SYBASE®

Reference Manual: Tables

Adaptive Server® Enterprise

12.5.1

DOCUMENT ID: DC36274-01-1251-02

LAST REVISED: November 2003

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

Contents

About This Book

The *Adaptive Server Reference Manual* includes four guides to Sybase® Adaptive Server® Enterprise and the Transact-SQL® language:

- *Building Blocks* describes the "parts" of Transact-SQL: datatypes, built-in functions, global variables, expressions and identifiers, reserved words, and SQLSTATE errors. Before you can use Transact-SQL sucessfully, you need to understand what these building blocks do and how they affect the results of Transact-SQL statements.
- *Commands* provides reference information about the Transact-SQL commands, which you use to create statements.
- *Procedures* provides reference information about system procedures, catalog stored procedures, extended stored procedures, and dbcc stored procedures. All procedures are created using Transact-SQL statements.
- *Tables* provides reference information about the system tables, which store information about your server, databases, users, and other details of your server. It also provides information about the tables in the dbccdb and dbccalt databases.

Audience

The *Adaptive Server Reference Manual* is intended as a reference tool for Transact-SQL users of all levels.

How to use this book

- Chapter 1, "System Tables," contains information about all of the system tables in the master database, the auditing database, and in any user databases (such as pubs2). Also described in this chapter is syblicenseslog, which is not a system database, but continues information about licenses.
- Chapter 2, "dbccdb Tables," contains information about the tables in the dbccdb and dbccalt databases.

Related documents

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

- The release bulletin for your platform – contains last-minute information that was too late to be included in the books.

A more recent version of the release bulletin may be available on the World Wide Web. To check for critical product or document information that was added after the release of the product CD, use the Sybase Technical Library.

- The *Installation Guide* for your platform – describes installation, upgrade, and configuration procedures for all Adaptive Server and related Sybase products.
- *What's New in Adaptive Server Enterprise?* – describes the new features in Adaptive Server version 12.5.1, the system changes added to support those features, and the changes that may affect your existing applications.
- *ASE Replicator User's Guide* – describes how to use the ASE Replicator feature of Adaptive Server to implement basic replication from a primary server to one or more remote Adaptive Servers.
- *Component Integration Services User's Guide* – explains how to use the Adaptive Server Component Integration Services feature to connect remote Sybase and non-Sybase databases.
- *Configuring Adaptive Server Enterprise* for your platform – provides instructions for performing specific configuration tasks for Adaptive Server.
- *EJB Server User's Guide* – explains how to use EJB Server to deploy and execute Enterprise JavaBeans in Adaptive Server.
- *Error Messages and Troubleshooting Guide* – explains how to resolve frequently occurring error messages and describes solutions to system problems frequently encountered by users.
- *Full-Text Search Specialty Data Store User's Guide* – describes how to use the Full-Text Search feature with Verity to search Adaptive Server Enterprise data.
- *Glossary* – defines technical terms used in the Adaptive Server documentation.
- *Historical Server User's Guide* – describes how to use Historical Server to obtain performance information for SQL Server® and Adaptive Server.
- *Java in Adaptive Server Enterprise* – describes how to install and use Java classes as data types, functions, and stored procedures in the Adaptive Server database.

- *Job Scheduler User's Guide* – provides instructions on how to install and configure, and create and schedule jobs on a local or remote Adaptive Server using the command line or a graphical user interface (GUI).
- *Monitor Client Library Programmer's Guide* – describes how to write Monitor Client Library applications that access Adaptive Server performance data.
- *Monitor Server User's Guide* – describes how to use Monitor Server to obtain performance statistics from SQL Server and Adaptive Server.
- *Performance and Tuning Guide* – is a series of four books that explains how to tune Adaptive Server for maximum performance:
  - *Basics* – the basics for understanding and investigating performance questions in Adaptive Server.
  - *Locking* – describes how the various locking schemas can be used for improving performance in Adaptive Server.
  - *Optimizer and Abstract Plans* – describes how the optimizer processes queries and how abstract plans can be used to change some of the optimizer plans.
  - *Monitoring and Analyzing* – explains how statistics are obtained and used for monitoring and optimizing performance.
- *Quick Reference Guide* – provides a comprehensive listing of the names and syntax for commands, functions, system procedures, extended system procedures, datatypes, and utilities in a pocket-sized book.
- *Reference Manual* – is a series of four books that contains the following detailed Transact-SQL® information:
  - *Building Blocks* – Transact-SQL datatypes, functions, global variables, expressions, identifiers and wildcards, and reserved words.
  - *Commands* – Transact-SQL commands.
  - *Procedures* – Transact-SQL system procedures, catalog stored procedures, system extended stored procedures, and `dbcc` stored procedures.
  - *Tables* – Transact-SQL system tables and `dbcc` tables.

- *System Administration Guide* – provides in-depth information about administering servers and databases. This manual includes instructions and guidelines for managing physical resources, security, user and system databases, and specifying character conversion, international language, and sort order settings.
- *System Tables Diagram* – illustrates system tables and their entity relationships in a poster format. Available only in print version.
- *Transact-SQL User's Guide* – documents Transact-SQL, Sybase's enhanced version of the relational database language. This manual serves as a textbook for beginning users of the database management system. This manual also contains descriptions of the pubs2 and pubs3 sample databases.
- *Using Adaptive Server Distributed Transaction Management Features* – explains how to configure, use, and troubleshoot Adaptive Server DTM features in distributed transaction processing environments.
- *Using Sybase Failover in a High Availability System* – provides instructions for using Sybase's Failover to configure an Adaptive Server as a companion server in a high availability system.
- *Utility Guide* – documents the Adaptive Server utility programs, such as isql and bcp, which are executed at the operating system level.
- *Web Services User's Guide* – explains how to configure, use, and troubleshoot Web Services for Adaptive Server.
- *XA Interface Integration Guide for CICS, Encina, and TUXEDO* – provides instructions for using the Sybase DTM XA interface with X/Open XA transaction managers.
- *XML Services in Adaptive Server Enterprise* – describes the Sybase native XML processor and the Sybase Java-based XML support, introduces XML in the database, and documents the query and mapping functions that comprise XML Services.

Other sources of information

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

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

- The Technical Library CD contains product manuals and is included with your software. The DynaText reader (included on the Technical Library CD) allows you to access technical information about your product in an easy-to-use format.

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

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

  To access the Technical Library Product Manuals Web site, go to Product Manuals at http://www.sybase.com/support/manuals/.

Sybase certifications on the Web

Technical documentation at the Sybase Web site is updated frequently.

❖ **Finding the latest information on product certifications**

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

❖ **Creating a personalized view of the Sybase Web site (including support pages)**

Set up a MySybase profile. MySybase is a free service that allows you to create a personalized view of Sybase Web pages.

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

Sybase EBFs and software maintenance

❖ **Finding the latest information on EBFs and software maintenance**

1. Point your Web browser to the Sybase Support Page at http://www.sybase.com/support.

2 Select EBFs/Maintenance. Enter user name and password information, if prompted (for existing Web accounts) or create a new account (a free service).

3 Select a product.

4 Specify a time frame and click Go.

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

Conventions

The following sections describe conventions used in this manual.

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

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

Table 1: Font and syntax conventions for this manual

Element	Example
Command names, command options, utility names, utility options, and other keywords are in "command" font (Arial, 8 point).	select sp_configure
Database names, datatypes, file names and path names are in "database object" font (Arial, 8 point).	master database
Book names, file names, variables, and path names are in italics.	*System Administration Guide* *sql.ini* file *column_name* *$SYBASE/ASE* directory
Variables, or words that stand for values that you fill in, are in "variable" font (Italics).	select *column_name* from *table_name* where *search_conditions*
Type parentheses as part of the command.	compute *row_aggregate* (*column_name*)
Double colon, equals sign indicates that the syntax is written in BNF notation. Do not type this symbol. Indicates "is defined as".	::=
Curly braces mean that you must choose at least one of the enclosed options. Do not type the braces.	{cash, check, credit}

Element	Example
Brackets mean that to choose one or more of the enclosed options is optional. Do not type the brackets.	[cash | check | credit]
The comma means you may choose as many of the options shown as you want. Separate your choices with commas as part of the command.	cash, check, credit
The pipe or vertical bar (|) means you may select only one of the options shown.	cash | check | credit
An ellipsis (...) means that you can repeat the last unit as many times as you like.	buy thing = price [cash | check | credit] [, thing = price [cash | check | credit]]... You must buy at least one thing and give its price. You may choose a method of payment: one of the items enclosed in square brackets. You may also choose to buy additional things: as many of them as you like. For each thing you buy, give its name, its price, and (optionally) a method of payment.

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

  sp_dropdevice [*device_name*]

  For a command with more options:

  select *column_name*
  from *table_name*
  where *search_conditions*

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

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

  ```
  select * from publishers
  ```

- Examples of output from the computer appear as follows:

```
pub_id    pub_name                city          state
-------   ---------------------   -----------   -----
0736      New Age Books           Boston        MA
0877      Binnet & Hardley        Washington    DC
1389      Algodata Infosystems    Berkeley      CA

(3 rows affected)
```

In this manual, most of the examples are in lowercase. However, you can disregard case when typing Transact-SQL keywords. For example, SELECT, Select, and select are the same.

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

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

System Tables

System tables are tables supplied by Sybase.

The topics in this chapter include:

Locations of system tables

System tables may be located in:

- The master database,
- The sybsecurity database,
- The sybsystemdb database, or
- All databases.

Most tables in the master database are system tables. Some of these tables also occur in user databases. They are automatically created when the create database command is issued.

System tables in *master*

The following system tables occur *only* in the master database:

System table	Contents
syscharsets	One row for each character set or sort order
sysconfigures	One row for each configuration parameter that can be set by users
syscurconfigs	Information about configuration parameters currently being used by Adaptive Server
sysdatabases	One row for each database on Adaptive Server
sysdevices	One row for each tape dump device, disk dump device, disk for databases, and disk partition for databases

System table	Contents
sysengines	One row for each Adaptive Server engine currently online
syslanguages	One row for each language (except U.S. English) known to the server
syslisteners	One row for each type of network connection used by current Adaptive Server
syslocks	Information about active locks
sysloginroles	One row for each server login that possesses a system role
syslogins	One row for each valid Adaptive Server user account
syslogshold	Information about the oldest active transaction and the Replication Server® truncation point for each database
sysmessages	One row for each system error or warning
sysmonitors	One row for each monitor counter
sysprocesses	Information about server processes
sysremotelogins	One row for each remote user
sysresourcelimits	One row for each resource limit
syssecmechs	Information about the security services available for each security mechanism that is available to Adaptive Server
sysservers	One row for each remote Adaptive Server
syssessions	Only used when Adaptive Server is configured for Sybase's Failover in a high availability system. syssessions contains one row for each client that connects to Adaptive Server with the failover property (for example, isql -Q)
syssrvroles	One row for each server-wide role
systimeranges	One row for each named time range
systransactions	One row for each transaction
sysusages	One row for each disk piece allocated to a database

System tables in *sybsecurity*

The following system tables occur *only* in the sybsecurity database:

System Table	Contents
sysauditoptions	One row for each global audit option
sysaudits_01 – sysaudits_08	The audit trail. Each audit table contains one row for each audit record

System table in *sybsystemdb*

The following system table occurs *only* in the sybsystemdb database:

System Table	Contents
syscoordinations	One row for each remote participant of a distributed transaction

System tables in all databases

The following system tables occur in all databases:

System table	Contents
sysalternates	One row for each Adaptive Server user mapped to a database user
sysattributes	One row for each object attribute definition
syscolumns	One row for each column in a table or view, and for each parameter in a procedure
syscomments	One or more rows for each view, rule, default, trigger, and procedure, giving SQL definition statement
sysconstraints	One row for each referential and check constraint associated with a table or column
sysdepends	One row for each procedure, view, or table that is referenced by a procedure, view, or trigger
sysgams	Allocation bitmaps for an entire database
sysindexes	One row for each clustered or nonclustered index, one row for each table with no indexes, and an additional row for each table containing text or image data
sysjars	One row for each Java archive (JAR) file that is retained in the database. Uses row-level locking
syskeys	One row for each primary, foreign, or common key; set by user (not maintained by Adaptive Server)
syslogs	Transaction log
sysobjects	One row for each table, view, procedure, rule, trigger default, log, and (in tempdb only) temporary object
syspartitions	One row for each partition (page chain) of a partitioned table
sysprocedures	One row for each view, rule, default, trigger, and procedure, giving internal definition
sysprotects	User permissions information
sysqueryplans	Abstract query plans and SQL text
sysreferences	One row for each referential integrity constraint declared on a table or column
sysroles	Maps server-wide roles to local database groups
syssegments	One row for each segment (named collection of disk pieces)
sysstatistics	One or more rows for each indexed column on a user table. May also contain rows for unindexed column
systabstats	One row for each table, plus one row for each nonclustered index
systhresholds	One row for each threshold defined for the database
systypes	One row for each system-supplied and user-defined datatype
sysusermessages	One row for each user-defined message

System table	Contents
sysusers	One row for each user allowed in the database
sysxtypes	One row for each extended, Java-SQL datatype. Uses row-level locking

About the *sybdiagdb* database

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

About the *syblicenseslog* table

The syblicenseslog table is described in syblicenseslog on page 95. It is not technically a system table, but you may need to consult it for license information related to shutting down Adaptive Server.

