What’s New in Sybase Adaptive Server Enterprise?
## Contents

### About This Book

### CHAPTER 1 New Features in Adaptive Server Version 12.5

<table>
<thead>
<tr>
<th>Feature</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>New features in version 12.5</td>
<td>1</td>
</tr>
<tr>
<td>Relaxed server limits</td>
<td>2</td>
</tr>
<tr>
<td>Varying logical page sizes</td>
<td>2</td>
</tr>
<tr>
<td>Number of columns and column size</td>
<td>3</td>
</tr>
<tr>
<td>Maximum length of expressions, variables, and stored procedure arguments</td>
<td>3</td>
</tr>
<tr>
<td>Number of logins</td>
<td>3</td>
</tr>
<tr>
<td>Performance implications of new limits</td>
<td>4</td>
</tr>
<tr>
<td>Dynamic reconfiguration</td>
<td>4</td>
</tr>
<tr>
<td>Dynamic memory allocation</td>
<td>4</td>
</tr>
<tr>
<td>Changes to how memory is allocated</td>
<td>5</td>
</tr>
<tr>
<td>SQLJ stored procedures and functions (Java)</td>
<td>5</td>
</tr>
<tr>
<td>XML in the database</td>
<td>6</td>
</tr>
<tr>
<td>Union in views</td>
<td>6</td>
</tr>
<tr>
<td>Internet directory services (LDAP)</td>
<td>7</td>
</tr>
<tr>
<td>Secure Socket Layers (SSL)</td>
<td>7</td>
</tr>
<tr>
<td>Enterprise Java Beans (EJB Server)</td>
<td>8</td>
</tr>
<tr>
<td>External file system support</td>
<td>8</td>
</tr>
<tr>
<td>Row-level access control</td>
<td>9</td>
</tr>
<tr>
<td>Suspending and resuming updates to databases</td>
<td>9</td>
</tr>
<tr>
<td>Java.net support</td>
<td>10</td>
</tr>
<tr>
<td>Compressed backups</td>
<td>10</td>
</tr>
<tr>
<td>Unichar support</td>
<td>11</td>
</tr>
<tr>
<td>Changes to CIS</td>
<td>12</td>
</tr>
<tr>
<td>SQL debug utility</td>
<td>12</td>
</tr>
<tr>
<td>Migration utility</td>
<td>13</td>
</tr>
</tbody>
</table>

### CHAPTER 2 System Changes in Adaptive Server Version 12.5

<table>
<thead>
<tr>
<th>Feature</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>System changes in version 12.5</td>
<td>15</td>
</tr>
<tr>
<td>Changes to configuration parameters</td>
<td>15</td>
</tr>
</tbody>
</table>
Changes to Transact-SQL commands ........................................ 17
New and changed system procedures ....................................... 19
New utility programs ................................................................. 20
Changes to current utility programs .......................................... 21
Changes to databases and system tables ................................. 21
New reserved words ................................................................ 24
New global variables ............................................................... 24
Changes to the documentation ................................................ 25
Changes that may effect existing applications ......................... 25
New Transact-SQL keywords ................................................... 25
Wide column and data truncation .......................................... 26
Changes that effect Client Server applications ....................... 26
Maximum number of expressions in a select statement ........ 28

CHAPTER 3 New Features and System Changes in Adaptive Server Version 12.0

New features in version 12 ...................................................... 29
Sybase's Failover for a high availability system ....................... 30
Rebuilding indexes .................................................................. 32
New Backup Server features .................................................. 32
High performance backup and restore ................................... 33
Non-rewinding tapes ............................................................ 33
UNIX file system support ....................................................... 33
Modifying an existing table's schema with alter table ............... 34
Suspending database updates with quiesce database ............... 35
Taking Adaptive Server engines offline .................................. 35
User settable process information .......................................... 35
Identity number gap for tables .............................................. 36
Diagnostic database ............................................................. 36
Java in Adaptive Server Enterprise ......................................... 37
Capabilities of Java in Adaptive Server ................................. 37
Number of tables in a query .................................................... 38
Query processing and optimization enhancements ................. 39
Merge joins ........................................................................... 39
Query costing improvements ................................................ 39
ANSI joins ............................................................................ 40
Dynamic execution of Transact-SQL ................................. 41
text and image datatype enhancements ............................ 41
Abstract plans ........................................................................ 41
Disabling triggers ............................................................... 42
Cache partitions ..................................................................... 43
Distributed Transaction Management .................................... 43
Enhancements to Component Integration Services (CIS) ....... 44
Network security ................................................................. 45
Chapter 6: New Features in SQL Server Release 11.0

New Features in SQL Server Release 11.0 ........................................ 111

New features in release 11.0 ....................................................... 111
User-defined caches ............................................................. 112
Data storage changes ........................................................... 115
Transaction log changes ....................................................... 117
Isolation level 0...................................................................... 119
Lock manager changes ........................................................ 119
Housekeeper task ................................................................. 122
SQL Server configuration ...................................................... 122
Lock escalation...................................................................... 123
Multiple network engines ..................................................... 123
Improvements to showplan .................................................... 124
Query and data modification changes................................... 125
Upgrading database dumps .................................................... 125
Tape device determination by Backup Server....................... 126
IDENTITY column changes................................................... 127
New text and image global variables..................................... 128
System changes in release 11.0 .................................................. 128
New online database command............................................ 128
Changes to existing commands ............................................ 129
New set options..................................................................... 129
New system procedures........................................................ 130
Changes to system procedures............................................. 131
New system tables ................................................................ 131
Changes to existing system tables ....................................... 131
Changes that may affect existing applications ...................... 132
New Transact-SQL keywords in release 11.0 ....................... 132
Changes to SQL Server configuration................................. 133
Subquery changes ................................................................ 137
Changes to showplan output in release 11.0 ......................... 141
New caching strategies may affect performance 142
Upgrading database dumps 142
Partitions and physical data placement 142

Index 145
About This Book

**Audience**

This manual is intended for customers who are upgrading to Adaptive Server™ version 12.5.

**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 document also covers the new features added and system changes made in releases 11.0, 11.5, 11.9.2, and 12.0. This information is provided for users who are upgrading to Adaptive Server version 12.5 from earlier releases.

This manual consists of the following chapters:

- Chapter 1, “New Features in Adaptive Server Version 12.5” describes the features added in release 12.5.
- Chapter 2, “System Changes in Adaptive Server Version 12.5” describes system additions and changes introduced in release 12.5, and how these additions and changes might affect your existing applications.
- Chapter 3, “New Features and System Changes in Adaptive Server Version 12.0” describes the Adaptive Server system changes made to support the new features. It also discusses how these additions and changes might affect your existing applications.
- Chapter 4, “New Features in Adaptive Server Version 11.9.2” describes the features added and system changes made in version 11.9.2.
- Chapter 5, “New Features in Adaptive Server Release 11.5” describes the features added and system changes made in release 11.5, and how these additions and changes might affect your existing applications.
- Chapter 6, “New Features in SQL Server Release 11.0” describes features and system changes introduced in release 11.0.
The following documents comprise the Sybase Adaptive Server Enterprise documentation:

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

- *What’s New in Adaptive Server Enterprise?* – describes the new features in Adaptive Server version 12, the system changes added to support those features, and the changes that may affect your existing applications.

- *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.

- *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.

- *Adaptive Server Reference Manual* – contains detailed information about all Transact-SQL commands, functions, procedures, and datatypes. This manual also contains a list of the Transact-SQL reserved words and definitions of system tables.


- The *Utility Guide* – documents the Adaptive Server utility programs, such as *isql* and *bcp*, which are executed at the operating system level.
About This Book

- **Error Messages and Troubleshooting Guide** – explains how to resolve frequently occurring error messages and describes solutions to system problems frequently encountered by users.

- **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.

- **Java in Adaptive Server Enterprise** – describes how to install and use Java classes as datatypes and user-defined functions in the Adaptive Server database.

- **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.

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

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

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

**Other sources of information**

Use the Sybase Technical Library CD and the Technical Library Product Manuals Web site to learn more about your product:

- Technical Library CD contains product manuals and is included with your software. The DynaText browser (downloadable from Product Manuals at http://www.sybase.com/detail/1,3693,1010661,00.html) 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.

- 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 the Technical Documents Web site (formerly known as Tech Info Library), the Solved Cases page, and Sybase/Powersoft newsgroups.

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.
❖ For 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.
4 Select the Certification Report filter, specify a time frame, and click Go.
5 Click a Certification Report title to display the report.

❖ For the latest information on EBFs and Updates
1 Point your Web browser to Technical Documents at http://www.sybase.com/support/techdocs/.
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 Specify a time frame and click Go.
4 Select a product.
5 Click an EBF/Update title to display the report.

❖ To create 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.

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.

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:

\textit{this font}

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

\texttt{/usr/u/sybase}

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

\texttt{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 \texttt{.cshrc}. The initialization file for the Bourne shell is called \texttt{.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:

\begin{table}[h]
\centering
\begin{tabular}{|l|p{10cm}|}
\hline
\textbf{Key} & \textbf{Definition} \\
\hline
command & Command names, command option names, utility names, utility flags, and other keywords are in \textit{Arial bold}. \\
\hline
\textit{variable} & Variables, or words that stand for values that you fill in, are in \textit{Italics}. \\
\hline
\{ \} & Curly braces indicate that you choose at least one of the enclosed options. Do not include braces in your option. \\
\hline
\[ \] & Brackets mean choosing one or more of the enclosed options is optional. Do not include brackets in your option. \\
\hline
( ) & Parentheses are to be typed as part of the command. \\
\hline
\| & The vertical bar means you can select only one of the options shown. \\
\hline
, & 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. \\
\hline
\end{tabular}
\caption{SQL syntax conventions}
\end{table}

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.
CHAPTER 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.

In addition to adding support for multiple logical page sizes, wider columns, a larger number of rows, Adaptive Server version 12.5 introduces several new licensed features for specialized use. These include:

- Extended Full-Text Search Engine
- Secure Sockets Layer (SSL)
- Extended File System
- Java
- EJB Server

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 12.5

The new features in Adaptive Server version 12.5 are:

<table>
<thead>
<tr>
<th>Name</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Relaxed server limits&quot;</td>
<td>2</td>
</tr>
<tr>
<td>&quot;Dynamic reconfiguration&quot;</td>
<td>4</td>
</tr>
<tr>
<td>&quot;SQLJ stored procedures and functions (Java)&quot;</td>
<td>5</td>
</tr>
<tr>
<td>&quot;XML in the database&quot;</td>
<td>6</td>
</tr>
<tr>
<td>&quot;Union in views&quot;</td>
<td>6</td>
</tr>
<tr>
<td>&quot;Internet directory services (LDAP)&quot;</td>
<td>7</td>
</tr>
</tbody>
</table>
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
- Number of arguments for stored procedures
- Number of expressions in a select statement
- Number of users per database
- 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 a 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 an APL table can be as large as 4010 bytes, less the row format overheads. See Table 1-1 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 1-1 lists the limits for the number of logins, users, and groups for Adaptive Server.

<table>
<thead>
<tr>
<th>Item</th>
<th>Version 12.0 limit</th>
<th>Version 12.5 limit</th>
<th>New range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of logins per server (SUID)</td>
<td>64K</td>
<td>2 billion plus 32K</td>
<td>-32768 to 2 billion</td>
</tr>
<tr>
<td>Number of users per database</td>
<td>48K</td>
<td>2 billion less 1032193</td>
<td>-32768 to 16383; 1048577 to 2 Billion</td>
</tr>
</tbody>
</table>
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 parameter that effect memory are now dynamic, and the server does not have to be restarted for them to take effect. For a full list of the configuration parameters that have changed from static to dynamic, see Table 2-3 in Chapter 2, “System Changes in Adaptive Server Version 12.5.”
CHAPTER 1  New Features in Adaptive Server Version 12.5

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 release 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 allows you to install Java classes in the database and then reference them as datatypes in SQL. Version 12.0 also allows 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 Transarc-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 OrderXML class, 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 ResultSetXML class, 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 were not 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, which allow 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 are an infrastructure that provide:

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

You 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 Socket Layers (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 his user name and password over the secure, encrypted connection.
- A comparison of the digital signature on the server certificate can determine whether the data received by the client was modified before reaching the intended recipient.

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 multi-tier 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:

```sql
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. You can specify up to 16 bytes. If you do not specify any, the tab character is used as the default.
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 enforces row-level protection through access rules. Access rules 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:

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

or,

