



What's New in Adaptive Server Enterprise?

**Adaptive Server® Enterprise
12.5.1**

DOCUMENT ID: DC37429-01-1251-02

LAST REVISED: November 2003

Copyright © 1989-2003 by Sybase, Inc. All rights reserved.

This publication pertains to Sybase software and to any subsequent release until otherwise indicated in new editions or technical notes. Information in this document is subject to change without notice. The software described herein is furnished under a license agreement, and it may be used or copied only in accordance with the terms of that agreement.

To order additional documents, U.S. and Canadian customers should call Customer Fulfillment at (800) 685-8225, fax (617) 229-9845.

Customers in other countries with a U.S. license agreement may contact Customer Fulfillment via the above fax number. All other international customers should contact their Sybase subsidiary or local distributor. Upgrades are provided only at regularly scheduled software release dates. No part of this publication may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without the prior written permission of Sybase, Inc.

Sybase, the Sybase logo, AccelaTrade, ADA Workbench, Adaptable Windowing Environment, Adaptive Component Architecture, Adaptive Server, Adaptive Server Anywhere, Adaptive Server Enterprise, Adaptive Server Enterprise Monitor, Adaptive Server Enterprise Replication, Adaptive Server Everywhere, Adaptive Server IQ, Adaptive Warehouse, Anywhere Studio, Application Manager, AppModeler, APT Workbench, APT-Build, APT-Edit, APT-Execute, APT-FORMS, APT-Translator, APT-Library, AvantGo, AvantGo Application Alerts, AvantGo Mobile Delivery, AvantGo Mobile Document Viewer, AvantGo Mobile Inspection, AvantGo Mobile Marketing Channel, AvantGo Mobile Pharma, AvantGo Mobile Sales, AvantGo Pylon, AvantGo Pylon Application Server, AvantGo Pylon Conduit, AvantGo Pylon PIM Server, AvantGo Pylon Pro, Backup Server, BizTracker, ClearConnect, Client-Library, Client Services, Convoy/DM, Copernicus, Data Pipeline, Data Workbench, DataArchitect, Database Analyzer, DataExpress, DataServer, DataWindow, DB-Library, dbQueue, Developers Workbench, Direct Connect Anywhere, DirectConnect, Distribution Director, e-ADK, E-Anywhere, e-Biz Integrator, E-Whatever, EC Gateway, ECMAP, ECRTT, eFulfillment Accelerator, Embedded SQL, EMS, Enterprise Application Studio, Enterprise Client/Server, Enterprise Connect, Enterprise Data Studio, Enterprise Manager, Enterprise SQL Server Manager, Enterprise Work Architecture, Enterprise Work Designer, Enterprise Work Modeler, eProcurement Accelerator, EWA, Financial Fusion, Financial Fusion Server, Gateway Manager, GlobalFIX, ImpactNow, Industry Warehouse Studio, InfoMaker, Information Anywhere, Information Everywhere, InformationConnect, InternetBuilder, iScript, Jaguar CTS, jConnect for JDBC, Mail Anywhere Studio, MainframeConnect, Maintenance Express, Manage Anywhere Studio, M-Business Channel, M-Business Network, M-Business Server, MDI Access Server, MDI Database Gateway, media.splash, MetaWorks, My AvantGo, My AvantGo Media Channel, My AvantGo Mobile Marketing, MySupport, Net-Gateway, Net-Library, New Era of Networks, ObjectConnect, ObjectCycle, OmniConnect, OmniSQL Access Module, OmniSQL Toolkit, Open Biz, Open Client, Open ClientConnect, Open Client/Server, Open Client/Server Interfaces, Open Gateway, Open Server, Open ServerConnect, Open Solutions, Optima++, PB-Gen, PC APT Execute, PC Net Library, PocketBuilder, Pocket PowerBuilder, Power++, power.stop, PowerAMC, PowerBuilder, PowerBuilder Foundation Class Library, PowerDesigner, PowerDimensions, PowerDynamo, PowerJ, PowerScript, PowerSite, PowerSocket, Powersoft, PowerStage, PowerStudio, PowerTips, Powersoft Portfolio, Powersoft Professional, PowerWare Desktop, PowerWare Enterprise, ProcessAnalyst, Rapport, Report Workbench, Report-Execute, Replication Agent, Replication Driver, Replication Server, Replication Server Manager, Replication Toolkit, Resource Manager, RW-DisplayLib, S-Designor, SDF, Secure SQL Server, Secure SQL Toolset, Security Guardian, SKILLS, smart.partners, smart.parts, smart.script, SQL Advantage, SQL Anywhere, SQL Anywhere Studio, SQL Code Checker, SQL Debug, SQL Edit, SQL Edit/TPU, SQL Everywhere, SQL Modeler, SQL Remote, SQL Server, SQL Server Manager, SQL SMART, SQL Toolset, SQL Server/CFT, SQL Server/DBM, SQL Server SNMP SubAgent, SQL Station, SQLJ, STEP, SupportNow, S.W.I.F.T. Message Format Libraries, Sybase Central, Sybase Client/Server Interfaces, Sybase Financial Server, Sybase Gateways, Sybase MPP, Sybase SQL Desktop, Sybase SQL Lifecycle, Sybase SQL Workgroup, Sybase User Workbench, SybaseWare, Syber Financial, SyberAssist, SyBooks, System 10, System 11, System XI (logo), SystemTools, Tabular Data Stream, TradeForce, Transact-SQL, Translation Toolkit, UltraLite.NET, UNIBOM, Unilib, Uninull, Unisep, Unistring, URK Runtime Kit for UniCode, Viewer, Visual Components, VisualSpeller, VisualWriter, VQL, WarehouseArchitect, Warehouse Control Center, Warehouse Studio, Warehouse WORKS, Watcom, Watcom SQL, Watcom SQL Server, Web Deployment Kit, Web.PB, Web.SQL, WebSights, WebViewer, WorkGroup SQL Server, XA-Library, XA-Server and XP Server are trademarks of Sybase, Inc. 03/03

Unicode and the Unicode Logo are registered trademarks of Unicode, Inc.

All other company and product names used herein may be trademarks or registered trademarks of their respective companies.

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

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

Contents

About This Book	ix
CHAPTER 1 New Features in Adaptive Server Version 12.5.1	1
Dynamic data cache	2
Automatic database expansion	2
Improvements to the Adaptive Server plug-in for Sybase Central ...	4
Dynamic listeners	5
Globalization enhancements	6
UTF-8 sort order	6
Unicode parser	6
SQL derived tables	6
Date and time datatypes	7
XML mapping	7
Native XML processor	8
LDAP user authentication	9
HA agent on VCS 3.5	10
Fast recovery	11
Database mount and unmount	11
Job Scheduler	12
New cache wizard section for sp_sysmon	13
Web Services for Adaptive Server	14
Enhanced compatibility with Microsoft SQL extensions	14
Adaptive Server 12.5.1 backward compatibility	15
CHAPTER 2 System Changes in Adaptive Server Version 12.5.1	17
System changes in version 12.5.1	17
New and changed configuration parameters	17
New and changed Transact-SQL commands	18
New functions	19
New and changed system procedures	19
New and changed system tables	20
New databases	21
New reserved words	21

	Changes to the documentation	21
CHAPTER 3	New Features in Adaptive Server Version 12.5.0.3.....	23
	New features in version 12.5.0.3.....	23
	Monitoring system tables in Adaptive Server	24
	Index selection	26
	Active-passive configuration for Sun Cluster 3.0.....	26
	Asynchronous log service and optimistic index locking.....	27
	Using sampling for update statistics.....	28
	Multiple temporary databases	28
	Housekeeper enhancements	29
	Chinese character set (GB18030) support.....	30
	SQLJ aliases for XQL.....	30
	SQLX mapping functions.....	30
	New configuration parameters in sp_monitorconfig	30
	Faster checkpoints	31
	Changes to checkstorage.....	31
	Changes to CIS	32
	Improved performance for select into.....	33
	New set parameters	34
	Improved performance for create database, alter database, load database, and dbcc checkalloc	34
	Number of databases per transaction	34
	New global variables	34
	New functions.....	35
	System changes.....	35
	New and changed configuration parameters.....	36
	New and changed Transact-SQL commands	37
	New and changed system procedures	37
CHAPTER 4	New Features in Adaptive Server Version 12.5.0.1.....	43
	New features in version 12.5.0.1	43
	Options for Adaptive Server version 12.5.0.1.....	44
	Using grant dbcc and revoke dbcc	45
	Increasing the size of devices with disk resize.....	45
	ASE Replicator	45
	Synchronizing EJB Servers.....	46
	XQL result sets.....	47
	XQL query engine	48
	Proxy table space.....	49
	Creating files and directories with the insert command.....	49
	Migration utility	50
	High availability on Sun Cluster 3.0.....	50

New character sets.....	50
System changes.....	51
New configuration parameters	51
New and changed Transact-SQL commands	51
Changed system procedures	51
New utility programs	52
Changes to databases and system tables	52

CHAPTER 5 New Features in Adaptive Server Version 12.5..... 53

New features in version 12.5.....	53
Relaxed server limits	54
Dynamic reconfiguration.....	55
SQLJ stored procedures and functions (Java)	56
XML in the database	57
Union in views	58
Internet directory services (LDAP)	58
Secure Sockets Layer (SSL)	58
Enterprise Java Beans (EJB Server).....	59
External file system support	59
Row-level access control.....	60
Suspending and resuming updates to databases	60
Java.net support.....	61
Compressed backups.....	61
unichar support.....	62
Changes to Component Integration Services (CIS)	63
SQL debug utility	63
Migration utility	64
System changes in version 12.5	64
Changes that may affect existing applications	73
Changes that affect Open Client applications	74
Maximum number of expressions in a select statement	76

CHAPTER 6 New Features and System Changes in Adaptive Server Version 12.0 77

New features in version 12.0.....	77
Sybase Failover for a high availability system	78
Rebuilding indexes.....	80
New Backup Server features	81
High performance backup and restore	81
Non-rewinding tapes	81
UNIX file system support.....	82
Modifying an existing table's schema with alter table	82
Suspending database updates with quiesce database	83

Taking Adaptive Server engines offline.....	83
User settable process information.....	83
Removed limit for number of fragments per database	84
Identity number gap for tables.....	84
Diagnostic database.....	85
Java in Adaptive Server Enterprise.....	85
Capabilities of Java in Adaptive Server.....	86
Number of tables in a query	87
Query processing and optimization enhancements	87
Merge joins.....	87
Query costing improvements.....	88
ANSI joins	88
Dynamic execution of Transact-SQL	89
text and image datatype enhancements	90
Abstract plans	90
Disabling triggers	91
Cache partitions	91
Distributed Transaction Management	92
Enhancements to Component Integration Services (CIS)	92
Network security.....	93
Performance considerations.....	94
Security features supported in Adaptive Server 12.0	94
User-defined login security.....	94
Increased maximum number of users and logins.....	95
Setting and changing the maximum login attempts.....	95
Locking and unlocking logins and roles.....	96
Displaying password information.....	96
Checking passwords for at least one character	96
Setting and changing minimum password length.....	97
Setting the expiration interval for a password	97
Concrete identification.....	98
System changes in version 12.0	98
New and changed configuration parameters.....	99
New and changed Transact-SQL commands	100
New and changed system procedures	101
New utility programs.....	103
New and changed databases and system tables.....	104
New reserved words.....	105
Changes to the documentation	106
Changes that may affect existing applications	106
Inserts on datarow-locking tables.....	106
New Transact-SQL keywords for 12.0	107
dsync on by default for UNIX file devices.....	107

Index.....	109
-------------------	------------

About This Book

Audience

This manual is for customers who are upgrading to Adaptive Server® version 12.5.x.

How to use this book

What's New in Adaptive Server Enterprise? is an introduction to the new Sybase® Adaptive Server Enterprise features and the commands, system procedures, system tables, and documentation that supports them.

This manual consists of the following chapters:

- Chapter 1, “New Features in Adaptive Server Version 12.5.1,” describes the features added to the 12.5.0.3 version of Adaptive Server.
- Chapter 2, “System Changes in Adaptive Server Version 12.5.1,” describes new and changed configuration parameters, Transact-SQL commands, system procedures, databases and system tables, and reserved words.
- Chapter 3, “New Features in Adaptive Server Version 12.5.0.3,” describes the features added to the 12.5.0.1 version of Adaptive Server.
- Chapter 4, “New Features in Adaptive Server Version 12.5.0.1,” describes the features added to the 12.5 version of Adaptive Server.
- Chapter 5, “New Features in Adaptive Server Version 12.5,” describes the features added to the 12.5.1 version of Adaptive Server.
- Chapter 6, “New Features and System Changes in Adaptive Server Version 12.0” describes the version 12.0 Adaptive Server new features, system changes, and how these additions and changes might affect your existing applications.

Related documents

The Sybase® 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.
- *ASE Replicator User's Guide* – describes how to use the ASE 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.
- *Configuring Adaptive Server Enterprise* for your platform – provides instructions for performing specific configuration tasks for Adaptive Server.
- *EJB Server User's Guide* – explains how to use EJB Server to deploy and execute Enterprise JavaBeans in Adaptive Server.
- *Error Messages and Troubleshooting Guide* – explains how to resolve frequently occurring error messages and describes solutions to system problems frequently encountered by users.
- *Full-Text Search Specialty Data Store User's Guide* – describes how to use the Full-Text Search feature with Verity to search Adaptive Server Enterprise data.
- *Glossary* – defines technical terms used in the Adaptive Server documentation.
- *Historical Server User's Guide* – describes how to use Historical Server to obtain performance information for SQL Server® and Adaptive Server.
- *Java in Adaptive Server Enterprise* – describes how to install and use Java classes as data types, 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).

- *Monitor Client Library Programmer's Guide* – describes how to write Monitor Client Library applications that access Adaptive Server performance data.
- *Monitor Server User's Guide* – describes how to use Monitor Server to obtain performance statistics from SQL Server and Adaptive Server.
- *Performance and Tuning Guide* – is a series of four books that explains how to tune Adaptive Server for maximum performance:
 - *Basics* – the basics for understanding and investigating performance questions in Adaptive Server.
 - *Locking* – describes how the various locking schemas can be used for improving performance in Adaptive Server.
 - *Optimizer and Abstract Plans* – describes how the optimizer processes queries and how abstract plans can be used to change some of the optimizer plans.
 - *Monitoring and Analyzing* – explains how statistics are obtained and used for monitoring and optimizing performance.
- *Quick Reference Guide* – provides a comprehensive listing of the names and syntax for commands, functions, system procedures, extended system procedures, datatypes, and utilities in a pocket-sized book.
- *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* – provides in-depth information about administering servers and databases. This manual includes instructions and guidelines for managing physical resources, security, user and system databases, and specifying character conversion, international language, and sort order settings.
- *System Tables Diagram* – illustrates system tables and their entity relationships in a poster format. Available only in print version.

-
- *Transact-SQL User's Guide* – documents Transact-SQL, Sybase's 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.
 - *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's Failover to configure an Adaptive Server as a companion server in a high availability system.
 - *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 Sybase Technical Library CD and the Technical Library 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 Technical Library CD. It is included with your software. To read or print documents on the Getting Started CD you need Adobe Acrobat Reader (downloadable at no charge from the Adobe Web site, using a link provided on the CD).
- The Technical Library CD contains product manuals and is included with your software. The DynaText reader (included on the Technical Library CD) allows you to access technical information about your product in an easy-to-use format.

Refer to the *Technical Library Installation Guide* in your documentation package for instructions on installing and starting the Technical Library.

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

To access the Technical Library Product Manuals Web site, go to Product Manuals at <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

- 1 Point your Web browser to Technical Documents at <http://www.sybase.com/support/techdocs/>.
- 2 Select Products from the navigation bar on the left.
- 3 Select a product name from the product list and click Go.
- 4 Select the Certification Report filter, specify a time frame, and click Go.
- 5 Click a Certification Report title to display the report.

❖ 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.

- 1 Point your Web browser to Technical Documents at <http://www.sybase.com/support/techdocs/>.
- 2 Click MySybase and create a MySybase profile.

Sybase EBFs and software updates

❖ Finding the latest information on EBFs and software updates

- 1 Point your Web browser to the Sybase Support Page at <http://www.sybase.com/support>.
- 2 Select EBFs/Updates. Enter user name and password information, if prompted (for existing Web accounts) or create a new account (a free service).
- 3 Select a product.
- 4 Specify a time frame and click Go.

-
- 5 Click the Info icon to display the EBF/Update report, or click the product description to download the software.

Conventions

The following style conventions are used in this manual:

- In a sample screen display, commands you should enter exactly as shown are given in:

`this font`

- In a sample screen display, words which you should replace with the appropriate value for your installation are shown in:

this font

- In the regular text of this document, the names of files and directories appear in italics:

/usr/u/sybase

- The names of programs, utilities, procedures, and commands appear in bold type:

bcp

- Commands for both the C and Bourne shells are provided in this document when they differ. The initialization file for the C shell is called *.cshrc*. The initialization file for the Bourne shell is called *.profile*. If you are using a different shell, such as the Korn shell, refer to your shell-specific documentation for the correct command syntax.

The conventions for syntax statements in this manual are as follows:

Table 1: SQL syntax conventions

Key	Definition
command	Command names, command option names, utility names, utility flags, and other keywords are in Arial bold.
<i>variable</i>	Variables, or words that stand for values that you fill in, are in Italics.
{ }	Curly braces indicate that you choose at least one of the enclosed options. Do not include braces in your option.
[]	Brackets mean choosing one or more of the enclosed options is optional. Do not include brackets in your option.
()	Parentheses are to be typed as part of the command.
	The vertical bar means you can select only one of the options shown.
,	The comma means you can choose as many of the options shown as you like, separating your choices with commas to be typed as part of the command.

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.

New Features in Adaptive Server Version 12.5.1

This chapter describes the new features and system changes introduced with Sybase Adaptive Server Enterprise version 12.5.1.

The new features in Adaptive Server version 12.5.1 are:

Feature
“Dynamic data cache” on page 2
“Automatic database expansion” on page 2
“Improvements to the Adaptive Server plug-in for Sybase Central” on page 4
“Dynamic listeners” on page 5
“Globalization enhancements” on page 6
“SQL derived tables” on page 6
“Date and time datatypes” on page 7
“XML mapping” on page 7
“Native XML processor” on page 8
“LDAP user authentication” on page 9
“HA agent on VCS 3.5” on page 10
“Fast recovery” on page 11
“Database mount and unmount” on page 11
“Job Scheduler” on page 12
“New cache wizard section for sp_sysmon” on page 13
“Web Services for Adaptive Server” on page 14
“Enhanced compatibility with Microsoft SQL extensions” on page 14
“Adaptive Server 12.5.1 backward compatibility” on page 15

Dynamic data cache

Adaptive Server 12.5.1 allows you to dynamically alter the data cache without having to restart the server, as was required in earlier versions. Using dynamic `sp_cacheconfig` actions, you release memory so that it can be reallocated immediately.

Some of the actions you can perform dynamically with `sp_cacheconfig` are:

- Adding a new cache
- Adding memory to an existing cache
- Deleting a cache, either log or mix
- Changing a cache type

If Adaptive Server cannot allocate all the memory requested, it allocates the available memory. The additional memory is allocated when you restart Adaptive Server.

Reducing cache size is not a dynamic action. Rather than using `sp_cacheconfig` to reduce cache size as a static action, remove the cache and recreate it with the correct size.

For more information, see the *System Administration Guide*.

Automatic database expansion

Adaptive Server Enterprise version 12.5.1 provides automatic expansion of databases and devices. Databases can be configured to expand automatically when they run out of space.

The automatic database expansion stored procedure `sp_dbextend` allows you to install thresholds that identify which devices have room and then appropriately alter the database—and the segment where the threshold was fired—on these devices.

`sp_dbextend` is the user interface for the automatic expansion process. DBAs use it to:

- Define site-specific rules for individual segments and devices
- Change user-specified policies and system-supplied default behaviours
- Examine the current state of the rules

DBAs can also use this interface to execute trial simulations of the database expansion machinery, and if necessary, to manually perform a database expansion without waiting for the automatic expansion process to be scheduled.

After you set up the database for automatic expansion, when a database grows to its free space threshold, internal mechanisms fire, increasing the size of the database by the amount of space specified by expansion policies. The automatic expansion process measures the amount of room left on all devices bound to the database. If there is sufficient room on the device, the database continues to grow. If any devices are configured for expansion, those devices are expanded next, followed by expansion of the database on those devices. The automatic expansion process runs as a background task and generates informational messages in the server's error log about its progress.

The automatic expansion process requires new installation procedures. You install it using the *installdbextend* script, which loads new rows into *master.dbo.sysattributes*—rows that describe defaults for autoexpansion in a database or in a device. You can install the automatic expansion process in one or more databases or one or more devices.

You can configure automatic expansion to run with server-wide default expansion policies, or you can customize it for individual segments in specified databases. You can install thresholds on key segments on which tables with critical data reside, allowing you to have a fine degree of control over how Adaptive Server meets the data space requirements for different kinds of tables. If your site has key tables with large volumes of inserts, you can bind these tables to specific segments, with site-specific rules of extending that segment. This enables you to avoid outages that can occur in a production environment due to large loads to such key tables.