Rules for using system tables

This section describes rules, restrictions and usage information for system tables.

Permissions on system tables

Permissions for use of the system tables can be controlled by the database owner, just like permissions on any other tables. By default, when Adaptive Server is installed, the installmodel script grants select access to "public" (all users) for most system tables and for most fields in the tables. However, no access is given for some system tables, such as systhresholds, and no access is given for certain fields in other system tables. For example, all users, by default, can select all columns of sysobjects except audflags. To determine the current permissions for a particular system table, execute:

sp_helprotect *system_table_name*

For example, to check the permissions of systhresholds in my_database, execute:

```
use my_database
go
sp_helprotect systhresholds
go
```

Locking schemes used for system tables

Unless noted otherwise, system tables use allpages locking.

Reserved columns

The word “reserved” in the column description means that the column is not currently used by Adaptive Server.

Updating system tables

All direct updates on system tables are by default not allowed —even for the database owner. Instead, Adaptive Server supplies system procedures to make any normally needed updates and additions to system tables.

You can allow direct updates to the system tables if it becomes necessary to modify them in a way that cannot be accomplished with a system procedure. To accomplish this, a System Security Officer must reset the configuration parameter called allow updates to system tables with the system procedure sp_configure. For more information, see the *System Administration Guide*.

There are entries in some of the master database tables that should not be altered by any user under any circumstances. For example, do not attempt to modify syslogs with a delete, update, or insert command. In addition, an attempt to delete all rows from syslogs will put Adaptive Server into an infinite loop that eventually fills up the entire database.

Triggers on system tables

You cannot create triggers on system tables. If you try to create a trigger on a system table, Adaptive Server returns an error message and cancels the trigger.

Aggregate functions and virtual tables

Aggregate functions cannot be used on virtual tables such as syslocks and sysprocesses.

sysalternates

All databases

Description

sysalternates contains one row for each Adaptive Server user mapped (or aliased) to a user of the current database. When a user tries to access a database, Adaptive Server looks for a valid uid entry in sysusers. If none is found, it looks in sysalternates.suid. If the user's suid is found there, he or she is treated as the database user whose suid is listed in sysalternates.altsuid.

On the Adaptive Server distribution media, there are no entries in sysalternates.

Columns

The columns for sysalternates are:

Name	Datatype	Description
suid	int	Server user ID of user being mapped
altsuid	int	Server user ID of user to whom another user is mapped

Indexes

Unique clustered index On suid.

sysattributes

All databases

Description

System attributes define properties of objects such as databases, tables, indexes, users, logins, and procedures. sysattributes contains one row for each of an object's attribute definitions (configured by various system procedures). master..sysattributes defines the complete set of valid attribute values and classes for Adaptive Server as a whole. It also stores attribute definitions for server-wide objects, such as databases and logins.

sysattributes should only be accessed indirectly using system procedures. The permissions required for modifying sysattributes depend on the system procedure you use.

Columns

The columns for sysattributes are:

Name	Datatype	Description
class	smallint	The attribute class ID. This describes the category of the attribute. In master..sysattributes, the special class 1 identifies all valid attributes for Adaptive Server. Class 0 identifies valid *classes* of attributes.
attribute	smallint	The attribute ID.
object_type	char(2)	A one- or two-letter character ID that defines the type of object to associate with the attribute.
object_cinfo	varchar(30) null	A string identifier for the object (for example, the name of an application). This field is not used by all attributes.
object	int null	The object identifier. This may be an object ID, user ID, or database ID, depending on the type of object. If the object is a part of a table (for example, an index), then this column contains the object ID of the associated table.
object_info1	int null	Defines additional information required to identify the object. This field is not used by all attributes. The contents of this field depend on the attribute that is defined.
object_info2	int null	Defines additional information required to identify the object. This field is not used by all attributes. The contents of this field depend on the attribute that is defined.
object_info3	int null	Defines additional information required to identify the object. This field is not used by all attributes. The contents of this field depend on the attribute that is defined.
int_value	int null	An integer value for the attribute (for example, the display level of a user).
char_value	varchar(255) null	A character value for the attribute (for example, a cache name).
text_value	text null	A text value for the attribute.
image_value	image null	An image value for the attribute.
comments	varchar(255) null	Comments or additional information about the attribute definition.

Table 1-1 shows the representation of temporary database groups and bindings as these appear in the sysattributes system table. Only relevant columns are shown. All other columns are NULL. Groups are represented in rows where attribute has a value of "0". Login and application bindings, as well as database to group bindings, are represented in rows where attribute has a value of "1". See Chapter 12, "Multiple Temporary Databases" in *Performance and Tuning: Optimizer* for more information about multiple temporary databases.

Table 1-1: sysattributes representation

class	attribute	object_type	object_cinfo	object	object_cinfo1	int_value	char_value
16	0	GR	group name	NULL	NULL	group id	NULL
16	0	D	database name	group ID	NULL	NULL	NULL
16	1	LG	NULL	user ID	0 for soft, 1 for hard	0 for database, 1 for group	database or group name
16	1	AP	application name	NULL	0 for soft, 1 for hard	0 for database, 1 for group	database or group name

Indexes

Unique clustered index On class, attribute, object_type, object, object_info1, object_info2, object_info3, object_cinfo.

Nonclustered index On object_type, object, object_info1, object_info2, object_info3, object_cinfo.

sysauditoptions

sybsecurity database

Description

sysauditoptions contains one row for each server-wide audit option and indicates the current setting for that option. Other types of auditing option settings are stored in other tables. For example, database-specific option settings are stored in sysdatabases, and object-specific option settings are stored in sysobjects. The default value for each option is 0, or "off." sysauditoptions can be accessed only by System Security Officers.

Columns

The columns for sysauditoptions are:

Name	Datatype	Description
num	smallint	Number of the server-wide option.
val	smallint	Current value; one of the following: 0 = off 1 = pass 2 = fail 3 = on
minval	smallint	Minimum valid value for this option.
maxval	smallint	Maximum valid value for this option.
name	varchar(30)	Name of option.
sval	varchar(30)	String equivalent of the current value: for example, "on", "off", "nonfatal".
comment	varchar(255)	Description of option.

sysaudits_01 – sysaudits_08

sybsecurity database

Description

These system tables contain the audit trail. Only one table at a time is active. The active table is determined by the value of the current audit table configuration parameter. An installation can have up to eight audit tables. For example, if your installation has three audit tables, the tables are named sysaudits_01, sysaudits_02, and sysaudits_03. An audit table contains one row for each audit record.

Columns

The columns for sysaudits_01 – sysaudits_08 are:

Name	Datatype	Description
event	smallint	Type of event being audited. See Table 1-3 on page 12.
eventmod	smallint	Further information about the event. Possible values are: 0 = no modifier for this event 1 = the event passed permission checking 2 = the event failed permission checking
spid	smallint	Server process ID of the process that caused the audit record to be written.
eventtime	datetime	Date and time of the audited event.
sequence	smallint	Sequence number of the record within a single event; some events require more than one audit record.
suid	smallint	Server login ID of the user who performed the audited event.
dbid	int null	Database ID in which the audited event occurred or the object/stored procedure/trigger resides, depending on the type of event.
objid	int null	ID of the accessed object or stored procedure/trigger.
xactid	binary(6) null	ID of the transaction containing the audited event. For a multi-database transaction, this is the transaction ID from the database where the transaction originated.
loginname	varchar(30) null	Login name corresponding to the *suid*.
dbname	varchar(30) null	Database name corresponding to the *dbid*.
objname	varchar(30) null	Object name corresponding to the *objid*.
objowner	varchar(30) null	Name of the owner of *objid*.
extrainfo	varchar(255) null	Additional information about the audited event. This field contains a sequence of items separated by semicolons. See Table 1-2.

The extrainfo column contains a sequence of items separated by semicolons as shown in Table 1-2:

Table 1-2: Items in the extrainfo column

Item	Contents
Roles	Lists the roles that are active. The roles are separated by blanks.

Item	Contents
Subcommand	The name of the subcommand or command option that was used for the event. For example, for the alter table command, the options add column or drop constraint might be used. Multiple subcommands or options are separated by commas.
Previous value	The value prior to the update if the event resulted in the update of a value.
Current value	The new value if the event resulted in the update of a value.
Other information	Additional security-relevant information that is recorded for the event.
Proxy information	The original login name, if the event occurred while a set proxy was in effect.
Principal information	The principal name from the underlying security mechanism, if the user's login is the secure default login, and the user logged into Adaptive Server via unified login. The value of this field is NULL, if the secure default login is not being used.

An example of an extrainfo column for the security-relevant event of changing an auditing configuration parameter might be:

```
sso_role;suspend auditing when full;1;0;;;;
```

This extrainfo column indicates that a System Security Officer changed the configuration parameter suspend auditing when full from 1 (suspend all processes that involve an auditing event) to 0 (truncate the next audit table and make it the current audit table). The other columns in the audit record give other pertinent information. For example, the record contains the server user id (suid) and the login name (loginname).

The event column values that pertain to each audit event are listed in Table 1-3.

Table 1-3: Values in event and extrainfo column

Event	Audit option	Command or access audited	extrainfo
1	adhoc	User-defined audit record	extrainfo is filled by the *text* parameter of sp_addauditrecord
2	alter	alter database	• *Roles* – Current active roles • *Subcommand*– ALTER SIZE • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if a set proxy is in effect

Event	Audit option	Command or access audited	extrainfo
3	alter	alter table	• *Roles* – Current active roles • *Subcommand* – ADD COLUMN, REPLACE COLUMN, ADD CONSTRAINT, or DROP CONSTRAINT • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if a set proxy is in effect
4	bcp	bcp in	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if set proxy in effect
6	bind	sp_bindefault	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* – Name of default • *Proxy information* – Original login name, if set proxy in effect
7	bind	sp_bindmsg	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* – Message ID • *Proxy information* – Original login name, if set proxy in effect
8	bind	sp_bindrule	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* – Name of the rule • *Proxy information* – Original login name, if set proxy in effect

Event	Audit option	Command or access audited	extrainfo
9	create	create database	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if set proxy in effect
10	create	create table	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if set proxy in effect
11	create	create procedure	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if set proxy in effect
12	create	create trigger	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if set proxy in effect
13	create	create rule	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if set proxy in effect

Event	Audit option	Command or access audited	extrainfo
14	create	create default	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Other information* – NULL • *Current value* – NULL • *Proxy information* – Original login name, if set proxy in effect
15	create	sp_addmessage	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* – Message Number • *Proxy information* – Original login name, if set proxy in effect
16	create	create view	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if set proxy in effect
17	dbaccess	Any access to the database by any user	• *Roles* – Current active roles • *Subcommand* – USE CMD or OUTSIDE REFERENCE • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if set proxy in effect
18	delete	delete from a table	• *Roles* – Current active roles • *Subcommand* – DELETE • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if set proxy in effect

Event	Audit option	Command or access audited	extrainfo
19	delete	delete from a view	• *Roles* – Current active roles • *Subcommand* – DELETE • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if set proxy in effect
20	disk	disk init	• *Roles* – Current active roles • *Subcommand* – disk init • *Previous value* – NULL • *Current value* – NULL • *Other information* – Name of the disk • *Proxy information* – Original login name, if set proxy in effect
21	disk	disk refit	• *Roles* – Current active roles • *Subcommand* – disk refit • *Previous value* – NULL • *Current value* – NULL • *Other information* – Name of the disk • *Proxy information* – Original login name, if set proxy in effect
22	disk	disk reinit	• *Roles* – Current active roles • *Subcommand* – disk reinit • *Previous value* – NULL • *Current value* – NULL • *Other information* – Name of the disk • *Proxy information* – Original login name, if set proxy in effect
23	disk	disk mirror	• *Roles* – Current active roles • *Subcommand* – disk mirror • *Previous value* – NULL • *Current value* – NULL • *Other information* – Name of the disk • *Proxy information* – Original login name, if set proxy in effect