```sql
quiesce database tag_name release
```
Java.net support

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 email 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 client or server process.

Compressed backups

Adaptive Server Enterprise version 12.5 provides a feature that 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`, will also operate on `unichar` and `univarchar`.

Note however, that 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 ASE configuration option. To use these new data types, 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 CIS

Adaptive Server Enterprise version 12.5 includes many enhancements to Component Integration Services.

Component Integration Services is fully compatible with the new features of Adaptive Server Enterprise documented in this book.

- Distributed query optimization
- Transaction management
- Extended data access
- Login name/password mapping to remote systems
- XNL - Extensible new limits
- Unicode support - new datatypes for support of 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

`sqldbg` is a command-line utility that debugs stored procedures and triggers. You can:

- attach `sqldbg` to a task
- set, enable, and disable breakpoints
- step through a task one line at a time
- step into and out of procedures
- detach `sqldbg` 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
This chapter describes the system changes introduced in Adaptive Server version 12.5.

Topics include:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>System changes in version 12.5</td>
<td>15</td>
</tr>
<tr>
<td>Changes that may effect existing applications</td>
<td>25</td>
</tr>
</tbody>
</table>

System changes in version 12.5

This section summarizes the version 12.5 changes:

- “Changes to configuration parameters” on page 15
- “Changes to Transact-SQL commands” on page 17
- “New and changed system procedures” on page 19
- “Changes to databases and system tables” on page 21
- “New reserved words” on page 24
- “Changes to the documentation” on page 25

Changes to configuration parameters

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

New configuration parameters

Table 2-1 summarizes the new configuration parameters.
System changes in version 12.5

**Table 2-1: New configuration parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>allocate max shared memory</td>
<td>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.</td>
</tr>
<tr>
<td>cis bulk insert array size</td>
<td>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.</td>
</tr>
<tr>
<td>dynamic allocation on demand</td>
<td>Determines when memory is allocated for changes to dynamic memory configuration parameters.</td>
</tr>
<tr>
<td>enable enterprise java beans</td>
<td>Enables or disables the EJB Server.</td>
</tr>
<tr>
<td>enable file access</td>
<td>Enables or disables access through proxy tables to the External File System. Requires a license for ASE_XFS.</td>
</tr>
<tr>
<td>enable full-text search</td>
<td>Enables or disables Enhances Full-Text Search services. Requires a license for ASE_EFTS.</td>
</tr>
<tr>
<td>enable row level access control</td>
<td>Enables or disables row level access control.</td>
</tr>
<tr>
<td>enable ssl</td>
<td>Enables or disables Secure Sockets Layer session-based security</td>
</tr>
<tr>
<td>enable surrogate processing</td>
<td>Enables or disables the processing and maintains the integrity of surrogate pairs in Unicode data.</td>
</tr>
<tr>
<td>enable unicode normalization</td>
<td>Enables or disables Unilib character normalization.</td>
</tr>
<tr>
<td>heap memory per user</td>
<td>Specifies the heap memory per user for Adaptive Server.</td>
</tr>
<tr>
<td>max memory</td>
<td>Specifies the maximum amount of total logical memory that you can configure Adaptive Server to allocate.</td>
</tr>
<tr>
<td>number of engines at startup</td>
<td>Specifies the number of engines that are brought online at startup.</td>
</tr>
<tr>
<td>number of java sockets</td>
<td>Specifies the maximum amount of total physical memory that you can configure Adaptive Server to allocate.</td>
</tr>
<tr>
<td>procedure cache size</td>
<td>Specifies the size of the procedure cache in 2K pages.</td>
</tr>
<tr>
<td>total logical memory</td>
<td>Specifies the amount memory that Adaptive Server is configured to use.</td>
</tr>
<tr>
<td>total physical memory</td>
<td>Displays the amount of memory that is being used by Adaptive Server at a given moment in time.</td>
</tr>
</tbody>
</table>

**Changes to configuration parameters**

Table 2-2 summarizes changes to existing configuration parameters.

**Table 2-2: Changed configuration parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>procedure cache percent</td>
<td>Removed from Adaptive Server version 12.5. You now determine the size of the procedure cache with procedure cache size.</td>
</tr>
<tr>
<td>total memory</td>
<td>Displays the total logical memory for the current configuration of Adaptive Server</td>
</tr>
</tbody>
</table>
Table 2-3 lists the configuration parameters that were static, but are now dynamic.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>size of process object heap</td>
<td>Now a server-wide setting and not assigned to a specific task.</td>
</tr>
<tr>
<td>max cis remote connections</td>
<td>Removed from Adaptive Server version 12.5. The max number is now determined by the max number of file descriptors available to a single process for a given operating system</td>
</tr>
</tbody>
</table>

Table 2-3: Configuration parameters changed from static to dynamic

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

Changes to Transact-SQL commands

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

Table 2-4 summarizes the new Transact-SQL commands.

Table 2-4: New commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>create function</td>
<td>Creates a SQLJ function, which is a Java method with a SQL name.</td>
</tr>
<tr>
<td>(SQLJ)</td>
<td></td>
</tr>
<tr>
<td>drop function</td>
<td>Removes a SQLJ function from the database.</td>
</tr>
</tbody>
</table>

Changed Transact-SQL commands

Table 2-5 summarizes the changes to existing Transact-SQL commands.

Table 2-5: Changed commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>disk init</td>
<td>Allows the following unit specifiers for the device size:</td>
</tr>
<tr>
<td></td>
<td>‘k’ or ‘K’ (kilobytes), ‘m’ or ‘M’ (megabytes), and ‘g’ or ‘G’ (gigabytes)</td>
</tr>
<tr>
<td>disk reinit</td>
<td>Allows the following unit specifiers for the device size:</td>
</tr>
<tr>
<td></td>
<td>‘k’ or ‘K’ (kilobytes), ‘m’ or ‘M’ (megabytes), and ‘g’ or ‘G’ (gigabytes)</td>
</tr>
<tr>
<td>create table</td>
<td>Supports the new size_in_bytes parameter, which allows you to specify the size of an in-row Java-SQL column.</td>
</tr>
<tr>
<td>alter table</td>
<td>Allows the following unit specifiers for the device size:</td>
</tr>
<tr>
<td></td>
<td>‘k’ or ‘K’ (kilobytes), ‘m’ or ‘M’ (megabytes), and ‘g’ or ‘G’ (gigabytes)</td>
</tr>
<tr>
<td>alter database</td>
<td>Allows the following unit specifiers for the device size:</td>
</tr>
<tr>
<td></td>
<td>‘k’ or ‘K’ (kilobytes), ‘m’ or ‘M’ (megabytes), and ‘g’ or ‘G’ (gigabytes)</td>
</tr>
<tr>
<td>set</td>
<td>You can update as many as 1024 columns in the set clause using literals, variables, or expressions returned from a subquery</td>
</tr>
<tr>
<td>select...for browse</td>
<td>You cannot use the select...for browse option on tables containing more than 255 columns</td>
</tr>
<tr>
<td>select select_list into...at pathname</td>
<td>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.</td>
</tr>
<tr>
<td>compute</td>
<td>If a compute clause includes a group by clause:</td>
</tr>
<tr>
<td></td>
<td>• The compute clause cannot contain more than 255 aggregates.</td>
</tr>
<tr>
<td></td>
<td>• The group by clause cannot contain more than 255 columns.</td>
</tr>
<tr>
<td></td>
<td>Columns included in a compute clause cannot be longer than 255 bytes.</td>
</tr>
<tr>
<td>like</td>
<td>The character string indicated by the like keyword cannot be longer than 255 bytes</td>
</tr>
<tr>
<td>declare cursor</td>
<td>You can include as many as 1024 columns in an update clause of a client’s declare cursor statement</td>
</tr>
<tr>
<td>+ operator</td>
<td>Returns result strings up to a length of 16384 bytes</td>
</tr>
<tr>
<td>dump database</td>
<td>Includes compression syntax for making a compressed dump.</td>
</tr>
<tr>
<td>load database</td>
<td>Includes compression syntax for loading a compressed dump.</td>
</tr>
</tbody>
</table>
New and changed set command options

Table 2-6 summarizes the new and changed options for the set command.

<table>
<thead>
<tr>
<th>Command</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>dump transaction</td>
<td>Includes compression syntax for making a compressed dump</td>
</tr>
<tr>
<td>load transaction</td>
<td>Includes compression syntax for loading a compressed dump</td>
</tr>
<tr>
<td>print statement</td>
<td>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.</td>
</tr>
<tr>
<td>create procedure</td>
<td>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.</td>
</tr>
<tr>
<td>union in view</td>
<td>You are now allowed to use the union command in a view</td>
</tr>
</tbody>
</table>

New Transact-SQL functions

Table 2-7 summarizes the new Transact-SQL functions.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lockscheme()</td>
<td>Returns the locking scheme of the specified object as a string.</td>
</tr>
<tr>
<td>pagesize()</td>
<td>Returns the page size, in bytes, for the specified object</td>
</tr>
</tbody>
</table>

New and changed system procedures

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

New system procedures

Table 2-8 summarizes the new system procedures.
System changes in version 12.5

Table 2-8: New system procedures

<table>
<thead>
<tr>
<th>System Procedure</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>sp_ssladmin</td>
<td>Adds, deletes, or displays a list of server certificates for Adaptive Server</td>
</tr>
<tr>
<td>sp_extengine</td>
<td>Allows you to start or stop EJB Server, or view status information about EJB Server, from the command line.</td>
</tr>
</tbody>
</table>

Changed system procedures

Table 2-9 summarizes the changes made to existing system procedures.

Table 2-9: Changed system procedures

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>sp_addserver</td>
<td>Adds functionality for Adaptive Server's new limits.</td>
</tr>
<tr>
<td>sp_dropserver</td>
<td>Adds functionality for Adaptive Server's new limits.</td>
</tr>
<tr>
<td>sp_havrfy</td>
<td>Adds functionality for Adaptive Server’s new limits.</td>
</tr>
<tr>
<td>sp_helpjava</td>
<td>Supports the new depends keyword, which allows you to view a list of database objects that depend on the specified Java class or JAR.</td>
</tr>
<tr>
<td>sp_configure</td>
<td>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.</td>
</tr>
<tr>
<td>sp_serveroption</td>
<td>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.</td>
</tr>
</tbody>
</table>

New utility programs

Table 2-10 summarizes the new utility programs added to Adaptive Server version 12.5.

Table 2-10: New utility programs

<table>
<thead>
<tr>
<th>Utility</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>ddigen</td>
<td>A Java-based tool that generates definitions for server- and database-level objects in Adaptive Server.</td>
</tr>
<tr>
<td>certpk12</td>
<td>Export or import a PKCS #12 file into a certificates file and a private key</td>
</tr>
</tbody>
</table>
Chapter 2  System Changes in Adaptive Server Version 12.5

Changes to current utility programs

Table 2-11 summarizes the new utility programs added to Adaptive Server version 12.5.

<table>
<thead>
<tr>
<th>Utility</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>certreq</td>
<td>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.</td>
</tr>
<tr>
<td>dataxtr</td>
<td>The GUI data-migration tool to move data and database schema from pre-12.5 Adaptive Server databases into 12.5 databases.</td>
</tr>
<tr>
<td>certauth</td>
<td>Converts a server certificate request to a CA- (certificate authority) signed certificate.</td>
</tr>
<tr>
<td>pwdcrypt</td>
<td>Creates and prints an encrypted LDAP password in the libtcl.cfg.</td>
</tr>
<tr>
<td>sqldbgr</td>
<td>Allows you to debug sql text running on Adaptive Server.</td>
</tr>
</tbody>
</table>

Changes to databases and system tables

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

New databases

There are no new databases for this release of Adaptive Server.

New system tables

There are no new system tables for this release of Adaptive Server.

Changed system tables

Table 2-12 summarizes the system tables that have been changed in this release.
### System changes in version 12.5

#### Table 2-12: Changed system tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>syscolumns</td>
<td>New status2 column. Indicates parameter mode of a SQLJ stored procedure, and the return type of a SQLJ function.</td>
</tr>
<tr>
<td></td>
<td>New accessrule column. The object ID of the access rule in sysprocedures. See “Row-Level Access Locking” for more information.</td>
</tr>
<tr>
<td>sysconstraints</td>
<td>New spare1 column. Not for customer use.</td>
</tr>
<tr>
<td>syslogins</td>
<td>New procid column. Stores the login trigger registered with the login script option in sp_modifylogin.</td>
</tr>
<tr>
<td>sysprocesses</td>
<td>New loggeindatetimetype column. Shows the time and date when the client connected to Adaptive Server. See “Row-Level Access Locking” for more information.</td>
</tr>
<tr>
<td></td>
<td>New ipaddr column. IP address of the client where the login is made. See “Row-Level Access Locking” for more information.</td>
</tr>
<tr>
<td>sysservers</td>
<td>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 set to 1,000 ms.</td>
</tr>
<tr>
<td>systypes</td>
<td>New accessrule column. The object ID of the access rule in sysprocedures. See “Row-Level Access Locking” for more information.</td>
</tr>
</tbody>
</table>

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

#### Changes to sysobjects

Table 2-13 describes changes to the sysobjects table. See “sysobjects” in the system tables chapter of the Adaptive Server Enterprise Reference Manual for more information on the columns used in sysobjects.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Datatype</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>char(2)</td>
<td>F = SQLJ function</td>
</tr>
<tr>
<td>sysstat2</td>
<td>int</td>
<td>Addition of:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• New bit (0x20000000) that represents a SQLJ stored procedure.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• New bit (0x01000000) that represents an access rule.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• New bit (0x04000000) that represents an OR access rule.</td>
</tr>
</tbody>
</table>

#### Changed status in existing columns

Table 2-14 summarizes the changes in existing system table columns in this release:
### Table 2-14: Changed status in existing columns

<table>
<thead>
<tr>
<th>Table name</th>
<th>Column name</th>
<th>Old datatype</th>
<th>New datatype</th>
</tr>
</thead>
<tbody>
<tr>
<td>sysalternates</td>
<td>altsuid</td>
<td>smallint</td>
<td>int</td>
</tr>
<tr>
<td></td>
<td>suid</td>
<td>smallint</td>
<td>int</td>
</tr>
<tr>
<td>syscolumns</td>
<td>colid</td>
<td>tinyint</td>
<td>smallint</td>
</tr>
<tr>
<td></td>
<td>length</td>
<td>tinyint</td>
<td>int</td>
</tr>
<tr>
<td>syscomments</td>
<td>colid</td>
<td>tinyint</td>
<td>smallint</td>
</tr>
<tr>
<td></td>
<td>colid2</td>
<td>tinyint</td>
<td>smallint</td>
</tr>
<tr>
<td>sysconfigures</td>
<td>status</td>
<td>tinyint</td>
<td>int</td>
</tr>
<tr>
<td>sysconstraints</td>
<td>colid</td>
<td>tinyint</td>
<td>int</td>
</tr>
<tr>
<td>syscurconfigs</td>
<td>status</td>
<td>smallint</td>
<td>int</td>
</tr>
<tr>
<td>syddatabases</td>
<td>suid</td>
<td>smallint</td>
<td>int</td>
</tr>
<tr>
<td>sysloginroles</td>
<td>srid</td>
<td>smallint</td>
<td>int</td>
</tr>
<tr>
<td></td>
<td>suid</td>
<td>smallint</td>
<td>int</td>
</tr>
<tr>
<td>syslogins</td>
<td>suid</td>
<td>smallint</td>
<td>int</td>
</tr>
<tr>
<td>syssobjects</td>
<td>uid</td>
<td>smallint</td>
<td>int</td>
</tr>
<tr>
<td>sysprocedures</td>
<td>sequence</td>
<td>smallint</td>
<td>int</td>
</tr>
<tr>
<td>sysprocesses</td>
<td>gid</td>
<td>smallint</td>
<td>int</td>
</tr>
<tr>
<td></td>
<td>suid</td>
<td>smallint</td>
<td>int</td>
</tr>
<tr>
<td></td>
<td>uid</td>
<td>smallint</td>
<td>int</td>
</tr>
<tr>
<td>sysprotects</td>
<td>uid</td>
<td>smallint</td>
<td>int</td>
</tr>
<tr>
<td>sysqueryplans</td>
<td>uid</td>
<td>smallint</td>
<td>int</td>
</tr>
<tr>
<td>sysreferences</td>
<td>fokey1 ... 16</td>
<td>tinyint</td>
<td>smallint</td>
</tr>
<tr>
<td></td>
<td>refkey1 ... 16</td>
<td>tinyint</td>
<td>smallint</td>
</tr>
<tr>
<td>sysremotelogins</td>
<td>suid</td>
<td>smallint</td>
<td>int</td>
</tr>
<tr>
<td>sysroles</td>
<td>id</td>
<td>smallint</td>
<td>int</td>
</tr>
<tr>
<td>syssrvroles</td>
<td>lrid</td>
<td>smallint</td>
<td>int</td>
</tr>
<tr>
<td>syssystems</td>
<td>uid</td>
<td>smallint</td>
<td>int</td>
</tr>
<tr>
<td></td>
<td>length</td>
<td>tinyint</td>
<td>int</td>
</tr>
<tr>
<td>sysusermessages</td>
<td>uid</td>
<td>smallint</td>
<td>int</td>
</tr>
<tr>
<td>syssusers</td>
<td>gid</td>
<td>smallint</td>
<td>int</td>
</tr>
<tr>
<td></td>
<td>uid</td>
<td>smallint</td>
<td>int</td>
</tr>
<tr>
<td></td>
<td>suid</td>
<td>smallint</td>
<td>int</td>
</tr>
</tbody>
</table>
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 release 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. Reserved words cannot be used as object names or column names.

New global variables

Table 2-15 summarizes the global variables for the server limits of logins, users, and groups.

<table>
<thead>
<tr>
<th>Name of variable</th>
<th>What it displays</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>@@minuserid</td>
<td>Lowest user ID</td>
<td>-32768</td>
</tr>
<tr>
<td>@@maxuserid</td>
<td>Highest user ID</td>
<td>2147483647</td>
</tr>
<tr>
<td>@@invaliduserid</td>
<td>Invalid user ID</td>
<td>-1</td>
</tr>
<tr>
<td>@@invaliduserid</td>
<td>Invalid server user ID</td>
<td>-2</td>
</tr>
<tr>
<td>@@guestuserid</td>
<td>Guest user ID</td>
<td>2</td>
</tr>
<tr>
<td>@@maxsuid</td>
<td>Highest server user ID</td>
<td>2147483647</td>
</tr>
</tbody>
</table>
Changes to the documentation

The following changes were made to the Adaptive Server documentation set:

- *Managing and Monitoring Adaptive Server Enterprise* has been discontinued.
- The *Utility Guide* is no longer in two volumes, one for UNIX and one for Windows NT. It is now a generic book that includes all utilities, regardless of platform.

Changes that may effect existing applications

This section describes system changes introduced by Adaptive Server version 12.5 that may affect your applications if you are upgrading from a previous release. Topic covered are:

- “New Transact SQL keywords” on page 25
- “Wide column and data truncation” on page 26
- “Changes that effect Client Server applications” on page 26
- “Maximum number of expressions in a select statement” on page 28

New Transact SQL keywords

See “New reserved words” on page 24 for information about the Adaptive Server version 12.5 additions to the reserved word list.
Wide column and data truncation

Previous releases of Adaptive Server had a column-length limit of 255 bytes. The 12.5 release of Adaptive Server allows you to create columns using char, varchar, binary, and varbinary that are up to 16294 bytes long, depending on the logical page size your server uses. Because of this, data that was truncated at 255 character for earlier releases of Adaptive Server are 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. Previous releases of Adaptive Server truncate this to 255 characters, and the sum of col1 and col2 might match col3. However, for the 12.5 release of Adaptive Server, 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 over this length is truncated.

Changes that effect Client Server 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 features that a client/server connection supports. In particular, 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 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.

- If you are running a pre-version 12.5 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-version 12.5 functionality is fine, but the new limits are not enabled. Adaptive Server sends the data according to the pre-12.5 limits, and will truncate any data that is beyond these limits.

- If you are running pre-version 12.5 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.

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

*Figure 2-1: 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 a 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 will truncate the data to 255 bytes and issue 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)
Changes that may effect existing applications

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.

In addition to improving overall manageability and performance, Adaptive Server version 12 introduces several new licensed features for specialized use. These include:

- 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

The new features and system changes in Adaptive Server version 12 are:

<table>
<thead>
<tr>
<th>Category</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability and Manageability</td>
<td>“Sybase’s Failover for a high availability system” on page 30</td>
</tr>
<tr>
<td></td>
<td>“Rebuilding indexes” on page 32</td>
</tr>
<tr>
<td></td>
<td>“High performance backup and restore” on page 33</td>
</tr>
<tr>
<td></td>
<td>“Non-rewinding tapes” on page 33</td>
</tr>
<tr>
<td></td>
<td>“UNIX file system support” on page 33</td>
</tr>
<tr>
<td></td>
<td>“Modifying an existing table’s schema with alter table” on page 34</td>
</tr>
<tr>
<td></td>
<td>“Suspending database updates with quiesce database” on page 35</td>
</tr>
</tbody>
</table>
Sybase’s Failover for a high availability system

A high availability cluster includes two machines that are configured so that, 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’s Failover product enables Adaptive Server to work in a high availability cluster in an active-active configuration. That is, both the node 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’s 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 a system failure or a scheduled maintenance. The secondary companion is an independent Adaptive Server, and can have its own applications running. Figure 3-1 describes an asymmetric setup:

**Figure 3-1: 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 3-2 describes a symmetric setup:
Rebuilding indexes

In Adaptive Server 12, you can use the `reorg rebuild` command to rebuild individual indexes while the table itself is accessible for read and update activities. For details, 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

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

---

*Figure 3-2: Symmetric configuration in a high availability system*
High performance backup and restore

When you start Backup Server with the \texttt{backupserver} utility command, you can use the \texttt{-m} parameter to improve the performance of the dump and load commands by configuring more shared memory for the Backup Server. The \texttt{-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.

\textbf{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 \texttt{-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 \textit{System Administration Guide}.

Non-rewinding tapes

In previous versions of Adaptive Server, Backup Server rewound the tape before each dump operation. In version 12, 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 \textit{System Administration Guide}.

UNIX file system support

With previous versions of Adaptive Server, Sybase recommended not placing 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 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 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:

```sql
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 changes the datatype of the `type` column in the `titles` table from `char(12)` to `varchar(20)` and makes it nullable:

```sql
alter table titles
    modify type varchar(20) null
```
Suspending database updates with *quiesce database*

Adaptive Server version 12 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`, `clientapplname` columns of the `sysprocesses` table.

For more information, see the “Managing Adaptive Server Logins and Database Users” chapter of the *System Administration Guide*. 
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 the *System Administration Guide* and the command and 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 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 stand-alone 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 data types 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 previous versions, a query and any views used in a query could reference at most 16 user tables, and the 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 introduces several query processing enhancements.

**Merge joins**

Earlier versions of Adaptive Server perform all join queries using nested-loop joins. Version 12 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

Previous versions of Adaptive Server only provided syntax for a Transact-SQL join, which included the =* and *= symbols for specifying an right or a left join, respectively. Adaptive Server 12 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:

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

  For example:

  ```sql
  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:

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

  For example:

  ```sql
  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 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 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 previous 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 new `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 information about using abstract plans, 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 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 re-enable 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 it accesses the cache. With a large number of engines and a high transaction rate, contention for buffer cache spinlocks can slow performance. Adaptive Server version 12 allows you to divide each cache into partitions, each with its own spinlock. See “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 introduces several distributed transaction management features to:

- 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 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 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 – A 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
- RPC’s 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.

Releases of Adaptive Server earlier than 12 could obtain network service information only by reading the interfaces file in the Sybase release directory. Adaptive Server 12 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 booted. 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

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

- **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 (UDLS) gives you more control over security-related features of Adaptive Server. In Adaptive Server 12, 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 was 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
User-defined login security

- 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

You use the `sp_displaylogin` system procedure 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 previous releases, 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 procedure 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 releases prior to 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

This section provides a summary of the version 12 changes, as follows:

- “Changes to configuration parameters” on page 50
- “Changes to Transact-SQL commands” on page 52
- “New and changed system procedures” on page 54
- “Changes to Databases and System Tables” on page 56
- “New reserved words” on page 58
- “Changes to the documentation” on page 58

Changes to configuration parameters

This section describes the new and changed configuration parameters in version 12.

New configuration parameters

Table 3-1 summarizes the new configuration parameters:
### Table 3-1: New configuration parameters

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

### Changes to configuration parameters

Table 3-2 summarizes the changes to existing configuration parameters.
System changes in version 12

### Table 3-2: Changed configuration parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>enable cis</td>
<td>The default value has been changed from 0 (off) to 1 (on).</td>
</tr>
</tbody>
</table>

### Changes to Transact-SQL commands

This section describes the new and changed Transact-SQL commands in version 12.

### New Transact-SQL commands

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

#### Table 3-3: New commands

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

### Changed Transact-SQL commands

Table 3-4 summarizes the changes to existing Transact-SQL commands.

#### Table 3-4: Changed commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>alter role</td>
<td>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.</td>
</tr>
<tr>
<td>alter table</td>
<td>Includes new off row and in row options to specify storage for Java-SQL columns.</td>
</tr>
<tr>
<td>create role</td>
<td>Includes new lock, unlock, and all overrides options to set the lock status of a role. Includes new passwd expiration, min passwd length, 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.</td>
</tr>
<tr>
<td>create table</td>
<td>Includes new off row and in row options to specify storage for Java-SQL columns.</td>
</tr>
</tbody>
</table>
New and changed set command options

The following table summarizes the new and changed options for the `set` command:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbcc</td>
<td>Includes new commands for heuristically completing distributed transactions.</td>
</tr>
<tr>
<td></td>
<td>Includes new <code>rebuild_text</code> 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.</td>
</tr>
<tr>
<td>disk init</td>
<td>Includes the new <code>dsync</code> option to control buffered writes to devices.</td>
</tr>
<tr>
<td>disk reinit</td>
<td>Includes the new <code>dsync</code> option to control buffered writes to devices.</td>
</tr>
<tr>
<td>execute</td>
<td>Includes the new <code>execute()</code> syntax to dynamically execute Transact-SQL commands.</td>
</tr>
</tbody>
</table>

New Transact-SQL functions

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

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>license_enabled</td>
<td>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.</td>
</tr>
</tbody>
</table>
System changes in version 12

New and changed system procedures

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

New system procedures

Table 3-7 summarizes the new system procedures:

<table>
<thead>
<tr>
<th>System Procedure</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>sp_deviceattr</td>
<td>Changes the dsync setting of an existing database device file.</td>
</tr>
<tr>
<td>sp_transactions</td>
<td>Reports detailed information about active transactions.</td>
</tr>
<tr>
<td>sp_companion</td>
<td>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</td>
</tr>
<tr>
<td>sp_ha_admin</td>
<td>Performs administrative tasks that address second point of failures on Adaptive Servers configured with Sybase’s Failover in a high availability system.</td>
</tr>
<tr>
<td>sp_helpjava</td>
<td>Displays information about Java classes and JARs installed in the database.</td>
</tr>
</tbody>
</table>

Changed system procedures

Table 3-8 summarizes the changes made to existing system procedures.
Table 3-8: Changed system procedures

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>sp_addlogin</strong></td>
<td>Specifies the password expiration interval, the minimum password length, and the maximum number of failed logins allowed for a specified login at creation.</td>
</tr>
</tbody>
</table>
| **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:  
  • 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 3-9 describes the new utility programs added to Adaptive Server version 12.
Table 3-9: New utility programs

<table>
<thead>
<tr>
<th>System Procedure</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>extractjava (UNIX)</td>
<td>Copies a retained JAR and the Java classes it contains from Adaptive Server to a client file.</td>
</tr>
<tr>
<td>extrjava (Windows NT)</td>
<td></td>
</tr>
<tr>
<td>installjava (UNIX)</td>
<td>Installs a JAR and the Java classes it contains from a client file to Adaptive Server.</td>
</tr>
<tr>
<td>instjava (Windows NT)</td>
<td></td>
</tr>
</tbody>
</table>

Changes to Databases and System Tables

This section describes the changes made to databases and system tables to support version 12 features.