Automatic expansion procedures allow you to simulate the expansion processes based on your chosen policies. This simulation allows you to make sure the expansion processes work without jeopardizing your production environment.

Note The automatic expansion procedure does not create new devices; it only alters the size of the database and segment on existing devices to which the segment currently maps.

For more information, see Chapter 30, “Automatic Database Expansion,” in the *System Administration Guide*.

Improvements to the Adaptive Server plug-in for Sybase Central

The Adaptive Server plug-in for Sybase Central has been enhanced for 12.5.1. The Adaptive Server plug-in, along with Sybase IQ and Adaptive Server Anywhere, now runs on Sybase Central version 4.1. This version of Sybase Central provides better performance, requires less memory, and improves the user interface.

The 12.5.1 Adaptive Server plug-in:

- Lets you view and export data from a table or a view using the new “Data” panel
- Lets you view SQL code of compiled objects—stored procedures, views, rules, and so on—using the new “Code” panel
- Lets you log SQL commands in a window or file so that a record exists of actions by the plug-in
- Provides robust interfaces file support
- Provides an enhanced table editor to support column properties and constraint bindings
- Lets you manage multiple tempdbs
- Provides Quiesce and Mount/Unmount Database wizards
- Provides an enhanced Proxy Table wizard for managing proxy tables on data sources other than Adaptive Server
- Lets you manage the Job Scheduler
- Lets you shut down Adaptive Server
- Lets you resize disks
- Lets you use the Monitor Server graphics interface on Windows platforms

The Adaptive Server plug-in for 12.5.1 is backward compatible to Adaptive Server version 12.0. Sybase recommends that users managing pre-12.5.1 servers upgrade Sybase Central to version 4.1 and the Adaptive Server plug-in to 12.5.1.

Dynamic listeners

This version of Adaptive Server adds functionality to `sp_listener` that allows you to manage network listeners. `sp_listener` allows you to:

- Start additional listeners (the maximum number of listeners is 32)
- Stop listeners
- Suspend listeners
- Resume suspended listeners

The syntax for `sp_listener` is:

```
sp_listener "command", "server_name", engine | remaining
```

Or:

```
sp_listener "command", "[protocol:]machine:port", engine | remaining
```

Where:

- *command* is start, stop, suspend, resume, or status.
- *server_name* is the name of Adaptive Server.
- *engine* specifies the number of the engine affected by this command (this parameter is ignored by Windows NT). *engine* can be a single-engine number in quotes ("2"), a list ("3,5,6"), a range ("2-5"), or mix ("2,3-5,7").
- *remaining* specifies that the command is to take effect on all engines on which it can be meaningfully applied (that is, where the listener is in a state in which the command can take effect).
- *protocol* is the protocol used (tcp, tli, ssltcp, ssltli, winsock, sslnlwnsck, or sslwinsock).
- *machine:port* is the machine name and port number to which the listener listens (as specified in the interfaces file).

The number of listener ports is determined by the dynamic configuration parameter `number of network listeners`. The default value of this parameter is 1. The semantics for `sp_listener` is atomic: if a command cannot be completed successfully, it is aborted.

For more information, see the *Performance and Tuning Guide*.

Globalization enhancements

Adaptive Server 12.5.1 supports the UTF-8 sort order and the Unicode parser.

UTF-8 sort order

In versions of Adaptive Server earlier than 12.5.1, when using UTF-8, the two sort order choices were binary and no-case (ASCII-only). Adaptive Server version 12.5.1 provides the ability to perform nonbinary sort orders in UTF-8. As such, all sort orders available for the `unichar` and `univarchar` datatypes can be used for `char` or `varchar` data when the default character set is set to UTF-8.

Unicode parser

In versions of Adaptive Server earlier than 12.5.1, you had to set the server's default character set to UTF-8 to use the `unichar` and `univarchar` datatypes.

This restriction has been removed in version 12.5.1. You can now use `unichar` and `univarchar` datatypes with any default character set without first having to configure your default character set to UTF-8.

By incorporating syntax from the new SQL2002 standard, you can now use escape sequences to specify any Unicode character in a string literal. If a string literal (whether or not it makes use of the new syntax) contains characters that cannot be represented in the server's default character set, the literal is promoted from the `varchar` datatype to the `univarchar` datatype.

See Chapter 7, “Configuring Character Sets, Sort Orders, and Languages” in the *System Administration Guide* for more information.

SQL derived tables

A SQL derived table is created with a nested `select` statement, as in the following example:

```
select * from (select * from table_1) derived_table_1
```

An equivalent group of SQL statements that uses a view instead of a SQL derived table requires both `create` and `drop` statements:

```
create view view_1 as select * from table_1
select * from view_1
drop view view_1
```

The group of SQL statements using a view is more complicated, requiring three statements. Also, the benefits of creating a view for only one query are offset by the overhead of making a system catalog entry. The SQL derived tables feature eliminates this overhead by enabling queries to spontaneously create non-persistent tables without needing to drop the tables or make insertions into the system catalog. This makes a SQL derived table more desirable than a group of SQL statements using a view for ad hoc queries. For repeated queries, a SQL derived table used multiple times performs comparably to a SQL statement using a view with a cached definition. A SQL derived table differs from a temporary table in that a SQL derived table exists only for the duration of the query, while a temporary table exists until the server is brought down.

A system catalog entry occurs for a view that uses a SQL derived table:

```
create view view_1 as
select column_1 from
(select column_1 from table_1) derived_table_1
```

This is likewise true for a stored procedure that uses a SQL derived table:

```
create proc sp_foo @name varchar(40) as
select column_1 from
(select column_1 from table_1) derived_table_1
```

Date and time datatypes

Adaptive Server has various ways to identify date and time. Before version 12.5.1, only `datetime` and `smalldatetime` were available. As of version 12.5.1, date and time have been added as separate datatypes.

XML mapping

The `for xml` clause in select statements and the `forxmlj` function map SQL result sets to SQLX-XML documents, using the SQLX-XML format defined by the ANSI SQLX standard. For example:

```
1> select name, type from systypes
2> where name like "%var%"
3> for xml
4> go
```

```
-----
<resultset xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <row>
    <name>nvarchar</name>
  </row>
  <row>
    <name>univarchar</name>
  </row>
  <row>
    <name>varbinary</name>
  </row>
  <row>
    <name>varchar</name>
    <type>39</type>
  </row>
</resultset>
```

For more information, see *XML Services in Adaptive Server Enterprise*.

Native XML processor

The native XML processor supports SQL extensions that perform XML query functions on XML documents. The native XML processor can query both stored XML documents and XML documents mapped from SQL by the `for xml` clause or `forxmlj` function.

Integration of the native XML processor in Adaptive Server 12.5.1 provides remarkable performance improvements over the Java-based XQL processor of earlier versions of Adaptive Server.

The native XML processor supports standard XML documents and standard XPath queries, which are a subset of the new XQuery language. *XML Services in Adaptive Server* describes the subsets of those standards that the native XML processor supports.

The SQL extensions supported by the native XML processor are:

- `xmlextract` – a built-in function that applies an XML query expression to an XML document and returns the select result.

- `xmltest` – a SQL predicate that applies an XML query expression to an XML document and returns the Boolean result.
- `xmlparse` – a built-in function that parses an XML document for more efficient processing.
- `xmlrepresentation` – a built-in function that determines whether a given image column contains a parsed XML document.

For more information, see *XML Services in Adaptive Server Enterprise*.

LDAP user authentication

LDAP is an industry standard for accessing directory services over a network. With version 12.5.1, Adaptive Server extends its LDAP support to include storage of user information.

With LDAP services enabled:

- Adaptive Server authenticates clients with data from an LDAP server.
Users authenticate with passwords stored on an LDAP server rather than in the syslogins catalog. The LDAP server provides a centralized location for login accounts—both names and passwords.
- Adaptive Server servers share user login data stored on the LDAP server.
Information formerly stored in syslogins is now managed and stored on an LDAP server. It is cached locally to preserve referential integrity and for other, database-specific uses.
- With LDAP enabled, users have a single login and password throughout the enterprise.

Adaptive Server support for LDAP requires an ASE_DIRS license. To set up accounts, you must:

- Add user login accounts to the LDAP server.
- Construct an LDAP URL search string for user authentication to Adaptive Server using `sp_ldapadmin`.
- Set the Adaptive Server configuration parameter `enable ldap user auth` to authorize the use of LDAP.

After a user account is added to the LDAP server, Adaptive Server can modify local characteristics of that account. A System Administrator or a System Security Officer can add a row in syslogins using `sp_addlogin` to set login-specific values—such as a default database or the granting of roles.

To aid migration of user accounts to the LDAP server, enable `ldap user auth` provides a setting that allows authentication through either the LDAP server or syslogins.

For more information, see the *System Administration Guide*.

HA agent on VCS 3.5

Adaptive Server version 12.5.1 introduces a new HA agent that adds support for active-active high availability on Veritas Cluster 3.5.

The new agent has these components:

- Resource definition file
- Agent binary with monitor function
- Agent scripts
- Agent installation tool

The new HA agent delivers several benefits:

- It provides a complete, independent solution and requires only an Adaptive Server HA license. Enterprise Agent for Sybase, developed by Veritas, is not necessary.
- It is easily installed using the included installation tool.
- It eliminates the need to overwrite Enterprise Agent scripts for Sybase products.
- It is smoothly and tightly integrated with Adaptive Server.

For more information, see *Using Sybase Failover in a High Availability System*.

Fast recovery

During a server restart after a planned or unplanned shutdown, or during HA failover, a significant amount of time can be spent on database recovery. The fast recovery feature enables databases to be available earlier and minimizes database downtime. It can aid in recovery from planned shutdowns, unplanned crashes, and HA failover.

Fast recovery enables the following changes to Adaptive Server:

- Brings engines online earlier after all system databases are recovered.
- Enhances performance of database recovery.
- Recovers multiple databases in parallel by making use of available server resources. This is controlled by the configuration parameter `max concurrently recovered db`.
- Provide for multiple checkpoint tasks at runtime that can run concurrently to minimize the work that needs to be done at recovery time. This is controlled by the configuration parameter `number of checkpoint tasks`.
- New global variable `@@recovery_state` to determine if Adaptive Server is in recovery.
- New option `strict | relax` to the stored procedure `sp_dbrecovery_order`.

For more information see “Fast Recovery” in the *System Administration Guide*.

Database mount and unmount

The purpose of the mount and unmount commands is to make it possible to transport databases from one source Adaptive Server to another destination Adaptive Server.

When you copy a database you must use a command outside of Adaptive Server, such as UNIX `dd` or `ftp`, to create a byte-for-byte copy of all pages in a set of one or more databases.

When you move a set of databases from a source Adaptive Server to a destination Adaptive Server, you are physically moving the underlying devices.

For more information, see Chapter 22, “Database Mount and Unmount,” in the *System Administration Guide*.

Job Scheduler

Job Scheduler eases the management of Adaptive Server by providing the ability to define and schedule database administration tasks. With Job Scheduler, jobs that normally require interaction from a database administrator can be scheduled to run unattended at the appropriate times, freeing the database administrator to attend to other issues.

Job Scheduler allows you to create and schedule jobs, and also share jobs and schedules. One database administrator can create a job and other database administrators can then schedule and run that job on another server. Jobs can be created from scratch, command line or GUI, loaded from a SQL batch file, or generated from a predefined template.

Job Scheduler captures the results and output of jobs, and records that information in log tables. Job Scheduler keeps a history of scheduled jobs. However, to keep a limit on the size of the history table, Job Scheduler is self-monitoring and removes outdated, unnecessary history records.

Job Scheduler is comprised of the following components:

- An internal Adaptive Server task
- An external process called the JS Agent
- The *sybmgmtdb* database and stored procedures
- The graphical user interface
- Predefined templates from which the database administrator may create and schedule useful, time-saving jobs

The internal ASE task determines when scheduled jobs should run and creates a historical record of jobs that are run. It starts the JS Agent process and feeds JS Agent the necessary information to retrieve job information and run the job on the specified ASE.

The JS Agent retrieves the job information from Job Scheduler's own database, called *sybmgmtdb*. Then it logs in to the target ASE and issues the job commands. When the job completes, JS Agent logs any result or output to the log tables in the *sybmgmtdb* database.

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

The GUI assists the user in creating and scheduling jobs, viewing job status and job history and in controlling jobs. The GUI also provides an administration feature to turn on and off the ASE internal task and therefore the ability of Job Scheduler to process and execute scheduled jobs.

Templates are an important tool in defining tasks for self-management of the database, such as database backups, reorganization rebuilds, modification of configuration parameters, and statistics updates and monitoring. They are implemented as batch Transact-SQL commands for which parameter values can be provided. Database administrators can use templates to generate jobs, which may then be scheduled to run at desired times.

For more information, see the *Job Scheduler User's Guide for Adaptive Server Enterprise*.

New cache wizard section for *sp_sysmon*

sp_sysmon for Adaptive Server 12.5.1 has been enhanced. A new cache wizard parameter aids in the monitoring and configuring of data caches for enhanced performance. The cache wizard allows you to identify:

- “Hot” objects (objects that are often accessed). The output is ranked by the number of logical reads in a named cache or default data cache.
- The usage of the cache and buffer pools.
- The percentage of hits at a cache, buffer pool, and object level.
- The effectiveness of large I/O.
- The effectiveness of Asynchronous Prefetch (APF).
- The cache occupancy by the various objects.

The cache wizard section appears in the *sp_sysmon* output only when you include the cache wizard parameter. You can include two parameters with cache wizard, *topN* and *filter*. *topN* ranks the objects while printing the report and *filter* prints cache names containing this pattern.

For more information about the cache wizard, see *Performance and Tuning: Monitoring and Analyzing*.

Web Services for Adaptive Server

A Web service is a self-contained, modular application that can be accessed and used over a network connection using the SOAP, WSDL, and XML open standards. Adaptive Server Enterprise Web Services enables Adaptive Server Enterprise to both provide and use Web services.

Adaptive Server Enterprise Web Services consists of a Producer component, which enables client applications to access SQL and stored procedures in Adaptive Server using SOAP.

For more information, see the *Web Services User's Guide for Adaptive Server Enterprise*.

Enhanced compatibility with Microsoft SQL extensions

Adaptive Server 12.5.1 provides enhanced compatibility with Microsoft SQL extensions.

Newly supported features let you:

- Use default in the insert statement. For example:

```
create table test_table (col1 tinyint,  
                        col2 tinyint default 10, col3 tinyint)  
insert test_table values (11, default, 43)
```

- Use set for variable and value assignments. For example:

```
declare @a tinyint ; set @a = 10
```

- Use square brackets surrounding identifier names. For example:

```
create table [toto2] (a tinyint)
```

- Use the new built-in function cast() and those added in version 12.5.0.3: len(), left(), day(), month(), year(), str_replace(), newid(), square().
- Use derived tables, which can be used wherever a view can be used, and which can be used in the from list of select, select into, create view, and insert statements.
- Migrate more easily between Adaptive Server and MS-SQL Server.

Adaptive Server 12.5.1 backward compatibility

Before using the new Adaptive Server 12.5.1 features, make sure that you read the Technote “Adaptive Server 12.5.1 Backward Compatibility,” which is available at <http://www.sybse.com/support/techdocs>.

You can downgrade to previous 12.5.x versions of Adaptive Server, but to do so you must take the precautions described in the technical note *before* using any of the new 12.5.1 features.

System Changes in Adaptive Server Version 12.5.1

This chapter describes the system changes introduced in Adaptive Server version 12.5.1.

System changes in version 12.5.1

This section provides a summary of the version 12.5.1 changes, as follows:

- “New and changed configuration parameters” on page 17
- “New and changed Transact-SQL commands” on page 18
- “New functions” on page 19
- “New and changed system procedures” on page 19
- “New and changed system tables” on page 20
- “New databases” on page 21
- “New reserved words” on page 21
- “Changes to the documentation” on page 21

New and changed configuration parameters

This section describes the new and changed configuration parameters in version 12.5.1.

Table 2-1 summarizes the new configuration parameters:

Table 2-1: New configuration parameters

Parameter	Function
enable ldap user auth	Enables the storage of user login information on the LDAP server.
enable job scheduler	Enables the Job Scheduler.

Table 2-2 summarizes the changes to existing configuration parameters.

Table 2-2: Changed configuration parameters

Parameter	Change
number of user connections	One connection is reserved for temporary administrative tasks, and has a total login duration of 15 minutes. The reserved connection ensures the availability of connections for database administrators when there are no free connections and number of user connections needs to be increased.

New and changed Transact-SQL commands

This section describes the new and changed Transact-SQL commands in version 12.5.1.

Table 2-3 summarizes the new Transact-SQL commands.

Table 2-3: New commands

Command	Function
mount	Use the mount command to add information for devices and other attributes for the database to the destination or secondary Adaptive Server. The mount command decodes the information in the manifest file and makes the set of databases available online.
unmount	The unmount command shuts down the database and drops it from the Adaptive Server. The devices are also deactivated and dropped.

Table 2-4 summarizes the changes to existing Transact-SQL commands:

Table 2-4: Changed commands

Command	Change
select	A SQL-derived table can be used in the from clause of a select statement. This includes a select statement that appears in an insert or create view statement. For more information, see the <i>Adaptive Server Enterprise Reference Manual</i> .
checkpoint	Enhancements: <ul style="list-style-type: none"> • Allows specification of an optional list of target databases or the keyword all. • Performs an automatic checkpoint for sp_dboption, which requires checkpoint to continue. • Permission to execute checkpoint is granted to the oper_role.

New functions

Table 2-5 describes new function for 12.5.1.

Table 2-5: New functions

Function	Description
current_date	Returns the current date as a date datatype.
current_time	Returns the current time as a time datatype.

New and changed system procedures

This section describes the new system procedures added and changes made to existing system procedures.

Table 2-6 summarizes the new system procedures:

Table 2-6: New system procedures

System procedure	Function
sp_dbextend	User interface for the automatic expansion process.
sp_ldapadmin	Creates an LDAP URL search string, lists an LDAP URL search string, or verifies an LDAP URL search string or login.
sp_listener	Dynamically starts and stops listeners on Adaptive Server on any given port on a per-engine basis.

Table 2-7 summarizes the changes made to existing system procedures.

Table 2-7: Changed system procedures

Procedure	Change
sp_addserver	<i>Component Integration Services only</i> – You can use <i>pname</i> to specify the hostname or IP address and the port of the server you wish to connect to.
sp_bindcache and sp_unbindcache	Although you can still use sp_bindcache and sp_unbindcache on a system tempdb, the binding of the system tempdb is now non-dynamic.
sp_cacheconfig	Dynamically starts and stops listeners on Adaptive Server on any given port on a per-engine basis.
sp_dbcc_evaluatedb	The output for sp_dbcc_evaluatedb is now more compact than before.
sp_dbcc_faultreport	Provides a new parameter @hard_only, which specifies a report of hard faults only.
sp_dboption	Now includes the checkpoint command. Users no longer need to execute checkpoint after executing sp_dboption.
sp_serveroption	The procedure has the following additional options: <ul style="list-style-type: none">• external engine auto start• server cost• server login
sp_sysmon	Now includes the cache wizard option, which allows you to monitor and configure data caches.

New and changed system tables

There are no new or changed system tables for Adaptive Server 12.5.1. For more information about system tables, see the *Adaptive Server Reference Manual*.

New databases

Table 2-8 lists the databases that are new in this version.

Table 2-8: New databases

Database	Function
sybmgmtdb	Job Scheduler has its own database, called sybmgmtdb. Any results or output are logged to the log tables in the sybmgmtdb database. All the job, schedule, scheduled job information, and data needed by the Job Scheduler Task for internal processing is stored in the sybmgmtdb database. Most access to data in the sybmgmtdb database is through stored procedures. The stored procedures make the data available to the GUI, the Job Scheduler Agent and the command-line interface. Only the Job Scheduler Task accesses data directly from the sybmgmtdb database.

New reserved words

There are no new reserved words for Adaptive Server 12.5.1.

The *Adaptive Server Reference Manual* lists all of the reserved words for version 12.5.1. Reserved words cannot be used as object names or column names.

You must change all database names that are new reserved words before you can upgrade from an earlier version of Adaptive Server. You can change table, view, and column names, or use delimited identifiers. Once you upgrade to version 12.5.1, you cannot use database objects whose names are new reserved words until you modify your procedures, SQL scripts, and applications.

Changes to the documentation

Adaptive Server version 12.5.1 includes the following new books, which describe separately licensed Adaptive Server features:

- *Web Services User's Guide* – describes the Web Services feature, and provides instructions for configuration, administration, and troubleshooting.

- *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. This book includes XML information previously described in *Java in Adaptive Server*.
- *Job Scheduler User's Guide* – provides instructions on installing and configuring, and creating and scheduling jobs on a local or remote Adaptive Server using the command line or a graphical user interface (GUI).

New Features in Adaptive Server Version 12.5.0.3

This chapter describes the new features and system changes introduced with Sybase Adaptive Server Enterprise version 12.5.0.3.