Event	Audit option	Command or access audited	extrainfo
24	disk	disk unmirror	• *Roles* – Current active roles • *Subcommand* – disk unmirror • *Previous value* – NULL • *Current value* – NULL • *Other information* – Name of the disk • *Proxy information* – Original login name, if set proxy in effect
25	disk	disk remirror	• *Roles* – Current active roles • *Subcommand* – disk remirror • *Previous value* – NULL • *Current value* – NULL • *Other information* – Name of the disk • *Proxy information* – Original login name, if set proxy in effect
26	drop	drop database	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if set proxy in effect
27	drop	drop table	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if set proxy in effect
28	drop	drop procedure	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if set proxy in effect

Event	Audit option	Command or access audited	extrainfo
29	drop	drop trigger	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if set proxy in effect
30	drop	drop rule	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if set proxy in effect
31	drop	drop default	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if set proxy in effect
32	drop	sp_dropmessage	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* – Message number • *Proxy information* – Original login name, if set proxy in effect
33	drop	drop view	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL
34	dump	dump database	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if set proxy in effect

Event	Audit option	Command or access audited	extrainfo
35	dump	dump transaction	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if set proxy in effect
36	errors	Fatal error	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* – Error number.Severity.State • *Proxy information* – Original login name, if set proxy in effect
37	errors	Non-fatal error	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* – Error number.Severity.State • *Proxy information* – Original login name, if set proxy in effect
38	exec_procedure	Execution of a procedure	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* – All input parameters • *Proxy information* – Original login name, if set proxy in effect
39	exec_trigger	Execution of a trigger	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if set proxy in effect

Event	Audit option	Command or access audited	extrainfo
40	grant	grant	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if set proxy in effect

41	insert	insert into a table	• *Roles* – Current active roles
			• *Subcommand*
			  • If insert – INSERT
			  • If select into – INSERT INTO followed by the fully qualified object name
			• *Previous value* – NULL
			• *Current value* – NULL
			• *Other information* – NULL
			• *Proxy information* – Original login name, if set proxy in effect

42	insert	insert into a view	• *Roles* – Current active roles • *Subcommand* – INSERT • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if a set proxy is in effect
43	load	load database	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if set proxy in effect
44	load	load transaction	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if set proxy in effect

Event	Audit option	Command or access audited	extrainfo
45	login	Any login to Adaptive Server	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* – Host name of the machine from which login was done • *Proxy information* – Original login name, if set proxy in effect
46	logout	Any logouts from Adaptive Server	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* – Host name of the machine from which login was done • *Proxy information* – Original login name, if set proxy in effect
47	revoke	revoke	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if set proxy in effect
48	rpc	Remote procedure call from another server	• *Roles* – Current active roles • *Subcommand* – Name of client program • *Previous value* – NULL • *Current value* – NULL • *Other information* – Server name, host name of the machine from which the RPC was done. • *Proxy information* – Original login name, if set proxy in effect
49	rpc	Remote procedure call to another server	• *Roles* – Current active roles • *Subcommand* – Procedure name • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if set proxy in effect

Event	Audit option	Command or access audited	extrainfo
50	security	Server start	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* -d*masterdevicename* -i*interfaces file path* -S*servername* -e*errorfilename* • *Proxy information* – Original login name, if set proxy in effect
51	security	Server shutdown	• *Roles* – Current active roles • *Subcommand* – shutdown • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if set proxy in effect
55	security	Role toggling	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – “on” or “off” • *Current value* – “on” or “off” • *Other information* – Name of the role being set • *Proxy information* – Original login name, if set proxy in effect
61	table_access	Table access	• *Roles* – Current active roles • *Subcommand* – SELECT, SELECT INTO, INSERT, UPDATE, DELETE, REFERENCE, READTEXT, or WRITETEXT • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if set proxy in effect
62	select	select from a table	• *Roles* – Current active roles • *Subcommand* – SELECT INTO, SELECT, or READTEXT • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if set proxy in effect

Event	Audit option	Command or access audited	extrainfo
63	select	select from a view	• *Roles* – Current active roles • *Subcommand* – SELECT, SELECT INTO, or READTEXT • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if set proxy in effect
64	truncate	truncate table	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if set proxy in effect
67	unbind	sp_unbindefault	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if set proxy in effect
68	unbind	sp_unbindrule	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if set proxy in effect
69	unbind	sp_unbindmsg	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if set proxy in effect

Event	Audit option	Command or access audited	extrainfo
70	update	update to a table	• *Roles* – Current active roles • *Subcommand* – UPDATE or WRITETEXT • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if set proxy in effect
71	update	update to a view	• *Roles* – Current active roles • *Subcommand* – UPDATE or WRITETEXT • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if set proxy in effect
73	**Note** This event is audited automatically. It is not controlled by an audit option.	Turning the auditing parameter on with sp_configure	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if set proxy in effect
74	**Note** This event is audited automatically. It is not controlled by an audit option.	Turning the auditing parameter off with sp_configure	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if set proxy in effect
76	security	Regeneration of a password by a System Security Officer (SSO)	• *Roles* – Current active roles • *Subcommand* – Setting SSO password • *Previous value* – NULL • *Current value* – NULL • *Other information* – Login name • *Proxy information* – Original login name, if set proxy in effect

Event	Audit option	Command or access audited	extrainfo
80	security	proc_role within a system procedure	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* – Required roles • *Proxy information* – Original login name, if set proxy in effect
81	dbcc	dbcc	• *Roles* – Current active roles • *Subcommand* – The dbcc subcommand name • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if set proxy in effect
82	security	sp_configure	• *Roles* – Current active roles • *Subcommand* – Name of the configuration parameter • *Previous value* – Old parameter value if command is setting a new value • *Current value* – New parameter value if command is setting a new value • *Other information* – Number of configuration parameter, if a parameter is being set; name of configuration file, if a configuration file is being used to set parameters • *Proxy information* – Original login name, if set proxy in effect
83	security	online database	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if set proxy in effect
84	setuser	setuser	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* – Name of the user being set • *Proxy information* – Original login name, if a set proxy is in effect

Event	Audit option	Command or access audited	extrainfo
85	func_obj_access, func_dbaccess	Accesses to objects and databases via Transact-SQL functions	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if set proxy in effect
85	security	valid_user	• *Roles* – Current active roles • *Subcommand* – valid_user • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if set proxy in effect
88	security	set proxy or set session authorization	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – Previous suid • *Current value* – New suid • *Other information* – NULL • *Proxy information* – Original login name, if set proxy or set session authorization had no parameters; otherwise, NULL.
92	cmdtxt	All actions of a particular user, or by users with a particular role	• *Roles* – Current active roles • *Subcommand* – NULL • *Previous value* – NULL • *Current value* – NULL • *Other information* – NULL • *Proxy information* – Original login name, if set proxy in effect

syscharsets

master database only

Description — syscharsets contains one row for each character set and sort order defined for use by Adaptive Server. One of the sort orders is marked in master..sysconfigures as the default sort order, which is the only one actually in use.

Columns — The columns for syscharsets are:

Name	Datatype	Description
type	smallint	The type of entity this row represents. Numbers from 1001 to 1999 represent character sets. Numbers from 2000 to 2999 represent sort orders.
id	tinyint	The ID for a character set or sort order. A sort order is defined by the combination of the sort order ID and the character set ID (csid). The character set is defined by id, which must be unique. Sybase reserves ID numbers 0–200.
csid	tinyint	If the row represents a character set, this field is unused. If the row represents a sort order, this is the ID of the character set that sort order is built on. A character set row with this ID must exist in this table.
status	smallint	Internal system status information bits.
name	varchar(30)	A unique name for the character set or sort order. Must contain only the 7-bit ASCII letters A-Z or a-z, digits 0-9, and underscores (_), and begin with a letter.
description	varchar(255)	An optional description of the features of the character set or sort order.
definition	image	The internal definition of the character set or sort order. The structure of the data in this field depends on the type.
sortfile	varchar(30) null	The name of the sort order file.

Indexes — **Unique clustered index** On id, csid, type

Unique nonclustered index On name

syscolumns

All databases

Description — syscolumns contains one row for every column in every table and view, and a row for each parameter in a procedure.

Columns — The columns for syscolumns are:

Name	Datatype	Description
id	int	ID of table to which this column belongs or of procedure with which this parameter is associated
number	smallint	Sub-procedure number when the procedure is grouped (0 for non-procedure entries)
colid	smallint	Column ID
status	tinyint	Bits 0–2 (values 1, 2, and 4) indicate bit positioning if the column uses the bit datatype. If the column uses the text/image datatype, bits 0 and 1 indicate replication status as follows: • 01 = always replicate • 10 = replicate only if changed • 00 = never replicate Bit 3 (value 8) indicates whether NULL values are legal in this column. Bit 4 (value 16) indicates whether more than one check constraint exists for the column. Bits 5 and 6 are used internally. Bit 7 (value 128) indicates an identity column. Bit 8 is unused.
type	tinyint	Physical storage type; copied from systypes
length	int	Physical length of data; copied from systypes or supplied by user
offset	smallint	Offset into the row where this column appears; if negative, this is a variable-length column
usertype	smallint	User type ID; copied from systypes
cdefault	int	ID of the procedure that generates default value for this column
domain	int	Constraint ID of the first rule or check constraint for this column
name	sysname	Column name
printfmt	varchar(255) null	Reserved
prec	tinyint null	Number of significant digits
scale	tinyint null	Number of digits to the right of the decimal point

Name	Datatype	Description
remote_type	int null	Maps local names to remote names. Required by the access methods of Component Integration Services to allow the software to pass native column datatype information in parameters to servers of class access_server.
remote_name	varchar(30) null	Maps local names to remote names. Required by the access methods of Component Integration Services to construct a query using the proper column names for a remote table.
xstatus	int	The status of a column with extended datatypes. The values are: • 0 = in row • 1 = off row xstatus is NULL for columns that do not have an extended datatype.
xtype	int	ID of the class. Used if a column in a table or a parameter in a procedure has a Java class as its datatype. When used, fields are not NULL, and the value of type is 0x39. Refer to *Java in Adaptive Server Enterprise* for more information.
xdbid	int	The database ID of the class. For system classes, the value is -1. Otherwise, the value is the current database ID. Used if a column in a table or a parameter in a procedure has a Java class as its datatype. Fields are not NULL, and the value of type is 0x39. Refer to *Java in Adaptive Server Enterprise* for more information.
accessrule	int null	The object ID of the access rule in sysprocedures. See "Row-level access control" in Chapter 11, "Managing User Permissions" of the *System Administration Guide* for more information.
status2	int	Indicates parameter mode of a SQLJ stored procedure, and the return type of a SQLJ function.

Indexes **Unique clustered index** On id, number, colid

syscomments

All databases

Description

syscomments contains entries for each view, rule, default, trigger, table constraint, and procedure. The text column contains the original definition statements. If the text column is longer than 255 bytes, the entries will span rows. Each object can occupy up to 65,025 rows.

Columns

The columns for syscomments are:

Name	Datatype	Description
id	int	Object ID to which this text applies
number	smallint	Sub-procedure number when the procedure is grouped (0 for non-procedure entries)
colid	smallint	Sequence of 255 rows for the object
texttype	smallint	0 for system-supplied comment (for views, rules, defaults, triggers, and procedures); 1 for user-supplied comment (users can add entries that describe an object or column)
language	smallint	Reserved
text	varchar(255) null	Actual text of SQL definition statement
colid2	smallint	Indicates next sequence of rows for the object (see colid above); object can have up to 255 sequences of 255 rows each
status	smallint null	

Note Do not delete the definition statements from the text column of syscomments. These statements are required for the Adaptive Server upgrade process. To encrypt a definition statement, run the system procedure sp_hidetext. To see if a statement created in release 11.5 or later was deleted, run sp_checksource. If the statement was deleted, you must either recreate the object that created the statement or reinstall the application that created the object, which will re-create the statement.