New Databases

Table 3-10 lists the databases that are new in this release:

Table 3-10: New databases

<table>
<thead>
<tr>
<th>Database</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>sybdiagdb</td>
<td>Holds diagnostic configuration data for debugging purposes. It is for Sybase Technical Support personnel only and should not be used by customers.</td>
</tr>
</tbody>
</table>

New system tables

Table 3-11 lists the system tables that are new in this release:

Table 3-11: New system tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>syscolumns</td>
<td>Adds the xtype and xdbid columns. These columns are used only for Adaptive Servers enabled for Java.</td>
</tr>
<tr>
<td>syscoordinations</td>
<td>(sybsystemdb database only.) Contains information about remote Adaptive Servers participating in distributed transactions (remote participants) and their coordination states.</td>
</tr>
<tr>
<td>sysdevices</td>
<td>status column utilizes a new control bit 0x4000 (16384 Decimal) to control the dsync setting for a database device file.</td>
</tr>
<tr>
<td>sysjars</td>
<td>Contains one row for each Java archive file retained in the database. This table is only used for Adaptive Servers enabled for Java.</td>
</tr>
<tr>
<td>syslocks</td>
<td>Adds the loid column.</td>
</tr>
<tr>
<td>sysprocesses</td>
<td>Adds the block_xloid column. The blocked column is renamed to block_xloid.</td>
</tr>
<tr>
<td>sysqueryplans</td>
<td>Stores query text and abstract plans for a query.</td>
</tr>
</tbody>
</table>
CHAPTER 3  New Features and System Changes in Adaptive Server Version 12.0

Table 3-12 lists the system tables that have been changed in this release:

**Table 3-12: Changed system tables**

<table>
<thead>
<tr>
<th>Table</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>sysattributes</td>
<td>Stores information for the configuration parameters that set or change the password expiration interval, maximum failed login attempts, and the minimum password length.</td>
</tr>
<tr>
<td>sysconfigures</td>
<td>Stores the server-wide values set with the configuration parameters minimum password length and maximum failed logins for logins and roles.</td>
</tr>
<tr>
<td>syslogins</td>
<td>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.</td>
</tr>
<tr>
<td>syservroles</td>
<td>The following columns have been added:</td>
</tr>
<tr>
<td></td>
<td>- <strong>pwdate</strong> – records the date and time when the password of a role was last changed. The datatype of this column is datetime.</td>
</tr>
<tr>
<td></td>
<td>- <strong>status</strong> – 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.</td>
</tr>
<tr>
<td></td>
<td>The values are:</td>
</tr>
<tr>
<td></td>
<td>0 – role is not expired or locked</td>
</tr>
<tr>
<td></td>
<td>2 – role is not expired but is locked</td>
</tr>
<tr>
<td></td>
<td>4 – role is expired but is not locked</td>
</tr>
<tr>
<td></td>
<td>6 – role is expired and locked</td>
</tr>
<tr>
<td>sysusers</td>
<td>In previous releases, the <strong>sysusers</strong> system table in each database had a unique clustered index on <strong>suid</strong>. However, in Adaptive Server 12.0 the uniqueness of <strong>sysusers.suid</strong> does not exist, so the clustered index has been dropped.</td>
</tr>
</tbody>
</table>

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

---

**Changed System Tables**

Table 3-12 lists the system tables that have been changed in this release:

**Table 3-12: Changed system tables**

<table>
<thead>
<tr>
<th>Table</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>syssessions</td>
<td>Contains one row for each client that connects to Adaptive Server with the failover property (for example, <em>isql -Q</em>). This table is only used for Adaptive Servers that are configured for Sybase’s Failover in a high availability system.</td>
</tr>
<tr>
<td>systransactions</td>
<td>(master database only.) Contains information about Adaptive Server transactions</td>
</tr>
<tr>
<td>sysxtypes</td>
<td>Contains one row for each extended, Java-SQL datatype. This table is only used for Adaptive Servers enabled for Java.</td>
</tr>
</tbody>
</table>
System changes in version 12

New reserved words

The *Adaptive Server Reference Manual* lists all of the reserved words for version 12. 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 release of the server. You can change table, view, and column names or use delimited identifiers. Once you upgrade to version 12, 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 includes several new books that describe separately-licensed Adaptive Server features such as Java, High Availability, and Distributed Transaction Management. See “Related documents” on page x for a brief description of these books.

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 that may affect your applications if you are upgrading from an earlier release.

New Transact-SQL Keywords for 12

See “New reserved words” on page 58.

$dsync$ On By Default for UNIX File Devices

Beginning with version 12, 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.
New Features in Adaptive Server Version 11.9.2

This chapter describes the new functionality in version 11.9.2. Topics include:

- “New locking schemes” on page 61
- “Changes to table-level locking” on page 63
- “Changes to statistics and query optimization” on page 63
- “Enhancements to the create index command” on page 64
- “Changes and additions to Transact-SQL syntax” on page 64
- “Configurable database recovery order” on page 66
- “Fault verification for dbcc checkstorage faults” on page 66
- “License use monitor” on page 66
- “Task-to-engine affinity” on page 67
- “Dynamic SQL performance improvements” on page 67
- “Direct updates through joins” on page 67
- “Component Integration Services changes” on page 68
- “Character set changes” on page 68
- “Changes that may affect existing applications” on page 68

New locking schemes

Adaptive Server version 11.9.2 provides two new locking schemes to improve the concurrency and performance of Adaptive Server:

- Datapages locking
- Datarows locking, also known as row-level locking
Together, these new locking schemes are referred to as data-only locking.

The pre-11.9.2 locking scheme continues to be supported; it is called **allpages locking**, and it is the default locking scheme when you first install or upgrade to version 11.9.2. A System Administrator can specify any locking scheme as the server-wide default.

Users can specify a locking scheme for a newly created table, using the `create table` command, and can change the locking scheme for an existing table to any other locking scheme using the `alter table` command.

Some of the changes made to support these new locking schemes include:

- Additional types of locks
- Changes to table and index structures for tables using the new locking schemes
- New configuration parameters and changes to existing configuration parameters
- Additions and changes to Transact-SQL command syntax, including:
  - Additions to `create table` and `alter table` to allow specifying the locking scheme
  - A new `reorg` command to manage space in tables that use the new locking schemes
  - Changes to `select into` syntax to allow specifying the locking scheme on the created table
- Changes to system procedures to allow reporting on and configuring of new functionality

**Note** The number of locks available to all processes on the server is limited by the configuration parameter `number of locks`. Changing to data-only locking affects the number of locks required during query processing.
Changes to table-level locking

Performing an update or delete via a table scan does not acquire a table-level exclusive lock as it did in earlier versions. In version 11.9.2, the scan acquires an exclusive intent table lock when the transaction starts. When the scan locates a row that must be updated, it acquires an exclusive page or row lock, depending on the locking scheme. The only exception is an unindexed update or delete to a data-only-locked table at transaction isolation level 3; these updates acquire an exclusive table lock. In previous versions, any update or delete that does not use an index acquires a table-level exclusive lock at the start of the transaction.

In most cases, performing the scan without using a table level lock increases concurrency. However, some applications may experience one of these problems:

- Applications that did not deadlock in earlier versions may experience deadlocks in 11.9.2.
- Tasks that update a large number of rows may use a large number of locks if they cannot perform lock promotion due to conflicting locks.

Workarounds:

- If deadlocks are creating problems, consider converting the table to use datarows locking.
- If deadlocks or excessive numbers of locks are creating problems, consider using the lock table command to acquire an exclusive table lock before you begin the transaction.

Changes to statistics and query optimization

Adaptive Server version 11.9.2 increases the flexibility of the statistics used by the query optimizer. Optimizer statistics are now kept in two system tables, systabstats and sysstatistics. A new utility, optdiag, can extract statistics and allows editing of statistics. optdiag makes many of these statistics viewable for the first time.

Statistics are now kept on a per-column basis, rather than a per-index basis. Query costing in the optimizer has been enhanced to use column-level statistics for search arguments and joins, even if indexes do not exist on the column.
Enhancements to the create index command

The update statistics command now supports storing statistics for unindexed columns. New update index statistics syntax facilitates creating statistics on all columns used in an index, and update all statistics now generates statistics for all columns in a table. A new command, delete statistics, can be used to drop column-level statistics, since drop index no longer removes statistics.

New statistics are kept to support the new locking schemes. Additional statistics track the clustering of data rows and data pages in physical storage and improve the cost estimates made by the optimizer. Some queries on tables using new locking schemes are optimized differently than queries on tables using the old locking scheme.

Enhancements to the create index command

There are two enhancements to create index:

- The create index command allows the user to specify ascending or descending order for each column in the index. In earlier versions, all indexes were created in ascending order for all columns. Scans that needed to read the data in reverse order could scan the index backward, but if the required order was a mix of ascending and descending order on the keys, the query needed to perform a sort step. Performance can be improved by matching the index ordering to the ordering used by most queries.

- Indexes can include as many as 31 key columns, an increase from 16 in previous versions. The maximum total number of bytes allowed for index keys is 600.

Changes and additions to Transact-SQL syntax

Adaptive Server version 11.9.2 provides new commands and options that affect locking:

- readpast option
- lock table command
- Server-level and session level options that specify how long tasks wait for locks
• Support for repeatable reads transaction isolation level (level 2) for data-only-locked tables

**New readpast concurrency option**

select, delete, update, readtext, and writetext commands now include the readpast option. This option allows the command to skip all pages or rows that are not immediately accessible due to incompatible locks.

**New lock table command**

A new command, `lock table`, explicitly locks entire tables in either shared or exclusive mode.

**Specifying a wait time for locks**

In earlier versions of Adaptive Server and SQL Server, all commands that cannot immediately acquire a lock block, or wait, until the required lock is available. In version 11.9.2, the maximum length of time that a task waits can be specified on a server-wide or session-level basis. Processes that fail to acquire the lock within the specified waiting period fail with an error message.

• For information on the new configuration parameter, `lock wait period`, see the *System Administration Guide*.
• For information on the new `set` option, `set lock wait`, see the *Adaptive Server Reference Manual*. 
Repeatable read transaction isolation

Version 11.9.2 adds support for repeatable read transactions (isolation level 2).

Configurable database recovery order

You can now specify the order in which user databases are recovered when you boot the server. This allows your most critical databases to be recovered first.

Fault verification for dbcc checkstorage faults

The dbcc checkstorage command, introduced in Adaptive Server version 11.5, checks the consistency of database objects without blocking access by other tasks. Since user activity can produce soft faults, dbcc checkstorage attempts to resolve the fault by a second check; if the fault persists, it is recorded in the dbccdb database. Since heavy database use during dbcc checkstorage operations can lead to lengthy fault reports, the dbcc checkverify command introduced in version 11.9.2 rechecks these faults, while holding a lock on the object, in order to eliminate the soft faults from the list of faults.

License use monitor

System Administrators can determine whether the usage on their Adaptive Server system exceeds license agreements by using the license monitor feature. Once a day, it records the maximum number of users in a system table in the master database.
**Task-to-engine affinity**

For servers running multiple engines, task-to-engine affinity improves performance by reducing the number of times the task context has to be switched between engines. The only visible effects of this are during system tuning. When there are fewer tasks running than there are engines, you may see some engines showing high utilization, while others are idle for the entire interval. For example, in earlier versions, while a single task runs a `select *` query, you would generally see each of 6 engines performing about 16% of the work. In version 11.9.2, `sp_sysmon` might report:

<table>
<thead>
<tr>
<th>Engine Busy Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Engine 0</td>
</tr>
<tr>
<td>Engine 1</td>
</tr>
<tr>
<td>Engine 2</td>
</tr>
<tr>
<td>Engine 3</td>
</tr>
<tr>
<td>Engine 4</td>
</tr>
<tr>
<td>Engine 5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Summary</th>
<th>Total</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>97.2 %</td>
<td>16.2 %</td>
</tr>
</tbody>
</table>

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

**Dynamic SQL performance improvements**

In pre-11.9.2 releases, the simultaneous use of Dynamic SQL by a large number of users performs poorly, largely due to contention on system tables in tempdb. In version 11.9.2, the contention problems have been eliminated, and the performance of the prepare and deallocate statements has been significantly improved. There are no user interface changes. All changes are internal, so applications do not need to be modified.

**Direct updates through joins**

In version 11.9.2, many restrictions on when direct updates can be performed have been removed.
Component Integration Services changes

Component Integration Services for Adaptive Server version 11.9.2 adds the following enhancements for remote table access:

- Enhances the performance of queries that include remote tables
- Simplifies the mapping of local proxy tables to remote tables
- Adds new server classes for Adaptive Server Enterprise, Adaptive Server Anywhere™, and Adaptive Server IQ™
- Adds a new trace flag that, when enabled, makes any query that is part of a declare cursor command, and that references proxy tables, read-only by default

Character set changes

Adaptive Server version 11.9.2 includes support for the European currency symbol, or “Euro”. Adaptive Server can also perform character set conversions that cause a change in the data length.

Changes that may affect existing applications

Changes in 11.9.2 that may affect user applications and system administration are:

- Changes to statistics and query optimization can cause changes to query plans, even for tables where the locking scheme is not changed.
- More logging is performed than in previous versions.
- For standby or report-processing servers, new syntax is required to dump the transaction log for the production server and bring the database online. It is no longer possible to load a dump that contains open transactions, bring the database online, and then load a subsequent dump of the log.
• The update statistics command may run more slowly, because it now performs a table scan as well as scanning indexes. Concurrency is improved, because update statistics with a table name or index name operates at transaction isolation level 0 on data-only-locked tables, and does not acquire table locks on allpages-locked tables.

• The update all statistics command now generates statistics for all columns in a table. It performs a complete table scan for each column that is not the leading column in an index. If your system runs scripts that include update all statistics commands, you may want to replace update all statistics commands with update statistics and update partition statistics commands.

Effects of changing to data-only-locking

Changing tables to data-only locking may have these effects:

• The number of locks required for your applications may increase, and you may need to change the number of locks configuration parameter.

• On data-only-locked tables with clustered indexes, the default behavior of disallowing duplicate rows and the ignore_dup_row option are not enforced during inserts and updates. This changes the behavior of commands that insert duplicate rows and may change the behavior of create clustered index if the table contains duplicate rows.

• On data-only-locked tables, the sorted_data option to create index can be used only immediately following a bulk copy operation that copies into an existing table. Use of the sorted_data option is prohibited once additional page allocation operations have been made for the table.

• Bulk copy into data-only-locked tables requires the version of bcp and the bulk copy libraries shipped with Adaptive Server version 11.9.2. Older versions of bcp and the bulk-copy libraries can still be used to copy into allpages-locked tables. See the Release Bulletin for the required version number.

• When using parallel sort for data-only-locked tables, the number of worker processes must be configured to equal or exceed the number of partitions, even for empty tables. The database option select into/bulkcopy/pllsort must also be enabled.
Changes that may affect existing applications

Query optimization changes and forced query plans

Changes in query optimization provide greater accuracy than in previous versions in costing some queries, especially those performing nonclustered index scans. In version 11.9.2, you can create statistics on unindexed columns, which can further improve query optimization. You should review any queries that are using forced query plans, especially index or join order forcing, to determine whether forcing is still needed or whether the optimizer can provide an improved plan.

Index forcing and data-only-locked tables

The choice of index to use for a query can be forced with index clause and an index name, as in this command:

```sql
select title_id, type, price
from titles (index title_id_ix)
```

Adaptive Server accepts an index ID in place of the keyword index and the index name. The following statement forces the use of a clustered index on an allpages-locked table:

```sql
select title_id, type, price
from titles (1)
```

When you convert a table with a clustered index to data-only locking, the index ID of the clustered index changes. If you execute a query that specifies index ID 1, the optimizer still uses the clustered index on the table. You should carefully check all query plans that force index ID 1.

**Note** Sybase strongly recommends the use of index names, rather than index IDs, for forcing index selection.

Performance after loading pre-11.9 databases

When pre-11.9 databases are loaded into an 11.9.2 Adaptive Server, the upgrade process inserts rows into the statistics tables, based on available statistics. The upgrade process does not scan the tables and indexes to generate the new types of statistics that can be used by the improved query optimizer. For statistics that are new in this version, the upgrade process inserts computed or magic values.
If you load a pre-11.9 database, run `update statistics` on the tables as soon as possible. If you experience performance problems after loading a pre-11.9 database, run `update statistics` for the tables in the application before you attempt other solutions.

**Ordering of results with data-only locked tables**

Clustered indexes on data-only-locked tables may not return rows in clustered key order if there is no `order by` clause. Bulk copy is not guaranteed to copy out a table in clustered key order.

Queries on unpartitioned allpages-locked tables with clustered indexes return rows in the clustered key order if the query does not include a sort (that is, if there are no clauses such as `order by` or `distinct`). Bulk copy also copies the rows out of allpages-locked tables in clustered key order for both partitioned and unpartitioned tables.

**Delay of cursor compilation until cursor open**

In pre-11.9.2 versions of SQL Server and Adaptive Server, a cursor is optimized and compiled when it is declared. For cursors with local variables or parameters, this can result in a suboptimal query plan, since default values are used to estimate the number of rows to be returned. The problem of poor plans is most common with client cursors, sent using Open Client™ Client-Library™ or Embedded SQL™.

In version 11.9.2, query optimization is delayed until the cursor is opened. The values of the variables are known at cursor open time, and query optimization may be considerably improved.

**Visible effects of delayed cursor optimization**

Besides improved query plans, the only user-visible changes are as follows:

- If you are using `showplan`, the plan is no longer displayed at the declare cursor statement, but at the open statement.
- The query is normalized at the declare cursor statement, so error messages about syntax errors, missing tables, and so forth, are reported at that time.
Changes that may affect existing applications

- Some error messages do not appear until the query is compiled and optimized, including those for cursors that require unique indexes. If error messages result at cursor open time (such as the lack of a needed index), the cursor must be deallocated and redeclared. You cannot just correct the problem and reopen the cursor.

- In pre-11.9.2 versions, changing isolation levels between cursor declare time and cursor open time generated an error message when the cursor was opened. In version 11.9.2, the following sequence of commands does not generate an error, and the cursor is compiled at transaction isolation level 0:

```
*start at isolation level 1 */
declare curs1 cursor for
select t.title_id, au_lname
from titles t , authors a, titleauthor ta
where t.title_id = ta.title_id
and a.au_id = ta.au_id
go
set transaction isolation level 0
go
open curs1
go
fetch curs1
go
```

**Note** Delayed optimization does not apply to cursors that are declared in a stored procedure (called server cursors). Server cursors are compiled and optimized when they are declared.
This chapter describes the new features and system changes introduced in Sybase Adaptive Server Enterprise release 11.5.

In addition to providing many new server features, Adaptive Server 11.5 provides a more comprehensive product by incorporating previous products and tools:

- SQL Server Manager™ has been replaced by Sybase Central™
  
  For more information, see “Sybase Central” on page 103.

- OmniConnect™ has been replaced by Component Integration Services
  
  Many of the OmniConnect release 10.5 features have been incorporated into the Component Integration Services (CIS) feature of Adaptive Server 11.5. For more information on CIS, see “Component Integration Services” on page 78.

### New features in release 11.5

The new features in Adaptive Server release 11.5 are:

<table>
<thead>
<tr>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Asynchronous prefetch” on page 74</td>
</tr>
<tr>
<td>“Auditing enhancements” on page 75</td>
</tr>
<tr>
<td>“case expression” on page 77</td>
</tr>
<tr>
<td>“Component Integration Services” on page 78</td>
</tr>
<tr>
<td>“create index enhancements” on page 82</td>
</tr>
<tr>
<td>“dbcc enhancements” on page 82</td>
</tr>
<tr>
<td>“Descending index scan optimization” on page 85</td>
</tr>
<tr>
<td>“Directory services” on page 86</td>
</tr>
<tr>
<td>“Engine affinity and execution precedence” on page 86</td>
</tr>
</tbody>
</table>
Asynchronous prefetch

Asynchronous prefetch improves performance by anticipating the pages that will be required for certain well-defined classes of database activities whose access patterns are predictable. The I/O requests for these pages are issued before the query needs them so that most pages are in cache by the time query processing needs to access the page.

Asynchronous prefetch can improve performance for:

- Sequential scans, such as table scans, clustered index scans, and covered nonclustered index scans
- The update statistics command
- Accesses via nonclustered indexes
Some dbcc checks
Recovery

Asynchronous prefetch can improve the performance of queries that access large numbers of pages, such as decision support system (DSS) applications, as long as the I/O subsystems on the machine are not saturated.

Asynchronous prefetch cannot help (or may help only slightly) when the I/O subsystem is already saturated or when Adaptive Server is CPU-bound. It can be used in some online transaction processing (OLTP) applications, but to a much lesser degree, since the queries in OLTP applications usually perform fewer I/O operations.

**Note** The default setting for global async prefetch limit is 10, meaning that asynchronous prefetch is enabled when you install or upgrade to release 11.5, and that all pools can use up to 10 percent of the buffers for asynchronous reads.

Table 5-1 shows where to find additional information on asynchronous prefetch.