New features in version 12.5.0.3

The new features in Adaptive Server version 12.5.0.3 are:

Name
“Monitoring system tables in Adaptive Server” on page 24
“Index selection” on page 26
“Active-passive configuration for Sun Cluster 3.0” on page 26
“Asynchronous log service and optimistic index locking” on page 27
“Using sampling for update statistics” on page 28
“Multiple temporary databases” on page 28
“Housekeeper enhancements” on page 29
“Chinese character set (GB18030) support” on page 30
“SQLJ aliases for XQL” on page 30
“SQLX mapping functions” on page 30
“New configuration parameters in sp_monitorconfig” on page 30
“Faster checkpoints” on page 31
“Changes to checkstorage” on page 31
“Changes to CIS” on page 32
“Improved performance for select into” on page 33
“New set parameters” on page 34
“Improved performance for create database, alter database, load database, and dbcc checkalloc” on page 34
“Number of databases per transaction” on page 34
“New global variables” on page 34
“New functions” on page 35

Monitoring system tables in Adaptive Server

Adaptive Server 12.5.0.3 includes a set of system tables that contain monitoring and diagnostic information. The information in these tables provides you with a statistical snapshot of the state of Adaptive Server, which allows you to analyze the server for performance improvements. You can query these system tables in much the same way you currently query any other tables in Adaptive Server. For example, to display statistical information about I/O on Sybase devices:

```
select * from monDeviceIO
Reads      APFReads      Writes      DevSemaphoreRequests  DevSemaphoreWaits  IOTime
LogicalName PhysicalName
-----
1563      7      7891      3      0      134900
master      /dev/vx/rdisk/sybase/master_vol01
59      0      15      2      0      800
engcomdb_data_vol01      /dev/vx/rdisk/sybase/engcomdb_data_vol01
5      0      13      2      0      100
engcomdb_log_vol01      /dev/vx/rdisk/sybase/engcomdb_log_vol01
126255  59657      8604      2      0      1408700
qts_db_data_vol01      /dev/vx/rdisk/sybase/qts_db_data_vol01
31      0      9879      2      0      128400
qts_db_log_vol01      /dev/vx/rdisk/sybase/qts_db_log_vol01
51      0      19      2      0      400
sadb_data_vol01      /dev/vx/rdisk/sybase/sadb_data_vol01
5      0      12      2      0      200
sadb_log_vol01      /dev/vx/rdisk/sybase/sadb_log_vol01
56      0      25      2      0      900
scratchdb_vol01      /dev/vx/rdisk/sybase/scratchdb_vol01
0      0      0      2      0      0
rmdb_data_vol01      /dev/vx/rdisk/sybase/rmdb_data_vol01
0      0      0      2      0      0
rmdb_log_vol01      /dev/vx/rdisk/sybase/rmdb_log_vol01
52658  424      99512      2      0      2231300
sysprocsdev      /dev/vx/rdisk/sybase/sybsystemprocs_vol01
146      0      3569      2      0      13700
tempdb_data      /tmp/tempdb_data
4      0      814      2      0      400
tempdb_log      /tmp/tempdb_log
```


Where `monDeviceIO` is the system table that contains statistical information about disk I/O (physical name, reads, writes, and so on). To perform this query, the user does not need to know if Monitor Server or any other monitoring agent is executing on the same host server; the `monDeviceIO` table contains all the information they need.

The monitoring tables are not created by default, but must be created using the `installmon` script in the `$SYBASE/ASE-12_5/scripts` directory.

Note You must have the `mon_role` role to query these tables.

Using Transact-SQL to monitor performance

Providing monitoring information as tables enables you to use Transact-SQL to monitor Adaptive Server. For example, to identify the Transact-SQL statements that are currently consuming CPU, enter:

```
select s.SPID, s.CpuTime, t.LineNumber, t.SQLText
from monProcessStatement s, monProcessSQLText t
where s.SPID=t.SPID
order by s.CpuTime, s.SPID, t.LineNumber desc
```

You can use this same query to find the SQL statements that are using the most physical I/O by substituting `CpuTime` for `PhysicalReads`.

The information in each monitoring table can be sorted, selected, joined, inserted into another table, and treated much the same as the information in a regular Adaptive Server table.

The tables are read-only because they are in-memory tables that are generated as they are queried.

Access to these tables is restricted to users with the `mon_role` role.

The definitions for the monitoring tables have a similar syntax to CIS definitions, which allow remote procedures to be defined as local tables. For example, the following syntax is used to create the `monNetworkIO` table on a server named “loopback”:

```
create existing table monNetworkIO (
    PacketsSent int,
    PacketsReceived int,
    BytesSent int,
    BytesReceived int,
)
external procedure
```

```
at "loopback...$monNetworkIO"
```

See the *Performance and Tuning Guide* for more information.

Index selection

Index selection allows you to determine which indexes are actively being used and those that are rarely used. The index selection feature is used with the monitoring tables feature.

See the *Performance and Tuning Guide* for more information.

Active-passive configuration for Sun Cluster 3.0

Adaptive Server for Sun Cluster 3.0 offers either an active-passive or an active-active setup.

An active-active setup includes two Adaptive Servers installed on two nodes of the cluster that are configured as companion servers. When the primary companion fails, the secondary companion takes over its workload. The active-active setup for Sun Cluster 3.0 is described in Chapter 11 of *Using Sybase Failover in a High Availability System*.

If Adaptive Server, or the node on which it is running, fails, the server and all its devices fail over to a secondary node. At the time of failover, all existing client connections are lost. Clients must reestablish their connections and resubmit their uncommitted transactions as soon as Adaptive Server is running on the secondary node. The failover of client connections (that is, reestablishing their connections after failover) can be done automatically by using the self-referencing *hafailover* entry in the interfaces file and setting the *CS_HAFAILOVER* connection property.

If your site includes more than two nodes in an active-passive setup, the server running on the primary node can fail over to any of the secondary nodes, as long as it can successfully host Adaptive Server and all its associated resources. The node list determines the order in which Adaptive Server fails over to the other nodes in the cluster.

Table 3-1 discusses the differences between an active-active setup and an active-passive setup.

Table 3-1: Differences between active-active and active-passive

Active-active	Active-passive
<i>Setup:</i> Two Adaptive Servers are configured as companion servers, each with an independent resource group. These companions run on the primary and secondary nodes, respectively, as individual servers until failover.	<i>Setup:</i> One Adaptive Server, running either on the primary node or on the secondary node. Only one Adaptive Server resource group exists, which is hosted by the primary node before failover and the secondary node after failover.
<i>Failover:</i> When failover occurs, the secondary companion takes over the devices, client connections, and so on from the primary companion. The secondary companion services the failed-over clients, as well as any new clients, until the primary companion fails back and resumes its activities.	<i>Failover:</i> When failover occurs, the Adaptive Server along with its associated resources and resource group is relocated to, and restarted on, the secondary node.
<i>Failback:</i> Failback is a planned event during which the primary companion takes back its devices and client connections from the secondary companion to resume its services.	<i>Failback:</i> Failback is a planned failover or relocation of the Adaptive Server resource group to the another node. Failback is not required, but can be used for administration.
<i>Client Connection failover:</i> During failover, clients connect to the secondary companion to resubmit their transactions. During failback clients connect to the primary companion to resubmit their transactions. Clients with the failover property reestablish their connections automatically	<i>Client Connection failover:</i> During failover and failback, clients connect to the same Adaptive Server to resubmit uncommitted transactions. Clients with the failover property reestablish their connections automatically.

Asynchronous log service and optimistic index locking

Complex, multitier applications that consolidate the various parts of an enterprise system increase the load on high-volume database servers. When you address this by upgrading database hardware, from 4 to 8 CPU systems to 16, 32, or 64 CPU systems, you can drastically increase contention on such key resources as the log, the spinlocks that guard the log, and the address locks. Asynchronous log service (ALS) and optimistic index locking help resolve contention, increasing Adaptive Server scalability in systems using four or more online engines.

Optimistic index locking can resolve increased contention on some important resources, such as the spinlocks that guard address locks on the root page of an index.

Applications where this amount of contention might occur are typically those in which:

- Access to a specified index constitutes a significant portion of the transaction profile, and many users are concurrently executing the same workload.
- Different transactions, such as ad hoc and standardized queries, use the same index concurrently.

Using sampling for *update statistics*

The Adaptive Server query optimizer uses statistics to optimize queries.

Run the update statistics commands on tables (data sets) to update information about the distribution of key values in specified indexes or columns, for all columns in an index, or for all columns in a table. The commands revise histograms and density values for column-level statistics.

Multiple temporary databases

Adaptive Server version 12.5.0.3 allows you to create and manage multiple temporary databases in addition to the system tempdb, which was the only temporary database in the server in earlier versions of Adaptive Server.

Multiple temporary databases, also referred to as **tempdbs**, reduce contention on system catalogs and logs in the system tempdb. They allow you to:

- Create temporary databases on fast-access devices
- Drop a temporary database to reclaim storage
- Partition tasks that create temporary objects into using specific tempdbs, which prevents these tasks from interfering with other sessions that need to use temporary database space

The multiple temporary database feature is fully enabled for:

- New installations
- Installations that are upgrading from an Adaptive Server earlier than version 12.5
- Databases that are loaded from an Adaptive Server earlier than version 12.5

The tempdb database is the system-created temporary database. Before Adaptive Server version 12.5.0.3, tempdb was the only temporary database in the server. Temporary tables and work tables are created in tempdb.

Housekeeper enhancements

Adaptive Server version 12.5.0.3 includes the following enhancements to the housekeeper utility:

- In versions earlier than 12.5.0.3, one housekeeper performed five tasks. The work is now 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`.

- Washing buffers is an optional task and runs only at idle times. You can turn off this task using the configuration parameter `housekeeper free write percent`. The housekeeper wash task is the only housekeeper task for which you use this configuration parameter.
- `HK_CHORES` runs at idle times only and does not use a common configuration parameter. It manages miscellaneous chores, such as:
 - Automatically flushing table statistics.
 - Automatically 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`.
- 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.

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.

Chinese character set (GB18030) support

Adaptive Server 12.5.0.3 supports the Chinese character set (GB18030).

SQLJ aliases for XQL

Adaptive Server includes a query processor for the XML query language (XQL), written in Java. The SQLJ alias feature simplifies SQL references to Java static methods by allowing you to create SQLJ alias names for them, rather than requiring you to reference them by their fully qualified Java names.

SQLX mapping functions

Adaptive Server Enterprise 12.5.0.3 includes a set of SQLX mapping functions, for mapping between SQL result sets and XML documents. These mapping functions are implemented with a set of Java classes that you can invoke in a client environment, by placing the *jar* file on your CLASSPATH, or in Adaptive Server, by installing the classes in the server.

New configuration parameters in *sp_monitorconfig*

sp_monitorconfig adds the all parameter. The new syntax for *sp_monitorconfig* is:

```
sp_monitorconfig "configname" [, "result_tbl_name"]
```

or

```
sp_monitorconfig "all" [, "result_tbl_name"]
```

configname is all or part of the configuration parameter name whose monitoring information is being queried. Until Adaptive Server 12.5, the only valid configuration parameters were open indexes, open objects, open databases, aux scan descriptors, txn to pss ratio, number of dtx participants, and all. Specifying all displays descriptor help information for all indexes, objects, databases, and auxiliary scan descriptors in the server.

The number of configuration parameters sp_monitorconfig supports has been increased from 6 to 34.

Faster checkpoints

In Adaptive Server Enterprise 12.5.0.3, the checkpoint task in a specified database executes up to 200% faster, according to internal benchmarks. This increase in checkpoint speed can cause significant gains in database recovery speeds. The checkpoint feature enhancement is automatic, and requires no action on the user's part.

Issuing a larger batch I/O of dirty buffers causes this increase in checkpoint speed. The new default value of the I/O batch size is 100.

Changes to *checkstorage*

Adaptive Server 12.5.0.3 provides the following changes to dbcc checkstorage. You can:

- Execute dbcc stored procedures from any database.
- Generate reports that provide recommended actions to address faults reported by checkstorage.
- Generate reports on aborted checkstorage and checkverify operations.
- Limit checkverify processing to a specific object.
- Exclude tables, faults, and combinations of faults and tables from processing by checkverify and sp_dbcc_faultreport.
- Create default configuration values for the dbcc configuration parameters and view them with sp_dbcc_configreport.

- Delete values of the dbcc configuration parameters.
- Avoid configuring an explicit cache for checkstorage operations.
- Avoid creating scan or text workspaces for every database on which checkstorage reports. You can create both or either one, the scan or the workspaces. Default scan and text workspaces are created when you run the *installdbccdb* script.
- Avoid running *sp_dbcc_updateconfig* to reregister a workspace that you have dropped and re-created.
- Avoid manually expanding the dbccdb workspaces.

Changes to CIS

CIS for Adaptive Server version 12.5.0.3 includes the discussed changes in this section.

set bulk array size

set bulk array size establishes the number of rows that will be buffered in local server memory before they are transferred with the bulk copy interface. The syntax for set bulk array size is:

set bulk array size *number*

The value of *number* indicates the number of rows to buffer. If the rows being transferred contain text, image, or Java Abstract Data Types (Java ADTs), the bulk copy interface ignores the current setting for array size and uses a value of 1. Also, the array size actually used never exceeds the value of @@*bulkbatchsize*. If @@*bulkbatchsize* is smaller than the array size, then the smaller value is used.

set bulk batch size

set bulk batch size establishes the number of rows transferred to a remote server via select into *proxy_table* when the bulk interface is used. The bulk interface is available to all Adaptive Server versions, as well as DirectConnect for Oracle version 12.5.1.

This set command is used with Component Integration Services for transferring rows to a remote server using select into.

Global variables

There are several new global variables for CIS.

@@bulkarraysize

@@bulkarraysize shows the current setting of bulk array size. The following isql session illustrates the use of *@@bulkarraysize*:

```
1> select @@bulkarraysize
2> go

-----
50      <== default value

(1 row affected)
```

@@bulkbatchsize

@@bulkbatchsize shows the current setting of bulk batch size. The following isql sessions illustrates the use of *@@bulkbatchsize*:

```
1> select @@bulkbatchsize
2> go

-----
0       <== default value

(1 row affected)
```

Improved performance for *select into*

Adaptive Server Enterprise version 12.5.0.3 enables large-page allocation for the target table in *select into*, by allocating pages by extent rather than individual page, and thus issuing fewer logging requests for the target table. Instead of logging and updating information for each new page in sysindexes, Adaptive Server updates only once, to the last page in sysindexes, as the last operation in *select into*.

Although *select into* statements use a 2K I/O buffer pool by default, the optimizer can choose the largest pool possible, depending on the size of the buffer pools available in the target database cache. If you configure Adaptive Server with large buffer pools, it uses large I/O buffer pools for writing the target table pages to disk.

New set parameters

Adaptive Server 12.5.0.3 includes the `identity_update` parameter for the `set` option, which supports updating the value of the identity column.

Improved performance for *create database*, *alter database*, *load database*, and *dbcc checkalloc*

The size of large I/O disk buffers used by `create database`, `alter database`, `load database`, and `dbcc checkalloc` to issue multiple parallel I/O is now one allocation, (256 pages), not one extent (8 pages). This increase means that the server requires more memory allocation to issue large buffers. The server now requests that more memory be configured, but this memory is not allocated until a command requests large I/O buffers. A disk buffer that required memory for 8 pages in earlier versions now needs memory for 256 pages. The limit on the configuration parameter number of large io buffers, with a default value of 6, has increased from 32 to 256. At the same time, the internal limit on the number of large I/O buffers a single command can create has been increased from 8 to 32. These increases allow you to both create larger buffers, and use more buffers per command. Whatever value you assign to the configuration parameter `dynamic allocation on demand`, memory for large I/O buffers is always allocated on demand. This memory pool is shrinkable: if the number of large I/O buffers decreases, memory requirements per buffer decrease accordingly.

Number of databases per transaction

In Adaptive Server Enterprise version 12.5.0.3, the maximum number of databases Adaptive Server allows per user transaction is the maximum number of databases that can be created within Adaptive Server. Adaptive Server returns an error message if it is not configured for more than 16 databases within a transaction.

New global variables

Adaptive Server 12.5.0.3 adds the following global variables.

Table 3-2: New global variables

Global variable	Definition
<i>@@datefirst</i>	Set using <code>set datefirst n</code> where <code>n</code> is a value between 1 and 7. Returns the current value of <i>@@datefirst</i> , indicating the specified first day of each week (1 for Sunday, 2 for Monday, and so on), expressed as <code>tinyint</code> . The default value in Adaptive Server is 1, Sunday (based on the <code>us_language</code> default).
<i>@@lock_timeout</i>	Set using <code>set lock wait n</code> . Returns the current <i>lock_timeout</i> setting, in milliseconds. <i>@@lock_timeout</i> returns the value of <code>n</code> . The default value is no timeout. If no <code>set lock wait n</code> is executed at the beginning of the session, <i>@@lock_timeout</i> returns -1.
<i>@@max_precision</i>	Returns the precision level used by decimal and numeric datatypes set by the server. This value is a fixed constant of 38.

New functions

The functions added for Adaptive Server version 12.5.0.3 are:

- `year`
- `month`
- `day`
- `str_replace`
- `square`
- `left`
- `len`
- `next_identity`
- `identity_burn_max`

System changes

This section describes the system changes in Adaptive Server version 12.5.0.3.

New and changed configuration parameters

Table 3-3 summarizes the new configuration parameters in version 12.5.0.3.

Table 3-3: New configuration parameters

Parameter	Function
deadlock pipe active	Controls whether Adaptive Server collects deadlock messages.
deadlock pipe max messages	Determines the number of deadlock messages Adaptive Server stores.
enable monitoring	Controls whether Adaptive Server collects the monitoring table data.
errorlog pipe active	Controls whether Adaptive Server collects error log messages.
errorlog pipe max messages	Determines the number of error log messages Adaptive Server stores.
max sql text monitored	Specifies the total number of bytes Adaptive Server allocates for each user task to store SQL text.
object lockwait timing	Controls whether Adaptive Server collects timing statistics for requests of locks on objects.
per object statistics active	Controls whether Adaptive Server collects statistics for each object.
plan text pipe active	Determines whether Adaptive Server collects query plan text.
plan text pipe max messages	Determines the number of query plan text messages Adaptive Server stores.
process wait events	Controls whether Adaptive Server collects statistics for each wait event for every task.
sql text pipe active	Controls whether Adaptive Server collects SQL text.
sql text pipe max messages	Specifies the number of SQL text messages Adaptive Server stores.
statement pipe active	Controls whether Adaptive Server collects statement-level statistics.
statement pipe messages	Determines the number of statement statistics messages Adaptive Server stores.
statement statistics active	Controls whether Adaptive Server collects the monitoring table statement-level statistics.
sql batch capture	Controls whether Adaptive Server collects SQL text.
wait event timing	Controls whether Adaptive Server collects statistics for individual wait events.

Table 3-4 summarizes changed configuration parameters in 12.5.0.3.

Table 3-4: Changed configuration parameters

Parameter	Changes
enable housekeeper GC	<p>Provides these parameter values:</p> <ul style="list-style-type: none"> • 0 – disables the housekeeper garbage collection task. • 1 – enables lazy garbage collection. • 2 – reserved for future use. • 3 – reserved for future use. • 4 – enables aggressive garbage collection for both the housekeeper garbage collection task and the <code>delete</code> command. • 5 – enables aggressive garbage collection for the housekeeper garbage collection task, and lazy garbage collection by the <code>delete</code> command.

New and changed Transact-SQL commands

Table 3-5 summarizes the new Transact-SQL commands in version 12.5.0.3.

Table 3-5: New commands

Command	Description
set identity_update	Updates the value of the <code>IDENTITY</code> column on a table.

Table 3-6 summarizes the changes to existing Transact-SQL commands in version 12.5.0.3.

Table 3-6: Changed commands

Command	Change
update statistics	Adds the <code>sampling=percent</code> option, which specifies a sampling percent.
dbcc	<p>Adds these new parameters:</p> <ul style="list-style-type: none"> • <code>pravailabletempdbs</code> – prints the global list of available temporary databases. • <code>addtempdb</code> – adds a temporary database to the global list of available temporary databases.
create database	Adds the temporary parameter, which creates a temporary database.

New and changed system procedures

Table 3-7 summarizes the changes made to existing system procedures.

Table 3-7: Changed system procedures

Procedure	Change
sp_monitorconfig	Adaptive Server 12.5.0.3 adds the all parameter, which displays descriptor help information for all 34 indexes, objects, databases, and auxiliary scan descriptors in the server.
sp_dboption	<ul style="list-style-type: none"> Adds the async log service option, which enables, disables, or configures the asynchronous log service feature. Can prevent user-created temporary databases from being set to single-user mode.
sp_helpdb	<ul style="list-style-type: none"> Adds a column to display asynchronous log service information. Reports whether or not a database is a user-created temporary database in the status column.
sp_help	Adds a column that displays optimistic index lock information.
sp_chgattribute	Adds the optimistic_index_lock option, which acquires an optimistic index lock on a table.
sp_bindcache	Can prevent bbinding of individual tables to a named cache in user-created temporary databases.
sp_dbcc_summary_report	Adds the display_recommendations parameter, which enables reporting the recommendations generated by sp_dbcc_recommndations.
sp_dbcc_faultreport	<p>Displays faults in ordered list by table and index. Adds these new parameters:</p> <ul style="list-style-type: none"> exclusion_mode – ignores or applies exclusion lists. exclusion_faults – list of fault types to be excluded from reporting. exclusion_tables – list of tables to be excluded from reporting. exclusion_combo – list of fault/table combinations to be excluded from reporting. display_recommendations – enables reporting the recommendations generated by sp_dbcc_recommendations.
sp_dbcc_checkverify	<p>Adds these new parameters:</p> <ul style="list-style-type: none"> ignore_exclusions – enables or disables the exclusion list. tblename – limits checkverify processing to a specific .object.
sp_dbcc_updateconfig	Provides default configuration values for dbcc configuration values. Accepts delete as a value for the str1 parameter. Disables enable automatic workspace expansion for databases that are enabled by default.
sp_dbcc_configreport	Adds the defaults parameter, which enables only the display of configured default values.