You can protect the text of a database object against unauthorized access by restricting select permission on the text column of the syscomments table to the owner of the object and the System Administrator. This restriction, which applies to direct access through select statements as well as access through stored procedures, is required in order to run Adaptive Server in the evaluated configuration. To enact this restriction, a System Security Officer must reset the parameter called allow select on syscomments.text column with the system procedure sp_configure. For information, see the *System Administration Guide*.

Indexes

Unique clustered index On id, number, icolid2, colid, texttype

sysconfigures

master database only

Description — sysconfigures contains one row for each configuration parameter that can be set by the user.

Columns — The columns for sysconfigures are:

Name	Datatype	Description
config	smallint	Configuration parameter number.
value	int	The user-modifiable value for the parameter with integer datatype. Its value is 0 for the parameters with character datatype.
comment	varchar(255)	Name of the configuration parameter.
status	int	Value that represents the type of configuration parameter. For details see Table 1-4.
name	varchar(80) null	Name of the configuration parameter (the same value as comment).
parent	smallint null	Configuration parameter number of the parent; if more than one parent, the additional parent numbers are stored in sysattributes.
value2	varchar(255) null	The user-modified value for the parameter with the character datatype. Its value is NULL for parameters with integer datatype. It is also used to store the pool size of a buffer pool.
value3	int null	Stores the wash size of a buffer pool.
value4	int null	Stores the asynchronous prefetch percents of a buffer pool, or -1 where an unspecified or default value.

Table 1-4 provides information about the status column.

Table 1-4: Status column description

Status type	Decimal	Hex	Description
CFG_NO_OPTIONS	0	0x0	Parameter has no options.
CFG_SYSTEM_OPTION	1	0x01	Parameter is a system option.
CFG_SYSTEM_GROUP	2	0x02	Parameter is a system group.
CFG_STATIC	4	0x04	Parameter is static.
CFG_DYNAMIC	8	0x08	Parameter is dynamic.
CFG_CALCULATED	16	0x10	Parameter is calculated.
CFG_READONLY	32	0x20	Parameter is readonly.
CFG_MEMORY_USED	64	0x40	Parameter consumes memory.
CFG_CONFIG_FILE	128	0x80	Parameter is externally visible.
CFG_SYSTEM_TAB	256	0x100	Parameter is only externally visible in system table.
CFG_EXTRAS_OPTION	512	0x200	Parameter is for CFG_EXTRAS not DS_CONFIG.
CFG_CFGBLK	1024	0x400	Parameter is stored in the configuration block.

Status type	Decimal	Hex	Description
CFG_CACHE_GROUP	2048	0x800	Parameter is a cache group.
CFG_CACHE_OPTION	4096	0x1000	Parameter is a cache option.
CFG_BUFFER_POOL_GROUP	8192	0x2000	Parameter is a buffer pool group.
CFG_BUFFER_POOL_OPTION	16384	0x4000	Parameter is a buffer pool option.
CFG_INTERNAL	32768	0x8000	Parameter is for internal use only.
CFG_FNOF_LPAGESIZE	65536	0x100000	Parameter entry depends on logical pagesize.

Indexes

Unique clustered index On name, parent, config

Nonclustered index On config, config

sysconstraints

All databases

Description

sysconstraints has one row for each referential constraint and check constraint associated with a table or column.

Whenever a user declares a new check constraint or referential constraint using create table or alter table, Adaptive Server inserts a row into the sysconstraints table. The row remains until a user executes alter table to drop the constraint. Dropping a table by executing drop table removes all rows associated with that table from the sysconstraints table.

Columns

The columns for sysconstraints are:

Name	Datatype	Description
colid	smallint	Column number in the table
spare1	tinyint	Unused
constrid	int	Object ID of the constraint
tableid	int	ID of the table on which the constraint is declared
error	int	Constraint specific error message
status	int	The type of constraint: • 0x0040 = a referential constraint • 0x0080 = a check constraint
spare2	int	Unused

Indexes

Clustered index On tableid, colid

Unique nonclustered index On constrid

syscoordinations

sybsystemdb database only

Description

syscoordinations contains information about remote Adaptive Servers participating in distributed transactions (remote participants) and their coordination states.

Columns

The columns for syscoordinations are:

Name	Datatype	Description
participant	smallint	Participant ID
starttime	datetime	Date the transaction started
coordtype	tinyint	Value indicating the coordination method or protocol in the systransactions table definition
owner	tinyint	Row owner (for internal use)
protocol	smallint	Reserved for internal use
state	smallint	Value indicating the current state of the remote participant (see Table 1-5)
bootcount	int	Reserved for internal use
dbid	smallint	Database ID at the start of the transaction.
logvers	tinyint	Reserved for internal use
spare	smallint	Reserved for internal use
status	tinyint	Reserved for internal use
xactkey	binary(14)	Unique Adaptive Server transaction key
gtrid	varchar(255)	Global transaction ID for distributed transactions coordinated by Adaptive Server (reserved for internal use)
partdata	varbinary(255)	Reserved for internal use
srvname	varchar(30)	Name of local server (null for remote servers)

Table 1-5 lists the values for the state column:

Table 1-5: syscoordinations state values

***state* value**	**Participant state**
1	Begun
4	Prepared
7	Committed
9	In Abort Tran

Indexes

Unique clustered index On xactkey, participant, owner

syscurconfigs

master database only

Description

syscurconfigs is built dynamically when queried. It contains an entry for each of the configuration parameters, as does sysconfigures, but with the current values rather than the default values. In addition, it contains four rows that describe the configuration structure.

Columns

The columns for syscurconfigs are:

Name	Datatype	Description
config	smallint	Configuration parameter number.
value	int	The current run value for the parameter with integer datatype. Its value is 0 for the parameters with character datatype.
comment	varchar(255)	Amount of memory used by each configuration parameter, represented in a string format. Values marked with a hash mark (#) share memory with other parameters.
status	int	Either of the following: • 1 – dynamic • 0 – parameter takes effect when Adaptive Server is restarted
value2	varchar(255)	The current run value for the parameter with the character datatype. Its value is NULL for parameters with the integer datatype.
defvalue	varchar(255)	Default value of the configuration parameter.
minimum_value	int	Minimum value of the configuration parameter.
maximum_value	int	Maximum value of the configuration parameter.
memory_used	int	Integer value for the amount of memory used by each configuration parameter.
display_level	int	Display level of the configuration parameter. The values are 1, 5, and 10.
datatype	int	Datatype of the configuration parameter.
message_num	varchar(20)	Message number of the sp_helpconfig message for this configuration parameter.
apf_percent	varchar(10)	The current run value for the asynchronous prefetch percent for a buffer pool. Valid only for rows that represent buffer pools.

Name	Datatype	Description
unit	varchar(20)	Unit of the parameter. Values are: • not applicable – parameter has no units • number – number of items • clock ticks – number of clock ticks • microseconds • milliseconds • seconds • minutes • hours • days • bytes • kilobytes • megabytes • memory pages (2k) • virtual pages (2k) • logical pages • percent • ratio • switch – a Boolean value • id – ID number • name • rows
type	varchar(10)	Specifies whether a configuration parameter is declared dynamic or static in its structure definition. Values are: • dynamic – takes effect immediately • static – takes effect after restarting Adaptive Server

sysdatabases

master database only

Description

sysdatabases contains one row for each database in Adaptive Server. When Adaptive Server is installed, sysdatabases contains entries for the master database, the model database, the sybsystemprocs database, and the tempdb database. If you have installed auditing, it also contains an entry for the sybsecurity database.

Columns

The columns for sysdatabases are:

Name	Datatype	Description
name	sysname	Name of the database
dbid	smallint	Database ID
suid	int	Server user ID of database owner
status	smallint	Control bits; those that the user can set with sp_dboption are so indicated in Table 1-6
version	smallint	Unused
logptr	int	Pointer to transaction log
crdate	datetime	Creation date
dumptrdate	datetime	Date of the last dump transaction
status2	intn	Additional control bits. See Table 1-7
audflags	intn	Audit settings for database
deftabaud	intn	Bit-mask that defines default audit settings for tables
defvwaud	intn	Bit-mask that defines default audit settings for views
defpraud	intn	Bit-mask that defines default audit settings for stored procedures
def_remote_type	smallint	Identifies the default object type to be used for remote tables if no storage location is provided via the stored procedure sp_addobjectdef
def_remote_loc	varchar(255)	Identifies the default storage location to be used for remote tables if no storage location is provided via the stored procedure sp_addobjectdef
status3	intn	Additional control bits.
status4	intn	Additional control bits.

Table 1-6 lists the bit representations for the status column.

Table 1-6: Status control bits in the sysdatabases table

Decimal	Hex	Status
1	0x01	Upgrade started on this database
2	0x02	Upgrade has been successful
4	0x04	• select into/bulkcopy • Can be set by user
8	0x08	• trunc log on chkpt • Can be set by user
16	0x10	• no chkpt on recovery • Can be set by user
32	0x20	Database created with for load option, or crashed while loading database, instructs recovery not to proceed
64	0x04	Recovery started for all databases to be recovered
256	0x100	• Database suspect • Not recovered • Cannot be opened or used • Can be dropped only with dbcc dbrepair
512	0x200	• ddl in tran • Can be set by user
1024	0x400	• read only • Can be set by user
2048	0x800	• dbo use only • Can be set by user
4096	0x1000	• single user • Can be set by user
8192	0x2000	• allow nulls by default • Can be set by user

Table 1-7 lists the bit representations for the status2 column.

Table 1-7: status2 control bits in the sysdatabases table

Decimal	Hex	Status
1	0x0001	abort tran on log full; can be set by user
2	0x0002	no free space acctg; can be set by user
4	0x0004	auto identity; can be set by user
8	0x0008	identity in nonunique index; can be set by user
16	0x0010	Database is offline

† If this bit appears after recovery completes, the server may be under-configured for open databases. Use sp_configure to increase this parameter.

Decimal	Hex	Status
32	0x0020	Database is offline until recovery completes
64	0x0040	The table will have an auto identity feature, and a unique constraint on the identity column.
128	0x0080	Database has suspect pages
256†	0x0100	Table structure written to disk
512	0x0200	Database is in the process of being upgraded
1024	0x0400	Database brought online for standby access
2048	0x0800	When set by the user, prevents cross-database access via an alias mechanism
-32768	0xFFFF8000	Database has some portion of the log which is not on a log-only device

† If this bit appears after recovery completes, the server may be under-configured for open databases. Use sp_configure to increase this parameter.

Table 1-8 lists the bit representations for the status3 column.

Table 1-8: status3 control bits in the sysdatabases table

Decimal	Hex	Status
1	0x0001	Database is a user-created proxy database
2	0x0002	Database is an HA-created proxy database.
4	0x0004	Database has an HA-created proxy database.
8	0x0008	Disallow access to the database, since database is being shut down.
16	0x0010	Database is a failed-over database.
32	0x0020	Database is a mounted database of the type master.
64	0x0040	Database is a mounted database.
128	0x0080	Writes to the database are blocked by the quiesce database command.
256	0x0100	User-created tempdb.
512	0x0200	Disallow external access to database in the server in failed-over state.
1024	0x0400	Ued for user-provided option to enable or disable asynchronous logging service threads. User enables this through sp_dboption enbale async logging service option set to true on a particular database.
4096	0x1000	Database has been shut down successfully.
8192	0x2000	A drop database is in progress.

Indexes

Unique clustered index On name

Unique nonclustered index On dbid

sysdepends

All databases

Description

sysdepends contains one row for each procedure, view, or table that is referenced by a procedure, view, or trigger.

Columns

The columns for sysdepends are:

Name	Datatype	Description
id	int	Object ID
number	smallint	Procedure number
depid	int	Dependent object ID
depnumber	smallint	Dependent procedure number
status	smallint	Internal status information
selall	bit	On if object is used in select * statement
resultobj	bit	On if object is being updated
readobj	bit	On if object is being read

Indexes

Unique clustered index On id, number, depid, depnumber

sysdevices

master database only

Description

sysdevices contains one row for each tape dump device, disk dump device, disk for databases, and disk partition for databases. There are four entries in sysdevices in the Adaptive Server distribution media: one for the master device (for databases), one for a disk dump device, and two for tape dump devices.

