode, remote logins and 494
update statistics syntax 575
updating See also changing
allow updates to system tables configuration parameter and 14
system procedures and 602
text after character set change 323
upgrade version configuration parameter 245
us_english language 105
Index

usage
disk resize 273
statistics 480
use message confidentiality server option 518
use message integrity server option 518
use security services configuration parameter 245, 510
user connections
memory allocated per 192–194
user databases
See also databases; permissions
master database control of 25
system tables for 28
user-defined messages 348
user errors 349, 349–352
user groups. See groups; public group
user IDs 404
displaying 430
finding 431
number 1, Database Owner 14
user log cache size configuration parameter 246
user log cache spinlock ratio configuration parameter 247
user mistakes. See errors; severity levels, error
user names 431, 568
changing 423
finding 431
preferences 397
user objects. See database objects
user_id system function 433
user_name system function 433
user-defined roles
activating 415
deactivating 415
dropping 417
granting with grant role 588
number of 409
planning 408
Users
creating 55
guest 54
users
See also aliases; groups; logins; remote logins
adding 392–396
aliases 425
application name, setting 424
client host name, setting 424
client name, setting 424
currently on database 429
currently on server 429
deleting 55
dropping from databases 417
dropping from groups 424
dropping from servers 419
errors by 349, 349–352
guest 398, 569
IDs 404, 431
information on 428–481
license use monitoring 477
number of user connections and 194
number or 400
permissions to all or specific 586, 601
remote 490–494
single-user mode 89, 284
views for specific 601
visiting 400
users, object. See database object owners
using proxy authorization 591
UTF-16 297
utility commands
See also Utility Programs manual
character sets and 340
utility, housekeeper, aggressive 361

V
variables in error messages 345
verification, user-access 488, 492
verifying Kerberos authentication 536
Vietnamese
character set support 295
views
See also database objects
dependent 604
ownership chains 603
permissions on 571, 600–602
security and 600
virtual
address 269
page numbers 266
visitor accounts 400
vstart option
Index

disk init 269

W
wait event timing configuration parameter 247
wash, housekeeper task 135
Western Europe
  character set support 295
  window of vulnerability 88
Windows NT LAN Manager security mechanism 509
with grant option option, grant 572
with nowait option, shutdown 369, 370
With override, database option 54
write operations
  physical 255
writetext command
  select into/bulkcopy/pllsort database option 283

X
X/Open XA 113
xact 248
.xfh files 327
XP Server
  freeing memory from 128
  priority 127
xp_cmdshell context configuration parameter 249
xp_cmdshell system extended stored procedure 15