New system procedures

Table 3-7 summarizes the new system procedures.

Table 3-8: New system procedures

Procedure	Change
sp_helptext	Displays the source text of a compiled object .
sp_tempdb	Creates the default temporary database group and binds temporary databases to the default temporary database group.
sp_dbcc_recommendations	Analyzes faults reported by the checkstorage operation and summarizes them.
sp_dbcc_help_fault	Provides a description of the specified fault type and the recommended fix.
sp_dbcc_exclusions	Allows the user to create and manage persistent exclusion lists for use by checkverify and sp_dbcc_faultreport.

New and changed utility programs

Table 3-9 summarizes changed utility programs in Adaptive Server version 12.5.0.3.

Table 3-9: Changed utility programs

Utility	Function
sybmigrate	<p>In Adaptive Server 12.5.0.3, sybmigrate selects dependent objects for migration when you use the auto-select feature.</p> <p>These changes have been made to the command line configuration:</p> <ul style="list-style-type: none"> • -rn diff – a difference report feature that compares source and target database objects. • -rn password – a new report feature to create a file for the changed passwords.

New global variables

Table 3-10 describes new global variables for 12.5.1.

Table 3-10: New global variables

Global variable	Definition
<code>@@datefirst</code>	<p>Set using <code>set datefirst n</code> where <code>n</code> is a value between 1 and 7. Returns the current value of <code>@@datefirst</code>, indicating the specified first day of each week (1 for Sunday, 2 for Monday, and so on), expressed as <code>tinyint</code>.</p> <p>The default value in Adaptive Server is 1, Sunday (based on the <code>us_language</code> default).</p>
<code>@@lock_timeout</code>	<p>Set using <code>set lock wait n</code>. Returns the current <code>lock_timeout</code> setting, in milliseconds. <code>@@lock_timeout</code> returns the value of <code>n</code>. The default value is no timeout. If no <code>set lock wait n</code> is executed at the beginning of the session, <code>@@lock_timeout</code> returns -1.</p>
<code>@@max_precision</code>	Returns the precision level used by decimal and numeric datatypes set by the server. This value is a fixed constant of 38.
<code>@@tempdbid</code>	Returns a valid temporary database ID of the session's assigned temporary database.

New functions

Table 3-11 describes the functions added for Adaptive Server version 12.5.0.3.

Table 3-11: New functions

Global variable	Definition
year	Returns an integer that represents the year in the datepart of a specified date.
month	Returns an integer that represents the month in the datepart of a specified date.
day	Returns an integer that represents the day in the datepart of a specified date.
str_replace	Replaces any instances of the second string expression (<i>string_expression2</i>) that occur within the first string expression (<i>string_expression1</i>) with a third expression (<i>string_expression3</i>).
square	Returns the square of a specified value expressed as a float.
left	Returns a specified number of characters on the left end of a character string.
len	Returns the number of characters, not the number of bytes, of a specified string expression, excluding trailing blanks.
newid	Generates human-readable, globally unique IDs (GUIDs) in two different formats, based on arguments you provide. The length of the human-readable format of the GUID value is either 32 bytes (with no dashes) or 36 bytes (with dashes).
tempdb_id	Reports the temporary database to which a given session is assigned.
next_identity	Retrieves the next identity value that is available for the next insert.
identity_burn_max	Tracks the identity burn max value for a given table.

New and changed system tables

This section describes new system tables to support version 12.5.0.3 features.

Table 3-12 describes new system monitoring tables.

Table 3-12: New system tables

System table	Description
monTables	Provides a description of all monitoring tables. No configuration options are required.
monTableParameters	Provides a description of all the optional parameters for each monitoring table. No configuration options are required.
monTableColumns	Describes all the columns for each monitoring table. No configuration options are required.
monState	Provides information regarding the overall state of Adaptive Server.
monEngine	Provides statistics regarding Adaptive Server engines. Requires the enable monitoring configuration parameter to be enabled.
monDataCache	Returns statistics relating to Adaptive Server data caches. Requires the enable monitoring configuration parameter to be enabled.
monProcedureCache	Returns statistics relating to Adaptive Server procedure cache. Requires the enable monitoring configuration parameter to be enabled.
monOpenDatabases	Provides state and statistical information pertaining to databases that are currently in use.

System table	Description
monSysWorkerThread	Returns server-wide statistics related to worker threads. Requires the enable monitoring configuration parameter to be enabled.
monNetworkIO	Returns network I/O statistics. Requires the enable monitoring configuration parameter to be enabled.
monErrorLog	Returns the most recent error messages from the Adaptive Server error log.
monLocks	Returns a list of all locks that are being held, and those that have been requested, by any process, for every object.
monDeadLock	Provides information pertaining to the most recent deadlocks that have occurred in Adaptive Server.
monWaitClassInfo	Provides a textual description for all of the wait classes. All events have been grouped into wait classes that classify the type of event that a process is waiting for.
monWaitEventInfo	Provides a textual description for every possible situation where a process is forced to wait within Adaptive Server.
monCachedObject	Returns statistics for all objects and indexes with pages currently in a data cache.
monCachePool	Provides statistics for all pools allocated for all caches.
monOpenObjectActivity	Provides statistics for all open objects.
monIOQueue	Provides device I/O statistics broken down into data and log I/O for normal and temporary databases on each device.
monDeviceIO	Returns statistical information relating to devices.
monSysWaits	Provides a server-wide view of where processes are waiting for an event.
monProcess	Provides detailed statistics about processes that are currently executing or waiting.
monProcessLookup	Provides information enabling processes to be tracked to an application, user, client machine, and so on.
monProcessActivity	Provides detailed statistics about process activity.
monProcessNetIO	Provides the network I/O activity for each process.
monProcessObject	Provides statistical information regarding objects that have been accessed by processes.
monProcessWaits	Provides a server-wide view of where processes are waiting for an event.
monProcessStatement	Provides information for currently executing statements.
monProcessSQLText	Provides the SQL text that is currently being executed.
monSysPlanText	Provides the most recent generated text plan.
monSysStatment	Provides statistics pertaining to the most recently executed statements.
monCachedProcedures	Provides statistics for all procedures currently stored in the procedure cache.
monSysSQLText	Provides the most recent SQL text that has been executed, or is currently being executed.
monProcessProcedures	Returns a list of procedures that are being executed by processes.

Table 3-13 describes changes to system tables to support Adaptive Server 12.5.0.3.

Table 3-13: Changed system tables

System table	Changes
sysattributes	Displays information about the representation of temporary database groups and bindings.
sysdatabases	Adds a new bit which indicates the temporary status of a database by the value of 0x00000100 (256 decimal) in the status3 field.

New Features in Adaptive Server Version 12.5.0.1

This chapter describes the new features and system changes introduced with Sybase Adaptive Server Enterprise version 12.5.0.1.

New features in version 12.5.0.1

The new features in Adaptive Server version 12.5.0.1 are:

Feature
“Options for Adaptive Server version 12.5.0.1” on page 44
“Using grant dbcc and revoke dbcc” on page 45
“Increasing the size of devices with disk resize” on page 45
“ASE Replicator” on page 45
“Synchronizing EJB Servers” on page 46
“XQL result sets” on page 47
“XQL query engine” on page 48
“Proxy table space” on page 49
“Creating files and directories with the insert command” on page 49
“Migration utility” on page 50
“High availability on Sun Cluster 3.0” on page 50
“New character sets” on page 50
“System changes” on page 51

Options for Adaptive Server version 12.5.0.1

Earlier versions of Adaptive Server were sold either at the Enterprise Level or the Workplace Level. Beginning with Adaptive Server version 12.5.0.1, Adaptive Server is available in the Enterprise Edition, the Small Business Edition, and the Developer's Edition. The Enterprise Edition of Adaptive Server is a full-featured server that can run all optional features. The Small Business Edition of Adaptive Server includes the features required by most small businesses, but excludes some of the more advanced features. The Developer's Edition is designed for you to design and build applications for Adaptive Server in a development environment.

In addition to offering the three editions of Adaptive Server, Sybase also offers a series of options. Some of these options are intended for an enterprise-level site, and are not available in the Small Business Edition. Table 4-1 describes the availability of each option. For a full description of each option, see your sales representative.

Table 4-1: Options available for Adaptive Server

Options	Enterprise Edition	Small Business Edition	Developer's Edition	Description
Security and Directory Services Edition	Yes	Yes	Yes	Systems continuously available with fault resilience
XML Management Edition	Yes	Yes	Yes	Semi-structured and unstructured-data handling
Content Management Edition	Yes	Yes	Yes	Semi-structured and unstructured-data management
e-business Edition	Yes	No	Yes	Data-storage, access, replication, and security via an e-business platform
High Availability	Yes	No	Yes	Systems continuously available
Data Transfer Management (DTM)	Yes	No	Yes	
Enhanced Full-Text Search (EFTS)	Yes	Yes	No	
SQL Expert	No	No	No	

Using *grant dbcc* and *revoke dbcc*

System Administrators can grant the execution of database consistency check (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.

Increasing the size of devices with *disk resize*

The disk resize command allows you to dynamically increase the size of your database devices, 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 devices on raw partitions and for file systems. The minimum amount of space that 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.

Only users with the sa role can execute the disk resize command.

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.

ASE Replicator

ASE Replicator extends Adaptive Server Enterprise capabilities and provides basic replication from a primary database to one or more replicate databases.

ASE Replicator:

- Replicates both tables and stored procedures
- Maintains database integrity and transactional consistency at multiple sites

- Provides guaranteed delivery of replicated data
- Manages database objects with a publish-and-subscribe model

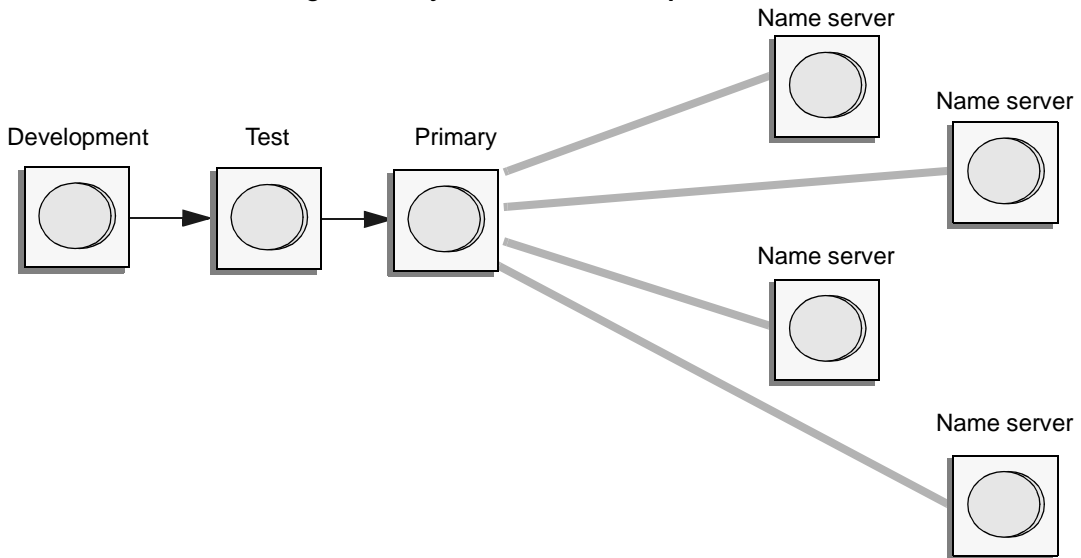
ASE Replicator uses Component Integration Services (CIS) to handle transaction and RPC distribution to replicate databases.

Synchronizing EJB Servers

Synchronization replicates application files and configuration information between EJB Servers. It is a useful alternative to importing and exporting packages.

For example, as Figure 4-1 illustrates, you can replicate new packages from a testing or development server to one or more production servers. As long as you can connect to both servers, synchronization is quicker and easier than exporting and importing package files.

Figure 4-1: Synchronization example



You can use the synchronization option to move repository information between servers. For example, you may want to move a development EJB Server's repository to a test server. In this case, the synchronization command is similar to export/import options without the JAR file; instead of moving packages from one server to another, you are moving repository information.

You can synchronize at the server or package level. The level you choose determines the available synchronization options. Options that are not available are dimmed, and cannot be selected.

XQL result sets

The Extensible Query Language (XQL) result set feature allows you to access query results as objects, rather than as strings, and to extract SQL base types easily from an XML document, without parsing results or writing code. The XQL engine is written in Java, so creating a Java object is easy. The engine encapsulates results in the Java class `com.sybase.xml.xql.XqlResults.SybXqlResult` and its methods.

You can:

- Iterate through XQL result sets to extract XML fragments
- Convert XML into SQL data, for insertion into database tables
- Get the results as a DOM (Document Object Model) object

Table 4-2 describes the methods added for 12.5.0.3.

Table 4-2: *com.sybase.xml.xql.Xql.results.SybXqlResult* specified methods

Method	Action
<code>next()</code>	Moves marker to the next item in the result set.
<code>getXML()</code>	Gets the current item in the result set and returns it as a <code>java.lang.String</code> object. If the internal marker is set to 0, this method returns the whole result set as <code>java.lang.String</code> .
<code>getDOM()</code>	Gets the current item in the result set and returns it as an <code>org.w3c.Document</code> <code>getDOM()</code> object. If the internal marker is set to 0, this method returns the whole result set as <code>org.w3c.Document</code> .
<code>getString()</code>	Gets the current item in the result set and returns it as a <code>java.lang.String</code> object. If the current item has nested elements, this method generates an exception.
<code>getInt()</code>	Gets the current item in the result set and returns it as an integer. If the current item has nested elements or cannot be cast as an integer, this method generates an exception.
<code>getDouble()</code>	Gets the current item in the result set and returns it as a double. If the current item has nested elements or cannot be cast as a double, this method generates an exception.
<code>getFloat()</code>	Gets the current item in the result set and returns it as a float. If the current item has nested elements or cannot be cast as a float, this method generates an exception.
<code>reset()</code>	Resets the internal marker to 0, which means that immediate calls to <code>getDOM()</code> or <code>getXML()</code> yield the entire result set as either <code>java.lang.String</code> or <code>org.w3c.Document</code> object. Calling <code>reset()</code> reinitializes the result set marker.
<code>isFirst()</code>	Returns true if the current item is the first item in the result set.
<code>isLast()</code>	Returns true if the current item is the last item in the result set.

XQL query engine

The XQL character processing engine now supports storage and querying of XML documents containing strings in native character sets.

The XML parser accepts XML documents as a stream of bytes containing characters in native code sets. The XML parsing engine accepts XML documents as either `java.io.InputStream` or as `java.lang.String`. The XML parser checks XML documents and indexes the XML document in any language or character set the XML standard supports.

The query engine parses queries made against the XML document, processes them, and returns query results as `java.lang.String`, which is Unicode. You can use the XQL engine either as part of Adaptive Server or as a standalone engine.

The processing engine returns results as `java.lang.String`, which is Unicode. You should then convert the results of your query from Java strings back to your native code set.

The query engine allows you to:

- Set the character set encoding. The XQL query engine in Adaptive Server mode processes documents whose character set encoding is the same as the character set of the Adaptive Server. If you use the query engine in a standalone server, character set encoding is set using the Java method `com.sybase.xml.xql.Xql.setEncoding()`.

Note `com.sybase.xml.xql.Xql.setEncoding()` is not supported in Adaptive Server. Use it only in standalone mode.

- Parse your XML document. Your XML document heading specifies the native character set of the data you enter, in `java.lang.String` or `java.io.InputStream`. The parser generates events while it parses an XML document. These events pass their information to the indexing layer as `java.lang.String`.

The new flexibility of the query engine requires no change in your parsing process if you use the Xerces parser. If you use a different parser, make sure your parser supports the native character set encoding of your XML document, which is specified in the encoding directive of the XML document heading.

- Query. The XQL query engine accepts queries as `java.io.InputStream` or `java.lang.String`. The `java.io.InputStream` parameter allows character set queries to be fed into the query parser without conversion.

Internally, the query engine translates the incoming stream of bytes to Unicode, using either the character encoding of Adaptive Server, or `com.sybase.xml.xql.Xql.setEncoding()` in standalone mode.

- Obtain results. You can use the sample code provided in this document to convert the results of your query from `java.lang.string` back to native code sets.

Proxy table space

Proxy tables in earlier versions of Adaptive Server occupy one extent (8 pages), as well as one extent for each index on a proxy table. In a pre-12.5.0.1 server with 16K logical page size, each proxy table uses 128K worth of space.

In Adaptive Server 12.5.0.1, proxy tables and indexes do not use extents; they use space only in the system catalogs, which Sybase estimates to be 1MB per 100 proxy tables (assuming an average of two indexes per table). The normal upgrade does not reclaim this unused space. To reclaim this space, first drop, then re-create the proxy table.

Creating files and directories with the *insert* command

You can use the insert command to create a new file directory. To do so, use only the filename, filetype, and content columns. Specify “DIR” as the file type to create filename as a directory. Otherwise, filename is created as a regular file. If filename contains subdirectories, and the subdirectories do not exist, they are created automatically.

Also:

- The value of `@@identity` is updated when an insert completes, if you created a regular file.
- If you attempt to create a file that already exists, an error is raised. You must first delete the file before you can overwrite the contents with an insert command.

You can use insert to transfer the contents of one directory to another, as in the following examples:

```
create proxy_table srctab external directory at '/work/sourcedir;R'
create proxy_table dsttab external directory at '/work1/destdir;R'
insert into dsttab (filename, filetype, content)
Select filename, filetype, content from srctab
```

Migration utility

Adaptive Server version 12.5 and later allows you to configure the page size of your server to use 2K, 4K, 8K, or 16K. However, traditional Adaptive Server upgrade procedures do not allow you to change the page size. Therefore, to convert your current Adaptive Server from one page size to another page size, you must first install a second Adaptive Server with the desired page size, and then use `sybmigrate` to transfer both schema and data from your original (source) Adaptive Server to the new (target) Adaptive Server.

Note `sybmigrate` runs only on Adaptive Server 12.5.0.1 and later.

Benefits of *sybmigrate*

`sybmigrate`:

- Aids users in changing the page sizes of their database applications.
- Provides a manageable and smooth migration process.
- Allows customers to take advantage of the variable page size feature for existing databases with user data, thus realizing the full benefit of Adaptive Server versions 12.5 and later.

High availability on Sun Cluster 3.0

Adaptive Server version 12.5.0.1 adds support for high availability for Sun Cluster 3.0.

New character sets

Adaptive Server version 12.5.0.1 adds support for the following character sets:

- `roman9`
- `iso15`
- `big5hk`

System changes

New configuration parameters

Table 4-3 summarizes the new configuration parameters in version 12.5.0.1.

Table 4-3: New configuration parameters

Parameter	Function
number of histogram steps	Specifies the number of histogram steps. If statistics do not exist on the column, the default value is 20. If statistics for a column already exist in sysstatistics, the default value is the current number of steps.

New and changed Transact-SQL commands

Table 4-4 summarizes the new Transact-SQL commands in version 12.5.0.1.

Table 4-4: New commands

Command	Description
grant dbcc	Allows the System Administrator to grant access on certain dbcc commands.
revoke dbcc	Allows the System Administrator to revoke access on some dbcc commands.

Table 4-5 summarizes the changes to existing Transact-SQL commands in version 12.5.0.1.

Table 4-5: Changed commands

Command	Change
insert	You can use the insert command to create a new file directory. To do so, use only the filename, filetype, and content columns. Specify “DIR” as the filetype, to create filename as a directory.

Changed system procedures

Table 4-6 summarizes the changes made to existing system procedures.

Table 4-6: Changed system procedures

Procedure	Change
sp_helptext	Adds the number parameter, which is an integer identifying an individual procedure, when <i>objname</i> represents a group of procedures. This parameter tells sp_helptext to display the source text for a specified procedure in the group.

Procedure	Change
sp_monitorconfig	sp_monitorconfig can now monitor more than 30 resources compared to the 6 resources it monitored in earlier versions.

New utility programs

Table 4-7 summarizes the new utility programs added to Adaptive Server version 12.5.0.1.

Table 4-7: New utility programs

Utility	Function
sybmigrate	Enables you to migrate database from a server using 2K logical pages to a server using 4, 8, or 16K logical pages.

Changes to databases and system tables

There are no new databases or system tables for Adaptive Server 12.5.0.1.

New Features in Adaptive Server Version 12.5

This chapter describes the new features and system changes introduced with Sybase Adaptive Server Enterprise version 12.5.