<table>
<thead>
<tr>
<th>For information On</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>How queries can benefit from asynchronous prefetch</td>
<td>Chapter 18, “Tuning Asynchronous Prefetch,” in the Performance and Tuning Guide</td>
</tr>
<tr>
<td>Configuring pool limits</td>
<td>Chapter 9, “Configuring Data Caches,” in the System Administration Guide</td>
</tr>
<tr>
<td>Configuring server-wide limits</td>
<td>Chapter 11, “Setting Configuration Parameters,” in the System Administration Guide</td>
</tr>
<tr>
<td>Tuning asynchronous prefetch limits</td>
<td>Chapter 24, “Monitoring Performance with sp_sysmon,” in the Performance and Tuning Guide</td>
</tr>
</tbody>
</table>

**Auditing enhancements**

The auditing system provides the following enhancements:

- Support of multiple audit tables. This capability allows you to archive and process audit data with no manual intervention and no loss of audit records.
Auditing enhancements

- A new, streamlined, user interface for the System Security Officer to control auditing. In the new interface, which replaces the current auditing interface, a single system procedure, `sp_audit`, provides the capability to set all auditing options.

- The addition of events that can be audited, including:
  - Creation of objects
  - Binding and unbinding of rules, defaults, and messages
  - All actions by individual users
  - All actions performed with a particular role active
  - A set of server-wide, security-relevant events

New system procedures

Use the system procedures listed in Table 5-2 to manage the auditing process.

<table>
<thead>
<tr>
<th>System Procedure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>sp_addauditrecord</code></td>
<td>Adds user-defined audit records (comments) into the audit trail. Users can add these records only if a System Security Officer enables ad hoc auditing with <code>sp_audit</code>.</td>
</tr>
<tr>
<td><code>sp_audit</code></td>
<td>Enables and disables all auditing options. This is the only system procedure required to establish the events to be audited.</td>
</tr>
<tr>
<td><code>sp_configure</code></td>
<td>Enables or disables auditing for Adaptive Server. The parameter takes effect immediately upon execution of <code>sp_configure</code>. Auditing occurs only when this parameter is enabled.</td>
</tr>
<tr>
<td>&quot;auditing&quot;</td>
<td>Establishes the size of the audit queue. Because this parameter affects memory allocation, it does not take effect until Adaptive Server is restarted.</td>
</tr>
<tr>
<td>&quot;current audit table&quot;</td>
<td>Sets the current audit table. This takes effect immediately upon execution of <code>sp_configure</code>.</td>
</tr>
<tr>
<td>&quot;suspend auditing when full&quot;</td>
<td>Controls the behavior of the audit process when an audit device becomes full. This takes effect immediately upon execution of <code>sp_configure</code>.</td>
</tr>
<tr>
<td><code>sp_displayaudit</code></td>
<td>Displays the active auditing options.</td>
</tr>
</tbody>
</table>

**Note** The system procedures `sp_auditoption`, `sp_auditdatabase`, `sp_auditobject`, `sp auditsproc`, and `sp auditlogin` are obsolete. Use `sp_audit` for setting the options that you set with those system procedures in pre-11.5 releases.
Table 5-3 shows the system procedures that were used to set auditing options in pre-11.5 releases and the types of auditing options that replace them in Adaptive Server release 11.5. The auditing options in the groups mentioned in the second column of Table 5-3 are described in the discussion of sp_audit in the Adaptive Server Reference Manual.

Table 5-3: Comparison of pre-11.5 and 11.5 options

<table>
<thead>
<tr>
<th>Option set in pre-11.5 releases</th>
<th>Option to set in 11.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>sp_auditoption</td>
<td>“global” auditing options</td>
</tr>
<tr>
<td>sp_auditdatabase</td>
<td>“database-specific” auditing options</td>
</tr>
<tr>
<td>sp_auditobject</td>
<td>“object-specific” auditing options</td>
</tr>
<tr>
<td>sp_auditsproc</td>
<td>“exec” auditing options</td>
</tr>
<tr>
<td>sp_auditlogin</td>
<td>“user-specific” auditing options</td>
</tr>
</tbody>
</table>

For example, in pre-11.5 releases, you used the following command to audit the actions of deleting, updating and inserting information in the authors table of the pubs2 database:

```
sp_auditobject "authors", "pubs2", "both", "dui"
```

In release 11.5, these actions are covered by the delete, update, and insert options in the “object-specific” group. You use the following commands to achieve the same results as you did using the preceding, obsolete commands:

```
sp_audit "delete", "all", "authors", "on"
sp_audit "update", "all", "authors", "on"
sp_audit "insert", "all", "authors", "on"
```

To install auditing, see the Adaptive Server configuration guide for your platform. For more information about auditing, see Chapter 8, “Auditing,” in the Security Administration Guide.

case expression

Using case expression simplifies standard SQL expressions by allowing a when...then construct instead of an if...else construct. A case expression is more concise than an if...else statement and Adaptive Server’s response time is faster, since the optimizer processes fewer lines of SQL code. Also, queries written using case expressions are often easier to read and understand than those written in standard SQL. Because of the concise nature of case expressions, they are allowed anywhere value expressions are used.
A case expression is written using the following syntax:

```
case
  [when search_condition1 then result1]
  [when search_conditionn then resultn]
end
```

For more information about the syntax of case expressions, see the *Adaptive Server Reference Manual*.

For more information and examples of case expressions, see the Transact-SQL User’s Guide.

---

**Component Integration Services**

The Component Integration Services (CIS) feature allows you to connect to remote Sybase and non-Sybase databases. It presents a uniform view of enterprise data to client applications and provides location transparency to the data sources that you access.

When you access both Sybase and non-Sybase databases on different servers, you can use the host data files, tables, triggers, views, remote procedure calls, and most commands in database systems such as Adaptive Server, Informix, DB2, and Oracle.

Component Integration Services can be used by anyone who needs to access multiple data sources or legacy data. It can also be used by anyone who needs to migrate data from one server to another.

A single server is often used to access data on multiple external servers. Component Integration Services manages the data regardless of the location of the external servers. Data management is transparent to the client application.
Using Component Integration Services, you can:

- Access tables in remote servers as if the tables were local.
- Perform joins between tables in multiple, remote, and heterogeneous servers. For example, you can join tables between Adaptive Server and an Oracle DBMS, between Adaptive Server and Record Management System (RMS) files in OpenVMS, and between tables in multiple Adaptive Servers.
- Use the `select into` command to transfer the contents of one table into a new table on any supported remote server.
- Provide applications such as PowerBuilder®, Microsoft Access, and DataEase with transparent access to heterogeneous data.
- Maintain referential integrity across heterogeneous data sources.
- Access native remote server capabilities using the Component Integration Services passthrough mode.

For more information on using Component Integration Services, see the Component Integration Services User’s Guide.

### New commands

Table 5-4 lists the new commands that support the Component Integration Services feature:
### Component Integration Services

#### Table 5-4: Commands for CIS

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connect to...disconnect</td>
<td>Connects to the specified server and disconnects the connected server.</td>
</tr>
<tr>
<td>create existing table</td>
<td>Creates a proxy table, then retrieves and stores metadata from a remote table, and places the data into the proxy table.</td>
</tr>
<tr>
<td>set cis_rpc_handling</td>
<td>Determines whether Component Integration Services handles outbound remote procedure call (RPC) requests by default.</td>
</tr>
<tr>
<td>set transactional_rpc</td>
<td>Controls the handling of remote procedure calls.</td>
</tr>
</tbody>
</table>

#### New system procedures

Table 5-5 lists the new system procedures that support the Component Integration Services feature.
Table 5-5: New system procedures for CIS

<table>
<thead>
<tr>
<th>System Procedure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>sp_addexternlogin</strong></td>
<td>Creates an alternate login account and password to use when communicating with a remote server through CIS.</td>
</tr>
<tr>
<td>sp_addobjectdef</td>
<td>Specifies the mapping between a local table and an external storage location. sp_addobjectdef replaces the sp_addtabledef command.</td>
</tr>
<tr>
<td>sp_autoconnect</td>
<td>Defines a passthrough connection to a remote server for a specific user, which allows the named user to enter passthrough mode automatically at login.</td>
</tr>
<tr>
<td>sp_configure &quot;enable cis&quot;</td>
<td>Enables or disables CIS. Default is 0 (disabled).</td>
</tr>
<tr>
<td>sp_defaultloc</td>
<td>Defines a default storage location for objects in a local database.</td>
</tr>
<tr>
<td>sp_droplexternlogin</td>
<td>Drops the definition of a remote login previously defined by sp_addexternlogin.</td>
</tr>
<tr>
<td>sp_dropobjectdef</td>
<td>Deletes the external storage mapping provided for a local object.</td>
</tr>
<tr>
<td>sp_passthru</td>
<td>Allows the user to pass a SQL command buffer to a remote server.</td>
</tr>
<tr>
<td>sp_remolesql</td>
<td>Establishes a connection to a remote server, passes a query buffer to the remote server from the client, and relays the results back to the client.</td>
</tr>
</tbody>
</table>

Modified system procedures

The following system procedures have been modified to support the Component Integration Services feature:

- sp_addserver
- sp_dropserver
- sp_help
- sp_helpdb
- sp_helpserver
- sp_serveroption
create index enhancements

create index enhancements

In earlier releases of SQL Server™, you could use the create clustered index...with sorted_data to skip the sort step on data that was already in sorted order. However, create clustered index always copied the data pages to a new location on the data devices.

In release 11.5, create clustered index...with sorted_data does not sort or copy data pages except when certain optional clauses are specified to create index. If these clauses are specified for partitioned tables, a parallel sort must be performed.


dbcc enhancements

dbcc has been enhanced to improve its performance in parallel environments and to expand the way in which it detects, records, and reports errors. The dbcc checkstorage command allows you to check a database that is in use, with little or no impact on performance. It stores the results of the check in the dbccdb database. When the check is complete, you can use the new dbcc stored procedures to generate reports based on the data that is collected.

Advantages of using dbcc checkstorage

The dbcc checkstorage command:

- Uses a named cache so that running it on the target database does not interfere with concurrent use of that database
- Scales linearly with the aggregate I/O throughput
- Does not lock tables or pages for extended periods, which allows dbcc checkstorage to locate errors while allowing concurrent update activity
- Combines many of the checks provided by the other dbcc commands
- Provides a detailed description of space usage in the target database
- Separates the functions of checking and reporting, which allows custom evaluation and report generation
Records dbcc checkstorage activity and results in the dbccdb database, which allows trend analysis and a source of accurate diagnostic information for support.


System and stored procedures for creating dbccdb

Before you can use dbcc checkstorage, you must set up the dbccdb database. Table 5-6 lists the system and stored procedures for creating dbccdb.

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sp_dbcc_createws</td>
<td>Creates a workspace of the specified type and size on the specified segment and database.</td>
</tr>
<tr>
<td>sp_dbcc_updateconfig</td>
<td>Updates the configuration parameters for the specified database.</td>
</tr>
<tr>
<td>sp_plan_dbccdb</td>
<td>Recommends values for the size of dbccdb, suitable devices, sizes for the scan and text workspaces, cache size, and the number of worker processes for optimal operation.</td>
</tr>
</tbody>
</table>

Stored procedures for maintaining dbccdb

Table 5-7 lists the dbcc stored procedures for maintaining dbccdb.
Table 5-7: Stored procedures for maintaining dbccdb

<table>
<thead>
<tr>
<th>Stored Procedure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sp_dbcc_alterws</td>
<td>Changes the size of the specified workspace to the specified value, and initializes the workspace.</td>
</tr>
<tr>
<td>sp_dbcc_deletedb</td>
<td>Deletes all information on the specified database from dbccdb.</td>
</tr>
<tr>
<td>sp_dbcc_deletehistory</td>
<td>Deletes results of dbcc checkstorage operations on the specified database from dbccdb.</td>
</tr>
<tr>
<td>sp_dbcc_evaluatedb</td>
<td>Recommends values for configuration parameters, using the results from previous dbcc operations.</td>
</tr>
<tr>
<td>sp_dbcc_runcheck</td>
<td>Runs dbcc checkstorage on the specified database and generates a summary report or the report you specify.</td>
</tr>
</tbody>
</table>

Stored procedures for generating reports on dbccdb

Table 5-8 lists the dbcc stored procedures for generating reports on dbccdb.
Table 5-8: Stored procedures for generating reports on dbccdb

<table>
<thead>
<tr>
<th>Stored Procedure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sp_dbcc_configreport</td>
<td>Reports the configuration parameters for the specified database.</td>
</tr>
<tr>
<td>sp_dbcc_differentialreport</td>
<td>Compares the results of dbcc checkstorage operations completed on date1 and date2 for dbname..object_name.</td>
</tr>
<tr>
<td>sp_dbcc_faultreport</td>
<td>Reports faults that were recorded for dbname..object_name on or before the specified date.</td>
</tr>
<tr>
<td>sp_dbcc_fullreport</td>
<td>Reports configuration, statistics, and fault information for database..object_name on or before the specified date.</td>
</tr>
<tr>
<td>sp_dbcc_statisticsreport</td>
<td>Reports statistics information from the dbcc_counters table generated by the dbcc checkstorage operation on or before the specified date.</td>
</tr>
<tr>
<td>sp_dbcc_summaryreport</td>
<td>Reports all dbcc checkstorage operations completed for dbname on or before the specified date.</td>
</tr>
</tbody>
</table>

Descending index scan optimization

Descending index scan optimization is a performance enhancement that can improve the performance of queries that use the desc keyword in order by clauses to return result sets in descending order. In earlier releases of SQL Server, descending result sets required a worktable and a sort. In Adaptive Server 11.5, the optimizer can choose to use the index and avoid the sort step, if the strategy reduces the query cost.

Descending index scans can speed the use of both clustered and nonclustered index access, reduce the tempdb space required for temporary tables, save the CPU time required for sorts, and shorten the time that locks are held, if descending scans use holdlock or transaction isolation level 3. However, there can be an increased chance of deadlocking in some applications.

See Chapter 11, “Setting Configuration Parameters,” in the System Administration Guide for information on the allow backward scans configuration parameter, which can be used to disable this optimization if deadlocking is a problem. Descending scans are enabled by default.

This feature does not change the syntax for the order by clause; it only changes the way that order by clauses with the desc keyword can be optimized.
Directory services

Before Adaptive Server and a client application can communicate with each other, each must know where the other resides on the network. Prior to Adaptive Server 11.5, the only location for storing network service information (server name, service type, and network protocol, device, and address) was the SQL Server interfaces file.

Adaptive Server 11.5 provides an alternative to using the traditional interfaces file by supporting the ability to connect to network service information through third-party directory service providers.

The advantage of using a directory service is that when your network is changed (for example, when a new server is added or a server address is changed), you do not need to update multiple interfaces files. You need to update only the directory service being used by the servers and clients.

Table 5-9 shows the directory service provider and the platform it supports in Adaptive Server 11.5.

<table>
<thead>
<tr>
<th>Directory Service Provider</th>
<th>Platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Registry</td>
<td>Microsoft Windows NT</td>
</tr>
</tbody>
</table>

If Adaptive Server is not configured to use a directory service provider, it defaults to the Sybase interfaces file for obtaining network service information.

For information on how to configure Adaptive Server for directory services, see the Adaptive Server configuration guide for your platform.

Engine affinity and execution precedence

The new Logical Process Manager in release 11.5 gives you a measure of control over the order in which Adaptive Server executes requests from connections and sessions. Two new capabilities, engine affinity and execution precedence, offer system procedures that let you tell Adaptive Server the performance trade-offs you want to make among individual connections and sessions in mixed work-load environments.
For example, you can rank a client application, login, or stored procedure as preferred with respect to others using system procedures to assign execution attributes. Adaptive Server will consider your execution assignments when it places the entity in one of three priority run queues. In addition, you can use engine affinity to make suggestions about how to partition Adaptive Server engine resources among connections and sessions.

In a multiple-application environment, establishing execution precedence can improve performance for selected applications and “on the fly” for some running Adaptive Server processes.

You establish which client applications, logins, and stored procedures receive execution precedence when you create a hierarchy among execution objects (client applications, logins, and stored procedures) using a new set of Adaptive Server system procedures (see Table 5-10). Adaptive Server uses the hierarchy as a guideline for distributing processing resources among execution objects.

For more information, see Chapter 22, “Distributing Engine Resources Between Tasks,” in the Performance and Tuning Guide.

**New system procedures**

You implement and manage execution hierarchy for client applications, logins, and stored procedures using the five categories of system procedures listed in Table 5-10.
Expanded error logging

Table 5-10: New system procedures for managing execution precedence

<table>
<thead>
<tr>
<th>Category</th>
<th>System Procedure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User-defined execution class</td>
<td>sp_addexeclass</td>
<td>Creates or drops a user-defined class with custom attributes or changes the attributes of an existing class.</td>
</tr>
<tr>
<td></td>
<td>sp_dropexeclass</td>
<td></td>
</tr>
<tr>
<td>Execution class binding</td>
<td>sp_bindexeclass</td>
<td>Binds or unbinds both predefined and user-defined classes to client applications and logins.</td>
</tr>
<tr>
<td></td>
<td>sp_unbindexeclass</td>
<td></td>
</tr>
<tr>
<td>“On the fly”</td>
<td>sp_setpsexec</td>
<td>Sets or clears attributes of an active session only.</td>
</tr>
<tr>
<td></td>
<td>sp_clearpsexec</td>
<td></td>
</tr>
<tr>
<td>Engines</td>
<td>sp_addengine</td>
<td>Adds engines to or drops engines from engine groups; creates and drops engine groups.</td>
</tr>
<tr>
<td></td>
<td>sp_dropengine</td>
<td></td>
</tr>
<tr>
<td>Reporting</td>
<td>sp_showcontrolinfo</td>
<td>Reports on engine group assignments, application bindings, and execution class attributes.</td>
</tr>
<tr>
<td></td>
<td>sp_showexeclass</td>
<td></td>
</tr>
<tr>
<td></td>
<td>sp_showpsexec</td>
<td></td>
</tr>
</tbody>
</table>

Expanded error logging

Users can now exercise control over whether certain messages are always logged to the Adaptive Server error log.

By setting the log audit logon success or log audit logon failure configuration parameter, you can enable or disable the logging of successful or unsuccessful Adaptive Server logins.

You can also specify that a particular message, identified by its message identifier, should be logged, using the with_log parameter to the sp_addmessage or sp_altermessage system procedure. See the Adaptive Server Reference Manual for information about these system procedures.

These features affect logging to the Adaptive Server error log on all platforms, as well as logging to the Windows NT Event Log on Windows NT servers, if event logging is enabled.
New system procedures

Table 5-11 lists the new system procedures that support the expanded error logging feature.

**Table 5-11: New system procedures for expanded error logging**

<table>
<thead>
<tr>
<th>System Procedure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sp_altermessage</td>
<td>Enables and disables the logging of a system-defined or user-defined message in the Adaptive Server error log.</td>
</tr>
<tr>
<td>sp_configure</td>
<td>Specifies whether to log successful Adaptive Server logins to the Adaptive Server error log and, on Windows NT servers, to the Windows NT Event Log, if event logging is enabled.</td>
</tr>
</tbody>
</table>

Modified system procedures

The following system procedures have been modified to support the expanded error logging feature by using the `with_log` parameter:

- sp_addmessage
- sp_addserver

Extended Stored Procedures (ESPs)

This release supports both user-defined and system-defined extended stored procedures (ESPs).

ESPs provide a method for calling procedural language functions from within Adaptive Server. The procedural language in which the functions are written must be capable of calling C language functions and manipulating C data types.
Extended Stored Procedures (ESPs)

ESPs allow Adaptive Server to perform a task outside Adaptive Server in response to an event occurring within Adaptive Server. For example, a procedural task outside the relational database management system (RDBMS), such as selling a security, could be invoked by a trigger on a database table that fires when the value of the security reaches a certain value. Or an email notification or a network-wide broadcast could be sent in response to an event occurring within the RDBMS.

The interface to ESPs is similar to the interface to system procedures and user-defined stored procedures. The difference is that an ESP executes procedural language code, rather than Transact-SQL statements.

XP Server

Extended stored procedures are implemented by an Open Server™ application called XP Server, which runs on the same machine as Adaptive Server. Running ESPs in a separate process protects Adaptive Server from failing because of faulty ESP code. Adaptive Server and XP Server communicate through remote procedure calls (RPCs).

The function that implements the ESP is compiled and linked into a dynamic link library (DLL) or a shared library. Adaptive Server looks in the system tables for the function that has the same name as the requested ESP and passes the function name and the DLL name to XP Server.

XP Server is responsible for:

- Loading the DLL, if it is not already loaded.
- Invoking the function that implements the ESP.
- Passing the function’s return status, output parameters, and results back to Adaptive Server.

Adaptive Server passes the results of the function to the client.

Adaptive Server starts XP Server on the first ESP request and shuts down XP Server when Adaptive Server shuts down.