Columns

The columns for sysdevices are:

Name	Datatype	Description
low	int	*Not used for dump devices* – First virtual page number on database device
high	int	Last virtual page number on database device or dump device
status	smallint	Bitmap indicating type of device, default and mirror status. See Table 1-9.
cntrltype	smallint	Controller type: • database device=0 • disk dump device or streaming tape=2 • tape dump device=3–8
name	sysname	Logical name of dump device or database device
phyname	varchar(127)	Name of physical device
mirrorname	varchar(127)	Name of mirror device

The bit representations for the status column, shown below, are additive. For example, "3" indicates a physical disk that is also a default.

Table 1-9: Bit representations for the status column.

Decimal	Hex	Status
1	0x01	Default disk
2	0x02	Physical disk
4	0x04	*Not used* – Logical disk
8	0x08	Skip header
16	0x10	Dump device
32	0x20	Serial writes
64	0x40	Device mirrored
128	0x80	Reads mirrored
256	0x100	Secondary mirror side only
512	0x200	Mirror enabled
1024	0x400	Master device is mirrored
2048	0x800	*Used internally* – Mirror disabled
4096	0x1000	*Used internally* – Primary device needs to be unmirrored

Decimal	Hex	Status
8192	0x2000	*Used internally* – Secondary device needs to be unmirrored
16384	0x4000	UNIX file device uses dsync setting (writes occur directly to physical media)

Indexes **Unique clustered index** On name

sysengines

master database only

Description

sysengines contains one row for each Adaptive Server engine currently online.

Columns

The columns for sysengines are:

Name	Datatype	Description
engine	smallint	Engine number
osprocid	int	Operating system process ID (may be NULL)
osprocname	char(32)	Operating system process name (may be NULL)
status	char	One of: online, in offline, in create, in destroy, debug, bad status
affinitied	int	Number of Adaptive Server processes with affinity to this engine
cur_kpid	int	Kernel process ID of process currently running on this engine, if any
last_kpid	int	Kernel process ID of process that previously ran on this engine
idle_1	tinyint	Reserved
idle_2	tinyint	Reserved
idle_3	tinyint	Reserved
idle_4	tinyint	Reserved
starttime	datetime	Date and time engine came online
status	char(12)	Internal system status information.

sysgams

All databases

Description

sysgams stores the global allocation map (GAM) for the database. The GAM stores a bitmap for all allocation units of a database, with one bit per allocation unit. You cannot select from or view sysgams.

sysindexes

All databases

Description — sysindexes contains one row for each clustered index, one row for each nonclustered index, one row for each table that has no clustered index, and one row for each table that contains text or image columns.

Columns — The columns for sysindexes are:

Name	Datatype	Description
name	sysname	Index or table name
id	int	ID of a table, or ID of table to which index belongs
indid	smallint	• 0 if a table • 1 if a clustered index on an allpages-locked table • >1 if a nonclustered index or a clustered index on a data-only-locked table • 255 if text, image or Java off-row structure (LOB structure)
doampg	int	Page number for the object allocation map of a table
ioampg	int	Page number for the allocation map of an index or (LOB structure)
oampgtrips	int	Number of times OAM pages cycle in the cache without being re-used, before being flushed
status2	smallint	Internal system status information (see Table 1-11)
ipgtrips	int	Number of times index pages cycle in the cache, without being reused, before being flushed
first	int	If indid is 0 or 1, page number of the first data page. If indid is between 2 and 250, page number of first leaf-level index page.
root	int	If indid is 0 and table is an unpartitioned allpages-locked table, page number of last page of page chain; unused for other types of pages. If indid is between 1 and 250, page number of root of index tree.
distribution	int	Unused. Formerly used to store the page number of the distribution page for an index.
usagecnt	smallint	Reserved
segment	smallint	Number of segment in which object resides
status	smallint	Internal system status information (see Table 1-11)
maxrowsperpage	smallint	Maximum number of rows per page
minlen	smallint	Minimum size of a row
maxlen	smallint	Maximum size of a row
maxirow	smallint	Maximum size of a non-leaf index row
keycnt	smallint	Number of keys for a clustered index on an allpages-locked table; number of keys, plus 1 for all other indexes
keys1	varbinary(255)	Description of key columns if entry is an index

Name	Datatype	Description
keys2	varbinary(255)	Description of key columns if entry is an index
soid	tinyint	Sort order ID that the index was created with; 0 if there is no character data in the keys
csid	tinyint	Character set ID that the index was created with; 0 if there is no character data in the keys
base_partition	int	Partition number, incremented by alter table...unpartition commands
fill_factor	smallint	Value for the fillfactor of a table set with sp_chgattribute
res_page_gap	smallint	Value for the reservepagegap on a table
exp_rowsize	smallint	Expected size of data rows
keys3	varbinary(255)	Description of key columns if entry is an index
identitygap	intn	Identity gap for a table
crdate	datetime	Creation date

Table 1-10 lists the bit representations for the status column.

Table 1-10: Status bits in the sysindexes table status column

Decimal	Hex	Status
1	0x1	Abort current command or trigger if attempt to insert duplicate key
2	0x2	Unique index
4	0x4	Abort current command or trigger if attempt to insert duplicate row; always 0 for data-only-locked tables
16	0x10	Clustered index
64	0x40	Index allows duplicate rows, if an allpages-locked table; always 0 for data-only-locked tables
128	0x80	Sorted object; not set for tables without clustered indexes or for text objects
512	0x200	sorted data option used in create index statement
2048	0x800	Index on primary key
32768	0x8000	Suspect index; index was created under another sort order

Table 1-11 lists the bit representations for the status2 column.

Table 1-11: Status bits in the sysindexes table status2 column

Decimal	Hex	Status
1	0x1	Index supports foreign key constraint
2	0x2	Index supports primary key/unique declarative constraint
4	0x4	Index includes an IDENTITY column
8	0x8	Constraint name not specified
16	0x10	Large I/Os (prefetch) not enabled for table, index, or text chain
32	0x20	MRU cache strategy not enabled for table, index, or text chain

Decimal	Hex	Status
64	0x40	Ascending inserts turned on for the table
256	0x0100	Index is presorted and does not need to be copied to new extents
512	0x0200	Table is a data-only-locked table with a clustered index
8192	0x2000	Index on a data-only-locked table is suspect

Indexes **Unique clustered index** On id, indid

sysjars

All databases

Description

sysjars contains one row for each Java archive (JAR) file that is retained in the database. Uses row-level locking.

For more information about JAR files, Java classes, and Java datatypes, see *Java in Adaptive Server Enterprise*.

Columns

The columns for sysjars are:

Name	Datatype	Description
sensitivity	sensitivity	Used by the Secure Adaptive Server.
jid	int	The ID of the JAR.
jstatus	int	Internal status information. Unused.
jname	varchar(255)	The JAR name.
jbinary	image	The contents of the JAR: the Java classes.

Indexes

Unique placement index On jid

Unique nonclustered index On jname

syskeys

All databases

Description

syskeys contains one row for each primary, foreign, or common key.

Columns

The columns for syskeys are:

Name	Datatype	Description
id	int	Object ID
type	smallint	Record type
depid	int	Dependent object ID
keycnt	int	Number of non-null keys
size	int	Reserved
key1 ... key8	smallint	Column ID
depkey1 ... depkey8	smallint	Column ID
spare1	smallint	Reserved

Indexes

Clustered index On id

syslanguages

master database only

Description

syslanguages contains one row for each language known to Adaptive Server. us_english is not in syslanguages, but it is always available to Adaptive Server.

Columns

The columns for syslanguages are:

Name	Datatype	Description
langid	smallint	Unique language ID
dateformat	char(3)	Date order; for example, “dmy”
datefirst	tinyint	First day of the week—1 for Monday, 2 for Tuesday, and so on, up to 7 for Sunday
upgrade	int	Adaptive Server version of last upgrade for this language
name	varchar(30)	Official language name, for example, “french”
alias	varchar(30)	Alternate language name, for example, “français”
months	varchar(251)	Comma-separated list of full-length month names, in order from January to December—each name is at most 20 characters long
shortmonths	varchar(119)	Comma-separated list of shortened month names, in order from January to December—each name is at most 9 characters long
days	varchar(216)	Comma-separated list of day names, in order from Monday to Sunday—each name is at most 30 characters long

Indexes

Unique clustered index On langid

Unique nonclustered index On alias, name

syslisteners

master database only

Description

syslisteners contains a row for each network protocol available for connecting with the current Adaptive Server. Adaptive Server builds syslisteners dynamically when a user or client application queries the table.

Columns

The columns for syslisteners are:

Name	Datatype	Description
net_type	char(32)	Network protocol
address_info	char(255)	Information that uniquely identifies this Adaptive Server on the network, usually the name of the current Adaptive Server and an identifying number, such as the server's port number for the protocol

syslocks

master database only

Description — syslocks contains information about active locks. It is built dynamically when queried by a user. No updates to syslocks are allowed.

Columns — The columns for syslocks are:

Name	Datatype	Description
id	int	Table ID
dbid	smallint	Database ID
page	int	Page number
type	smallint	Type of lock (bit values for the type column are listed in Table 1-12)
spid	smallint	ID of process that holds the lock
class	varchar(30)	Name of the cursor this lock is associated with, if any
fid	smallint	The family (coordinating process and its worker processes) to which the lock belongs. fid values are listed in Table 1-13.
context	tinyint	Context type of lock request. context values are listed in Table 1-14.
row	smallint	Row number
loid	int	Unique lock owner ID

Table 1-12 lists the bit representations for the type column.

Table 1-12: type control bits in the syslocks table

Decimal	Hex	Status
1	0x1	Exclusive table lock
2	0x2	Shared table lock
3	0x3	Exclusive intent lock
4	0x4	Shared intent lock
5	0x5	Exclusive page lock
6	0x6	Shared page lock
7	0x7	Update page lock
8	0x8	Exclusive row lock
9	0x9	Shared row lock
10	0xA	Update row lock
11	0xB	Shared next key lock
256	0x100	Lock is blocking another process
512	0x200	Demand lock

Table 1-13 lists the values for the fid column:

Table 1-13: fid column values in the syslocks table

Value	Interpretation
0	The task represented by the spid is a single task executing a statement in serial.
Nonzero	The task (spid) holding the lock is a member of a family executing a statement in parallel. If the value is equal to the spid, it indicates that the task is the coordinating process in a family executing a query in parallel.

Table 1-14 lists the values for the context column:

Table 1-14: context column values in the syslocks table

Value	Interpretation
null	The task holding this lock is either executing a query in serial, or it is a query being executed in parallel in transaction isolation level 1.
0x1	The task holding the lock will hold the lock until the query is complete. A lock's context may be FAM_DUR (0x1H) when the lock is: • A table lock held as part of a parallel query. • Held by a worker process at transaction isolation level 3. • Held by a worker process in a parallel query and must be held for the duration of the transaction.
0x2	Range lock held by serializable read task
0x4	Infinity key lock
0x8	Lock acquired on an index pages of an allpages-locked table
0x10	Lock on a page or row acquired to delete a row
0x20	Address lock acquired on an index page during a shrink or split operation
0x40	Intent lock held by a transaction performing repeatable reads. Valid only for shared intent and exclusive intent locks on data-only locked tables.

sysloginroles

master database only

Description

sysloginroles contains a row for each instance of a server login possessing a system role. One row is added for each role granted to each login. For example, if a single server user is granted “sa_role”, “sso_role”, and “oper_role”, three rows are added to sysloginroles associated with that user’s system user ID (suid).

Columns

The columns for sysloginroles are:

Name	Datatype	Description
suid	int	Server user ID
srid	int	Server role ID; one of the following: 0 – sa_role 1 – sso_role 2 – oper_role 4 – navigator_role 5 – replication_role
status	smallint	Reserved

Indexes

Clustered index On suid

syslogins

master database only

Description — syslogins contains one row for each valid Adaptive Server user account.

Columns — The columns for syslogins are:

Name	Datatype	Description
suid	int	Server user ID
status	smallint	Status of the account (see Table 1-15)
accdate	datetime	Date totcpu and totio were last cleared
totcpu	int	CPU time accumulated by login
totio	int	I/O accumulated by login
spacelimit	int	Reserved
timelimit	int	Reserved
resultlimit	int	Reserved
dbname	sysname	Name of database in which to put user when connection established
name	sysname	Login name of user
password	varbinary(30)	Password of user (encrypted)
language	varchar(30)	User's default language
pwdate	datetime	Date the password was last changed
audflags	int	User's audit settings
fullname	varchar(30)	Full name of the user
srvname	varchar(30)	Name of server to which a passthrough connection must be established if the AUTOCONNECT flag is turned on.
logincount	smallint	Number of failed login attempts; reset to 0 by a successful login.
procid	int	Stores the login trigger registered with the login script option in sp_modifylogin.