New features in version 12.5

The new features in Adaptive Server version 12.5 are:

Feature
“Relaxed server limits” on page 54
“Dynamic reconfiguration” on page 55
“SQLJ stored procedures and functions (Java)” on page 56
“XML in the database” on page 57
“Union in views” on page 58
“Internet directory services (LDAP)” on page 58
“Secure Sockets Layer (SSL)” on page 58
“Enterprise Java Beans (EJB Server)” on page 59
“External file system support” on page 59
“Row-level access control” on page 60
“Suspending and resuming updates to databases” on page 60
“Java.net support” on page 61
“Compressed backups” on page 61
“unichar support” on page 62
“Changes to Component Integration Services (CIS)” on page 63
“SQL debug utility” on page 63
“Migration utility” on page 64

Relaxed server limits

Adaptive Server version 12.5 includes increases in the following limits:

- Number of available logical page sizes
- Number of columns and column sizes
- Maximum length of expressions, variables, and stored procedure arguments
- Number of logins per server

Varying logical page sizes

Adaptive Server version 12.5 does not use the buildmaster binary to build the master device. Instead, Sybase has incorporated the buildmaster functionality in the dataserver binary.

The dataserver command allows you to create master devices and databases with logical pages of size 2K, 4K, 8K, or 16K. Larger logical pages allow you to create larger rows, which can improve your performance because Adaptive Server accesses more data each time it reads a page. For example, a 16K page can hold 8 times the amount of data as a 2K page, an 8K page holds 4 times as much data as a 2K page, and so on, for all the sizes for logical pages.

Number of columns and column size

The maximum number of columns you can create in a table is:

- 1024 for fixed-length columns in both all-pages-locked (APL) and data-only-locked (DOL) tables
- 254 for variable-length columns in an APL table
- 1024 for variable-length columns in an DOL table

The maximum size of a column depends on:

- Whether the table includes any variable- or fixed-length columns.
- The logical page size of the database. For example, in a database with 2K logical pages, the maximum size of a column in an APL table can be as large as a single row, about 1962 bytes, less the row format overheads. Similarly, for a 4K page, the maximum size of a column in a APL table can be as large as 4010 bytes, less the row format overheads. See Table 5-1 for more information.

Maximum length of expressions, variables, and stored procedure arguments

The maximum size for expressions, variables, and arguments passed to stored procedures is 16384 (16K) bytes, for any page size. This can be either character or binary data. You can insert variables and literals up to this maximum size into text columns without using the writetext command.

Number of logins

Table 5-1 lists the limits for the number of logins, users, and groups for Adaptive Server.

Table 5-1: Limits for number of logins, users, and groups

Item	Version 12.0 limit	Version 12.5 limit	New range
Number of logins per server (SUID)	64K	2 billion plus 32K	-32768 to 2 billion
Number of users per database	48K	2 billion less 1032193	-32768 to 16383; 1048577 to 2 billion
Number of groups per database	16K	1032193	16384 to 1048576

Performance implications of new limits

The new limits for Adaptive Server mean that the server may have to handle large volumes of data for a single query, DML operation, or command. For example, if you use a data-only-locked (DOL) table with a char(2000) column, Adaptive Server must allocate memory to perform column copying while scanning the table. Increased memory requests during the life of a query or command means a potential reduction in throughput.

Dynamic reconfiguration

Adaptive Server 12.5 changes the way Adaptive Server allocates memory.

Dynamic memory allocation

Adaptive Server version 12.5 allows you to allocate total physical memory dynamically. Many of the configuration parameters that consume memory were static in pre-12.5 versions of Adaptive Server, and the server needed to be restarted when more memory was required. For example, when you changed the number of user connections, you had to restart the server for this to take effect. Many of the configuration parameters that affect memory are now dynamic, and the server does not have to be restarted for them to take effect.

Changes to how memory is allocated

In earlier versions of Adaptive Server, the size of the procedure cache was based on a percentage of the available memory. After you configured the data cache, whatever was left over was allocated to the procedure cache. For Adaptive Server 12.5, both the data cache and the procedure cache are specified as absolute values. The sizes of the caches do not change until you reconfigure them.

A new configuration parameter, `max memory`, allows you to establish a maximum setting, beyond which you cannot configure Adaptive Server's total physical memory.

If you upgrade to version 12.5 Adaptive Server or higher, pre-12.5 Adaptive Server configuration values are used to calculate the new values for the procedure cache size. Adaptive Server computes the size of the default data cache during the upgrade and writes this value to the configuration file. If the computed sizes of the data cache or procedure cache are less than the default sizes, they are reset to the default. During the upgrade, `max memory` is set to the value of total logical memory specified in the configuration file.

SQLJ stored procedures and functions (Java)

Adaptive Server version 12.0 allowed you to install Java classes in the database and then reference them as datatypes in SQL. Version 12.0 also allowed you to invoke Java static methods directly from SQL.

Adaptive Server version 12.5 extends these capabilities. You can now wrap Java static methods in SQL names and create SQLJ stored procedures or functions that you can use as you would Transact-SQL stored procedures or built-in functions. This new functionality:

- Allows Java methods to return output parameters and result sets to the calling environment.
- Allows you to take advantage of traditional SQL syntax, metadata, and permission capabilities.
- Complies with Part 1 of the ANSI SQLJ standard specification.
- Allows you to use existing Java methods as SQLJ procedures and functions on the server, on the client, and on any SQLJ-compliant, third-party database.

XML in the database

Adaptive Server version 12.5 provides an overview of XML, the eXtensible Markup Language, suitable for data exchange because it is more complete and disciplined than SGML. XML allows you to define your own application-oriented markup tags. This feature, written entirely in Java, also includes methods for storing XML documents and generating them from SQL data. You can:

- Select raw data from Adaptive Server using XQL and display it as an XML document. You can also store XML documents, and the data extracted from them, in Adaptive Server, and you can use Adaptive Server to search XML documents you store on the Web.
- Use the Java XML parser Sybase provides, which allows you to install your query engine as either a standalone program or inside Adaptive Server. You can also use any query engine that is compliant with SAX 2.0.
- Use the sample queries in the OrderXMLclass, a Java sample application that Sybase supplies, which uses XML documents to query, parse, and save customer order data. The OrderXML class is written specifically to process XML documents for order data.
- Use the ResultSetXMLclass, a subclass of the JXml class, which Sybase supplies to help you generate an XML document that represents a SQL result set. You can also access and update XML documents that represent SQL result sets. You can generate a ResultSet in either a client environment or inside Adaptive Server, search, and store it.

Union in views

In earlier versions of Adaptive Server, union operators in select statements that define views have not been supported. Adaptive Server version 12.5 removes this restriction.

You might use this feature, for example, to split a large table into subtables. The data can be partitioned between the subtables based on ranges of data values in one of the columns. You can then define a view that uses union all to combine selects of all the subtables into a single result set. You can then issue select statements on the view containing the union.

You can create views that reference Oracle, DB2, and Informix tables, for example, as well as local tables.

Internet directory services (LDAP)

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

LDAP directories provide:

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

Use the *libtcl.cfg* and the *libtcl64.cfg* files to specify the LDAP server name, port number, DIT base, user name, and password to authenticate the connection to an LDAP server. LDAP defines the communication protocol and the contents of messages exchanged between clients and servers.

Secure Sockets Layer (SSL)

SSL is an industry standard for sending wire- or socket-level encrypted data over secure network connections.

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 its 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.

Adaptive Server implements the SSL protocol as a filter that is appended to the master and query lines of the interfaces file. The address and port numbers where Adaptive Server accepts connections are configurable so you can specify multiple networks, different protocols, and alternate ports.

Enterprise Java Beans (EJB Server)

Sybase provides the Enterprise JavaBean (EJB) Server, a component transaction server, with Adaptive Server version 12.5. EJB Server provides the framework for creating, deploying, and managing middle-tier business logic in the form of EJBs in a multitier environment.

In a three-tier environment, the client provides the user interface logic, the business rules are separated to the middle tier, and the database is the information repository. The client does not access the database directly. Instead, the client makes a call to the EJB Server on the middle tier, which then accesses the database.

EJB Server is designed to reside on the same machine as the database engine it serves. Because the servers are on the same machine, EJB Server can communicate with the database using Adaptive Server's high-speed, shared-memory JDBC driver. This approach ensures:

- High-speed communication and data transfer, even for large data sets
- Secure data transmission because the transfer of information from the database to the middle tier does not take place over the network

External file system support

Adaptive Server enables SQL access to file system directories and their underlying files. The supported syntax is:

```
create [existing] table fname  
external file at "directory_pathname" column delimiter "delimiter"
```

The directory path name must reference a file system directory that is visible to and searchable by Adaptive Server. A proxy table maps column names to attributes of files that exist within the directory. Individual columns, except for the last one, are delimited by the column delimiter specified in the create table syntax.

Row-level access control

Database Owners and table owners can restrict access to a table's data rows by defining access rules and binding those rules to the table. Access to data can be further controlled by setting application contexts and creating login triggers.

These features can be grouped under the concept of row-level access control. Row-level access control enables the Database Owner or table owner to control the rows in a table that users can access, based on their identification or profile and the privileges the user has from the application level. Adaptive Server enforces row-level access control for all data manipulation languages (DMLs), which prevents users from bypassing the access control to get to the data.

Domain rules allow table owners to control the values that users can enter into a particular column that is using a base datatype, or any column that is using a user-defined datatype. Rules are enforced during inserts and updates.

Adaptive Server enables row-level protection through access rules, which are enforced on select, update, and delete operations. Adaptive Server enforces the access rules on all columns that are read in a query, even if the columns are not included in the select list. In other words, for a given query, Adaptive Server enforces the domain rule on the table that is updated and the access rule on the tables that are read. Using access rules does not cause performance degradation.

Suspending and resuming updates to databases

`quiesce database ...hold` allows you to delay updates to one or more databases while you perform a disk unmirroring or external copy of each database. Because no writes are performed during this time, the external copy (the secondary image) of the database is identical to the primary image. While the database is in the quiescent state, read transactions from the database are allowed. To resume updates to the database, issue `quiesce database...release`. You can issue `quiesce database` from one isql connection and then log in with another isql connection and issue `quiesce database...release`.

The syntax for quiesce database is:

```
quiesce database tag_name hold database_name [, database_name]  
[for external copy]
```

or,

```
quiesce database tag_name release
```

Where *tag_name* is a user-defined label for the list of databases to hold or release, and *database_name* is the name of the database for which you are suspending updates.

Java.net support

With java.net, Adaptive Server version 12.5 allows you to create client-side Java networking applications within the server. You can create a network Java client application that connects to any server, which enables Adaptive Server to function as a client to external servers. You can:

- Download documents from any URL address on the Internet.
- Send e-mail messages from inside the server.
- Connect to an external server to save or edit a document.
- Access documents using XML.

Sybase supports certain java.net classes, and provides examples that allow you to set up java.net as either a client or a server process.

Compressed backups

Adaptive Server Enterprise version 12.5 allows you to compress databases and transaction logs using Backup Server. This new feature is part of the dump command through its new compress option.

The partial syntax for dump database ... compress and dump transaction ... compress commands is:

```
dump database dbname  
to [compress::[compression_level::]]stripe_device  
...[stripe on [compress::[compression_level::]]stripe_device] ...  
  
dump transaction dbname  
to [compress::[compression_level::]]stripe_device  
...[stripe on [compress::[compression_level::]]stripe_device]...
```

The default level of compression is 1. You can however, use *compression_level* to specify up to a level-9 compression to accommodate various requirements and specifications.

Adaptive Server also lets you load databases and transaction logs that you dumped using the dump ... compress option, by using the load ... compress option.

The partial syntax for load database .. compress and load transaction .. compress is:

```
load database dbname
from [compress::]stripe_device
...[stripe on compress::stripe_device]...

load transaction dbname
from [compress::]stripe_device
...[stripe on compress::stripe_device]...
```

unichar support

Adaptive Server version 12.5 adds two new datatypes using the UTF-16 encoding of the Unicode character. The new unichar and univarchar datatypes are independent of the existing char and varchar datatypes, but mirror their behavior. Unichar is a fixed-width, non-nullable data type (like char) and univarchar is a variable-width, nullable data type (like varchar). The set of built-in string functions that operate on char and varchar, also operate on unichar and univarchar.

Unlike the existing char and varchar, the new unichar and univarchar only store UTF-16 characters and have no connection to the default character set ID or default sort order ID Adaptive Server configuration option. To use these new datatypes, the default character set for the server must be set to UTF-8.

The main advantage of these new datatypes is efficiency. The UTF-16 character types are approximately 33% more space efficient than UTF-8 for Asian characters.

Each unichar/univarchar character requires two bytes of storage; a unichar/univarchar column consists of 16-bit Unicode values. The following command creates a table with one unichar column for 10 Unicode values requiring 20 bytes of storage:

```
Create table unitbl (unicol unichar (10))
```

The length of a unichar/univarchar column is limited by the size of a data page in Adaptive Server, just as in char/varchar columns.

Changes to Component Integration Services (CIS)

CIS is fully compatible with the new features of Adaptive Server Enterprise documented in this chapter.

- Distributed query optimization
- Transaction management
- Extended data access
- Login name/password mapping to remote systems
- Extensible new limits
- New datatypes for support of the Unicode character set
- LDAP
- SSL
- Union in views

For more information about using these features with CIS, see the *Component Integration Services User's Guide*.

SQL debug utility

sqldbgr is a command line utility that debugs stored procedures and triggers. You can:

- Attach sqldbgr to a task
- Set, enable, and disable breakpoints
- Step through a task one line at a time
- Step into and out of procedures
- Detach sqldbgr from stored procedures or triggers once the debugging is complete

Migration utility

Adaptive Server version 12.5 extends the page size of user databases from 2K to 4K, 8K, or 16K. You can use the migration utility to migrate data from a 12.5 server with one logical page size, to another 12.5 server with a different logical page size. The migration utility migrates all user table data and the following objects to the target server:

- Defaults
- User-defined datatypes
- Rules
- Tables
- Views
- Triggers
- Indexes
- Stored procedures
- Extended stored procedures
- Users
- Logins
- Roles
- Remote servers

System changes in version 12.5

This section summarizes the version 12.5 system changes:

- “New and changed configuration parameters” on page 65
- “New and changed Transact-SQL commands” on page 67
- “New and changed system procedures” on page 68
- “Changed databases and system tables” on page 70
- “New reserved words” on page 72
- “Changes to the documentation” on page 73

New and changed configuration parameters

This section describes the new and changed configuration parameters in version 12.5.

Table 5-2 summarizes the new configuration parameters.

Table 5-2: New configuration parameters

Parameter	Function
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.
cis bulk insert array size	Controls the size of the array when performing a bulk transfer of data from one Adaptive Server to another. During the transfer, CIS buffers rows internally, and asks the Open Client bulk library to transfer them as a block.
dynamic allocation on demand	Determines when memory is allocated for changes to dynamic memory configuration parameters.
enable enterprise java beans	Enables or disables the EJB Server.
enable file access	Enables or disables access through proxy tables to the External File System. Requires a license for ASE_XFS.
enable full-text search	Enables or disables Enhances Full-Text Search services. Requires a license for ASE_EFTS.
enable row level access control	Enables or disables row-level access control.
enable ssl	Enables or disables Secure Sockets Layer session-based security.
enable surrogate processing	Enables or disables the processing and maintains the integrity of surrogate pairs in Unicode data.
enable unicode normalization	Enables or disables Unilib character normalization.
heap memory per user	Specifies the heap memory per user for Adaptive Server.
max memory	Specifies the maximum amount of total logical memory that you can configure Adaptive Server to allocate.
number of engines at startup	Specifies the number of engines that are brought online at start-up. This replaces the minimum online engines parameter.
number of java sockets	Specifies the maximum amount of total physical memory that you can configure Adaptive Server to allocate.
procedure cache size	Specifies the size of the procedure cache in 2K pages.
total logical memory	Specifies the amount of memory that Adaptive Server is configured to use.
total physical memory	Displays the amount of memory that is being used by Adaptive Server at a given moment in time.

Table 5-3 summarizes changes to existing configuration parameters.

Table 5-3: Changed configuration parameters

Parameter	Change
procedure cache percent	Removed from Adaptive Server version 12.5. You now determine the size of the procedure cache with procedure cache size.
total memory	Displays the total logical memory for the current configuration of Adaptive Server.
size of process object heap	Now a server-wide setting and not assigned to a specific task.
max cis remote connections	Removed from Adaptive Server version 12.5. The maximum number is now determined by the maximum number of file descriptors available to a single process for a given operating system.

Table 5-4 lists the configuration parameters that were static, but are now dynamic.

Table 5-4: Configuration parameters changed from static to dynamic

Configuration parameter	Configuration parameter
addition network memory	number of pre-allocated extents
audit queue size	number of user connections
cpu grace time	number of worker processes
deadlock pipe max messages	open index hash spinlock ratio
default database size	open index spinlock ratio
default fill factor percent	open object spinlock ratio
disk i/o structures	partition groups
errorlog pipe max messages	partition spinlock ratio
max cis remote connections	permission cache entries
memory per worker process	plan text pipe max messages
number of alarms	print recovery information
number of aux scan descriptors	process wait events
number of devices	size of global fixed heap
number of dtx participants	size of process object heap
number of java sockets	size of shared class heap
number of large i/o buffers	size of unilib cache
number of locks	sql text pipe max messages
number of mailboxes	statement pipe max messages
number of messages	tape retention in days
number of open databases	time slice
number of open indexes	user log cache spinlock ratio
number of open objects	

New and changed Transact-SQL commands

This section describes the new and changed Transact-SQL commands in version 12.5.

Table 5-5 summarizes the new Transact-SQL commands.

Table 5-5: New commands

Command	Function
create function (SQLJ)	Creates a SQLJ function, which is a Java method with a SQL name.
drop function	Removes a SQLJ function from the database.

Table 5-6 summarizes the changes to existing Transact-SQL commands.

Table 5-6: Changed commands

Command	Change
disk init	Allows the following unit specifiers for the device size: 'k' or 'K' (kilobytes), 'm' or 'M' (megabytes), and 'g' or 'G' (gigabytes).
disk reinit	Allows the following unit specifiers for the device size: 'k' or 'K' (kilobytes), 'm' or 'M' (megabytes), and 'g' or 'G' (gigabytes).
create table	Supports the new size_in_bytes parameter, which allows you to specify the size of an in-row Java-SQL column.
alter table	Allows the following unit specifiers for the device size: 'k' or 'K' (kilobytes), 'm' or 'M' (megabytes), and 'g' or 'G' (gigabytes).
alter database	Allows the following unit specifiers for the device size: 'k' or 'K' (kilobytes), 'm' or 'M' (megabytes), and 'g' or 'G' (gigabytes).
set	You can update as many as 1024 columns in the set clause using literals, variables, or expressions returned from a subquery.
select...for browse	You cannot use the select...for browse option on tables containing more than 255 columns.
select select_list into...at pathname	Allows you to specify a proxy table as the target of a select into command. The at indicates that the new table is a proxy table.
compute	If a compute clause includes a group by clause: <ul style="list-style-type: none"> The compute clause cannot contain more than 255 aggregates. The group by clause cannot contain more than 255 columns. Columns included in a compute clause cannot be longer than 255 bytes.
like	The character string indicated by the like keyword cannot be longer than 255 bytes.
declare cursor	You can include as many as 1024 columns in an update clause of a client's declare cursor statement.
+ operator	Returns result strings up to a length of 16384 bytes.
dump database	Includes compression syntax for making a compressed dump.

Command	Change
load database	Includes compression syntax for loading a compressed dump.
dump transaction	Includes compression syntax for making a compressed dump.
load transaction	Includes compression syntax for loading a compressed dump.
print statement	The total length of the format string + the output cannot exceed 1K. However, Adaptive Server does not issue an error message if you exceed this length.
create procedure (SQLJ)	Creates a SQLJ stored procedure, which is a Java method with a SQL name. Adaptive Server executes a SQLJ stored procedure in the same way it executes a Transact-SQL stored procedure.
create access rule	Allows you to create a rule determining how a user connects to a server.
union in view	You are now allowed to use the union command in a view.

Table 5-7 summarizes the new and changed options for the set command.

Table 5-7: New and changed set command options

set option	Description
stringsize	Limits the number of bytes displayed when Adaptive Server returns a printable representation of a Java object. The default is 50 bytes.

Table 5-8 summarizes the new Transact-SQL functions.

Table 5-8: New Transact-SQL functions

Function	Description
lockscheme()	Returns the locking scheme of the specified object as a string.
pagesize()	Returns the page size, in bytes, for the specified object.

New and changed system procedures

This section describes the new system procedures added and changes made to existing system procedures.

Table 5-9 summarizes the new system procedures.

Table 5-9: New system procedures

System Procedure	Function
sp_ssladmin	Adds, deletes, or displays a list of server certificates for Adaptive Server.
sp_extengine	Allows you to start or stop EJB Server, or view status information about EJB Server, from the command line.

Table 5-10 summarizes the changes made to existing system procedures.

Table 5-10: Changed system procedures

Procedure	Change
sp_addserver	Adds functionality for Adaptive Server's new limits.
sp_dropserver	Adds functionality for Adaptive Server's new limits.
sp_havrfy	Adds functionality for Adaptive Server's new limits.
sp_helpjava	Supports the new depends keyword, which allows you to view a list of database objects that depend on the specified Java class or JAR.
sp_configure	sp_configure allows you to specify the value for configuration parameters in unit specifiers. The unit specifiers are p or P for pages, m or M for megabytes, and g or G for gigabytes.
sp_serveroption	Supports the new external engine auto start clause, which enables or disables automatic startup of EJB Server whenever Adaptive Server starts up. The default is true.