The installation or configuration program for your platform adds the default <hostname>_XP entry to the interfaces file and the sysservers table. Additional XP Server entries must be added to the interfaces file using dsedit. Additional XP Server entries to the sysservers table must be manually added. There is no run_server file associated with XP Server.
See the Transact-SQL User’s Guide for detailed information about using and developing your own ESPs.

To develop your own ESPs, you need to purchase the Open Server product.

**System-defined ESPs**

Table 5-12 lists the system-defined ESPs.

**Table 5-12: System-defined ESPs**

<table>
<thead>
<tr>
<th>System Procedure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>xp_cmdshell</code></td>
<td>Executes a native operating system command on the host system running Adaptive Server.</td>
</tr>
<tr>
<td><code>xp_enumgroups</code></td>
<td>Displays groups for a specific Windows NT domain. (Windows NT only)</td>
</tr>
</tbody>
</table>

In addition, several Windows NT-specific ESPs provide Windows NT features, such as event logging and Adaptive Server mail support.


**New system procedures for ESPs**

Table 5-13 lists the system procedures that support ESPs.
Increased referential integrity limits

### Table 5-13: New system procedures for ESP

<table>
<thead>
<tr>
<th>System Procedure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sp_addextendedproc</td>
<td>Creates an ESP.</td>
</tr>
<tr>
<td>sp_configure</td>
<td></td>
</tr>
<tr>
<td>&quot;esp execution priority&quot;</td>
<td>Sets the priority of the XP Server thread for ESP execution.</td>
</tr>
<tr>
<td>&quot;esp execution stacksize&quot;</td>
<td>Sets the size of the stack, in bytes, to allocate for ESP execution.</td>
</tr>
<tr>
<td>&quot;esp unload dll&quot;</td>
<td>Specifies whether DLLs that support ESPs should automatically be unloaded from XP Server memory after the ESP call has completed.</td>
</tr>
<tr>
<td>&quot;xp_cmdshell_context&quot;</td>
<td>Sets the security context for the operating system command to be executed using the xp_cmdshell system ESP.</td>
</tr>
<tr>
<td>sp_dropextendedproc</td>
<td>Drops an ESP.</td>
</tr>
<tr>
<td>sp_freedll</td>
<td>Unloads a DLL that was loaded into XP Server memory to support the execution of an ESP.</td>
</tr>
<tr>
<td>sp_helpextendedproc</td>
<td>Displays ESPs in the current database with their associated DLL files.</td>
</tr>
</tbody>
</table>

Supporting commands

The following Transact-SQL commands have been enhanced to support ESPs:

- `create procedure`
- `drop procedure`
- `execute`

Increased referential integrity limits

The referential integrity relaxed limit feature increases the maximum number of table references allowed in a query from 16 to 192. The new limit eliminates the need to write special triggers to compensate for the previous 16-table reference limitation.
When a table has references to other tables, Adaptive Server scans the referenced tables when the table is being modified. Adaptive Server uses scan descriptors to manage the table scans. A scan descriptor is an internal data structure that manages the scan of the tables being queried. To make efficient use of Adaptive Server resources, you can configure Adaptive Server to use an appropriate number of scan descriptors for your system.

For more information on using referential integrity, see the Transact-SQL User’s Guide.

For information on configuring scan descriptors, see the number of aux scan descriptors configuration parameter in Chapter 11, “Setting Configuration Parameters,” of the System Administration Guide.

## Metadata cache management

Adaptive Server allows you to manage individual metadata caches for:

- User indexes
- Objects such as procedures, triggers, views, rules, and defaults
- Databases

Managing individual metadata caches for these objects is beneficial for a server that contains a large number of user indexes and objects and where there is high concurrency among users.

A metadata cache is a reserved area of memory for tracking information for indexes, objects, or databases. When you configure a metadata cache for indexes, objects, or databases, Adaptive Server accesses the information directly from the in-memory structure that describes them in the sysindexes, sysobjects, or sysdatabases row.

This method improves performance because Adaptive Server bypasses expensive calls that require disk access. Configuring metadata caches also improves performance by reducing synchronization and spinlock contention when Adaptive Server has to retrieve index, object, or database information at run time.

For more information on configuring metadata caches, see “Metadata Caches” on page 11-61 in the System Administration Guide.
Monitor access to SQL batch text

Monitor access to SQL batch text

The monitor access to SQL batch text feature allows a user with the System Administrator role to display the SQL batch text of any client command that is running on Adaptive Server. Although you configure Adaptive Server to save the SQL batch text, the batches are viewable only through Adaptive Server Monitor Server.

Monitor access includes sp_showplan, which allows you to view the query plan for the query that is currently running. Monitor Server is not required to retrieve the query plans.

Both monitor access and sp_showplan are particularly useful for determining why a client session is hanging or using large amounts of CPU time or physical I/O.

Allocating memory for batch text

Monitor access to SQL batch text is turned off by default when Adaptive Server is installed, so you must configure Adaptive Server to allocate memory for this feature. Consider the following to help you determine how much memory to allocate per user:

- SQL batches exceeding the allocated amount of memory are truncated without warning. Therefore, if you do not allocate enough memory for the batch statements, the text you want to view might be the section of the batch that is truncated.

- The more memory you allocate for SQL text from shared memory, the less chance there is that the problem statement will be truncated from the batch copied to shared memory. However, the memory you allocate for SQL text reduces the amount of memory for data and procedure caches.

New columns in sysprocesses

The procedure you are viewing may be nested in a batch of SQL text. To aid in the viewing of nested procedure, new columns have been added to the sysprocesses table to specify the object ID, current statement number and line number of the current statement of the procedure that is currently running.

Table 5-14 shows the new columns added to the sysprocesses table.
Parallel bulk copy

Parallel bulk copy allows you to copy data in parallel to Adaptive Server from multiple files. Parallel bulk copy substantially increases performance during bcp sessions because large bulk copy jobs can be split into multiple sessions and run concurrently.

Before you copy data into your database using parallel bulk copy, you must first partition the table that will contain the data. You can then issue multiple bcp sessions, with each session copying its data from a different source file into a different partition. If you are copying data from one, large operating system file, you divide this file into smaller files and then start multiple parallel bulk copy sessions.

Figure 5-2 illustrates how parallel bulk copy works.
You can also use the -F and -L flags to designate the first and last row of the same host file for each of the multiple parallel bulk copy sessions.

For more information about parallel bulk copy, see bcp in the Utility Programs manual for your platform.

Parallel queries and enhanced partitioning

When you configure Adaptive Server for parallel query processing, the optimizer evaluates each query to determine whether it is eligible for parallel execution. If it is, the query is divided into components that process simultaneously. The results are combined and delivered to the client in a shorter period of time than it would take to process the query serially as a single component.

For the same throughput, processing a query in parallel requires more work from Adaptive Server and additional resources than processing a query in serial. It also involves evaluating more complex trade-offs to achieve optimal performance. Fully enabled parallel query processing requires multiple processes, engines, and partitions, resulting in increased overhead for Adaptive Server, additional CPU requirements, and increased disk I/O.
You can configure various levels of parallelism, each providing a performance gain and requiring corresponding trade-offs in physical resources. Chapter 13, “Introduction to Parallel Query Processing,” in the Performance and Tuning Guide introduces the Adaptive Server parallel query processing model and concepts. It also discusses the trade-offs between resources and performance gains for different levels of parallelism.

For more information

In the Performance and Tuning Guide, see:

- Chapter 9, “Understanding Query Plans,” for information on new showplan messages added for parallel query execution.
- Chapter 14, “Parallel Query Optimization,” for details on how the Adaptive Server optimizer determines eligibility for parallel execution.
- Chapter 15, “Parallel Sorting,” to understand parallel sorting topics.
- Chapter 17, “Controlling Physical Data Placement,” for information on partitioned tables, creating clustered indexes on partitioned tables, and parallel processing.

In the System Administration Guide, see:

- Chapter 11, “Setting Configuration Parameters,” for details on how you can configure various levels of parallelism, each providing a performance gain and requiring corresponding trade-offs in physical resources.

**Point-in-time recovery**

Point-in-time recovery allows you to recover a database by rolling it forward to a particular time in its transaction log. You do this by using the until_time option of the load transaction command.

This feature is useful if, for example, a user inadvertently drops an important table from a database; you can use the until_time option to recover the changes to that database up to a time just before the table was dropped.

You can use point-in-time recovery for any database that can be loaded and dumped. However, you cannot use it for databases in which the data and logs are on the same device, such as the master database or for any database, such as tempdb, whose log has been truncated since the last dump database.
Proxy authorization

Adaptive Server provides the proxy authorization capability, which allows one user to assume the identity of another user on a server-wide basis. A System Security Officer can grant the ability to assume the security context of another user to selected logins. If a login has permission to use proxy authorization, the login can impersonate any other login in Adaptive Server.

**Warning!** The ability to assume another user’s identity is extremely powerful and must be strictly limited. A user with this permission could assume the identity of the “sa” login, which would give the user unlimited power within Adaptive Server. Plan to limit this permission to trusted administrators and applications, and to audit their server activity.

A System Security Officer or a System Administrator might want to assume the permissions of another user to make sure the permissions are correct for a user or to perform maintenance on a user’s database objects.

An application server can use proxy authorization to login to Adaptive Server with a generic login, which the application server uses to execute procedures and commands for several users.

A System Security Officer uses the grant set proxy or grant set session authorization command to give a user permission to use proxy authorization. The user with this permission can then execute either set proxy or set session authorization to become another user. A user executing set proxy or set session authorization operates with both the login and server user ID of the user being impersonated. The login and server user ID are active across the entire server in all databases.

**Note** set proxy and set session authorization are identical in function and can be used interchangeably. The only difference between them is that set session authorization is SQL92 compatible, and set proxy is a Transact-SQL extension.
For more information about proxy authorization, see the Security Administration Guide.

Recovery fault isolation

Prior to this release, when recovery detected corruption in a database, it would make the entire database inaccessible. The database remained unavailable to users until the suspect pages had been repaired or removed from the database.

With recovery fault isolation, a System Administrator has the option of isolating the corrupt pages marked suspect by recovery while the rest of the database is brought online and made available to users.

Recovery fault isolation provides a System Administrator the ability to:

- Specify whether an entire database or just the suspect pages become inaccessible when recovery detects corruption
- Specify whether a database with suspect pages is brought online in read_only mode
- List databases that have suspect pages
- List suspect pages by page ID, index ID, and object name
- Bring suspect pages online for the System Administrator while they are being repaired
- Bring suspect pages online for all database users after they have been repaired

The ability to isolate the suspect pages while bringing the rest of the database online provides a greater degree of flexibility in dealing with data corruption. Problems can be diagnosed, and sometimes corrected, while most of the database is accessible to users. The extent of the damage can be assessed and emergency repairs can be scheduled for a convenient time.

A System Administrator can force offline pages online temporarily, for System Administrator use only, to aid in the analysis and repair of the suspect pages.

Recovery fault isolation applies only to user databases. System databases are always taken entirely offline if they have any suspect pages.
Relaxed LRU cache replacement policy

New system procedures

Table 5-15 lists the system procedures that support recovery fault isolation:

<table>
<thead>
<tr>
<th>System Procedure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sp_forceonline_db</td>
<td>Provides access to all the suspect pages in a database.</td>
</tr>
<tr>
<td>sp_forceonline_page</td>
<td>Provides access to an individual suspect page.</td>
</tr>
<tr>
<td>sp_listsuspect_db</td>
<td>Lists all databases that have pages marked suspect by recovery.</td>
</tr>
<tr>
<td>sp_listsuspect_page</td>
<td>Lists all pages in a database that are marked suspect by recovery.</td>
</tr>
<tr>
<td>sp_setsuspect_granularity</td>
<td>Displays and sets the recovery fault isolation mode. This mode governs whether recovery marks an entire database or only the corrupt pages suspect.</td>
</tr>
<tr>
<td>sp_setsuspect_threshold</td>
<td>Sets the maximum number of suspect pages that recovery will allow in a database before marking the entire database suspect.</td>
</tr>
</tbody>
</table>

Side effects of recovery fault isolation

- Transactions that attempt to access offline data directly or indirectly will fail.
- You cannot use `dump database` on a database that has offline pages. This prevents users from dumping a corrupt database.
- You cannot use `dump transaction with no_log` or `dump transaction with truncate_only` on a database that has offline pages.

For more information on recovery fault isolation, see “Fault Isolation During Recovery” on page 20-7 in the System Administration Guide.

Relaxed LRU cache replacement policy

Relaxed LRU (least recently used) cache replacement policy is a performance feature designed to reduce overhead in named caches where little or no buffer replacement is occurring.
Relaxed LRU replacement policy removes the cache linkage overhead required by caches maintained on an MRU/LRU (most recently used/least recently used) basis. Relaxed LRU replacement policy is meant to be used when both of the following conditions are true:

- There is little or no replacement of buffers in the cache, and
- The data is not updated or is updated rarely.

In addition to saving the overhead to maintain the MRU/LRU chain, relaxed LRU replacement policy can improve CPU performance on multiple-CPU servers.

See Chapter 9, “Configuring Data Caches,” in the System Administration Guide for information on configuring caches to use LRU cache replacement policy.

See Chapter 16, “Memory Use and Performance,” in the Performance and Tuning Guide for information on how to select caches where relaxed LRU replacement policy can improve performance.

Resource limits

Adaptive Server provides resource limits to help System Administrators prevent queries and transactions from monopolizing server resources. A resource limit is a set of parameters specified by a System Administrator to prevent an individual login or application from:

- Exceeding estimated or actual I/O costs, as determined by the optimizer
- Returning more than a set number of rows
- Exceeding a given elapsed time

The set of parameters for a resource limit includes the time of day to enforce the limit and the type of action to take. For example, you can prevent huge reports from running during critical times of the day or kill a session whose query produces unwanted Cartesian products.

To specify times of the day or week, you can create named time ranges. A time range is a contiguous block of time within a single day across one or more contiguous days of the week. It is defined by its starting and ending periods. Contiguous days of the week are allowed to wrap around the end of one week to the beginning of the next.
Resource limits

Adaptive Server includes one predefined time range, the “at all times” range, which covers the period of midnight to midnight, Monday through Sunday. You can create, modify, and drop additional time ranges as necessary for resource limits.

For more information on resource limits, see Chapter 12, “Limiting Access to Server Resources,” in the System Administration Guide.

New system procedures

The system procedures in Table 5-16 support the resource limits feature.

Table 5-16: New system procedures for resource limits

<table>
<thead>
<tr>
<th>System Procedure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sp_add_resource_limit</td>
<td>Creates a resource limit.</td>
</tr>
<tr>
<td>sp_add_time_range</td>
<td>Creates a named time range.</td>
</tr>
<tr>
<td>sp_configure</td>
<td>&quot;allow resource limits&quot; Signals the server to allocate internal memory for</td>
</tr>
<tr>
<td></td>
<td>time ranges, resource limits, and internal server alarms. It also signals</td>
</tr>
<tr>
<td></td>
<td>the server to internally assign applicable ranges and limits to user</td>
</tr>
<tr>
<td></td>
<td>sessions. When this option is set to 1 (on), sp_configure displays the</td>
</tr>
<tr>
<td></td>
<td>optimizer’s cost estimate for a query. The default is 0 (off).</td>
</tr>
<tr>
<td>sp_drop_resource_limit</td>
<td>Drops a resource limit.</td>
</tr>
<tr>
<td>sp_drop_time_range</td>
<td>Drops a named time range.</td>
</tr>
<tr>
<td>sp_help_resource_limit</td>
<td>Displays resource limits in the current server.</td>
</tr>
<tr>
<td>sp_modify_resource_limit</td>
<td>Modifies a resource limit.</td>
</tr>
<tr>
<td>sp_modify_time_range</td>
<td>Modifies a named time range.</td>
</tr>
</tbody>
</table>

Supporting commands

The following commands support the resource limits feature:

- set showplan
- set statistics io
- statistics time
- select @@rowcount
Source text verification and encryption

In releases prior to Adaptive Server release 11.5, the source text of compiled objects was saved in syscomments only so that it could be returned to a user who executed sp_helptext. Since this was the only reason for saving the source text, users often deleted it to save disk space and to remove confidential information from this public area.

In Adaptive Server release 11.5, you should not remove the source text from syscomments. Deleting the source text from syscomments will cause problems when upgrading to the next release of Adaptive Server. The upgrade process in future releases of Adaptive Server will require that the source text of compiled objects be present in syscomments at the time of the upgrade. Before upgrading to the next release of Adaptive Server, you need to restore missing source text to syscomments.

You can use the sp_checksource system procedure to verify that source text is present in syscomments for each compiled object. To prevent users from seeing the source text in syscomments, you can encrypt the text with the sp_hidetext system procedure.

For more information, see “Compiled Objects” on page 1-3 in the Transact-SQL User’s Guide. Also see sp_checksource and sp_hidetext in the Adaptive Server Reference Manual.

Sybase Central

With the release of Adaptive Server, Sybase no longer supplies SQL Server Manager, which has been replaced by Sybase Central™, Sybase's common management interface, and the Adaptive Server Enterprise plug-in for Sybase Central.

The Adaptive Server plug-in for Sybase Central allows you to manage Adaptive Server installations using the Sybase Central graphical management tool. You can manage Adaptive Server running on any platform from a PC that is running Windows 95 or Windows NT.

To learn how to use Sybase Central to manage Adaptive Server, see Managing and Monitoring Sybase Adaptive Server Enterprise.
\textit{sp\_sysmon}

\textit{sp\_sysmon}

sp\_sysmon supports the new performance features in release 11.5, including the addition of these new sections in the \textit{sp\_sysmon} report:

- Application Queue Management
- Asynchronous Prefetch Usage
- Metadata Cache Management
- Monitor Access to Executing SQL
- Parallel Query Management
- Worker Process Management

The syntax for \textit{sp\_sysmon} has been changed:

- You can specify the interval using the form “hh:mm:ss”.
- You can use the \texttt{begin\_sample} argument to start \textit{sp\_sysmon} which continues to monitor until you use \texttt{end\_sample} to print the report.
- You can print only one section of the report at a time. This example specifies an interval of 15 minutes and prints only the Data Cache Management section:

  \begin{verbatim}
  sp\_sysmon "00:15:00, dcache
  \end{verbatim}


\section*{Two-phase commit enhancements}

The two-phase commit protocol allows client applications to coordinate transaction updates across two or more Adaptive Servers and treats those transactions as a single transaction. Two-phase commit guarantees that either all or none of the databases in the participating Adaptive Servers are updated.

The enhancements to two-phase commit are:

- Improved space and log management – The \texttt{spt\_committab} table has been moved from the \texttt{master} database to a new system database, \texttt{sybsystemdb}.
- Improved performance – You can use a random value for the commit identifier—\texttt{commid}. 
**spt_comittab moved to sybsystemdb**

The `spt_comittab` table in Adaptive Server stores information and tracks the completion status of each two-phase commit transaction. All activities made against `spt_comittab` are logged. Prior to 11.5, `spt_comittab` resided in the master database. However, the logging activity against `spt_comittab` for two-phase commit transactions can fill up the master transaction log, causing frequent space management problems.

In release 11.5, the `spt_comittab` table resides in a new system database called `sybsystemdb`, instead of the master database. Providing `spt_comittab` its own database makes space management and recovery of two-phase commit transactions easier, for two reasons:

- The `dump tran` command can be used against `sybsystemdb`.
- Unlike the master database, the `sybsystemdb` database is extensible.

**Commit identifier changed to a random value**

The value of the commit identifier, `commid` in the `spt_comittab` table, becomes a random value (between 1 and \(2^{31}-1\)), instead of a monotonically increasing value. This reduces page lock contention on the row where the maximum `commid` resides when several distributed transactions try to insert an entry into the `spt_comittab` table. `spt_comittab` continues to have a unique, clustered index.

For information on how to configure Adaptive Server for two-phase commit functionality, see the Adaptive Server configuration documentation for your platform.

**User-defined roles**

The user-defined roles feature allows a System Security Officer to simplify the task of granting and preventing access to data.

The System Security Officer can create roles, request object owners to grant privileges to each role, and grant the user-defined roles to individual employees, based on the functions they perform in the organization. The System Security Officer can also revoke user-defined roles granted to the employee.
Windows NT event log integration

This release has expanded support for logging Adaptive Server errors in the Windows NT Event Log. This is in addition to automatic logging in the Adaptive Server error log. This option is available on Windows NT servers only.

By default, Windows NT Event Logging of Adaptive Server messages is enabled, but you can disable it by setting the event logging configuration parameter to 0. You can also specify a remote Windows NT computer to receive event logging by setting the event log computer name configuration parameter to the name of the remote computer.

This feature provides an extended stored procedure that you can use to log user-defined events to the Windows NT Event Log from within Adaptive Server. See `sp_logdevice` in the Adaptive Server Reference Manual.