On the Adaptive Server distribution media, syslogins contains an entry in which the name is "sa", the suid is 1, and the password is null. It also contains the entry "probe" with an unpublished password. The login "probe" and the user "probe" exist for the two phase commit probe process, which uses a challenge and response mechanism to access Adaptive Server.

Table 1-15 lists the bit representations for the status column:

Table 1-15: status control bits in the syslogins table

Decimal	Hex	Status
1	0x1	Password contains fewer than 6 characters or is NULL
2	0x2	Account is locked
4	0x4	Password has expired

Indexes

Unique clustered index On suid

Unique nonclustered index On name

syslogs

All databases

Description

syslogs contains the transaction log. It is used by Adaptive Server for recovery and roll forward. It is not useful to users.

You cannot delete from, insert into, or update syslogs. Every data modification operation is logged, so before you can change syslogs, the change must be logged. This means that a change operation on syslogs adds a row to syslogs, which then must be logged, adding another row to syslogs, and so on, producing an infinite loop. The loop continues until the database becomes full.

Columns

The columns for syslogs are:

Name	Datatype	Description
xactid	binary(6)	Transaction ID
op	tinyint	Number of update operation

syslogshold

master database only

Description

syslogshold contains information about each database's oldest active transaction (if any) and the Replication Server truncation point (if any) for the transaction log, but it is not a normal table. Rather, it is built dynamically when queried by a user. No updates to syslogshold are allowed.

Columns

The columns for syslogshold are:

Name	Datatype	Description
dbid	smallint	Database ID.
reserved	int	Unused.
spid	smallint	Server process ID of the user that owns the oldest active transaction (always 0 for Replication Server).
page	int	Starting page number of active portion in syslogs defined by oldest transaction (or the truncation page in syslogs for Replication Server).
xactid	binary(6)	ID of the oldest active transaction (always 0x000000 for Replication Server).
masterxactid	binary(6)	ID of the transaction's master transaction (if any) for multi-database transactions; otherwise 0x000000 (always 0x000000 for Replication Server).
starttime	datetime	Date and time the transaction started (or when the truncation point was set for Replication Server).
name	char(67)	Name of the oldest active transaction. It is the name defined with begin transaction, "$user_transaction" if no value is specified with begin transaction, or "$chained_transaction" for implicit transactions started by the ANSI chained mode. Internal transactions started by Adaptive Server have names that begin with the dollar sign ($) and are named for the operation, or are named "$replication_truncation_point" for Replication Server.
xloid	int	

sysmessages

master database only

Description

sysmessages contains one row for each system error or warning that can be returned by Adaptive Server. Adaptive Server displays the error description on the user's screen.

Columns

The columns for sysmessages are:

Name	Datatype	Description
error	int	Unique error number
severity	smallint	Severity level of error
dlevel	smallint	Reserved
description	varchar(1024)	Explanation of error with placeholders for parameters
langid	smallint	Language; null for us_english
sqlstate	varchar(5)	SQLSTATE value for the error

Indexes

Clustered index On error, dlevel

Unique nonclustered index On error, dlevel, langid

sysmonitors

master database only

Description

sysmonitors contains one row for each monitor counter.

Columns

The columns for sysmonitors are:

Name	Datatype	Description
field_name	char(79)	Name of the counter
group_name	char(25)	Group this counter belongs to
field_id	smallint	Unique identifier for the row
value	int	Current value of the counter
description	varchar(255)	Description of the counter; not used

sysobjects

All databases

Description — sysobjects contains one row for each table, view, stored procedure, extended stored procedure, log, rule, default, trigger, check constraint, referential constraint, and (in tempdb only) temporary object.

Columns — The columns for sysobjects are:

Name	Datatype	Description
name	sysname	Object name
id	int	Object ID
uid	int	User ID of object owner
type	char(2)	One of the following object types: D – default F – SQLJ function L – log P – Transact-SQL or SQLJ procedure PR – prepare objects (created by Dynamic SQL) R – rule RI – referential constraint S – system table TR – trigger U – user table V – view XP – extended stored procedure
userstat	smallint	Application-dependent type information (32768 decimal [0x8000 hex] indicates to Data Workbench® that a procedure is a report)
sysstat	smallint	Internal status information (256 decimal [0x100 hex] indicates that table is read-only)
indexdel	smallint	Index delete count (incremented if an index is deleted)
schemacnt	smallint	Count of changes in the schema of an object (incremented if a rule or default is added)
sysstat2	int	Additional internal status information (see Table 1-16)
crdate	datetime	Date the object was created
expdate	datetime	Reserved
deltrig	int	Stored procedure ID of a delete trigger if the entry is a table. Table ID if the entry is a trigger.
instrig	int	Stored procedure ID of a table's insert trigger if the entry is a table
updtrig	int	Stored procedure ID of a table's update trigger if the entry is a table
seltrig	int	Reserved
ckfirst	int	ID of first check constraint on the table
cache	smallint	Reserved
audflags	int	Object's audit settings

Name	Datatype	Description
objspare	int	Spare
versionts	binary	
loginame	varchar(30)	Login name of the user who created the object

Table 1-16 lists the bit representations for the sysstat2 column:

Table 1-16: sysstat2 control bits in the sysobjects table

Decimal	Hex	Status
1	0x1	Table has a referential constraint
2	0x2	Table has a foreign key constraint
4	0x4	Table has more than one check constraint
8	0x8	Table has a primary key constraint
16	0x10	Stored procedure can execute only in chained transaction mode
32	0x20	Stored procedure can execute in any transaction mode
64	0x40	Table has an IDENTITY field
128	0x80	Not used
256	0x100	Not used
512	0x200	Table does not contain variable-length columns
1024	0x400	Table is remote
2048	0x800	Table is a proxy table created with the existing keyword
4096	0x1000	Not used
8192	0x2000	Table uses allpages locking scheme
16384	0x4000	Table uses datapages locking scheme
32768	0x8000	Table uses datarows locking scheme
65536	0x10000	Table was created in a version 11.9 or later version of the server
131072	0x20000	Table has a clustered index
262144	0x40000	Object represents an Embedded SQL procedure
524288	0x80000	Not used
16777216	0x1000000	Object represents an access rule.
33554432	0x2000000	Object represents a SQLJ stored procedure.
67108864	0x4000000	Object represents an OR access rule.

Indexes

Unique clustered index On id

Unique nonclustered index On name, uid

syspartitions

All databases

Description

syspartitions contains one row for each partition (page chain) of a partitioned table.

Columns

The columns for syspartitions are:

Name	Datatype	Description
state	smallint	Internal information about the state of the partition
id	int	Object ID of the partitioned table
partitionid	int	Partition ID number
firstpage	int	Page number of the partition's first page
controlpage	int	Page number of the partition's control page
spare	binary(32)	Reserved

Indexes

Unique clustered index On id, partitionid

sysprocedures

All databases

Description

sysprocedures contains entries for each view, default, rule, trigger, procedure, declarative default, and check constraint. The plan or sequence tree for each object is stored in binary form. If the sequence tree does not fit into one entry, it is broken into more than one row. The sequence column identifies the sub-rows.

Columns

The columns for sysprocedures are:

Name	Datatype	Description
type	smallint	Object type (see Table 1-17)
id	int	Object ID
sequence	int	Sequence number if more than one row is used to describe this object
status	smallint	Internal system status
number	smallint	Sub-procedure number when the procedure is grouped (0 for non-procedure entries)
version	int	

Table 1-17 lists the bit representations for the type column.

Table 1-17: type control bits in the sysprocedures table

Decimal	Hex	Status
1	0x1	Entry describes a plan (reserved)
2	0x2	Entry describes a tree

Indexes

Unique clustered index On id, type, sequence, number

sysprocesses

master database only

Description

sysprocesses contains information about Adaptive Server processes, but it is not a normal table. It is built dynamically when queried by a user. No updates to sysprocesses are allowed. Use the kill statement to kill a process.

Columns

The columns for sysprocesses are:

Name	Datatype	Description
spid	smallint	Process ID
kpid	int	Kernel process ID
enginenum	int	Number of engine on which process is being executed
status	char(12)	Process ID status. See Table 1-18.
suid	int	Server user ID of user who issued command
hostname	char(10)	Name of host computer
program_name	char(16)	Name of front-end module
hostprocess	char(8)	Host process ID number
cmd	char(16)	Command or process currently being executed. Evaluation of a conditional statement, such as an if or while loop, returns cond.
cpu	int	Cumulative CPU time for process in ticks
physical_io	int	Number of disk reads and writes for current command
memusage	int	Amount of memory allocated to process
blocked	smallint	Process ID of blocking process, if any
dbid	smallint	Database ID
uid	int	ID of user who executed command
gid	int	Group ID of user who executed command
tran_name	varchar(64)	Name of the active transaction
time_blocked	int	Time blocked in seconds
network_pktsz	int	Current connection's network packet size
fid	smallint	Process ID of the worker process' parent
execlass	varchar(30)	Execution class that the process is bound to
priority	varchar(10)	Base priority associated with the process
affinity	varchar(30)	Name of the engine to which the process has affinity
id	int	Object ID of the currently running procedure (or 0 if no procedure is running)
stmtnum	int	The current statement number within the running procedure (or the SQL batch statement number if no procedure is running)
linenum	int	The line number of the current statement within the running stored procedure (or the line number of the current SQL batch statement if no procedure is running)

Name	Datatype	Description
origsuid	int	Original server user ID. If this value is not NULL, a user with an suid of origsuid executed set proxy or set session authorization to impersonate the user who executed the command.
block_xloid	int	Unique lock owner ID of a lock that is blocking a transaction
clientname	varchar(30)	Name by which the user is know for the current session. This parameter is optional
clienthostname	varchar(30)	Name by which the host is known for the current session. This parameter is optional
clientapplname	varchar(30)	Name by which the application is known for the current session. This parameter is optional
sys_id	smallint	Unique identity of companion node
ses_id	int	Unique identity of each client session
loggedindatetime	datetime	Shows the time and date when the client connected to Adaptive Server. See "Row-level access control" in Chapter 11, "Managing User Permissions" of the *System Administration Guide* for more information..
ipaddr	varchar(15)	IP address of the client where the login is made. See "Row-level access control" in Chapter 11, "Managing User Permissions" of the *System Administration Guide* for more information..

Table 1-18 lists the values for the status column:

Table 1-18: sysprocesses status column values

Status	Meaning
alarm sleep	Waiting for alarm to wake process up (user executed a waitfor delay command)
background	A process, such as a threshold procedure, run by Adaptive Server rather than by a user process
infected	Server has detected a serious error condition; extremely rare
latch sleep	Waiting on a latch acquisition
lock sleep	Waiting on a lock acquisition
PLC sleep	Waiting to access a user log cache
recv sleep	Waiting on a network read
remote i/o	Performing I/O with a remote server
runnable	In the queue of runnable processes
running	Actively running on one of the server engines
send sleep	Waiting on a network send
sleeping	Waiting on a disk I/O, or some other resource (often indicates a process that is running, but doing extensive disk I/O)
stopped	Stopped process
sync sleep	Waiting on a synchronization message from another process in the family

sysprotects

All databases

Description — sysprotects contains information on permissions that have been granted to, or revoked from, users, groups, and roles.

Columns — The columns for sysprotects are:

Name	Datatype	Description
id	int	ID of the object to which this permission applies.
uid	int	ID of the user, group, or role to which this permission applies.
action	smallint	One of the following permissions: 151 = references 167 = set proxy or set session authorization 193 = select 195 = insert 196 = delete 197 = update 198 = create table 203 = create database 205 = grant 206 = revoke 207 = create view 221 = create trigger 222 = create procedure 224 = execute 228 = dump database 233 = create default 235 = dump transaction 236 = create rule 253 = connect
protecttype	tinyint	One of the following values: 0 = grant with grant 1 = grant 2 = revoke
columns	varbinary(133)	Bitmap of columns to which this select or update permission applies. The bits indicate the following: • 0 – indicates all columns • 1 – means permission applies to that column • NULL – means no information
grantor	int	User ID of the grantor. If the grantor is a System Administrator, the user ID of the object owner is used.

Indexes	**Unique clustered index**	On id, action, grantor, uid, protecttype

sysqueryplans

All databases

Description — sysqueryplans contains two or more rows for each abstract query plan. Uses datarow locking.