New and changed utility programs

Table 5-11 summarizes the new utility programs added to Adaptive Server version 12.5.

Table 5-11: New utility programs

Utility	Function
ddlgen	A Java-based tool that generates definitions for server- and database-level objects in Adaptive Server.
certpk12	Export or import a PKCS #12 file into a certificates file and a private key.
certreq	Creates a server certificate request and corresponding private key. This utility can be used in interactive mode, or you can provide all optional parameters on the command line.
dataxtr	The GUI migration tool used to migrate data from a 12.5 server with one logical page size, to another 12.5 server with a different logical page size.
certauth	Converts a server certificate request to a CA- (certificate authority) signed certificate.
pwdcrypt	Creates and prints an encrypted LDAP password in the <i>libtcl.cfg</i> .
sqldbgr	Allows you to debug SQL text running on Adaptive Server.

Table 5-12 summarizes the new utility programs added to Adaptive Server version 12.5.

Table 5-12: Changes to utility programs

Utility	Change
optdiag	optdiag and Adaptive Server must be the same version. That is, you cannot use a 12.0 version of optdiag with a 12.5 version of Adaptive Server.

Changed databases and system tables

This section describes the changes made to databases and system tables to support version 12.5 features.

There are no new databases or system tables for this version of Adaptive Server.

Table 5-13 summarizes the system tables that have been changed in this release.

Table 5-13: Changed system tables

Table	Change
syscolumns	New status2 column. Indicates parameter mode of a SQLJ stored procedure, and the return type of a SQLJ function.
	New accessrule column. The object ID of the access rule in sysprocedures.
sysconstraints	New spare1 column. Not for customer use.
syslogins	New procid column. Stores the login trigger registered with the login script option in sp_modifylogin.
sysprocesse	New loggedindatetime column. Shows the time and date when the client connected to Adaptive Server.
	New ipaddr column. IP address of the client where the login is made. See “Row-Level Access Locking” for more information.
sysservers	New srvcost column. Provides the network cost in milliseconds for accessing a server over a network. Used only by the Adaptive Server query optimizer for evaluating the cost of a query when accessing a proxy table. The default is 1,000 ms.
systypes	New accessrule column. The object ID of the access rule in sysprocedures. See “Row-Level Access Lockin” for more information.

For more information about system tables, see the *Reference Manual Volume 4: System Tables*.

Table 5-14 summarizes the changes in existing system table columns in this version:

Table 5-14: Changed status in existing columns

Table name	Column name	Old datatype	New datatype
sysalternates	altsuid	smallint	int
	suid	smallint	int
syscolumns	colid	tinyint	smallint
	length	tinyint	int
syscomments	colid	tinyint	smallint
	colid2	tinyint	smallint
sysconfigures	status	tinyint	int
sysconstraints	colid	tinyint	int
syscurconfigs	status	smallint	int
sysdatabases	suid	smallint	int
sysloginroles	srld	smallint	int
	suid	smallint	int
syslogins	suid	smallint	int
sysobjects	uid	smallint	int
sysprocedures	sequence	smallint	int
sysprocesses	gid	smallint	int
	origsuid	smallint	int
	suid	smallint	int
	uid	smallint	int
sysprotects	uid	smallint	int
sysqueryplans	uid	smallint	int
sysreferences	fokey1 ... 16	tinyint	smallint
	refkey1 ... 16	tinyint	smallint
sysremotelogins	suid	smallint	int
sysroles	id	smallint	int
sysrvroles	lrid	smallint	int
systhresholds	suid	smallint	int
systypes	uid	smallint	int
	length	tinyint	int
sysusermessages	uid	smallint	int
sysusers	gid	smallint	int
	uid	smallint	int
	suid	smallint	int

Table 5-15 describes changes to the sysobjects table. See “sysobjects” in *Reference Manual: Tables* of the Reference Series for more information on the columns used in sysobjects.

Table 5-15: Changes in the sysobjects table

Column name	Datatype	Description
type	char(2)	F = SQLJ function
sysstat2	int	Addition of: <ul style="list-style-type: none">• New bit (0x2000000) that represents a SQLJ stored procedure.• New bit (0x01000000) that represents an access rule.• New bit (0x04000000) that represents an OR access rule.

New reserved words

The following are the reserved words added for version 12.5 Adaptive Server:

- deterministic
- func
- function
- inout
- new
- out
- output
- returns
- stringsize
- syb_terminate

You must change all database names that are new reserved words before you can upgrade from an earlier version of the server. You can change table, view, and column names, or use delimited identifiers. Once you upgrade to version 12.5, you cannot use database objects whose names are new reserved words until you modify your procedures, SQL scripts, and applications.

Reference Manual Volume: 1 Building Blocks lists all of the reserved words for version 12.5. You cannot use reserved words as object names or column names.

New global variables

Table 5-16 summarizes the global variables for the server limits of logins, users, and groups.

Table 5-16: New global variables

Name of variable	What it displays	Value
@ @userid	Lowest user ID	-32768
@ @maxuserid	Highest user ID	2147483647
@ @invaliduserid	Invalid user ID	-1
@ @invalidusid	Invalid server user ID	-2
@ @guestuserid	Guest user ID	2
@ @maxsuid	Highest server user ID	2147483647
@ @minsuid	Lowest server user ID	-32768
@ @guestsuid	Guest server user ID	-1
@ @mingroupid	Lowest group user ID	16384
@ @maxgroupid	Highest group user ID	1048576
@ @probesuid	Probe server user ID	2
@ @maxpagesize	Displays the page size for the server	2048, 4096, 8128, or 16384
@ @boottime	Displays the time at which the server was started	N/A

Changes to the documentation

The following changes have been made to the Adaptive Server documentation set:

- *Managing and Monitoring Adaptive Server Enterprise* has been discontinued.
- The *Utility Guide* is no longer in two versions; it is now a generic book that includes all utilities, regardless of platform.

Changes that may affect existing applications

This section describes system changes introduced by Adaptive Server version 12.5 that may affect your applications if you are upgrading from an earlier version:

- “New Transact SQL keywords” on page 74
- “Wide column and data truncation” on page 74

- “Changes that affect Open Client applications” on page 74
- “Maximum number of expressions in a select statement” on page 76

New Transact SQL keywords

See “New reserved words” on page 72 for information about the Adaptive Server version 12.5 additions to the reserved word list.

Wide column and data truncation

Earlier versions of Adaptive Server had a column-length limit of 255 bytes. Version 12.5 allows you to create columns using char, varchar, binary, and varbinary datatypes that can be up to 16294 bytes long, depending on the logical page size your server uses. Because of this, data that was truncated at 255 character in earlier versions is no longer truncated. If your application depended on this truncation, the result set it receives may no longer be accurate. In the following example, col1 and col2 are each 200 characters long, and col3 is 255 characters long:

```
select * from t1 where col1 + col2 = col3
```

If col1 and col2 each hold 200 characters, their concatenation results in a 400-character string. Earlier versions truncate this to 255 characters, and the sum of col1 and col2 may have matched col3. However, for version 12.5, the product of col1 and col2 is 400 characters, and will never match a col3 that is 255 characters long.

Character and binary expressions can produce a result that up to 16384 bytes long. Data that is longer than this is truncated.

Changes that affect Open Client applications

Using Open Client with Adaptive Server’s new limits

Adaptive Server and Open Client interact depending on how you determine Open Client’s capabilities. Capabilities describe the types of requests that an application sends on a specific connection and the types of server responses that a server returns on a specific connection. Before you use Open Client with the wider limits allowed by Adaptive Server version 12.5, you must enable the Open Client capability for these limits. See your Open Client documentation for information about enabling the wide-table capabilities.

DB Library cannot use the wide limits for Adaptive Server version 12.5.

Depending on which version of Open Client your site is using, you may have to perform one of the steps below.

Adaptive Server and Open Client both use version 12.5

You should not have any problems if both Adaptive Server and Open Client are at version 12.5. Perform the following before you use Open Client 12.5 with Adaptive Server version 12.5:

- 1 Relink your Open Client application with the 12.5 libraries.
- 2 Establish the new version number with CS_VERSION.
- 3 If you are using jConnect, reconnect the drivers.

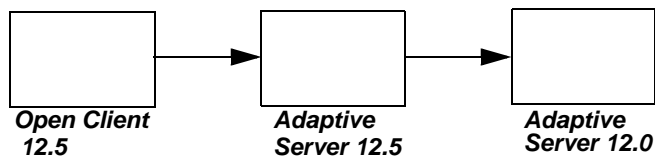
Adaptive Server version 12.5 with older Open Client

- If you are running a pre-12.5 version Open Client application with Adaptive Server version 12.5, and have relinked your application with 12.5 Open Client, but have not changed the version of your libraries, the pre-12.5 version functionality is fine, but the new limits are not enabled. Adaptive Server sends the data according to the pre-12.5 limits, and truncates any data that is beyond these limits.
- If you are running pre-12.5 version Open Client with Adaptive Server version 12.5, but have *not* relinked your libraries, Adaptive Server sends data according to the pre-12.5 version, with the same restrictions outlined above.

Open Client connected to a remote Adaptive Server

If you are connecting to a remote Adaptive Server in the manner described below, Open Client does not receive wide data.

Figure 5-1: Open Client connecting to a remote server



In this example, Open Client connects to the 12.5 Adaptive Server using the 12.5 capabilities, but because the database it is querying is in a 12.0 server, the data is returned using the 12.0 limits, and any wide data is truncated.

If you use an earlier version of Open Client with Adaptive Server

Earlier versions of Open Client never received more than 255 bytes per column from Adaptive Server. If you use earlier versions of Open Client with Adaptive Server version 12.5, Adaptive Server truncates the data to 255 bytes and issues a warning.

Older versions of Open Client work successfully with Adaptive Server version 12.5 if you do not use the new limits (wider columns and rows, larger logical page sizes, and so on).

Maximum number of expressions in a select statement

Adaptive Server version 12.5 has no explicit limit on the number of expressions in a select statement. The number is limited only by the available system memory.

New Features and System Changes in Adaptive Server Version 12.0

This chapter describes the new features and system changes introduced with Sybase Adaptive Server Enterprise version 12.0.

In addition to improving overall manageability and performance, Adaptive Server version 12.0 introduces several new licensed features for specialized use, including:

- Java in Adaptive Server Enterprise
- Sybase Failover for High Availability Environments
- Distributed Transaction Management
- Advanced Security

Before using any of these features, you must purchase the feature's license and configure the feature using the SySAM asset management system (see your installation guide for detailed instructions). To purchase one or more licensed features, contact your sales representative.

New features in version 12.0

The new features and system changes in Adaptive Server version 12.0 are:

Category	Feature
Availability and manageability	"Sybase Failover for a high availability system" on page 78
	"Rebuilding indexes" on page 80
	"New Backup Server features" on page 81
	"UNIX file system support" on page 82
	"Modifying an existing table's schema with alter table" on page 82
	"Suspending database updates with quiesce database" on page 83
	"Taking Adaptive Server engines offline" on page 83

Category	Feature
	“User settable process information” on page 83
	“Removed limit for number of fragments per database” on page 84
	“Identity number gap for tables” on page 84
	“Diagnostic database” on page 85
Performance and productivity	“Java in Adaptive Server Enterprise” on page 85
	“Number of tables in a query” on page 87
	“Query processing and optimization enhancements” on page 87
	“ANSI joins” on page 88
	“Dynamic execution of Transact-SQL” on page 89
	“text and image datatype enhancements” on page 90
	“Abstract plans” on page 90
	“Disabling triggers” on page 91
	“Cache partitions” on page 91
Distributed processing	“Distributed Transaction Management” on page 92
	“Enhancements to Component Integration Services (CIS)” on page 92
Security	“Network security” on page 93
	“User-defined login security” on page 94
	“Concrete identification” on page 98

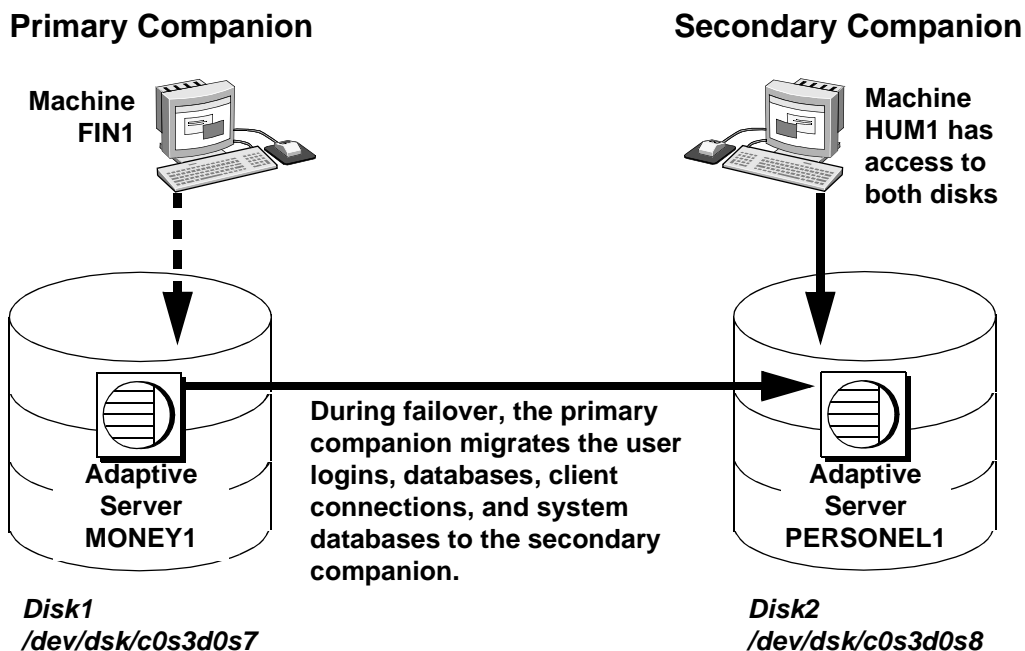
Sybase Failover for a high availability system

A high availability cluster includes two machines that are configured so if one machine (or application) is brought down, the second machine assumes the workload of both machines. Each of these machines is called one **node** of the high availability cluster. A high availability cluster is typically used in an environment that must always be available, for example, a banking system to which clients must connect continuously, 365 days a year.

Sybase Failover enables Adaptive Server to work in a high availability cluster in an active-active configuration. That is, both nodes in the cluster include Adaptive Servers managing independent workloads, and are capable of taking over each other's workload in the event of a failure. The Adaptive Server that takes over the workload is called a **secondary companion**, and the Adaptive Server that fails is called the **primary companion**. Together they are **companion servers**. This movement from one node to another is called **failover**. After the primary companion is ready to resume its workload, it is moved back to its original node. This movement is called **failback**. Clients connected to the failed Adaptive Server automatically reestablish their network connections via the second machine.

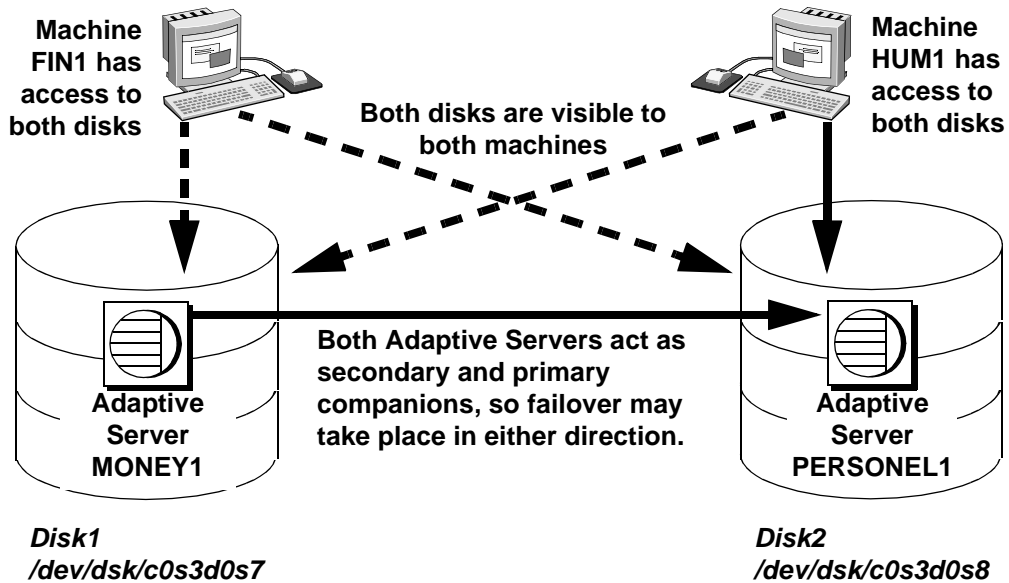
You can configure Sybase Failover for either an *asymmetric* or a *symmetric* setup. In an asymmetric setup, the primary Adaptive Server performs the work during day-to-day operations, while the secondary Adaptive Server is prepared to take over the work during system failure or scheduled maintenance. The secondary companion is an independent Adaptive Server, and can have its own applications running. Figure 6-1 describes an asymmetric setup.

Figure 6-1: Asymmetric configuration in a high availability system



A symmetric setup consists of two fully functional Adaptive Servers running on separate machines, with their own system devices, system databases, user databases, and user logins. However, in the case of failover, either of the Adaptive Servers acts as a primary or secondary companion for the other Adaptive Server. Figure 6-2 describes a symmetric setup.

Figure 6-2: Symmetric configuration in a high availability system



For information about running Sybase Failover in a high availability system, see *Using Sybase Failover in a High Availability System*.

Rebuilding indexes

In Adaptive Server 12.0, you can use the `reorg rebuild` command to rebuild individual indexes while the table itself is accessible for read and update activities. See the *System Administration Guide* and the `reorg` description in the *Adaptive Server Reference Manual*.

New Backup Server features

The following features are new for Backup Server:

- High performance backup and restore
- Non-rewinding tapes

High performance backup and restore

When you start Backup Server with the `backupserver` command, you can use the `-m` parameter to improve the performance of the dump and load commands by configuring more shared memory for the Backup Server. The `-m` parameter specifies the maximum amount of shared memory used by the Backup Server. You must also configure your operating system to ensure that this amount of shared memory is available to the Backup Server. After a dump or load operation is completed, its shared memory segments are released.

Note Configuring more shared memory improves dump and load performance only if the performance limits of the hardware setup have not been reached. Increasing the value of `-m` may not result in improved performance when dumping to a slow tape device such as QIC, but it can improve performance significantly when dumping to a faster device such as DLT.

For details about using Backup Server, see the *Utility Programs* manual for your platform and the *System Administration Guide*.

Non-rewinding tapes

In pre-12.0 versions of Adaptive Server, Backup Server rewound the tape before each dump operation. In version 12.0, the tape is positioned at the end of valid dump data, which saves time when you want to perform multiple dump operations. For details, see the *System Administration Guide*.

UNIX file system support

With pre-12.0 versions of Adaptive Server, Sybase recommended that you not place database devices on UNIX operating system files. The integrity of data on such devices could not be ensured, because writes to the file were buffered by the UNIX file system. Adaptive Server could not know when an update was reflected on the physical media, and therefore could not recover databases on the device under certain system failure scenarios. For this reason, UNIX raw partitions were required to ensure the integrity of database devices.

Adaptive Server version 12.0 introduces a new `dsync` setting for database device files, which 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.

For more information about the `dsync` flag, see the *Reference Manual* entries for `disk init`, `disk reinit`, `sp_deviceattr`, or `sp_helpdevice`.

Modifying an existing table's schema with *alter table*

Adaptive Server 12.0 includes new syntax for `alter table` that allows you to add, drop, or modify a table's columns. `alter table...modify` allows you to modify a column's datatype, length, locking scheme, or default value. `alter table` also allows you to add, drop, or modify an `IDENTITY` column and to add a null or non-null column.

`alter table` includes the following syntax for modifying a table's schema:

```
alter table table_name
[add column_name datatype [identity | null |
    not null] [, column_name datatype [identity | null | not null]]]
[drop column_name [, column_name]
    [modify column_name {[data_type]
    [[null] | [not null]]}
    [, column_name datatype [null | not null]]]
```

Where *table_name* is the table you are altering, and *datatype* is the datatype of the altered column. The following example changes the datatype of the type column in the titles table from `char(12)` to `varchar(20)` and makes it nullable:

```
alter table titles
```

```
modify type varchar(20) null
```

Suspending database updates with *quiesce database*

Adaptive Server version 12.0 provides the new *quiesce database* command to temporarily suspend updates to databases. You may want to suspend database updates to use an external database backup utility, or to separate a database from its mirror for reporting purposes.

For more information about this feature, see the *quiesce database* pages in the *Reference Manual*.

Taking Adaptive Server engines offline

You can dynamically change the number of engines in use by Adaptive Server with the *dbcc engine* command to take an engine offline or bring an engine online. This allows a System Administrator to reconfigure CPU resources as processing requirements fluctuate over time.

For information on using *dbcc engine (offline)*, see Chapter 16, “Managing Multiprocessor Servers,” in the *System Administration Guide*.

User settable 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*, and *clientapplname* columns of the *sysprocesses* table.