Supporting system ESPs and system procedures

The following system ESP and `sp_configure` parameters support the Windows NT Event Log Integration feature.
Windows NT MAPI support

Sybase Adaptive Server Enterprise for Windows NT can send and receive email messages using Microsoft’s Message Application Programming Interface (MAPI). This facility, called Sybmail, is currently available only on Windows NT.

Adaptive Server can send outgoing messages that consist of text or query results. Adaptive Server can receive queries as incoming messages.

To use Sybmail, Adaptive Server must have a Windows NT Mail account and a Windows NT Mail profile. Sybmail must also have a login in Adaptive Server.

See Configuring Adaptive Server for Windows NT for information on setting up and using Sybmail.

Supporting system ESPs and system procedures

The following system ESPs and system procedures support the Windows NT MAPI feature:

---

**Table 5-17: System support for Windows NT Event Log Integration**

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>xp_logevent</td>
<td>A system-defined ESP that provides for logging a user-defined event in the Windows NT Event Log.</td>
</tr>
<tr>
<td>sp_configure</td>
<td>Specifies the name of the Windows NT PC that logs Adaptive Server messages in its Windows NT Event Log.</td>
</tr>
<tr>
<td>&quot;event log computer name&quot;</td>
<td>Enables and disables the logging of Adaptive Server messages in the Windows NT Event Log.</td>
</tr>
</tbody>
</table>
Windows NT Performance Monitor integration

You can enable and disable the ability to monitor Adaptive Server statistics from the Windows NT Performance Monitor by setting the SQL Perfmon Integration configuration parameter. This parameter applies to Windows NT servers only.

Adaptive Server must be registered as an NT Service to support performance monitor integration. This occurs automatically when you start Adaptive Server using Sybase Central, the Services Manager, or the Services option in the Control Panel or when you have configured Windows NT to start Adaptive Server as an automatic service.

For more information on Windows NT Performance Monitor, see Chapter 9, “Administrative Tasks and Performance and Tuning,” in Configuring Adaptive Server for Windows NT.

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sp_configure</td>
<td>&quot;start mail session&quot; Enables and disables the automatic initiation of a Adaptive Server mail session when Adaptive Server is started.</td>
</tr>
<tr>
<td>sp_processmail</td>
<td>Reads, sends, and deletes messages in the Adaptive Server message inbox. Available on Windows NT only.</td>
</tr>
<tr>
<td>xp_deletemail</td>
<td>Deletes a message from the Adaptive Server message inbox.</td>
</tr>
<tr>
<td>xp_findnextmsg</td>
<td>Retrieves the message identifier of the next message in the Adaptive Server message inbox.</td>
</tr>
<tr>
<td>xp_readmail</td>
<td>Reads a message from the Adaptive Server message inbox.</td>
</tr>
<tr>
<td>xp_sendmail</td>
<td>Sends a message to the specified recipients using the MAPI interface.</td>
</tr>
<tr>
<td>xp_startmail</td>
<td>Starts an Adaptive Server mail session.</td>
</tr>
<tr>
<td>xp_stopmail</td>
<td>Stops an Adaptive Server mail session.</td>
</tr>
</tbody>
</table>

For more information on the system-defined ESPs, see Chapter 5, “System Extended Stored Procedures,” in the Adaptive Server Reference Manual.
Supporting system procedure

The following `sp_configure` configuration parameter supports the Windows NT Performance Monitor Integration feature:

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>sp_configure</code></td>
<td>&quot;SQL Perfmon Integration&quot; Enables and disables the ability to monitor Adaptive Server statistics from the Windows NT Performance Monitor.</td>
</tr>
</tbody>
</table>
CHAPTER 6

New Features in SQL Server
Release 11.0

This chapter describes the features and system changes introduced in SQL Server release 11.0. If you are upgrading to release 11.5 from a pre-11.0 release, these features will be new to you.

Topics covered are:
• “New features in release 11.0” on page 111
• “System changes in release 11.0” on page 128
• “Changes that may affect existing applications” on page 132

New features in release 11.0

The new features in release 11.0 are as follows:
• “User-defined caches” on page 112
• “Data storage changes” on page 115
• “Transaction log changes” on page 117
• “Isolation level 0” on page 119
• “Lock manager changes” on page 119
• “Housekeeper task” on page 122
• “SQL Server configuration” on page 122
• “Lock escalation” on page 123
• “Multiple network engines” on page 123
• “Improvements to showplan” on page 124
• “Query and data modification changes” on page 125
• “Upgrading database dumps” on page 125
User-defined caches

Release 11.0 allows System Administrators to split the SQL Server data cache into separate named data caches and to bind databases or database objects to those caches. Also, System Administrators can create pools within caches that perform large I/O to disk, improving performance for many queries.

Configuring caches

Named data caches are created with the new system procedure `sp_cacheconfig`. Memory pools for large I/O are configured within data caches with `sp_poolconfig`. Configuring caches and pools can also be done by editing a configuration file.

Entities are bound to caches with `sp_bindcache` and unbound with `sp_unbindcache` or `sp_unbindcache_all`. The system procedure `sp_helpcache` provides information about caches and cache bindings. `sp_cacheconfig` also provides information about caches and the pools within caches.

See Chapter 9, “Configuring Data Caches,” in the *System Administration Guide* for instructions on configuring named caches and binding database objects to them.

Binding objects to user-defined caches

In general, you can increase performance by binding the following objects to their own named caches:

- `sysindexes` table and its index
  
  Bind these two objects to the same named cache. This is the most frequently used table and its index. However, you do not have to make its named cache very large.

- `syslogs` table
  
  This helps reduce the contention on the buffer manager spinlocks. You should also configure a 4K memory pool in its named cache to benefit from the larger log I/O size, as defined by `sp_logiosize`. 
• tempdb database

  This helps performance if many temporary or work tables are generated by your applications.

In addition to the above objects, the most frequently used tables and indexes (based on your database and application designs) are good candidates for binding to their own caches. However, if you have only one table with no index, binding that table to a named cache will not improve performance.

**Note** Having too many named caches without increasing the total memory available to your server is generally not a good idea. Each additional named cache reduces the amount of memory available in the default cache.

---

**Cache strategies**

SQL Server release 11.0 provides new optimization strategies and new commands to control the use of those strategies for objects, sessions, and queries. Figure 6-1 shows the two cache strategies.

*Figure 6-1: Cache strategy choices*

SQL Server can:

• Read pages in at the start of the MRU (most recently used) chain in order to keep the pages in cache so that they can be accessed many times without performing additional I/O

• Read pages in near the end of the cache so that they will not force other pages out of the cache

Large I/O

When caches are created, all space is assigned to a 2K pool. This pool can be split to allow the cache to perform large I/O, reading up to eight data pages at once. Since most I/O time is spent performing queueing, seeking, and positioning operations, large I/O can greatly increase performance for queries that scan entire tables or ranges of tables.

SQL Server can also perform large I/O on the transaction log.

Changing the log I/O size

Using the new `sp_logiosize` system procedure, you can change the log I/O size used by SQL Server. This can improve performance by reducing the number of times that SQL Server writes transaction log pages to disk. The value you specify for `sp_logiosize` must correspond to an existing memory pool configured for the cache used by the database’s transaction log.

By default, SQL Server defines the log I/O size of a database as 4K. If the transaction log for a database is bound to the default cache or a user-defined cache that does not contain a 4K memory pool, SQL Server sets the log I/O size to 2K (a 2K memory pool is always present in any cache). For most workloads, a log I/O size of 4K performs much better than one of 2K, so each cache used by a transaction log should have a 4K memory pool.

New query tuning options

New query tuning options support the capabilities of named caches and provide additional control when tuning queries:

- The `set forceplan` command forces SQL Server to join the tables in a query in the order in which they are listed in the `from` clause.
- The `set table count` command allows you to specify the number of tables that are considered at once as joins are optimized.
- The new system procedure `sp_cachestrategy` disables and reenables cache strategies and large I/O for individual objects.
- New clauses for `select`, `delete`, and `insert` commands allow you to specify the index, cache strategy, or I/O size for a query.
- The `set prefetch` command allows you to enable or disable large I/O for a session.
See Chapter 10, “Advanced Optimizing Techniques,” in the *Performance and Tuning Guide* for information on query processing options that control caching strategies and I/O size.

**Data storage changes**

This section describes changes to the way SQL Server stores data.

**Maximum number of rows per page**

You can now configure the maximum number of rows stored on a data page or index leaf page. The `max_rows_per_page` value specified in a `create table`, `create index`, or `alter table` command restricts the number of rows allowed on a data page, a clustered index leaf page, or a nonclustered index leaf page. This reduces lock contention and improves concurrency for frequently accessed tables.

The `max_rows_per_page` value applies to the data pages of an unindexed table or the leaf pages of an index. The value of `max_rows_per_page` is stored in the `maxrowsperpage` column of `sysindexes`. (The `maxrowsperpage` column was named `rowpage` in previous releases of SQL Server.)

Unlike fillfactor, which is not maintained after creating a table or index, SQL Server retains the `max_rows_per_page` value when adding or deleting rows. For information about how to use `max_rows_per_page` to reduce lock contention and improve concurrency for your server, see the *Performance and Tuning Guide*.

**Page allocation**

SQL Server 10.0 and earlier releases searched the OAM page chain for unused pages before allocating new extents to objects. Release 11.0 provides a system-wide configuration parameter, `page utilization percent`, that allows SQL Server to allocate a new extent to an object without searching the OAM page chain, depending on the ratio of used and unused pages in the table to the table’s allocated extents.
Partitioned tables

**Note** 11.5 Note: Release 11.5 adds the ability to scan partitioned tables in parallel and to create clustered indexes.

By default, SQL Server stores a heap table’s data in one double-linked chain of database pages. When a transaction inserts a row into the table, it holds an exclusive page lock on the last page of the page chain while inserting the row. As multiple transactions attempt to insert rows into the same table at the same time, performance problems can occur. Because only one transaction at a time can obtain an exclusive lock on the last page, other, concurrent insert transactions block, as shown in Figure 6-2.

**Figure 6-2: Page contention during inserts**

In release 11.0, SQL Server provides the ability to partition heap tables. A partition is simply another term for a page chain. Partitioning a table creates multiple page chains (partitions) for the table and, therefore, multiple last pages for insert operations.

When a transaction inserts data into a partitioned table, SQL Server randomly assigns the transaction to one of the table’s partitions. Concurrent inserts are less likely to block, since multiple last pages are available for inserts, as shown in Figure 6-3.
If a table’s segment spans several physical disks, SQL Server distributes the table’s partitions across those disks when you create the partitions. This can improve I/O performance when SQL Server writes the table’s cached data to disk, since the I/O is distributed over several devices.

SQL Server release 11.0 manages partitioned tables transparently to users and applications. Partitioned tables appear exactly like unpartitioned tables, except when accessed via the `dbcc checktable` and `dbcc checkdb` commands or when viewed with the new `sp_helpartition` procedure.

See Chapter 17, “Controlling Physical Data Placement,” of the *Performance and Tuning Guide* for information about how to create and administer partitioned tables.

### Transaction log changes

This section describes the changes made to the transaction log for release 11.0.

### User log caches

There is one user log cache for each configured user connection. SQL Server uses these user log caches to buffer the user transaction log records, which reduces the contention at the end of the transaction log. When a user log cache becomes full or when another event occurs (such as when the transaction completes), SQL Server “flushes” all log records from the user log cache to the database transaction log. You can configure the size of all user log caches for your server using the `user log cache size` parameter of `sp_configure`; the default size is 2048 bytes.
New features in release 11.0

Since more than one process can access the contents of a user log cache when determining a possible “flush,” SQL Server uses spinlocks to protect the user log caches. A spinlock is an internal locking mechanism that prevents a second process from accessing the data structure being used by another process. SQL Server uses the user log cache spinlock ratio parameter of sp_configure to specify the ratio of user log caches per user log cache spinlock; the default is 20 user log caches for each spinlock.

For information about configuring user log cache size and user log cache spinlock ratio, see Chapter 11, “Setting Configuration Parameters,” in the System Administration Guide.

**syslogshold table**

You can query the new syslogshold system table to determine the oldest active transaction in each database. syslogshold exists in the master database, and each row in the table represents either:

- The oldest active transaction in a database or
- The Replication Server® truncation point for the database’s log.

A database may have no rows in syslogshold, a row representing one of the above, or two rows representing both of the above. For information about how Replication Sever truncation points affects the truncation of a database’s transaction log, see the Replication Server documentation.

Querying syslogshold can help you when the transaction log becomes too full, even with frequent log dumps. The dump transaction command truncates the log by removing all pages from the beginning of the log up to the page that precedes the page containing an uncommitted transaction record (the oldest active transaction). The longer this active transaction remains uncommitted, the less space is available in the transaction log, since dump transaction cannot truncate additional pages.

For information about how to query syslogshold to determine the oldest active transaction that is holding up your transaction dumps, see Chapter 21, “Backing Up and Restoring User Databases,” in the System Administration Guide.
Isolation level 0

You can specify isolation level 0 for the queries in transactions along with the isolation levels 1 and 3 supported in release 10.0. Isolation level 0 prevents other transactions from changing data that has already been modified by an uncommitted transaction. The other transactions are blocked from modifying that data until the transaction commits. However, other transactions can still read the uncommitted data (known as dirty reads).

Queries executing at isolation level 0 do not acquire any read locks for their scans, so they do not block other transactions from writing to the same data (and vice versa). Applications that are not impacted by dirty reads may see better concurrency and reduced deadlocks when accessing the same data using isolation level 0. However, transactions that require data consistency should not use isolation level 0.

You can specify isolation level 0 for the transactions of a session as follows:

```sql
set transaction isolation level 0
```

Since this command makes all queries execute at isolation level 0, you should be wary of using it for any transaction that requires data consistency. Instead, you can selectively choose isolation level 0 for a query in a transaction using the `at isolation` clause as follows:

```sql
select *
from titles
at isolation read uncommitted
```

For information about transactions and isolation levels, see Chapter 18, “Transactions: Maintaining Data Consistency and Recovery,” in the *Transact-SQL User’s Guide*.

Lock manager changes

This section describes the changes to locking behavior.
New features in release 11.0

Table, page, and address lock tables

SQL Server manages the acquiring and releasing of table locks by using an internal hash table with 101 rows (known as hash buckets) and of page and address locks by using internal hash tables with 1031 rows each. In release 11.0, these tables use one or more spinlocks to serialize access between processes that are running on different engines. A spinlock is an internal locking mechanism that prevents a second process from accessing the resource currently being used by another process. All processes trying to access the resource must wait (or spin) until the lock is released.

For SQL Servers running with multiple engines, you can set the ratio of spinlocks protecting each hash table using the table lock spinlock ratio, page lock spinlock ratio, and address lock spinlock ratio configuration parameters. The table locks hash table default ratio is 20 rows for each spinlock, and the page or address locks hash table default ratio is 100 rows for each spinlock. A SQL Server configured with only one engine has only one spinlock for each hash table, regardless of the values specified for these parameters.

For information about configuring the table lock spinlock ratio, page lock spinlock ratio, and address lock spinlock ratio parameters, see Chapter 11, “Setting Configuration Parameters,” in the System Administration Guide.

Deadlock checking

SQL Server performs deadlock checking after a minimum period of time for any process that is waiting (or sleeping) for a lock to be released. By default, this minimum period of time is 500 ms. Previous releases of SQL Server perform this deadlock check at the time the process begins to wait for a lock. This deadlock checking is a time--consuming overhead for applications that wait without a deadlock.

You can change the minimum amount of time (in milliseconds) that a process must wait before it initiates a deadlock check using the deadlock checking period configuration parameter. If you expect your applications to deadlock infrequently, you can delay deadlock checking even further and reduce the overhead cost. However, configuring deadlock checking period to a higher value produces longer delays before deadlocks are detected.

For information about deadlock checking, see Chapter 5, “Locking in Adaptive Server,” in the Performance and Tuning Guide.
New engine freelock lists

In release 11.0, when a process that is running on a multi-engine SQL Server requests a lock, it looks for one in its engine’s freelock list. If the engine freelock list is out of locks, SQL Server moves a certain number of locks from its global freelock list to the engine freelock list. For single-engine SQL Servers, the entire global freelock list is moved to the engine freelock list at server start-up time.

After an engine completes a process, all locks held by that process are released and returned to that engine’s freelock list. This reduces the contention of each engine accessing the global freelock list. However, if the number of locks released to the engine exceed the maximum number of locks allowed in the engine’s freelock list, SQL Server moves a number of locks to the global freelock list. This replenishes the number of locks that are available to other engines from the global list.

You can configure the maximum number of locks available to the engine freelock lists as a percentage of the total number of locks available to your server with the `max engine freelocks` configuration parameter. You can also configure the number of locks transferred back and forth between the engine and global freelock lists using the `freelock transfer block size` parameter. For information about these parameters, see Chapter 11, “Setting Configuration Parameters,” in the System Administration Guide.

Increasing the engine freelock lists

Each engine’s freelock list is a portion of the global freelock list allocated to the engines, defined by the `max engines freelocks` configuration parameter as a percentage of the total number of locks. By default, SQL Server assigns 10 percent of the total number of locks to all the engines’ freelock lists; the remainder stays in the global freelock list. For example, if your server is configured with 5000 locks, 500 locks are distributed evenly between each engine freelock list, and 4500 remain in the global freelock list.

For most SQL Server configurations, you can double the percentage of locks assigned to the engine freelock lists and improve the performance of your applications.
Housekeeper task

When SQL Server has no user tasks to process, a housekeeper task automatically begins writing dirty buffers from cache to disk. Because these writes are done during the server’s idle cycles, they are known as free writes.

The benefits of the housekeeper task are:

- Improved CPU utilization
- Decreased need for buffer washing during transaction processing
- Faster checkpoints
- Shorter recovery time

In applications that repeatedly update the same database page, the housekeeper task may initiate some database writes unnecessarily. System Administrators can use the housekeeper free write percentage configuration parameter to disable the housekeeper task or to control its side effects.

For more information about the housekeeper task, see Chapter 21, “How Adaptive Server Uses Engines and CPUs,” in the Performance and Tuning Guide.

SQL Server configuration

In previous releases, SQL Server was configured using sp_configure, reconfigure, and a number of undocumented dbcc tune and buildmaster -y parameters. In order to provide a single entry point for all SQL Server configuration (except for Buffer Manager), sp_configure has been redesigned to incorporate the configuration parameters previously implemented with dbcc tune and buildmaster -y. In addition, a number of new configuration parameters have been implemented.

For information about which old configuration names have new names, see “New names for existing configuration parameters” on page 134.

The sp_configure interface has been redesigned to include these new features:

- Display levels, which allow the user to select one of three views of the configuration parameters:
  - Basic level, appropriate for general tuning of SQL Server
  - Intermediate level, somewhat more inclusive than the Basic level
• Comprehensive level, appropriate for the most detailed level of SQL Server tuning
• Parameter hierarchy, which organizes the configuration parameters according to the area of SQL Server behavior to which they pertain.
• Configuration files, which allow users to replicate specific configurations, validate configurations before setting them, and create multiple configurations that can be easily switched.

For information about SQL Server configuration, see Chapter 11, “Setting Configuration Parameters,” in the System Administration Guide.

Lock escalation

In previous releases of SQL Server, once a statement accumulated more than 200 page locks on a table, SQL Server tried to issue a table lock on that object. If the table lock succeeded, the page locks were no longer necessary and were released.

The lock promotion HWM, lock promotion LWM, and lock promotion PCT configuration parameters and the sp_setpglockpromote system procedure configure the number of page locks that SQL Server acquires on a table before it attempts to escalate to a table lock on a server-wide, per database, and per table basis.

For more information, see Chapter 5, “Locking in Adaptive Server,” in the Performance and Tuning Guide.

Multiple network engines

If your symmetric multiprocessing (SMP) system supports network affinity migration, each SQL Server engine handles the network I/O for its connections in release 11.0. During login, SQL Server migrates the client connection task from engine 0 to the engine that is servicing the smallest number of connections. The client’s tasks run network I/O on that engine until the connection is terminated.
By distributing the network I/O among its engines, SQL Server can handle more user connections. The per-process limit on the maximum number of open file descriptors no longer limits the number of connections. Adding more engines linearly increases the maximum number of file descriptors. For more information about how multiple network engines work in an SMP environment, see Chapter 10, “Managing Multiprocessor Servers,” in the System Administration Guide.

**Improvements to **showplan**

The output for the set showplan command has been improved as follows:

- Indents and delimiters have been added to help readability.
- Line numbers and statement numbers help debug long batches and procedures.
- Additional messages include the keys used on indexes and messages that provide more information about access methods.
- Subquery types, nesting levels, and other subquery information help check subquery performance.
- Messages have been added for the new release 11.0 features to display the I/O size and the caching strategy.