Columns — The columns for sysqueryplans are:

Name	Datatype	Description
uid	int	User ID of user who captured the abstract plan.
gid	int	The abstract plan group ID under which the abstract plan was saved.
hashkey	int	The hash key over the SQL query text.
id	int	The unique ID if the abstract plan.
type	smallint	10 if the text column contains query text or 100 if the text column contains abstract plan text.
sequence	smallint	Sequence number if multiple rows are required for the text of the SQL query or abstract plan.
status	int	Reserved.
text	varchar(255)	The SQL text, if type is 10, or the abstract query plan text, if the type is 100.

Indexes

Unique clustered index On uid, gid, hashkey, id, type, sequence

Nonclustered unique index On id, type, sequence

sysreferences

All databases

Description — sysreferences contains one row for each referential integrity constraint declared on a table or column.

Columns — The columns for sysreferences are:

Name	Datatype	Description
indexid	smallint	ID of the unique index on referenced columns
constrid	int	Object ID of the constraint from sysobjects
tableid	int	Object ID of the referencing table
reftabid	int	Object ID of the referenced table
keycnt	tinyint	Number of columns in the foreign key
status	smallint	Reserved
frgndbid	smallint	Database ID of the database that includes the referenced table (the table with the foreign key).
pmrydbid	smallint	Database ID of the database that includes the referenced table (the table with the primary key).
spare2	int	Reserved
fokey1 ... fokey16	smallint	Column ID of the first to the 16th referencing column
refkey1 ... refkey16	smallint	Column ID of the first to the 16th referenced column
frgndbname	varchar(30)	Name of the database that includes the referencing table (the table with the foreign key); NULL if the referencing table is in the current database
pmrydbname	varchar(30)	Name of the database that includes the referenced table (the table with the primary key); NULL if the referenced table is in the current database

Indexes

Clustered index On tableid, frgndbname

Nonclustered index On constrid, frgndbname, indexid, pmrydbname, reftabid

sysremotelogins

master database only

Description

sysremotelogins contains one row for each remote user that is allowed to execute remote procedure calls on this Adaptive Server.

Columns

The columns for sysremotelogins are:

Name	Datatype	Description
remoteserverid	smallint	Identifies the remote server
remoteusername	varchar(30)	User's login name on remote server
suid	int	Local server user ID
status	smallint	Bitmap of options

Indexes

Unique clustered index On remoteserverid, remoteusername

sysresourcelimits

master database only

Description

sysresourcelimits contains a row for each resource limit defined by Adaptive Server. Resource limits specify the maximum amount of server resources that can be used by a Adaptive Server login or an application to execute a query, query batch, or transaction.

Columns

The columns for sysresourcelimits are:

Name	Datatype	Description
name	varchar(30) null	Login name
appname	varchar(30) null	Application name
rangeid	smallint	id column from systimeranges
limitid	smallint	id column from spt_limit_types
limitvalue	int	Value of limit
enforced	tinyint	Subset of the enforced column from spt_limit_types: • 1 = prior to execution • 2 = during execution • 3 = both
actiontotake	tinyint	Action to take on a violation: • 1 = issue warning • 2 = abort query batch • 3 = abort transaction • 4 = kill session
scope	tinyint	Scope of user limit (a bitmap indicating one or more of the following): • 1 = query • 2 = query batch • 4 = transaction
spare	tinyint	Reserved

Indexes

Clustered index On name, appname

sysroles

All databases

Description

sysroles maps server role IDs to local role IDs.

Columns

The columns for sysroles are:

Name	Datatype	Description
id	int	Server role ID (srid)
lrid	int	Local role ID
type	smallint	Unused
status	int	Unused

When a database permission is granted to a role, if an entry for the role does not exist in syssrvroles, Adaptive Server adds an entry to sysroles map the local role ID (lrid) to the server-wide role ID (srid) in syssrvroles.

Indexes

Unique clustered index On lrid

syssecmechs

master database only

Description

syssecmechs contains information about the security services supported by each security mechanism that is available to Adaptive Server. Unlike other system tables, it is not created during installation. Instead, it is built dynamically when queried by a user.

Columns

The columns for syssecmechs are:

Name	Datatype	Description
sec_mech_name	varchar(30)	Name of the security mechanism; for example, “NT LANMANAGER”
available_service	varchar(30)	Name of the security service supported by the security mechanism; for example, “unified login”

syssegments

All databases

Description

syssegments contains one row for each segment (named collection of disk pieces). In a newly created database, the entries are: segment 0 (system) for system tables; segment 2 (logsegment) for the transaction log; and segment 1 (default) for other objects.

Columns

The columns for syssegments are:

Name	Datatype	Description
segment	smallint	Segment number
name	sysname	Segment name
status	smallint	Indicates which segment is the default segment

sysservers

master database only

Description

sysservers contains one row for each remote Adaptive Server, Backup Server™, or Open Server™ on which this Adaptive Server can execute remote procedure calls.

Columns

The columns for sysservers are:

Name	Datatype	Description
srvid	smallint	ID number (for local use only) of the remote server
srvstatus	smallint	Bitmap of options. See Table 1-19.
srvname	varchar(30)	Server name
srvnetname	varchar(32)	Interfaces file name for the server
srvclass	smallint	Server category defined by the class parameter of sp_addserver. See Table 1-20.
srvsecmech	varchar(30)	Security mechanism
srvcost	smallint	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.

Table 1-19 lists the bit representations for the srvstatus column:

Table 1-19: status control bits in the sysservers table

Decimal	Hex	Status
0	0x0	Timeouts are enabled
1	0x1	Timeouts are disabled
2	0x2	Network password encryption is enabled
4	0x4	Remote server is read only
8	0x8	Use rpc security model A
16	0x10	Use rpc security model B
64	0x40	Use message confidentiality
128	0x80	Use message integrity
256	0x100	Mutual authentication

Table 1-20 lists the server categories for the srvclass column:

Table 1-20: Server categories is the sysservers table

srvclass	Server category
0	Local Server
1	sql_server class server
3	direct_connect class server
4	DB2 class server
6	sds class server
7	ASEnterprise class server
8	ASAnywhere class server
9	ASIQ class server

Indexes

Unique clustered index On srvid

Unique nonclustered index On srvname

syssessions

master database only

Description

syssessions is only used when Adaptive Server is configured for Sybase's Failover in a high availability system. syssessions contains one row for each client that connects to Adaptive Server with the failover property (for example, isql -Q). Clients that have an entry in syssessions during failover are moved to the secondary companion. Clients that do not have an entry in syssessions are dropped during failover. Clients that have an entry in syssessions during failback are moved to the primary companion. Clients that do not have an entry in syssessions during failback are dropped.

Columns

The columns for syssessions are:

Name	Datatype	Description
sys_id	smallint	Unique identity of companion node
ses_id	int	Unique identity of each client session
state	tinyint	Describes whether the session is active or inactive
spare	tinyint	Reserved for future functionality
status	smallint	Reserved for future functionality
dbid	smallint	Reserved for future functionality
name	varchar(30)	Same as client's login name as specified in syslogins

syssrvroles

master database only

Description

syssrvroles contains a row for each system or user-defined role.

Columns

The columns for syssrvroles are:

Name	Datatype	Description
srid	int	Server role ID
name	varchar(30)	Name of the role
password	varinary(30)	Password for the role (encrypted)
pwdate	datetime null	Date the password was last changed
status	smallint	Bitmap for role status. See Table 1-21
logincount	smallint	Number of failed login attempts; reset to 0 by a successful login.

Table 1-21 lists the bit representations for the status column:

Table 1-21: status control bits in the syssrvroles table

Decimal	Hex	Status
2	0x2	Role is locked
4	0x4	Role is expired

Indexes

Unique clustered index On srid

sysstatistics

All databases

Description

sysstatistics contains one or more rows for each indexed column on a user table. May also contain rows for unindexed column. Uses datarow locking.

Columns

The columns for sysstatistics are:

Name	Datatype	Description
statid	smallint	Reserved
id	int	Object ID of table
sequence	int	Sequence number if multiple rows are required for this set of statistics
moddate	datetime	Date this row was last modified
formatid	tinyint	Type of statistics represented by this row
usedcount	tinyint	Number of fields c0 to c79 used in this row
colidarray	varbinary(100)	An ordered list of column IDs
c0...c79	varbinary(255)	Statistical data

Indexes

Unique clustered index On id, statid, colidarray, formatid, sequence

systabstats

All databases

Description

systabstats contains one row for each clustered index, one row for each nonclustered index, and one row for each table that has no clustered index. Uses datarow locking.

Columns

The columns for systabstats are:

Name	Datatype	Description
indid	smallint	0 if a table; 1 if a clustered index on an allpages-locked table; >1 if a nonclustered index or a clustered index on a data-only-locked table; statistics on text or image objects (255) are not maintained in systabstats.
id	int	ID of table to which index belongs
activestatid	smallint	Reserved
indexheight	smallint	Height of the index; maintained if indid is greater than 1
leafcnt	int	Number of leaf pages in the index; maintained if indid is greater than 1
pagecnt	int	Number of pages in the table or index
rowcnt	float	Number of rows in the table; maintained for indid of 0 or 1
forwrowcnt	float	Number of forwarded rows; maintained for indid of 0 or 1
delrowcnt	float	Number of deleted rows
dpagecrcnt	float	Number of extent I/Os that need to be performed to read the entire table
ipagecrcnt	float	Number of extent I/Os that need to be performed to read the entire leaf level of a nonclustered index
drowcrcnt	float	Number of page I/Os that need to be performed to read an entire table
oamapgcnt	int	Number of OAM pages for the table, plus the number of allocation pages that store information about the table
extent0pgcnt	int	Count of pages that are on the same extent as the allocation page
datarowsize	float	Average size of the data row
leafrowsize	float	Average size of a leaf row for nonclustered indexes and clustered indexes data-only-locked tables
status	int	Internal system status information (see Table 1-22)

Name	Datatype	Description
spare1	int	Reserved
spare2	float	Reserved
rslastoam	int	Last OAM page visited by a reorg reclaim_space or reorg compact command
rslastpage	int	Last data or leaf page visited by a reorg reclaim_space or reorg compact command
frlastoam	int	Last OAM page visited by the reorg forwarded_rows command
frlastpage	int	Last data page visited by the reorg forwarded_rows command
conopt_thld	smallint	Concurrency optimization threshold
spare3	smallint	Reserved
emptypgcnt	int	Number of empty pages in extents allocated to the table or index
spare4	float	Reserved

Table 1-22 lists the bit representations for the status column:

Table 1-22: status bits in the systabstats table

Decimal	Hex	Status
1	0x1	Statistics are the result of upgrade (not update statistics)

Indexes

Unique clustered index On id, indid

systhresholds

All databases

Description

systhresholds contains one row for each threshold defined for the database.

Columns

The columns for systhresholds are:

Name	Datatype	Description
segment	smallint	Segment number for which free space is being monitored.
free_space	int	Size of threshold, in 2K pages (4K for Status).
status	smallint	Bit 1 equals 1 for the logsegment's last-chance threshold, 0 for all other thresholds.
proc_name	varchar(255)	Name of the procedure that is executed when the number of unused pages on segment falls below free_space.
suid	int	The server user ID of the user who added the threshold or modified it most recently.
currauth	varbinary(255)	A bit mask that indicates which roles were active for suid at the time the threshold was added or most recently modified. When the threshold is crossed, proc_name executes with this set of roles, less any that have been deactivated since the threshold was added or last modified.

Indexes

Unique clustered index On segment, free_space

systimeranges

master database only

Description

systimeranges stores named time ranges, which are used by Adaptive Server to control when a resource limit is active.

Columns

The columns for systimeranges are:

Name	Datatype	Description
name	varchar(30)	Unique name of the time range.
id	smallint	Unique identifier for the time range. 1 represents the "at all times" limit.
startday	tinyint	Day of week (1–7) for the beginning of the range. Monday = 1, Sunday = 7.
endday	tinyint	Day of week (1–7) for the end of the range. Monday = 1, Sunday = 7.
starttime	varchar(10)	Time of day for the beginning of the range.
endtime	varchar(10)	Time of day for the end of the range.

Indexes

Clustered index On id

systransactions

master database only

Description

systransactions contains information about Adaptive Server transactions, but it is not a normal table. Portions of the table are built dynamically when queried by a user, while other portions are stored in the master database. Updates to the dynamically-built columns of systransactions are not allowed.