For more information, see Chapter 10, “Managing Adaptive Server Logins and Database Users” in the *System Administration Guide*.

Removed limit for number of fragments per database

Adaptive Server 12.5 removes the limit for the number of fragments per database. You can now have any number of fragments per database.

Identity number gap for tables

The identity column contains a unique ID number, generated by Adaptive Server, for each row in a table. Because of the way the server generates ID numbers by default, you may have large gaps in the ID numbers. The *identity_gap* parameter gives you control over ID numbers, and potential gaps in them, for a specific table.

This parameter can be used when creating a table with *create table* or *select into*. To change the identity gap, use *identity_gap* with *sp_chgattribute*. To see the identity gap setting for a table, use *sp_help*.

For details about using *identity_gap*, see Chapter 7, “Creating Databases and Tables” in *Transact-SQL User’s Guide*, the *select* and *create table* commands and *sp_chgattribute* system procedure descriptions in the *Adaptive Server Reference Manual*.

Diagnostic database

Sybase Technical Support may create the sybdiagdb database on your system for debugging purposes. This database holds diagnostic configuration data. It should not be used by customers.

The following configname options have been added to sp_helpconfig for use with the sybdiagdb database:

- *number of ccbs* is the number of configurable action point control blocks available to aid debugging.
- *caps per ccb* is the maximum number of configurable action points that can be configured at any one time within one configurable action point.
- *average cap size* is the estimated number of bytes of memory required to store the information associated with a typical configurable action point.

Note The new options can be used by Sybase Technical Support, for diagnostic purposes only, with the sybdiagdb database.

Java in Adaptive Server Enterprise

Adaptive Server version 12.0 provides a runtime environment for Java, which means that you can execute Java code in the server. Adaptive Server's Java capabilities provide you with powerful new ways of managing and storing both data and logic, using a language that is both portable and widely available.

- You can use the Java programming language as an integral part of Transact-SQL.
- You can reuse Java code in the different layers of your application—client, middle-tier, or server—and use them wherever makes most sense to you.
- Java provides a more powerful language than stored procedures for building logic into the database.
- Java classes become rich, user-defined datatypes.
- Methods of Java classes provide new functions accessible from SQL.

- Java can be used in the database without jeopardizing the integrity, security, and robustness of the database. Using Java does not alter the behavior of existing SQL statements or other aspects of non-Java relational database behavior.

Capabilities of Java in Adaptive Server

Java in Adaptive Server provides you with:

- Java user-defined functions (UDFs)
- Java classes as datatypes in SQL

Java user-defined functions

You can install Java classes in the Adaptive Server database, and then invoke the methods of those classes, both from within the SQL system and from client systems.

The methods of an object-oriented language correspond to the functions of a procedural language. You can invoke Java methods as UDFs in, for example, select lists and where clauses. You can use methods from other sources as well as methods you have created and tested.

Java classes as datatypes

You can install pure Java classes in a SQL system, and then use those classes in a natural manner as datatypes in SQL. This capability adds a full object-oriented datatype extension mechanism to SQL. The objects that you create and store with this facility are readily transferable to any Java-enabled environment, either in another SQL system or standalone Java environment.

This capability of using Java classes in the database has two different but complementary uses:

- It provides a type extension mechanism for SQL, which you can use for data that is created and processed in SQL.
- It provides a persistent data capability for Java, which you can use to store data in SQL that is created and processed (mainly) in Java. Java in Adaptive Server provides a distinct advantage over traditional SQL facilities: you do not need to map Java objects into scalar SQL datatypes or store Java objects as untyped binary strings.

Number of tables in a query

The number of user tables and worktables that can be used for a query has been increased from earlier versions of Adaptive Server. In earlier versions, a query and any views used in a query could reference at most 16 user tables, and the maximum number of worktables that could be created was 12. The limits are now 50 user tables and 14 worktables. Limits on the number of referential integrity checks allowed (192) and the number of subqueries allowed in a query (16) have not changed.

You may need to reset the configuration parameter number of aux scan descriptors if you increase the number of tables referenced in queries.

For information on how queries are optimized when large numbers of tables are included, see Chapter 7, “Access Methods and Query Costing for Joins and Subqueries,” in the *Performance and Tuning Guide*.

For information on resetting scan descriptors, see number of aux scan descriptors in Chapter 17, “Setting Configuration Parameters,” in the *System Administration Guide*.

Query processing and optimization enhancements

Adaptive Server version 12.0 introduces several query processing enhancements.

Merge joins

Pre-12.0 versions of Adaptive Server performed all join queries using nested-loop joins. Version 12.0 adds merge joins as a query execution method for equijoins. Merge joins can produce dramatic improvements in query execution times, especially for joins involving several very large tables. By default, merge joins are not enabled. They can be enabled server-wide or for an individual session.

For information on configuring merge joins, see enable sort-merge joins and JTC in Chapter 17, “Setting Configuration Parameters,” in the *System Administration Guide*. For information on merge joins and query processing, see Chapter 7, “Access Methods and Query Costing for Joins and Subqueries,” in the *Performance and Tuning Guide*.

Query costing improvements

The following changes help improve the accuracy of optimizer costing and provide additional paths for query execution:

- Transitive closure can be applied to joins.
- Predicate factoring and transformation can improve costing for queries using `or`.
- Special like optimization for leading wildcards in strings.

These features are described in Chapter 5, “Understanding the Query Optimizer,” in the *Performance and Tuning Guide*.

ANSI joins

Pre-12.0 versions of Adaptive Server provided syntax only for a Transact-SQL join, which included the `=*` and `*=` symbols for specifying a right or a left join, respectively. Adaptive Server 12.0 includes the ANSI syntax for joining tables or views, which allows you to write either:

- Inner joins, in which the joined table includes only the rows of the inner and outer tables that meet the conditions of the `on` clause. The result set of a query that includes an inner join does not include any null supplied rows for the rows of the outer table that do not meet the conditions of the `on` clause. The syntax for an ANSI inner join is:

```
select select_list
from table1 inner join table2
on join_condition
```

For example:

```
select au_id, titles.title_id, title, price
from titleauthor inner join titles
on titleauthor.title_id = titles.title_id
and price > 15
```


- Outer joins, in which the joined table includes all the rows from the outer table whether or not they meet the conditions of the `on` clause. If a row does not meet the conditions of the `on` clause, values from the inner table are stored in the joined table as null values. The `where` clause of an ANSI outer join restricts the rows that are included in the query result. ANSI syntax also allows you to write nested outer joins. The syntax for an ANSI outer join is:

```
select select_list
from table1 {left | right} [outer] join
      table2
on predicate
[join restriction]
```

For example:

```
select au_fname, au_lname, pub_name
from authors left join publishers
on authors.city = publishers.city
```

Sybase recommends that applications use ANSI outer joins because they unambiguously specify whether the `on` or `where` clause contains the predicate, which is ambiguous when you use Transact-SQL outer joins.

For more information, see the *Transact-SQL User's Guide*.

Dynamic execution of Transact-SQL

Adaptive Server version 12.0 provides an extension to the `execute` command that enables users to define Transact-SQL commands dynamically at execution time. Applications and procedures can use the new `execute()` syntax in cases where table names, column names, and so forth are not known until the application or procedure executes.

For more information about this feature, see the `execute` pages in the *Reference Manual*.

text and image datatype enhancements

Adaptive Server version 12.0 improves the storage format for text and image datatypes. Changes to the storage format are transparent to end-user applications, but they enable Adaptive Server to perform random access when querying data. This improves query performance over earlier versions, which had to access text and image data sequentially.

The new storage format for text and image datatypes also enables Adaptive Server to asynchronously prefetch the data into an existing buffer pool. You can fine-tune the prefetch settings for text and image data using the `sp_configure` parameter `text prefetch size`. See the *System Administration Guide* for more information about text prefetch size.

Abstract plans

Adaptive Server can capture query text and save an abstract plan for a query in a new system table called `sysqueryplans`. Using a rapid hashing method, incoming SQL queries can be compared to stored query text, and if a match is found, the saved abstract plan is used to execute the query.

An abstract plan describes the execution plan for a query using a language created for that purpose. This language contains operators to specify the choices and actions that can be generated by the optimizer. For example, to specify an index scan on the `titles` table, using the index `title_id_ix`, the abstract plan says:

```
( i_scan title_id_ix titles)
```

Abstract plans provide a means for System Administrators and performance tuners to protect the overall performance of a server from changes to query plans. Changes in query plans can arise due to:

- Adaptive Server software upgrades that affect optimizer choices
- New Adaptive Server features that change query plans
- Changing tuning options such as the parallel degree, table partitioning, or indexing

Other uses include:

- The ability to look for specific types of plans, such as table scans or reformatting

- The ability to determine which plans use particular indexes
- The ability to specify full or partial plans for poorly-performing queries.

For more information, see Chapter 21, “Introduction to Abstract Plans,” in the *Performance and Tuning Guide*.

Disabling triggers

Loading a database from a previous dump causes any triggers defined in the database to fire, which can increase the amount of time required to load the database. Adaptive Server 12.0 introduces the disable trigger option of the alter table command to disable any triggers in a database before you load the database, reducing the time required to load the database. Use alter table... enable trigger to reenable the triggers after the load database is complete. alter table... disable trigger uses the following syntax:

```
alter table [database_name.[owner_name].]table_name
{enable | disable} trigger [trigger_name]
```

Cache partitions

In a multiengine environment, each task that needs to access the data cache holds a spinlock on the cache while accessing it. With a large number of engines and a high transaction rate, contention for buffer cache spinlocks can slow performance. Adaptive Server version 12.0 allows you to divide each cache into partitions, each with its own spinlock. See Chapter 19, “Configuring Data Caches” in the *System Administration Guide* for information on how to configure cache partitions. See “Memory Use and Performance” and “Monitoring Performance with sp_sysmon” in the *Performance and Tuning Guide* for information on how to check cache spinlock contention and other considerations when using cache partitions.

Distributed Transaction Management

Adaptive Server version 12.0 introduces several Distributed Transaction Management features, which:

- Bring Adaptive Server into full compliance with the X/Open XA protocol when acting as a resource manager, without requiring additional services such as XA-Server
- Provide support for distributed transactions coordinated by Microsoft Distributed Transaction Coordinator (MSDTC)
- Ensure consistent commit or rollback for all transactions that update Adaptive Server version 12.0 data via remote procedure calls (RPCs) and Component Integration Services (CIS)
- Provide the framework to support additional distributed transaction management protocols in the future

These capabilities are available as a separately-licensed Adaptive Server feature. See *Using Adaptive Server Distributed Transaction Management Features* for more information.

Enhancements to Component Integration Services (CIS)

Adaptive Server version 12.0 includes these new Component Integration Services (CIS) features:

- New dbcc commands – dbcc cis showcaps and dbcc cis setcaps allow examination and modification of server capabilities.
- Proxy databases – proxy database is created containing proxy tables. Metadata for proxy tables is imported automatically from the remote location which contains the actual tables. This metadata is then used to create proxy tables within the proxy database.
- Synchronizing proxy tables – proxy tables in a proxy database can be resynchronized using the alter database command.
- Java in the database
- Support for ANSI joins
- 50-table join limit

- New global variables
- Support for remote server logins
- Two-phase commit transaction management
- Support for execute immediate
- New set commands
- System proxy databases in support of high availability clusters
- create table syntax
- create existing table syntax
- create proxy_table
- RPCs as proxy tables

See the *Component Integration Services User's Guide* for more information.

Network security

Network security for Adaptive Server is achieved through the security control layer (SCL). The security features supported by Adaptive Server are provided by one or more of the security drivers that work with a particular security mechanism.

Pre-12.0 versions of Adaptive Server could obtain network service information only by reading the interfaces file in the Sybase release directory. Adaptive Server 12.0 can access network service information, such as the host name and port number, from sources other than the interfaces file. Adaptive Server sends and receives this configuration information through the directory control layer (DCL), which defines a generic directory service for applications to access the network service information.

The drivers for specific service providers are called directory drivers. If no directory drivers are installed, the DCL obtains network services information from the interfaces file. Directory drivers are platform-specific. For example, the DCE directory driver is implemented on several UNIX platforms, but the NT Registry driver is implemented only on the NT platform.

Performance considerations

Using the DCL should have little or no impact on performance. You may notice some change in performance when the server is started and when connections are created. On listening ports, there is no performance impact after the server is started. On server-to-server connections, there is no performance impact after the site handler connection is established. The performance of the DCL depends on the performance of the directory driver.

Security features supported in Adaptive Server 12.0

Depending upon the security mechanism you choose, Adaptive Server allows you to use one or more of these security services:

- **Unified login** – Adaptive Server receives a user’s previously authenticated credential (login and password) from the client application when the client makes a request of the server. Adaptive Server then verifies that credential with the security mechanism, so the user supplies a login name and password only once to gain access to Adaptive Server.
- **Integrity** – Adaptive Server uses a message integrity algorithm negotiated at login time with the client application to verify that data communications from the client have not been tampered with. Adaptive Server “signs” all outgoing data communications so the client can verify their integrity.

For details on setting up your system to use security features, see Chapter 10, “Using Network-Based Security,” in the *System Administration Guide*.

User-defined login security

User-defined login security gives you more control over security-related features of Adaptive Server. In Adaptive Server 12.0, the System Security Officer can:

- Add more user logins and roles than was possible in earlier versions.
- Specify 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.
- Lock and unlock roles manually.

- Ensure that all user passwords have at least one digit.
- Specify the minimum password length required server-wide or for a specific login or role.
- Display all security-related information for logins and roles.
- Associate a password expiration value with a specified login or role.

Auditing is part of the Adaptive Server security features, but it was not updated for version 12.0, so there are no documentation changes for auditing.

The features of UDLS are described in detail in the *System Administration Guide* and the *Adaptive Server Reference Manual*.

Increased maximum number of users and logins

The maximum number of logins and the maximum number of users per-database has been increased. You may notice the following new behaviors as a result of these changes:

- Negative values may be used for user IDs (uid).
- The server user ID (suid) associated with a group or a role in sysusers is no longer equal to the negation of their user ID (uid). In Adaptive Server 12.0, 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.

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.

You can use the following commands or system procedures to set or change the maximum number of login attempts:

- create role

- alter role
- sp_addlogin
- sp_modifylogin
- sp_configure

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.

You can use the following commands or system procedure to lock or unlock logins and roles:

- alter role
- dataserver -u
- sp_locklogin

Displaying password information

Use sp_displaylogin to display password information for logins and roles.

Checking passwords for at least one character

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.

Setting and changing minimum password length

In earlier versions, the minimum password length was a nonconfigurable, hard-coded value of six characters. In Adaptive Server 12.0, you can customize passwords to fit your needs such as using four-digit personal identification numbers (PINs) or anonymous logins with NULL passwords.

The System Security Officer can specify:

- A globally enforced minimum password length
- A per-login or per-role minimum password length

The per-login or per-role value overrides the server-wide value. Setting a minimum password length affects only new passwords created after setting the value. It does not affect existing passwords.

You can use the following commands or system procedures to set the minimum password length:

- `create role`
- `alter role`
- `sp_addlogin`
- `sp_modifylogin`
- `sp_configure`

Setting the expiration interval for a password

System Administrators and System Security Officers can:

- Specify the expiration interval for a login password at creation
- Change the expiration interval for a login password
- Specify the expiration interval for a role password at creation
- Change the expiration interval for a role password

When a password for a login or role is about to expire, a warning message asks the user to contact the System Security Officer.

Password expiration did not affect roles in versions earlier than Adaptive Server 12.0. Therefore, in Adaptive Server 12.0 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.

Use `sp_addlogin` to set the password expiration interval for a new login. 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.

Concrete identification

Concrete identification enables Adaptive Server to verify chains of ownership between procedures, views, and triggers and the objects they reference in other databases. Adaptive Server identifies users during a session by login name or server user ID (suid). 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.

For more information about concrete identification, see the *System Administration Guide*.

System changes in version 12.0

This section provides a summary of the version 12.0 changes, as follows:

- “New and changed configuration parameters” on page 99
- “New and changed Transact-SQL commands” on page 100
- “New and changed system procedures” on page 101
- “New and changed databases and system tables” on page 104
- “New reserved words” on page 105
- “Changes to the documentation” on page 106

New and changed configuration parameters

This section describes the new and changed configuration parameters in version 12.0.

Table 6-1 summarizes the new configuration parameters.

Table 6-1: New configuration parameters

Parameter	Function
check password for digit	Sets the server to check for at least one character or digit in a password.
dtm detach timeout period	Sets the maximum amount of time that detached transactions remain active in Adaptive Server.
dtm lock timeout period	Sets the maximum amount of time, in seconds, that a distributed transaction branch waits for lock resources to become available.
enable dtm	Enables or disables the Distributed Transaction Management feature.
enable HA	Enables Adaptive Server to be configured for Sybase's Failover in a high availability system.
enable java	Enables or disables Java on the server.
enable xact coordination	Enables or disables Adaptive Server distributed transaction coordination services.
maximum failed logins	Sets the server-wide maximum number of login attempts for logins and roles.
minimum password length	Specifies a server-wide value for minimum password length for both logins and roles.
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.
size of global fixed heap	Specifies the memory space for internal data structures.
size of process object fixed heap	Specifies the memory space for the Java VM for Java objects referenced during a session.
size of shared class heap	Specifies the shared memory space for all Java classes called into the Java VM.
strict dtm enforcement	Determines whether or not Adaptive Server transaction coordination services will strictly enforce the ACID properties of distributed transactions.
text prefetch size	Limits the number of pages of text and image data that can be prefetched into an existing buffer pool.
txn to pss ratio	Determines the total number of transaction descriptors available to the server.

Parameter	Function
xact coordination level	Defines the length of time between attempts to resolve transaction branches that were propagated to remote servers.

Table 6-2 summarizes the changes to existing configuration parameters.

Table 6-2: Changed configuration parameters

Parameter	Change
enable cis	The default value has been changed from 0 (off) to 1 (on).

New and changed Transact-SQL commands

This section describes the new and changed Transact-SQL commands in version 12.0.

Table 6-3 summarizes the new Transact-SQL commands.

Table 6-3: New commands

Command	Function
quiesce database	Suspends and resumes updates to a specified list of databases.
join	Indicates the join uses ANSI syntax for either an inner or outer join.
remove java	Removes one or more Java classes, packages, or JARs from the database.

Table 6-4 summarizes the changes to existing Transact-SQL commands.

Table 6-4: Changed commands

Command	Change
alter role	Includes new lock, unlock, and all overrides options to change the lock status of a role. Includes new passwd expiration, max failed_logins, and min passwd length options to change the password expiration interval, the minimum password length, and the maximum number of failed logins allowed for a specified role.
alter table	Includes new off row and in row options to specify storage for Java-SQL columns.
create role	Includes new lock, unlock, and all overrides options to set the lock status of a role. Includes new passwd expiration, min passwd length, and max failed_logins options to set the password expiration interval, the minimum password length, and the maximum number of failed logins allowed for a specified role.
create table	Includes new off row and in row options to specify storage for Java-SQL columns.

Command	Change
dbcc	Includes new commands for heuristically completing distributed transactions. Includes new rebuild_text command for rebuilding or creating internal data structures for text or image data. Adaptive Server uses these data structures to perform random access and asynchronous prefetch for queries against text or image data.
disk init	Includes the new dsync option to control buffered writes to devices.
disk reinit	Includes the new dsync option to control buffered writes to devices.
execute	Includes the new execute() syntax to dynamically execute Transact-SQL commands.

Table 6-5 summarizes the new and changed options for the set command:

Table 6-5: New and changed set command options

set option	Description
clientappliance	Assigns an application an individual name for the current session.
clienthostname	Assigns a host an individual name for the current session.
clientname	Assigns a client an individual name for the current session.
strict_dtm_enforcement	Determines whether the Adaptive Server propagates transactions to servers that do not support Adaptive Server transaction coordination services.
transactional_rpc	Determines whether or not Adaptive Server coordinates remote procedure calls and updates to remote server data.

Table 6-6 summarizes the new Transact-SQL functions.

Table 6-6: New Transact-SQL functions

Function	Description
license_enabled	Returns 1 if a feature's license is enabled, 0 if the license is not enabled, or null if you specify an invalid license name.

New and changed system procedures

This section describes the new system procedures added and changes made to existing system procedures.

Table 6-7 summarizes the new system procedures.

Table 6-7: New system procedures

System procedure	Function
sp_deviceattr	Changes the dsync setting of an existing database device file.
sp_transactions	Reports detailed information about active transactions.
sp_companion	Performs cluster operations such as configuring Adaptive Server as a secondary companion in a high availability system and moving a companion server from one failover mode to another. sp_companion is run from the secondary companion.
sp_ha_admin	Performs administrative tasks that address second point of failures on Adaptive Servers configured with Sybase's Failover in a high availability system.
sp_helpjava	Displays information about Java classes and JARs installed in the database.

Table 6-8 summarizes the changes made to existing system procedures.