In the following example, there is an index named `title_ix` on the `titles` table:

```sql
select title_id, price
from titles
where title = 'Computers and Privacy'
```

This is the pre-11.0 output from showplan:

```
STEP 1
The type of query is SELECT.
FROM TABLE titles
Nested iteration
Index : title_ix
```

This is the release 11.0 output from showplan:

```
QUERY PLAN FOR STATEMENT 1 (at line 1).

STEP 1
The type of query is SELECT.
```
FROM TABLE
titles
Nested iteration.
Index : title_ix
Ascending scan.
Positioning by key.
Keys are:
    title
Using I/O Size 2 Kbytes.
With LRU Buffer Replacement Strategy.

All showplan messages are documented in Chapter 9, “Understanding Query Plans,” of the Performance and Tuning Guide.

Query and data modification changes

This section describes the changes made to subquery processing and update behavior.

Subquery changes

The behavior of certain subqueries has changed. See “Changes that may affect existing applications” on page 132 for detailed information about these changes and how they may affect your existing applications.

Update changes

Update strategies for SQL Server have improved. More updates are now made in place or directly, with fewer writes to the transaction log. See the Performance and Tuning Guide for more information on performance and update operations.

Upgrading database dumps

When a SQL Server installation is upgraded to a new release, all databases associated with that server are automatically upgraded.

As a result, the database and transaction log dumps created with any previous SQL Server must be upgraded before they can be used with the current SQL Server.
SQL Server release 11.0 provides an automatic upgrade mechanism, on a per-database basis, for upgrading a database or transaction log dump from any SQL Server release 10.0 to the current SQL Server, thus making the dump compatible for use. This mechanism is entirely internal to SQL Server release 11.0 and requires no external programs. It provides the flexibility of upgrading individual dumps as needed.

The following tasks are not supported by this automatic upgrade functionality:

- Loading a SQL Server release 10.0 master database onto release 11.0.
- Installing new or modified stored procedures. Continue to use `installmaster`.
- Loading and upgrading dumps prior to SQL Server release 10.0.

Associated with this feature is a new `online database` command. This command marks a database available for public use after a normal load sequence, and if needed, upgrades a loaded database and/or transaction log dumps to the current version of SQL Server.

For more information on upgrading database dumps, see Chapter 21, “Backing Up and Restoring User Databases,” in the *System Administration Guide* and the `load database` and `dump database` commands in the *Reference Manual*.

For more information on the `online database` command, see the *Reference Manual*.

### Tape device determination by Backup Server

To become less device dependent and more flexible, Backup Server™ provides a method for determining the tape device characteristics (tape positioning for read, close, append, I/O size, file marks, and ability to overwrite a tape mark) for a dump operation.

When you issue a `dump database` or `dump transaction` command, Backup Server checks to see whether the device type of the specified dump device is known (supplied and supported internally) by SQL Server. If the device is not a known type, Backup Server checks the tape configuration file (default location is `$SYBASE/backup_tape.cfg`) for the device configuration. If the configuration is found, the dump command proceeds.

If the configuration is not found in the tape device configuration file, the dump command fails with the following error message:

```
Device not found in configuration file. INIT needs to
```
be specified to configure the device.

This means the device needs to be configured. Issue the `dump database` or `dump transaction` with `init` qualifier to configure the device. Using operating system calls, Backup Server attempts to determine the device’s characteristics, and if successful, stores the device characteristics in the tape configuration file.

If Backup Server cannot determine the dump device characteristics, it defaults to one dump per tape. The device can not be used if the configuration fails to write at least one dump file.

Tape configuration by Backup Server applies only to UNIX platforms.

For more information on tape device determination and the tape configuration file, see Chapter 21, “Backing Up and Restoring User Databases,” in the *System Administration Guide*.

**IDENTITY column changes**

The IDENTITY column feature allows you to create a column with system-generated values that uniquely identify each row in a table. Release 11.0 provides the following enhancements to IDENTITY columns:

- For tables with no unique indexes, the `identity in nonunique index database` option allows you to create a unique index by including an IDENTITY column in the index key. Unique indexes are required for isolation level 0 reads and updatable cursors.

- The `identity grab size` configuration parameter allows each SQL Server process in a multiprocessor environment to reserve a specified number of IDENTITY column values when you add rows to tables that contain IDENTITY columns.

- The `size of auto identity` configuration parameter sets the precision of the IDENTITY columns that SQL Server generates for tables created in databases where the `auto identity` database option is set to `true`.

For more information about the enhancements to IDENTITY columns, see Chapter 7, “Creating Databases and Tables,” in the *Transact-SQL User’s Guide*. 
New text and image global variables

The following five global variables have been added to release 11.0 to facilitate inserting or updating text and image data through Open Client™ applications:

- `@@textcolid`
- `@@textdbid`
- `@@textobjid`
- `@@textptr`
- `@@textts`

For information about these global variables, see Chapter 13, “Using Batches and Control-of-Flow Language,” in the Transact-SQL User’s Guide. For information about how to use these variables in a procedure for Open Client applications, see the Client-Library/C Reference Manual.

System changes in release 11.0

This section provides a summary of release 11.0 changes. The changes are:

- New online database command
- Changes to existing commands
- New set options
- New system procedures
- Changes to system procedures
- New system tables
- Changes to existing system tables

New online database command

Table 6-1 describes the new online database command:
CHAPTER 6  New Features in SQL Server Release 11.0

Table 6-1: New online database command

<table>
<thead>
<tr>
<th>Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>online database</td>
<td>Makes a database available for public use after a normal load sequence and, if needed, upgrades a loaded database and transaction log dumps to the current version of SQL Server.</td>
</tr>
</tbody>
</table>

Changes to existing commands

Table 6-2 summarizes the changes to existing commands:

<table>
<thead>
<tr>
<th>Name</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>alter table</td>
<td>The new partition clause allows you to create additional page chains for a table with no clustered index. The new unpartition clause allows you to concatenate all page chains for a partitioned table.</td>
</tr>
<tr>
<td>buildmaster</td>
<td>The buildmaster -r flag is no longer supported.</td>
</tr>
<tr>
<td>delete</td>
<td>New clauses allow specifying the index, cache strategy, and I/O size for a query.</td>
</tr>
<tr>
<td>reconfigure</td>
<td>The reconfigure command is no longer required to activate changes made with sp_configure. reconfigure is now non-operational. Scripts using reconfigure will still run, but should be changed at your earliest convenience, as reconfigure may not be supported in future releases.</td>
</tr>
<tr>
<td>select</td>
<td>New clauses allow specifying the index, cache strategy, and I/O size for a query. The new at isolation clause allows you to specify the isolation level for the query.</td>
</tr>
<tr>
<td>set showplan</td>
<td>Changes to output.</td>
</tr>
<tr>
<td>set transaction isolation level</td>
<td>New isolation level option: 0</td>
</tr>
<tr>
<td>update</td>
<td>New clauses allow you to specify the index, cache strategy, and I/O size for a query.</td>
</tr>
</tbody>
</table>

New set options

Table 6-3 summarizes the new set options:
Table 6-3: New set options

<table>
<thead>
<tr>
<th>Option</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>forceplan</td>
<td>Forces the optimizer to perform joins in the order in which tables are named in the from clause of a query.</td>
</tr>
<tr>
<td>prefetch</td>
<td>Disables or enables large I/O (data prefetch) for a session.</td>
</tr>
<tr>
<td>statistics</td>
<td>Prints statistics about the use of the subquery cache during the execution of a subquery.</td>
</tr>
<tr>
<td>subquerycache</td>
<td></td>
</tr>
<tr>
<td>table count</td>
<td>integer Specifies the number of tables that are to be optimized at the same time in a join query.</td>
</tr>
</tbody>
</table>

New system procedures

The following new system procedures provide some of the functionality for the features described earlier:

Table 6-4: New system procedures

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>sp_bindcache</td>
<td>Binds databases, tables, indexes, or text or image chains to data caches.</td>
</tr>
<tr>
<td>sp_cacheconfig</td>
<td>Configures named data caches and provides information about caches.</td>
</tr>
<tr>
<td>sp_cachestrategy</td>
<td>Enables and disables caching strategies and large I/O for specific tables and indexes.</td>
</tr>
<tr>
<td>sp_chgattribute</td>
<td>Changes the max_rows_per_page value for future space allocations of a table or index.</td>
</tr>
<tr>
<td>sp_displaylevel</td>
<td>Sets and displays a user’s display level. The display level determines which SQL Server configuration parameters are displayed in sp_configure output.</td>
</tr>
<tr>
<td>sp_dropglockpromote</td>
<td>Removes lock promotion values from a table or database.</td>
</tr>
<tr>
<td>sp_helpcache</td>
<td>Provides information about cache overhead requirements and provides information about caches and cache bindings.</td>
</tr>
<tr>
<td>sp_helppartition</td>
<td>Lists the first page and the control page for each partition of a partitioned table.</td>
</tr>
<tr>
<td>sp_logiosize</td>
<td>Changes the log I/O size used by SQL Server to a different memory pool when doing I/O for the transaction log of the current database.</td>
</tr>
<tr>
<td>sp_poolconfig</td>
<td>Configures pools within named caches to enable large I/O.</td>
</tr>
<tr>
<td>sp_procmode</td>
<td>Reports the subquery processing mode of an object.</td>
</tr>
<tr>
<td>sp_setpglockpromote</td>
<td>Sets or changes the lock promotion thresholds for a database, table, or for SQL Server.</td>
</tr>
<tr>
<td>sp_unbindcache</td>
<td>Unbinds a specific database, table, index, or text or image object from a data cache.</td>
</tr>
<tr>
<td>sp_unbindcache_all</td>
<td>Unbinds all objects bound to a cache.</td>
</tr>
</tbody>
</table>
Changes to system procedures

The following system procedures have been changed in this release:

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>sp_dboption</td>
<td>Has a new database option, identity in nonunique index, that allows you to create a unique index by including an IDENTITY column in the index key.</td>
</tr>
<tr>
<td>sp_configure</td>
<td>Includes new parameters and new optional subcommands, and displays new output.</td>
</tr>
</tbody>
</table>

New system tables

The following system tables are new in this release:

<table>
<thead>
<tr>
<th>Table</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>sysattributes</td>
<td>Defines attributes for objects such as databases, tables, indexes, users, logins, and procedures.</td>
</tr>
<tr>
<td>syslogshold</td>
<td>Stores information about the oldest active transaction and the Replication Server truncation point for each database</td>
</tr>
<tr>
<td>syspartitions</td>
<td>Stores internal information about table partitions.</td>
</tr>
</tbody>
</table>

Changes to existing system tables

Table 6-7 lists system tables that have been changed in this release.

<table>
<thead>
<tr>
<th>Table</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>sysindexes</td>
<td>Changed column: rowpage to maxrowsperpage</td>
</tr>
<tr>
<td></td>
<td>The root column entry for a table becomes obsolete when the table is partitioned. Root pages for individual partition are derived from information in the new syspartitions table.</td>
</tr>
<tr>
<td>sysconfigures</td>
<td>Added new columns:</td>
</tr>
<tr>
<td></td>
<td>• name contains the configuration parameter name</td>
</tr>
<tr>
<td></td>
<td>• parent contains the group to which the parameter belongs</td>
</tr>
<tr>
<td></td>
<td>• value2 is used for configuration parameters whose values are character strings</td>
</tr>
<tr>
<td></td>
<td>• value3 is not currently used</td>
</tr>
</tbody>
</table>
Changes that may affect existing applications

<table>
<thead>
<tr>
<th>Table</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>syscurconfigs</td>
<td>Added new columns:</td>
</tr>
<tr>
<td></td>
<td>• value2 is used for configuration parameters whose values are character strings</td>
</tr>
<tr>
<td></td>
<td>• defvalue contains default values for configuration parameters</td>
</tr>
<tr>
<td></td>
<td>• minimum_value contains the minimum legal value for a configuration parameter</td>
</tr>
<tr>
<td></td>
<td>• maximum_value contains the maximum legal value for a configuration parameter</td>
</tr>
<tr>
<td></td>
<td>• memory_used contains the per-unit amount of memory used by a configuration parameter</td>
</tr>
<tr>
<td></td>
<td>• display_level contains the display level associated with a configuration parameter</td>
</tr>
<tr>
<td></td>
<td>• datatype and message_num are for internal use</td>
</tr>
<tr>
<td>sysdatabases</td>
<td>Added new column:</td>
</tr>
<tr>
<td></td>
<td>• offline status bit is used by the upgrading database dumps feature</td>
</tr>
</tbody>
</table>

Changes that may affect existing applications

This section describes the system changes introduced in release 11.0 that may affect your applications if you are upgrading to release 11.5 from a pre-11.0 release. Topics covered are as follows:

• New Transact-SQL keywords in release 11.0
• Changes to SQL Server configuration
• Subquery changes
• Changes to showplan output in release 11.0
• New caching strategies may affect performance
• Upgrading database dumps
• Partitions and physical data placement

New Transact-SQL keywords in release 11.0

The following keywords are new reserved words in SQL Server 11.0. They cannot be used as object names or column names.

• errordata
• max_rows_per_page

132
• online
• partition
• unpartition

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

Changes to SQL Server configuration

The following changes to the SQL Server configuration interface will affect existing applications.

The reconfigure command

In previous releases, changes made with sp_configure had to be activated with the reconfigure command. This is no longer required. reconfigure is no longer operational. Existing scripts should continue to run, but should be changed at your earliest convenience because reconfigure may not be supported in future releases.

buildmaster -r no longer supported

The buildmaster -r flag, which was used to rewrite the configuration block with default values for SQL Server configuration parameters, is no longer supported.

To run SQL Server with its built-in default values:
1 Rename, move, or delete your configuration file.
2 Shut down and restart SQL Server.

When you do this, SQL Server uses the built-in default configuration values, and creates a configuration file server_name.cfg in the directory from which SQL Server was started. However, these values do not get written to sysconfigures. In order to have them written to sysconfigures, you need to shut down and restart SQL Server once again.
Changes that may affect existing applications

New names for existing configuration parameters

The following SQL Server configuration parameters (formerly called “configuration variables”) have new names:

Table 6-8: New configuration parameter names for release 11.0

<table>
<thead>
<tr>
<th>Old Name</th>
<th>New Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1204 (trace flag)</td>
<td>print deadlock information</td>
</tr>
<tr>
<td>T1603 (trace flag)</td>
<td>allow sql server async i/o</td>
</tr>
<tr>
<td>T1610 (trace flag)</td>
<td>tcp no delay</td>
</tr>
<tr>
<td>T1611 (trace flag)</td>
<td>lock shared memory</td>
</tr>
<tr>
<td>allow updates</td>
<td>allow updates to system tables</td>
</tr>
<tr>
<td>calignment</td>
<td>memory alignment boundary</td>
</tr>
<tr>
<td>cclkrate</td>
<td>sql server clock tick length</td>
</tr>
<tr>
<td>cfgcprot</td>
<td>permission cache entries</td>
</tr>
<tr>
<td>cguardsz</td>
<td>stack guard size</td>
</tr>
<tr>
<td>cindextrips</td>
<td>number of index trips</td>
</tr>
<tr>
<td>cmmaxnetworks</td>
<td>max number network listeners</td>
</tr>
<tr>
<td>cmmaxscheds</td>
<td>i/o polling process count</td>
</tr>
<tr>
<td>cnalarm</td>
<td>number of alarms</td>
</tr>
<tr>
<td>cnbikio</td>
<td>disk i/o structures</td>
</tr>
<tr>
<td>cnlanguinfo</td>
<td>number of languages in cache</td>
</tr>
<tr>
<td>cnmaxaio_engine</td>
<td>max async i/o per engine</td>
</tr>
<tr>
<td>cnmaxaio_server</td>
<td>max async i/o per server</td>
</tr>
<tr>
<td>cnmailbox</td>
<td>number of mailboxes</td>
</tr>
<tr>
<td>cnmsg</td>
<td>number of messages</td>
</tr>
<tr>
<td>coamtrips</td>
<td>number of oam trips</td>
</tr>
<tr>
<td>cpreallocext</td>
<td>number of pre-allocated extents</td>
</tr>
<tr>
<td>cpu flush</td>
<td>cpu accounting flush interval</td>
</tr>
<tr>
<td>cschedspins</td>
<td>runnable process search count</td>
</tr>
<tr>
<td>csortbufsize</td>
<td>number of sort buffers</td>
</tr>
<tr>
<td>csortpgcount</td>
<td>sort page count (Removed in release 11.5.)</td>
</tr>
<tr>
<td>ctimemax</td>
<td>cpu grace time</td>
</tr>
<tr>
<td>database size</td>
<td>default database size</td>
</tr>
<tr>
<td>default language</td>
<td>default language id</td>
</tr>
<tr>
<td>devices</td>
<td>number of devices</td>
</tr>
<tr>
<td>extent i/o buffers</td>
<td>number of extent i/o buffers (Removed in 11.5.)</td>
</tr>
<tr>
<td>fillfactor</td>
<td>default fill factor percent</td>
</tr>
<tr>
<td>i/o flush</td>
<td>i/o accounting flush interval</td>
</tr>
</tbody>
</table>
New configuration parameters

Release 11.0 introduced a number of new configuration parameters listed in Table 6-9. See Chapter 11, “Setting Configuration Parameters,” in the *System Administration Guide* for a complete description of these parameters.

<table>
<thead>
<tr>
<th>Old Name</th>
<th>New Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>language in cache</td>
<td>number of languages in cache</td>
</tr>
<tr>
<td>locks</td>
<td>number of locks</td>
</tr>
<tr>
<td>maximum network packet size</td>
<td>max network packet size</td>
</tr>
<tr>
<td>memory</td>
<td>total memory</td>
</tr>
<tr>
<td>mstart</td>
<td>shared memory starting address</td>
</tr>
<tr>
<td>nested trigger</td>
<td>allow nested triggers</td>
</tr>
<tr>
<td>open databases</td>
<td>number of open databases</td>
</tr>
<tr>
<td>open objects</td>
<td>number of open objects</td>
</tr>
<tr>
<td>password expiration interval</td>
<td>systemwide password expiration</td>
</tr>
<tr>
<td>pre-read packets</td>
<td>remote server pre-read packets</td>
</tr>
<tr>
<td>procedure cache</td>
<td>procedure cache percent</td>
</tr>
<tr>
<td>recovery flags</td>
<td>print recovery information</td>
</tr>
<tr>
<td>recovery interval</td>
<td>recovery interval in minutes</td>
</tr>
<tr>
<td>remote access</td>
<td>allow remote access</td>
</tr>
<tr>
<td>remote connections</td>
<td>number of remote connections</td>
</tr>
<tr>
<td>remote logins</td>
<td>number of remote logins</td>
</tr>
<tr>
<td>remote sites</td>
<td>number of remote sites</td>
</tr>
<tr>
<td>sql server code size</td>
<td>executable code size</td>
</tr>
<tr>
<td>tape retention</td>
<td>tape retention in days</td>
</tr>
<tr>
<td>user connections</td>
<td>number of user connections</td>
</tr>
</tbody>
</table>
Changes that may affect existing applications

Table 6-9: New SQL Server configuration parameters for release 11.0

<table>
<thead>
<tr>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>address lock spinlock ratio</td>
</tr>
<tr>
<td>configuration file</td>
</tr>
<tr>
<td>deadlock checking period</td>
</tr>
<tr>
<td>deadlock retries</td>
</tr>
<tr>
<td>event buffers per engine</td>
</tr>
<tr>
<td>freelock transfer block size</td>
</tr>
<tr>
<td>housekeeper free write percent</td>
</tr>
<tr>
<td>identity grab size</td>
</tr>
<tr>
<td>lock promotion HWM</td>
</tr>
<tr>
<td>lock promotion LWM</td>
</tr>
<tr>
<td>lock promotion PCT</td>
</tr>
<tr>
<td>max engine freelocks</td>
</tr>
<tr>
<td>o/s async i/o enabled</td>
</tr>
<tr>
<td>o/s file descriptors</td>
</tr>
<tr>
<td>page lock spinlock ratio</td>
</tr>
<tr>
<td>page utilization percent</td>
</tr>
<tr>
<td>partition groups</td>
</tr>
<tr>
<td>partition spinlock ratio</td>
</tr>
<tr>
<td>size of auto identity column</td>
</tr>
<tr>
<td>table lock spinlock ratio</td>
</tr>
<tr>
<td>total data cache size</td>
</tr>
<tr>
<td>user lock cache size</td>
</tr>
<tr>
<td>user lock cache spinlock ratio</td>
</tr>
</tbody>
</table>

New deadlock checking period parameter

Previous releases of SQL Server performed deadlock checking when the process began to wait for a lock. In release 11.0, SQL Server performs deadlock checking for a process only after it has waited a minimum of 500 ms. for a lock to be released. To return to the previous way of deadlock checking, set deadlock checking period to 0.

New page utilization percent parameter

The default behavior of previous releases of SQL Server is to search the OAM page chain for unused pages before allocating a new extent. To keep this behavior, set the page utilization percent to 100.
Compiled object sizes have grown

The size of compiled objects is larger in System 11, and was significantly larger in System 10. You may need to resize your procedure cache to maintain the same performance.

SQL Server code size has grown

More memory is used for the kernel and other internal structures. You may need to add more memory to maintain the same performance as your previous release.

Subquery changes

Several problems relating to subqueries have been fixed in SQL Server release 11.0. This section describes old and new behavior in detail. Examine applications that use subqueries to determine if your application relies on behavior that has been corrected.

Subquery processing in previous releases of SQL Server used “inside-out” processing. Release 11.0 subqueries are processed “outside-in” for greater efficiency. Upgrading to release 11.0 does not automatically change the processing mode of the subquery.

After upgrading to release 11.0, you must drop and re-create objects to take advantage of the new processing style. Use sp_procmode to identify objects that include subqueries. After upgrading to SQL Server release 11.0, you cannot create an object that uses release 10.0 processing.

It is best to test procedures that contain subqueries and make any changes that are necessary for your application before upgrading your production system.

Different results

Subqueries in objects created in SQL Server release 11.0 may return different results than subqueries in objects created on an earlier SQL Server release. To test performance after upgrade, create the objects in SQL Server release 11.0, test the results and performance, and make any changes that are necessary for your application.
Changes that may affect existing applications

Determining query processing mode

A new system procedure, `sp_procqmode`, reports on the subquery processing mode of an object.

Duplicates in subquery results

As of release 10.0, subqueries using `in` or any predicates no longer return duplicates.

`set dup_in_subquery on` was provided as an upgrade path to SQL Server release 10.0. It is no longer supported. If your application depends on the duplicate rows in the result set, rewrite the subquery as a join.

For example:

```
select a from s where b in
  (select c from t)
```

would become:

```
select a from s, b where s.b = t.c
```

Procedures created in SQL Server release 10.0 that use `dup_in_subquery` will continue to run until the procedure is dropped and re-created in release 11.0.

Changes in subquery restrictions

The following subquery restrictions have been removed:

- You can use `distinct with group by` in a subquery.

  The `distinct` keyword suppressed duplicate rows in the output of pre-release 10.0 subqueries. This keyword is no longer necessary, and is ignored, in subqueries.

- You can reference a correlated variable in the select list of a subquery. For example, the following expression subquery is now supported:

```
select t1.c1 from t1
  where t1.c2 = (select t2.c2 + t1.c3 from t2)
```

The following subquery restriction has been added:

- There is a maximum of 16 subqueries within a single side of a union.
Handling NULL results

Prior to release 11.0, a correlated expression subquery in the set clause of an update returned 0 instead of NULL when there were no matching rows. Release 11.0 correctly returns NULL when there are no matching rows, and an error is raised if the column does not permit nulls.

If you have applications that depend on the pre-release 11.0 behavior, you will need to rewrite them.

For example, the following trigger tries to update a column that does not permit NULL values:

```sql
update t1
    set c1 = (select max(c1)
              from inserted where t1.c2 = inserted.c2)
```

The correct trigger is:

```sql
update t1
    set c1 = (select isnull(max(c1), 0)
              from inserted
              where t1.c2 = inserted.c2)
```

or:

```sql
update t1
    set c1 = (select max(c1)
              from inserted
              where t1.c2 = inserted.c2)
    where exists (select * from inserted
                  where t1.c2 = inserted.c2)
```

The `where` clause updates table `t1.c1` to 0, if the subquery does not return any correlation values from the outer table `t1`.

Another example of this is the `totalsales_trig` in the `pubs2` sample database. In previous versions, the trigger read as follows:

```sql
create trigger totalsales_trig
    on salesdetail
    for insert, update, delete
    as
    /******** Save processing: return if there are no rows affected ********/
    if @@rowcount = 0
        begin
            return
        end
    /******** add all the new values ********/
```
Changes that may affect existing applications

/* use isnull: a null value in the titles table means
   "no sales yet" not "sales unknown"
*/
update titles
   set total_sales = isnull(total_sales, 0) +
      (select sum(qty)
       from inserted
       where titles.title_id = inserted.title_id)

/* remove all values being deleted or updated */
update titles
   set total_sales = isnull(total_sales, 0) -
      (select sum(qty)
       from deleted
       where titles.title_id = deleted.title_id)

sum(qty) is NULL if no row is returned from the table, so when a statement
changes the total_sales column, the trigger changes to NULL all the rows in
titles that do not qualify.

To guarantee that the subquery in the expression for the update returns a non-
null value, the corrected trigger is:

create trigger totalsales_trig
on salesdetail
for insert, update, delete
as
   /* Save processing: return if there are no rows affected */
   if @@rowcount = 0
      begin
         return
      end

   /* add all the new values */
   /* use isnull: a null value in the titles table means
      "no sales yet" not "sales unknown"
   */
   update titles
      set total_sales = isnull(total_sales, 0) +
         (select sum(qty)
          from inserted
          where titles.title_id = inserted.title_id)
      where title_id in (select title_id from inserted)

   /* remove all values being deleted or updated */
   update titles
      set total_sales = isnull(total_sales, 0) -
(select sum(qty)
    from deleted
    where titles.title_id = deleted.title_id)
where title_id in (select title_id from deleted)

Improved performance

Most subqueries will perform better after being dropped and re-created.

However, an expression subquery containing an aggregate where the outer table is very large and has few duplicate correlation values, and the inner table is small, may not perform as well in release 11.0. For example:

```sql
select * from huge_table where x=
    (select sum(a) from tiny_table
    where b = huge_table.y)
```

The workaround is:

```sql
select huge_table.y, s=sum(a)
    into #t
    from huge_table, tiny_table
    where b=huge_table.y
    group by huge_table.y

select huge_table.*
    from huge_table, #t
    where x=#t.x
    and huge_table.y=#t.y
```

Changes to showplan output in release 11.0

The output for the set option showplan has changed. If you have applications that use this generated output, they need to be changed to use the new output.

See Chapter 9, “Understanding Query Plans” in the Performance and Tuning Guide.
Changes that may affect existing applications

New caching strategies may affect performance

The optimizer may now choose a new caching strategy called “fetch and discard” strategy. Standard caching strategy reads pages in at the head of the MRU/LRU (most recently used/least recently used) chain in cache. This flushes other pages out of cache. Fetch-and-discard (or MRU) strategy reads pages into cache much closer to the end of the cache, so it does not flush other pages. But these pages remain in cache a much shorter time, so subsequent user queries that might get a “cache hit” by finding these pages in cache under the standard strategy will be less likely to find the page in cache.

When the optimizer estimates that a particular query will read a significant number of pages, and only need those pages once during a query, it may choose fetch and discard strategy. If your disk I/O increases in release 11.0, check the I/O strategy for queries using showplan to see the cache strategy in use. If the queries are using fetch-and-discard (MRU) strategy, you can add a clause to select, update, and delete commands to specify standard caching.


Upgrading database dumps

If you use scripts to perform database loads using the load database and load transaction commands, the scripts will fail unless you include the online database command at the end of your load sequence. You should also remove the following sp_dboption options from your scripts: no chkpt on recovery, dbo use only, and read only. These options are no longer needed with release 11.0, as load database sets the database status to offline, thus making it inaccessible by users.


Partitions and physical data placement

Note Release 11.5 adds the ability to create clustered indexes on partitioned tables and to access the tables in parallel. See the Performance and Tuning Guide for more information.
In release 10.0, a heap table’s data was always inserted at the end of a single page chain. This meant that a heap table (for example, a history table) would physically store its newest entries at the end of the page chain, and its oldest entries at the beginning. A simple `SELECT` statement or cursor scan against such a table could return rows in roughly the same order in which they were inserted.

In release 11.0 you can create partitioned tables that have multiple page chains. Inserts into partitioned tables can occur at the ends of many separate page chains. Data in such a table is not physically stored in sequential order. If you want to view rows from a partitioned table in sequential order, use the `ORDER BY` clause in your `SELECT` statement.

See the *Performance and Tuning Guide* for more information about partitions.
Changes that may affect existing applications
Index

Symbols
@@textcolid global variable 128
@@textdbid global variable 128
@@textobjid global variable 128
@@textptr global variable 128
@@textts global variable 128

A
abstract plans for queries 41
access control 9
Adaptive Server
  changes that may affect applications for 12.5 25
column size 3
  external file system support 8
  logical page sizes 2
  number of groups 4
  number of logins 3
  number of users 3
  SQLJ 5
  unichar support 11
  version 12.5 relaxed limits 2
  wide columns and data truncation 26
address locks 120
alter role command 52
alter table command 52
  enabling or disabling triggers 42
  modifying an existing tables schema 34
alter table command 18, 52
ANSI joins 40
application execution precedence, system procedures 87
assigning names to clients 35
at isolation clause 119
auditing, system procedures for 76
auto identity, sp_dboption 127

B
Backup Server
copying data in parallel 95
dump operations 33
high performance backup and restore 33
non-rewinding tapes 33
tape device determination 126
Bourne shell xiii
buildmaster -y utility 122

C
C shell xiii
cache strategies 113
case expression 77
changing identity gaps 36
check password for digit configuration parameter 51
checkpoints, housekeeper task and 122
clientappname command 19, 53
clienthostname command 53
clientname command 53
clients, assigning names to 35
column size 3
Component Integration Services (CIS) 43, 44, 78
compressed archive support 10
configname options for sp_helpconfig 36
configuration changes 122, 133
corruption 99
costing for queries 39
CPU utilization, housekeeper task and 122
create role command 52
create table command 36, 52
create table command 52
creating objects 50
.cshrc file xiii
cursors
  compilation 71
  optimization 71
Index

partitioned tables and partitioned tables and 142

D
data caches 112
data manipulation languages 9
data storage, max_rows_per_page and database
  for debugging 36
  integrity, ensuring 33
  object owners, tracking 50
  suspending 35
data types
  enhancements for datatypes 41
  java classes as 38
dbcc checkdb 117
dbcc checktable 117
dbcc command 53
dbcc engine command 35
dbcc tune 122
deadlock checking 120
delete command, tracking creator of 50
delete, new clauses for 114
direct updates 125
Directory control layer (DCL) 45
Directory Services 86
disabling triggers 42
disk init command 34, 53
disk reinit command 34, 53
Distributed Transaction Management 43
dsync command 34
dtm detach timeout period configuration parameter 51
dtm lock timeout period configuration parameter 51
dump compression
  defined 10
  syntax example 10
dump database command 100
  compress option 10
dump operations and Backup Server 33
dump transaction command 100, 118
  compress option 10
dumps, database, upgrading 125, 142
duplicate subquery results 138

e
 e-mail messages 107
 enable cis configuration parameter 52
 enable dtm configuration parameter 51
 enable HA configuration parameter 51
 enable java configuration parameter 51
 enable sort-merge joins and JTC configuration parameter 39
 enable xact coordination configuration parameter 51
 enabling triggers 42
 engine affinity 87
 engine freelock lists 121
 engines offline 35
 enhancements
  for datatypes 41
  enhancements for queries 39
 error log 88
 ESPs 89
 execute command 41, 53
 execution
  attributes 87
  objects 87
  of Transact-SQL commands 41
  precedence 86
  system procedures for 87
 expressions, maximum length 3
 Extended Stored Procedures. See ESPs
 external file system support 8
 extractjava (UNIX) utility 56
 extrjava (NT) utility 56

F
 Failover, using in a high availability system 30–31, 51
 files
  .cshrc xiii
  .profile xiii
 fillfactor, compared to max_rows_per_page 115
 freelock lists 121
 functions, Java 37

G
 global freelock lists 121
global variables
  @@textcolid 128
  @@textdbid 128
  @@textobjid 128
  @@textptr 128
  @@textts 128
granting permission for proxy authorization 98
groups, number of for 12.5 4

H
hash tables 120
heap tables 116
High availability, using Sybase's Failover 30–31, 51
high performance backup and restore 33
housekeeper task
  free writes and 122

I
IDENTITY columns, changes for 11.0 127
identity gap parameter 36
identity gaps, changing 36
identity in nonunique index
  sp_dboption 127
image datatypes 41
image global variables 128
impersonating another user 98
indexes, rebuilding 32
in-place updates 125
insert command, tracking creator of 50
insert, new clauses for 114
installjava (UNIX) utility 56
instjava (NT) utility 56
integrity
  database 33
  message 46
Interfaces file 45
Isolation levels 119

capabilities 37
classes as datatypes 38
datatypes 38
in Adaptive Server Enterprise 37–38
installing in Adaptive Server 5
user-defined functions 37
join 18, 52
joins
  merged 39
  using ANSI syntax 40

K
keywords
  new for release 11.0 132
  new for release 11.5 59
  new for release 12.5 24
Korn shell xiii

L
large I/O 113
LDAP, specifying in libtcl.cfg 7
libtcl*.cfg file 7
licensed feature 30, 37, 43
limits on tables in queries 38
load transaction command 97
lock escalation 123
lock manager 119
log audit logon failure configuration parameter 88
log audit logon success configuration parameter 88
log I/O size 114
logical page sizes 2
Logical Process Manager 86
login security, user-defined 46
logins
  number of for 12.5 3
  server 50
  unified 46

M
managing distributed transactions 43
Index

max_rows_per_page
  compared to fillfactor 115
maximum failed logins configuration parameter 51
memory pools 114
merge joins 39
message integrity 46
metadata caches 93
Microsoft Distributed Transaction Coordinator (MSDTC) 43
minimum password length configuration parameter 51
Monitor Access to Executing SQL 94
  configuring amount of text saved in memory 94
  monitor access to executing SQL 94
monitoring from Windows NT Performance Monitor 108
multiple network engines 123

N
  named data caches 112
  named time ranges 101, 102
network
    I/O engines 123
    security 45
non-rewinding tapes for Backup Server 33
number of aux scan descriptors configuration parameter 93
number of dtx participants configuration parameter 51
number of groups 4
number of logins 3
number of users 3

O
  OAM pages 115
  objects, creating 50
  oldest active transactions 118
  onfiguring caches 112
  online database command
    usage 126, 129, 142
  Open Client and Adaptive Server’s limits 26
  optimization for queries 39
  order by clause 142

P
  page allocation 115
  page locks 120
  page utilization percent 136
  parallel bulk copy
    and partitioned tables 95
    and the -F and -L flags 96
    copying in data in parallel 95
  parallel query processing 96
  parameter hierarchy 123
  parameters, name changes of 134
  partitioned tables 116
  data placement and 142
  performance
    application tradeoffs 86
    of network security 45
  plans, query 41
  point-in-time recovery 97
  pools 114
  procedural language functions, calling 89
  .profile file xiii
  proxy authorization
    granting permission for 98
    value for applications 98
    value for users 98

Q
  queries
    abstract plans for 41
    changes to 125
    enhancements to 39
    improvements in costing 39
    limiting with sp_add_resource_limit 101
    number of tables in 38
  query processing
    mode 138
    parallel 96
  query tuning options 114
  quiesce database command 9, 35, 52

R
raw partitions 33
Index

rebuilding indexes 32
reconfigure command 133
recovery 97, 99
recovery time, housekeeper task and 122
referential integrity enhancements 92
remote servers, connecting with Component Integration Services 78
remove java command 52
reorg rebuild command 32
reserved words
  new for release 11.0 132
  new for version 12.0 59
resource limits 101
rows per page 115

S
security
  network 45
  user-defined login 46
segments, partitions and 117
select command 40
  new clauses 114, 119
select into command 36
server user name and ID 50
set command 35
  clientapplname 19, 53
  clienthostname 53
  clientname 53
set dup_in_subquery 138
set forceplan 114
set options overview 129
set prefetch 114
set proxy command 98
set session authorization command 98
set showplan command 124
set table count 114
set transaction isolation level 119
shells xiii
showplan improvements 124
size of global fixed heap configuration parameter 51
size of global fixed heap configuration parameter 51
size of process object fixed heap configuration parameter 51
size of shared class heap configuration parameter 51
size of shared class heap configuration parameter 51
SMP systems 123
sp_addlogin system procedure 55
sp_addresssystem procedure 88
sp_altermessage system procedure 88
sp_bindcache 112, 130
sp_cacheconfig 112, 130
sp_cachestrategy 114, 130
sp_chgattribute 130
sp_chgattribute system procedure 36
sp_companion system procedure 54
sp_configure 122, 131, 136
  address lock spinlock ratio 120
  deadlock checking period 120, 136
  freelock transfer block size 121
  identity grab size 127
  lock promotion 123
  max engine freelocks 121
  page lock spinlock ratio 120
  page utilization percent 115
  table lock spinlock ratio 120
  user log cache size 117
  user log cache spinlock ratio 118
sp_configure command 41
sp_dboption 131
  auto identity 127
  identity in nonunique index 127
sp_deviceattr system procedure 34, 54
sp_displaylevel 130
sp_displaylogin system procedure 55
sp_displayroles system procedure 55
sp_droplockpromote 130
sp_ha_admin system procedure 54
sp_help system procedure 36
sp_helpartition 117, 130
sp_helpcache 112, 130
sp_helpconfig system procedure 36
sp_helppdevice system procedure 34, 55
sp_helppjava system procedure 54
sp_helpjava system procedure 54
sp_lock system procedure 55
sp_logiosize 114, 130
sp_modifysql system procedure 55

149
Index

sp_monitorconfig system procedure 55
sp_poolconfig system procedure 112, 130
sp_procqmode 130, 138
sp_setpglockpromote 123, 130
sp_transactions system procedure 54
sp_unbindcache 112, 130
sp_unbindcache_all 130
sp_who system procedure 20, 55
spinlocks 118, 120
SQL Perfmon Integration configuration parameter 108
SQLJ in Adaptive Server 5
SSL, defined 7
status column, syssrvroles 57
store procedures, maximum length 3
strict_dtm_enforcement configuration parameter 51
strict_dtm_enforcement command 53
subqueries 137, 139
  changes to 125
  NULL results 139
  restrictions 138
suid (server user ID), object creator and 50
support for UNIX file systems 33
suspect pages 99, 100
suspending databases 35
sybdiagdb database 36
Sybmail 107
syntax conventions xiii
sysattributes system table 57, 131
syscolumns system table 56
syscolumns system table 56
sysconfigures system table 57, 131
syscoordinations system table 56
syscurconfigures system table 132
sysdatabases system table 132
sysdevices system table 56
syservroles system table 57
sysindexes system table 131
sysjars system table 56
sysjars system table 56
syslocks system table 56
syslogins system table 57
syslogshold system table 118, 131
syspartition system table 131
sysprocesses system table 56
sysroutines system table 57
sysroutines system table 57

System procedures
sp_bindcache 130
sp_cacheconfig 130
sp_cachestrategy 130
sp_chgattribute 130
sp_config 122, 131
sp_dboption 131
sp_displaylevel 130
sp_dropglockpromote 130
sp_helppartition 130
sp_helppcache 130
sp_logiosize 114, 130
sp_poolconfig 130
sp_procqmode 130, 138
sp_setpglockpromote 123, 130
sp_unbindcache 130
sp_unbindcache_all 130
systransactions system table 57
sysusers system table 57
sysxtypes system table 57
sysxtypes system table 57

T

table locks 120
table scans, partitioned tables and 142
tables, number of, in a query 38
taking engines offline 35
tape device determination 126
text global variables 128
text prefetch size configuration parameter 41, 51
time ranges 101, 102
titles system table 41
transaction log 114, 117
transactional_rpc command 53
Transactions
  limiting with sp_add_resource_limit 101
transactions
  managing distributed 43
two-phase commit 104
Transact-SQL, dynamic execution of 41
triggers
  disabling 42
enabling 42
NULL results 139
truncation points 118
two-phase commit 104
txn to pss ratio configuration parameter 51

U
Unichar support 11
unified login 46
union operator in a view 6
UNIX file system support 33
update command tracking creator of 50
user log caches 117
user-defined
   caches 112
   functions for Java 37
   login security 46
users, number of for 12.5 3
Uupdate changes in subqueries 125

V
variables, maximum length 3
views, using union operators 6

W
wide column 26
Windows NT Event Log 88
Windows NT Mail account 107
Windows NT Mail profile 107
Windows NT Performance Monitor 108

X
X/Open XA protocol 43
xact coordination level configuration parameter 51
XP Server 90
   priority 92
xp_cmdshell configuration parameter 92