Table 6-8: Changed system procedures

Procedure	Change
sp_addlogin	Specifies the password expiration interval, the minimum password length, and the maximum number of failed logins allowed for a specified login at creation.
sp_displaylogin	Displays the login security related parameters configured for a login. The syntax has not changed; however, the output shows the following additional information: <ul style="list-style-type: none"> • Whether the account is locked • The date of the last password change • The password expiration interval • Whether the password has expired • The minimum length of the password • The maximum number of failed logins allowed before the login is locked • The current number of failed logins
sp_displayroles	Displays the login security-related parameters configured for a role. The output has changed as described for sp_displaylogin.
sp_helpdevice	Output displays the new dsync setting in the “description” column.
sp_lock	Output displays the new lock owner ID (loid) field.
sp_modifylogin	Use to change password expiration interval, the minimum password length, and the maximum number of failed logins allowed for a specified existing login.
sp_monitorconfig	Displays additional usage statistics for transaction descriptors and DTX participants.
sp_who	Output displays the new block_xloid field, which indicates the lock owner ID of the blocking transaction. The older blocked field in the output is renamed to blk_spid.

New utility programs

Table 6-9 describes the new utility programs added to Adaptive Server version 12.0.

Table 6-9: New utility programs

System procedure	Function
extractjava (UNIX) extrjava (Windows NT)	Copies a retained JAR and the Java classes it contains from Adaptive Server to a client file.
installjava (UNIX) instjava (Windows NT)	Installs a JAR and the Java classes it contains from a client file to Adaptive Server.

New and changed databases and system tables

This section describes the changes made to databases and system tables to support version 12.0 features.

Table 6-10 lists the new databases.

Table 6-10: New databases

Database	Function
sybdiagdb	Holds diagnostic configuration data for debugging purposes. This database is for Sybase Technical Support personnel only and should not be used by customers.

Table 6-11 lists the new system tables.

Table 6-11: New system tables

Table	Function
syscolumns	Adds the xtype and xdbid columns. These columns are used only for Adaptive Servers enabled for Java.
syscoordinations	(sybssystemdb database only.) Contains information about remote Adaptive Servers participating in distributed transactions (remote participants) and their coordination states.
sysdevices	status column utilizes a new control bit 0x4000 (16384 Decimal) to control the dsync setting for a database device file.
sysjars	Contains one row for each Java archive file retained in the database. This table is used only for Adaptive Servers enabled for Java.
syslocks	Adds the loid column.
sysprocesses	Adds the block_xloid column. The blocked column is renamed to block_xloid.
sysqueryplans	Stores query text and abstract plans for a query.
syssessions	Contains one row for each client that connects to Adaptive Server with the failover property (for example, isql -Q). This table is used only for Adaptive Servers that are configured for Sybase's Failover in a high availability system.
systransactions	(master database only.) Contains information about Adaptive Server transactions.

Table	Function
sysxtypes	Contains one row for each extended, Java-SQL datatype. This table is used only for Adaptive Servers enabled for Java.

Table 6-12 lists the system tables that have been changed.

Table 6-12: Changed system tables

Table	Change
sysattributes	Stores information for the configuration parameters that set or change the password expiration interval, maximum failed login attempts, and the minimum password length.
sysconfigures	Stores the server-wide values set with the configuration parameters minimum password length and maximum failed logins for logins and roles.
syslogins	The new column, logincount, indicates the number of failed logins. A successful login automatically resets the number of failed logins to 0. The datatype of this column is int.
syservroles	<p>The following columns have been added:</p> <ul style="list-style-type: none"> • pwdate – records the date and time when the password of a role was last changed. The datatype of this column is datetime. • status – indicates the status of the role. The datatype of this column is smallint. The status is shown as a numeric value that corresponds to the combination of expiration and lock status for the role. The values are: <ul style="list-style-type: none"> 0 – role is not expired or locked. 2 – role is not expired but is locked. 4 – role is expired but is not locked. 6 – role is expired and locked.
sysusers	Adaptive Server 12.0 no longer has the unique attribute of clustered index on sysusers.suid.

For more information about system tables, see the *Adaptive Server Reference Manual*.

New reserved words

The *Adaptive Server Reference Manual* lists all of the reserved words for version 12.0. Reserved words cannot be used as object names or column names.

You must change all database names that are new reserved words before you can upgrade from an earlier version of the server. You can change table, view, and column names or use delimited identifiers. Once you upgrade to version 12.0, you cannot use database objects whose names are new reserved words until you modify your procedures, SQL scripts, and applications.

Changes to the documentation

Adaptive Server version 12.0 includes several new books that describe separately-licensed Adaptive Server features such as Java, High Availability, and Distributed Transaction Management.

Information from the following two books has been moved into the *System Administration Guide* and *Performance and Tuning Guide*:

- *Security Features User's Guide*
- *Security Administration Guide*

Information from *Configuring Adaptive Server for UNIX* has been moved into the *Installation Guide*.

The Adaptive Server documentation set no longer includes the following documents:

- *Adaptive Server Enterprise Getting Started*
- *Introducing Adaptive Server Enterprise*
- *Master Index to Adaptive Server Documentation*
- *Navigating Adaptive Server Enterprise Documentation*
- Online sybsyntax database and sp_syntax procedure
- *Quick Reference Card*

Changes that may affect existing applications

This section describes system changes introduced by version 12.0 that may affect your applications if you are upgrading from an earlier version.

Inserts on datarow-locking tables

As of Adaptive Server 12.0, inserts on datarow-locking tables no longer block select statements under isolation level one or two.

New Transact-SQL keywords for 12.0

See “New reserved words” on page 105.

dsync on by default for UNIX file devices

Beginning with version 12.0, Adaptive Server initializes UNIX file system devices using the *dsync* option to ensure recovery. If you require earlier behavior or performance characteristics, change the *dsync* setting using *sp_deviceattr*. See *disk init* and *sp_deviceattr* in the *Adaptive Server Reference Manual* for more information.

Index

A

- abstract plans for queries 90
- access control 60
- accessing query results as objects 47
- active-active 27
 - failover in Sun Clusters 27
- active-passive
 - failover in Sun Clusters 27
 - fall back in Sun Clusters 27
 - Sun clusters 26
- Adaptive Server
 - changes that may affect applications for 12.5 73
 - column size 54
 - external file system support 59
 - logical page sizes 54
 - number of groups 55
 - number of logins 55
 - number of users 55
 - SQLJ 56
 - unichar* support 62
 - version 12.5 relaxed limits 54
 - wide columns and data truncation 74
- Adaptive Server Enterprise Web Services 14
- aggressive garbage collection 29
- alter role** command 100
- alter table** command 67, 100
 - enabling or disabling triggers 91
- ANSI joins 88
- ASE Replicator 45
- ASE Web Services 14
- assigning names to clients 83
- asynchronous log service 27
- @lock_timeout** 39
- @max_precision** 39
- automatic database expansion 2

B

- Backup Server
 - dump operations 81
 - high performance backup and restore 81
 - non-rewinding tapes 81
- backward compatibility 15
- Bourne shell xiv
- buffer pool, larger 33
- bytes, converting to Unicode 49

C

- C shell xiv
- cache wizard, for **sp_sysmon** 13
- changing identity gaps 84
- character processing engine, XQL 48
- check password for digit** configuration parameter 99
- checkpoint** command 19
- checkpoints 31
- checkstorage** 31
- Chinese character set 30
- client connection
 - failover 27
 - Sun Clusters failover 27
- clientapplname** command 68, 101
- clienthostname** command 101
- clientname** command 101
- clients, assigning names to 83
- cluster
 - active-active setup 26
 - active-passive setup 26
 - synchronization 46
- column size 54
- commands
 - disk resize** 45
- Component Integration Services (CIS) 32, 45, 49, 92
- compressed archive support 61
- configname** options for **sp_helpconfig** 85

configuration parameter

- deadlock pipe active** 36
- deadlock pipe max messages** 36
- dtm detach timeout period** 29
- dynamic allocation on demand** 34
- enable ldap user auth** 18
- enable monitoring** 36
- errorlog pipe active** 36
- errorlog pipe max messages** 36
- housekeeper free write percent** 29
- max sql text monitored** 36
- number of large io buffers increased** 34
- number of network listeners** 5
- number of user connections** 18
- object lockwait timing** 36
- per object statistics active** 36
- plan text pipe active** 36
- plan text pipe max messages** 36
- process wait events** 36
- SQL text pipe active** 36
- sql batch active** 36
- sql text pipe max messages** 36
- statement pipe active** 36
- statement pipe messages** 36
- statement statistics active** 36
- wait even timing** 36

costing for queries 88

create database command 37

create role command 100

create table command 84, 100

creating objects 98

.cshrc file xiv

D

data in native character set 48

data manipulation languages 60

database

- for debugging 85
- integrity, ensuring 82
- object owners, tracking 98
- suspending 83

database expansion 2

datatypes

- date 7

- enhancements for 90
- java classes as 86
- time 7

dbcc command 37, 101

- described 45
- discretionary access control 45

dbcc engine command 83

delete command, tracking creator of 98

directory control layer (DCL) 93

disabling triggers 91

discretionary access control of **dbcc** commands 45

disk init command 82, 101

disk reinit command 101

disk resize 45

Distributed Transaction Management 92

document heading, XML 48

DOM

- getting results as 47

dsync command 82

dtm detach timeout period configuration parameter 29, 99

dtm lock timeout period configuration parameter 99

dump compression

- defined 61
- syntax example 61

dump database command

- compress** option 61

dump operations and Backup Server 81

dump transaction command

- compress** option 61

dynamic listeners 5

E

enable cis configuration parameter 100

enable dtm configuration parameter 99

enable HA configuration parameter 99

enable java configuration parameter 99

enable job scheduler configuration parameter 18

enable ldap user auth configuration parameter 9, 18

enable sort-merge joins and JTC configuration parameter 87

enable xact coordination configuration parameter 99

enabling triggers 91

engine

- query processing 48
- query, XQL 48
- XQL character processing 48
- engines offline 83
- enhancements
 - for datatypes 90
 - for queries 87
- execute** command 89, 101
- execution
 - of Transact-SQL commands 89
- expressions, maximum length 55
- external file system support 59
- extracting SQL base types
 - from XML document 47
- extractjava** (UNIX) utility 104
- extrjava** (NT) utility 104

F

- Failover, using in a high availability system 78–80, 99
- fallback in Sun Clusters 27
- files
 - .cshrc xiv
 - .profile xiv
- functions
 - current_date** 19
 - current_time** 19
 - day** 40
 - identity_burn_max** 40
 - left** 40
 - len** 40
 - month** 40
 - newid** 40
 - next_identity** 40
 - square** 40
 - str_replace** 40
 - tempdb_id** 40
 - year** 40
- functions, Java 86

G

- getDOM()** Java method 47

- getDouble()** Java method 47
- getFloat()** Java method 47
- getInt()** Java method 47
- getString()** Java method 47
- getXML()**, Java method 47
- global variables 33, 34, 39
 - 33, 33
 - @datefirst** 35, 39
 - @lock_timeout** 35, 39
 - @max_precision** 35, 39
 - @tempdbid** 39
- grant dbcc** command 51
- groups, number of for 12.5 55

H

- high availability, on Sun Cluster 3.5 10
- high availability, using Sybase's Failover 78–80, 99
- high performance backup and restore 81
- housekeeper
 - chores 29
 - configuration parameter license information 29
 - wash 29
- housekeeper free write percent** configuration
 - parameter 29

I

- identity gap
 - changing 84
 - parameter 84
- image datatypes 90
- increase, configuration parameter **number of large io buffers** 34
- indexes
 - rebuilding 80
 - Unicode-compliant 48
- insert** command 51
 - Component Integration Services 49
 - tracking creator of 98
- installdbextend script 3
- installjava** (UNIX) utility 104
- instjava (NT)) utility 104
- integrity

- database 82
- message 94
- interfaces file 93
- isFirst()** Java method 47
- isLast()** Java method 47

J

- Java
 - capabilities 86
 - classes as datatypes 86
 - datatypes 86
 - in Adaptive Server Enterprise 85–86
 - installing in Adaptive Server 56
 - user-defined functions 86
- Java class
 - com.sybase.xml.xql.XqlResults.SybXqlResult** 47
- Java exception
 - double** 47
- Java method
 - getXML()** 47
 - java.io.InputStream** 48
 - java.lang.String** 48
 - query()**, invoking with literal 48
 - getDOM()** 47
 - getDouble()** 47
 - getFloat()** 47
 - getInt()** 47
 - getString()** 47
 - isFirst()** 47
 - isLast()** 47
 - next** 47
 - reset()** 47
 - setEncoding** 48
- Java methods in result set 47
- java.io.InputStream** 48
- java.lang.String** 48
- Job Scheduler 12
- join** 67, 100
- joins
 - merged 87
 - using ANSI syntax 88

K

- keywords
 - new for version 12.5 72
- Korn shell xiv

L

- large I/O buffers, improvement 34
- larger buffer pool 33
- lazy garbage collection 29
- LDAP
 - specifying in *libtcl.cfg* 58
 - user authorization 9
- libtcl*.cfg* file 58
- license information, configuration parameter 29
- licensed feature 78, 85, 92
- limits on tables in queries 87
- logical page sizes 54
- login security, user-defined 94
- logins
 - number of for 12.5 55
 - server 98
 - unified 94

M

- managing distributed transactions 92
- maximum failed logins configuration parameter 99
- memory allocation 2
- merge joins 87
- message integrity 94
- Microsoft Distributed Transaction Coordinator (MSDTC) 92
- minimum password length configuration parameter 99
- mon_role** 25
- monCachedObject* system table 41
- monCachedProcedures* system table 41
- monCachePool* system table 41
- monDataCache* system table 40
- monDeadLock* system table 41
- monDeviceIO* system table 41
- monEngine* system table 40
- monErrorLog* system table 41
- monIOQueue* system table 41

monitoring indexes 26
 monitoring tables 24–25
 CIS and 25
 mon_role 25
 Using Transact-SQL to monitor performance 25
monLocks system table 41
monNetworkIO system table 41
monOpenDatabase system table 40
monOpenObjectActivity system table 41
monProcedureCache system table 40
monProcess system table 41
monProcessActivity system table 41
monProcessLookup system table 41
monProcessNetIO system table 41
monProcessObject system table 41
monProcessProcedures system table 41
monProcessSQLText system table 41
monProcessStatement system table 41
monProcessWait system table 41
monState system table 40
monSysPlanText system table 41
monSysSQLText system table 41
monSysStatement system table 41
monSysWait system table 41
monSysWorkerThread system table 41
monTableColumns system table 40
monTableParameters system table 40
monTables system table 40
monWaitClassinfo system table 41
monWaitEventinfo system table 41
mount Transact-SQL command 18
 multiple temporary databases. See tempdbs.

N

native character sets
 of data 48
 querying in 48
 native code sets
 converting results back to 49
 native XML processor 8
 network security 93
 new global variables 34, 39
next Java method 47
 non-rewinding tapes for Backup Server 81

number of
 dtx participants configuration parameter 99
 groups 55
 logins 55
 users 55
 number of databases per transaction 34
number of histogram steps configuration parameter 51
number of user connections configuration parameter 18

O

objects
 accessing query results as 47
 creating 98
 Open Client and Adaptive Server's limits 74
 optimistic index locking 27
 optimization for queries 87
 option packages 44
 Content Management Edition 44
 descriptions 44
 e-Business Edition 44
 Security and Directory Services Edition 44
 XML Management edition 44
 overview of cluster synchronization 46

P

parser, XML, Xerces 48
 performance
 of network security 94
 performance improvements 33
 plans, query 90
 .profile file xiv
 proxy table 49

Q

queries
 abstract plans for 90
 enhancements to 87
 improvements in costing 88

Index

- in native character sets 48
- number of tables in 87
- XML documents 48
- query engine
 - XQL 48
- query results, accessing as objects 47
- quiesce database command 60, 83, 100

R

- raw partitions 82
- rebuilding indexes 80
- remove java** command 100
- reorg rebuild** command 80
- replication
 - with ASE Replicator 45
- repositories, moving information with synchronization 46
- reserved words
 - new for version 12.0 107
- reset()** Java method 47
- result set
 - Java methods in 47
- results
 - converting to native code sets 49
 - obtaining in Java string 49
- revoke dbcc** command 51

S

- sample code
 - for XML results 49
- sampling
 - statistics 28
- security
 - network 93
 - user-defined login 94
- select** command 19, 88
- select into** command 33, 84
- server user name and ID 98
- set bulk array size** 32
- set bulk batch size** 32
- set** command 83
 - clientappliance** 68, 101
 - clienthostname** 101
 - clientname** 101
 - identity_update** 34
- set identity_update** command 37
- shells xiv
- size of global fixed heap** configuration parameter 99
- size of process object fixed heap** configuration parameter 99
- size of shared class heap** configuration parameter 99
- source Adaptive Server, for migration 50
- sp_addlogin** system procedure 10, 103
- sp_addserver** system procedure 20
- sp_bindcache** system procedure 20, 38
- sp_cacheconfig** system procedure 20
- sp_chgattribute** system procedure 38, 84
- sp_companion** system procedure 102
- sp_configure** command 90
- sp_dbcc_checkverify** system procedure 38
- sp_dbcc_configreport** system procedure 38
- sp_dbcc_exclusions** system procedure 39
- sp_dbcc_faultreport** system procedure 20, 38
- sp_dbcc_help_fault** system procedure 39
- sp_dbcc_recommendations** system procedure 39
- sp_dbcc_summary_report** system procedure 38
- sp_dbcc_updateconfig** system procedure 38
- sp_dbccvaluatedb** system procedure 20
- sp_dbextend** system procedure 2, 20
- sp_dboption** system procedure 20, 38
- sp_deviceattr** system procedure 102
- sp_displaylogin** system procedure 103
- sp_displayroles** system procedure 103
- sp_ha_admin** system procedure 102
- sp_help** system procedure 38, 84
- sp_helpconfig** system procedure 85
- sp_helppdb** system procedure 38
- sp_helpdevice** system procedure 103
- sp_helpjava** system procedure 102
- sp_helptext** system procedure 39, 51
- sp_ldapadmin** system procedure 20
- sp_listener** system procedure 5, 20
- sp_lock** system procedure 103
- sp_modifylogin** system procedure 103
- sp_monitorconfig** system procedure 30, 38, 103
- sp_serveroption** system procedure 20
- sp_sysmon** system procedure 20
- sp_tempdb** system procedure 39
- sp_transactions** system procedure 102

sp_unbindcache system procedure 20
sp_who system procedure 69, 103
 SQL
 base types, extracting from XML document 47
 converting XML to 47
 derived tables 6
 SQL compatibility 14
 SQL mapping functions 30
 SQL2002 standard 6
 SQLJ aliases for XQL 30
 SQLJ in Adaptive Server 55
 SQLX
 standard 30
 support 7
 SSL, defined 58
 statistics
 sampling 28
status column, *sysrvroles* 105
 store procedures, maximum length 55
strict dtm enforcement
 configuration parameter 99
strict_dtm_enforcement
 command 101
 suid (server user ID), object creator and 98
 Sun Clusters 26
 active-active failover 27
 active-active fallback 27
 active-active setup 26
 active-passive fall back 27
 active-passive failover 27
 client connection failover 27
 support for UNIX file systems 82
 suspending databases 83
sybdiagdb database 85
sybmgmtdb database 12
sybmigrate
 Adaptive Server version level 50
 source Adaptive Server 50
 target Adaptive Server 50
sybmigrate utility program 39, 52
 synchronization
 moving information between repositories 46
 overview 46
 syntax conventions xiv
sysattributes system table 42, 105
syscolumns system table 104

sysconfigures system table 105
syscoordinations system table 104
sysdatabases system table 42
sysdevices system table 104
syservroles system table 105
sysjars system table 104
syslocks system table 104
syslogins system table 105
sysprocesses system table 104
sysessions system table 104
sysrvroles table 105
systransactions system table 104
sysusers system table 105
sysxtypes system table 105

T

Table 104
 tables
 number of, in a query 87
 proxy 49
 taking engines offline 83
 target Adaptive Server, for migration 50
 tempdbs
 and the system *tempdb* 29
 described 28
tempdbs
 described 28
 text datatypes 90
text prefetch size configuration parameter 90, 99
 three housekeepers 29
 titles system table 90
transactional_rpc command 101
 transactions, managing distributed 92
 Transact-SQL, dynamic execution of 89
 triggers
 disabling 91
 enabling 91
txn to pss ratio configuration parameter 99

U

unichar support 62
 Unicode parser 6

Index

- Unicode-compliant indexes 48
- unified login 94
- union operator in a view 58
- UNIX file system support 82
- unmount** Transact-SQL command 18
- update** command tracking creator of 98
- update statistics** 28
- update statistics** command 37
- updating statistics
 - use sampling 28
- updating statistics** 28
- upgrade, incorporating proxy tables 49
- user-defined
 - functions for Java 86
 - login security 94
- users, number of for 12.5 55
- UTF-8 sort order 6
- getting results as DOM 47
- iterating through document 47
- parser, accepting bytes 48
- results, sample code 49
- XML processor 8
- XQL
 - character processing engine 48
 - query engine 48

V

- variables
 - global, new 34
 - maximum length 55
- version for Adaptive Server on **sybmigrate** 50
- views, using union operators 58

W

- Web services 14
 - with Adaptive Server 14
- wide column 74

X

- X/Open XA protocol 92
- xact coordination level** configuration parameter 100
- Xerces parser 48
- XML
 - converting to SQL data 47
 - document heading 48
 - document, character encoding 48
 - document, extracting SQL base types from 47
 - document, querying 48