Columns

The columns for systransactions are:

Name	Datatype	Description
xactkey	binary(14)	Unique Adaptive Server transaction key
starttime	datetime	Date the transaction started
failover	int	Value indicating the transaction failover state (see Table 1-23)
type	int	Value indicating the type of transaction (see Table 1-24)
coordinator	int	Value indicating the coordination method or protocol (see Table 1-25)
state	int	Value indicating the current state of the transaction (see Table 1-26)
connection	int	Value indicating the connection state (see Table 1-27)
status	int	Internal transaction status flag
status2	int	Additional internal transaction status flags.
spid	smallint	Server process ID, or 0 if the process is detached
masterdbid	smallint	Starting database of the transaction
loid	int	Lock owner ID
namelen	smallint	Length of “xactname” below
xactname	varchar(255)	Transaction name or *XID*
srvname	varchar(30)	Name of the remote server (null for local servers)

Table 1-23: systransactions failover column values

failover value	Failover state
0	Resident Tx
1	Failed-over Tx
2	Tx by Failover-Conn

Table 1-24: systransactions type column values

type value	Transaction type
1	Local
3	External
98	Remote
99	Dtx_State

Table 1-25: systransactions coordinator column values

coordinator value	Coordination method or protocol
0	None
1	Syb2PC
2	ASTC
3	XA
4	DTC

Table 1-26: systransactions state column values

state value	Transaction state
1	Begun
2	Done Command
3	Done
4	Prepared
5	In Command
6	In Abort Cmd
7	Committed
8	In Post Commit
9	In Abort Tran
10	In Abort Savept
65537	Begun-Detached
65538	Done Cmd-Detached
65539	Done-Detached
65540	Prepared-Detached
65548	Heur Committed
65549	Heur Rolledback

Table 1-27: systransactions connection column values

***connection* value**	**Connection state**
1	Attached
2	Detached

systypes

All databases

Description

systypes contains one row for each system-supplied and user-defined datatype. Domains (defined by rules) and defaults are given, if they exist.

The rows that describe system-supplied datatypes cannot be altered.

Columns

The columns for systypes are:

Name	Datatype	Description
uid	int	User ID of datatype creator
usertype	smallint	User type ID
variable	bit	1 if datatype is variable length; 0 otherwise
allownulls	bit	Indicates whether nulls are allowed for this datatype
type	tinyint	Physical storage datatype
length	int	Physical length of datatype
tdefault	int	ID of system procedure that generates default for this datatype
domain	int	ID of system procedure that contains integrity checks for this datatype
name	sysname	Datatype name
printfmt	varchar(255)	Reserved
prec	tinyint	Number of significant digits
scale	tinyint	Number of digits to the right of the decimal point
ident	tinyint	1 if column has the IDENTITY property, 0 if it does not
hierarchy	tinyint	Precedence of the datatype in mixed mode arithmetic
accessrule	int	The object ID of the access rule in sysprocedures.
xtypeid	int	The internal class ID
xdbid	int	The dbid where a class is installed: • -1 = the system database • -2 = the current database

Table 1-28 lists each system-supplied datatype's name, hierarchy, type (not necessarily unique), and usertype (unique). The datatypes are ordered by hierarchy. In mixed-mode arithmetic, the datatype with the lowest hierarchy takes precedence:

Table 1-28: Datatype names, hierarchy, types, and usertypes

Name	hierarchy	type	usertype
floatn	1	109	14
float	2	62	8
datetimn	3	111	15
datetime	4	61	12
real	5	59	23
numericn	6	108	28
numeric	7	63	10
decimaln	8	106	27
decimal	9	55	26
moneyn	10	110	17
money	11	60	11
smallmoney	12	122	21
smalldatetime	13	58	22
intn	14	38	13
int	15	56	7
smallint	16	52	6
tinyint	17	48	5
bit	18	50	16
univarchar	19	155	35
unichar	20	135	34
reserved	21		
varchar	22	39	2
sysname	22	39	18
nvarchar	22	39	25
char	23	47	1
nchar	23	47	24
varbinary	24	37	4
timestamp	24	37	80
binary	25	45	3
text	26	35	19
image	27	34	20
date	28	123	37
time	29	124	38
daten	30	160	39
timen	31	29	40

Indexes	**Unique clustered index** On name
	Unique nonclustered index On usertype

sysusages

master database only

Description

sysusages contains one row for each **disk allocation piece** assigned to a database. Each database contains a specified number of database (logical) page numbers. Each disk piece includes the segments on the Adaptive Server distribution media, segments 0 and 1.

The create database command checks sysdevices and sysusages to find available disk allocation pieces. One or more contiguous disk allocation pieces are assigned to the database, and the mapping is recorded in sysusages.

Columns

The columns for sysusages are:

Name	Datatype	Description
dbid	smallint	Database ID
segmap	int	Bitmap of possible segment assignments
lstart	int	First database (logical) page number
size	int	Number of contiguous database (logical) pages
vstart	int	Starting virtual page number
pad	smallint	Unused
unreservedpgs	int	Free space not part of an allocated extent
crdate	datetime	Creation date

Indexes

Unique clustered index On dbid, lstart

Unique nonclustered index On vstart

sysusermessages

All databases

Description — sysusermessages contains one row for each user-defined message that can be returned by Adaptive Server.

Columns — The columns for sysusermessages are:

Name	Datatype	Description
error	varchar(1024)	Unique error number. Must be 20,000 or higher.
uid	int	Server user ID (suser_id) of the message creator.
description	varchar(1024)	User-defined message with optional placeholders for parameters.
langid	smallint	Language ID for this message; null for us_english.
dlevel	smallint	Stores the with_log bit, which is used to call the appropriate routine to log a message.

Indexes — **Clustered index** On error

Unique nonclustered index On error, langid

sysusers

All databases

Description — sysusers contains one row for each user allowed in the database, and one row for each group or role.

Columns — The columns for sysusers are:

Name	Datatype	Description
suid	int	Server user ID, copied from syslogins.
uid	int	User ID, unique in this database, is used for granting and revoking permissions. User ID 1 is "dbo".
gid	int	Group ID to which this user belongs. If uid = gid, this entry defines a group. Negative values may be used for user IDs (uid). Every suid associated with a group or a role in sysusers is set to -2 (INVALID_SUID).
name	sysname	User or group name, unique in this database.
environ	varchar(255)	Reserved.

On the Adaptive Server distribution media, master..sysusers contains some initial users: "dbo", whose suid is 1 and whose uid is 1; "guest", whose suid is -1 and whose uid is 2; and "public", whose suid is -2 and whose uid is 0. In addition, both system-defined and user-defined roles are listed in sysusers.

The user "guest" provides a mechanism for giving users not explicitly listed in sysusers access to the database with a restricted set of permissions. The "guest" entry in master means any user with an account on Adaptive Server (that is, with an entry in syslogins) can access master.

The user "public" refers to all users. The keyword public is used with the grant and revoke commands to signify that permission is being given to or taken away from all users.

Indexes — **Unique clustered index** On suid

Unique nonclustered index On name, uid

sysxtypes

All databases

Description

sysxtypes contains one row for each extended, Java-SQL datatype. Uses row-level locking.

Refer to *Java in Adaptive Server Enterprise* for more information about Java-SQL classes and datatypes.

Columns

The columns for sysxtypes are:

Name	Datatype	Description
sensitivity	sensitivity	Used by the Secure Adaptive Server.
xtname	varchar(255)	The name of the extended type.
xtid	int	System-generated ID for the extended type.
xtstatus	int	Internal status information. Unused.
xtmetatype	int	Unused.
xtcontainer	int	The ID of the JAR file containing the class. Can be NULL.
xtsource	text	Source code for the extended type. Unused.
xtbinary	image	Object code for the extended type. For Java classes, it contains the class file.

Indexes

Unique placement index On xtid

Unique non-clustered index On xtname

syblicenseslog

master database only

Description

syblicenseslog contains one row for each update of the maximum number of licenses used in Adaptive Server per 24-hour period. syblicenseslog is updated every 24 hours. If Adaptive Server is shut down at any time, License Use Manager logs the number of licenses currently being used in syblicenseslog before the shutdown is complete. The 24 hour period restarts when you start Adaptive Server.

Note syblicenseslog is not a system table. Its type is "U" and its object ID is greater than 100.

Columns

The columns for syblicenseslogs are:

Name	Datatype	Description
status	smallint	Status of the maximum number of licenses used; one of the following: 0 = number of licenses not exceeded 1 = number of licenses is exceeded -1 = housekeeper is unable to monitor number of licenses
logtime	datetime	Date and time the log was written
maxlicenses	int	Maximum number of licenses used during the 24-hour period

CHAPTER 2

dbccdb Tables

In addition to the standard system tables included in all databases, the dbcc management database, dbccdb, contains seven tables that define inputs to and outputs from dbcc checkstorage. It also contains at least two workspaces.

Topics include:

dbccdb workspaces

Workspaces are special tables in dbccdb that store intermediate results of the dbcc checkstorage operation. Workspaces differ from worktables in that they:

- Are preallocated contiguously to improve I/O performance
- Are persistent
- Do not reside in the tempdb database

When you create dbccdb, two workspaces, scan and text, are created automatically. They are preallocated as follows:

- *Scan workspace* – contains a row for each page of the target database. The allocation is approximately 1 percent of the database size. Each row consists of a single binary(18) column.
- *Text workspace* – contains a row for each table in the target database that contains text or image columns. The size of this table depends on the design of the target database, but it is usually significantly smaller than the scan workspace. Each row consists of a single binary(22) column.

If either allocation is larger than needed by dbcc checkstorage, the operation uses only what it requires. The allocation does not change. If the text workspace allocation is too small, dbcc checkstorage reports this, recommends a new size, and continues checking; however, not all text chains are checked. If the scan workspace allocation is too small, the dbcc checkstorage operation fails immediately.

You must have at least one scan and one text workspace, but you may create as many as you need. While in use, the workspaces are locked so that only one dbcc checkstorage operation can use them at any given time. You can execute concurrent dbcc checkstorage operations by supplying each one with a separate scan and text workspace.

For more information on creating workspaces, see the *System Administration Guide* and the *Adaptive Server Reference Manual*.

Ideally, workspaces should be accessed only through dbcc checkstorage, but this is not a requirement. dbcc checkstorage exclusively locks the workspaces it uses, and the content of the workspaces is regenerated with each execution of dbcc checkstorage. The workspaces do not contain any secure data.

Note While the contents of the workspaces are accessible through SQL, no interpretation of the binary values is available. Access through SQL might return data from different dbcc checks mixed together. The presence of a row in these tables does not ensure that it contains valid data. dbcc tracks valid rows only during execution. That information is lost when the operation completes.

Most of the update activity in dbccdb is performed in the text and scan workspaces. The workspaces are preallocated, and only one dbcc checkstorage operation can use the workspaces at any given time, so the workspaces are less susceptible to corruption than most user tables. Corruption in a workspace can cause the dbcc checkstorage operation to fail or behave erratically. If this happens, drop and re-create the corrupt workspace.

Checks of databases using different workspaces can proceed simultaneously, but the performance of each operation might be degraded as it competes for disk throughput.

To delete a workspace, in dbccdb, enter:

```
drop table workspace_name
```

dbccdb log

The results of each dbcc checkstorage operation are recorded in the dbccdb log. Updates to the text and scan workspaces are not recorded there.

The dbccdb log must be sized to handle updates to the tables. The log requirement is related to the number of tables and indexes in the target database. It is not related to the target database size.

To minimize the log requirement and the recovery time, use the truncate log on checkpoint option with dbccdb.

dbcc_config

Description

The dbcc_config table describes the currently executing or last completed dbcc checkstorage operation. It defines:

- The location of resources dedicated to the dbcc checkstorage operation
- Resource usage limits for the dbcc checkstorage operation

Columns

The columns for dbcc_config are:

Column name	Datatype	Description
dbid	smallint	Matches the dbid from a row in sysindatabases.
type_code	int	Matches the type_code from a row in the dbcc_types table. Valid values are 1–9.
value	int	Specifies the value of the item identified by type_code. Can be null only if the value of stringvalue is not null.
stringvalue	varchar(255)	Specifies the value of the item identified by type_code. Can be null only if the value of value is not null.

Primary key

Combination of dbid and type_code

See also

For information on initializing and updating dbcc_config, see the *System Administration Guide*.

dbcc_counters

Description

The dbcc_counters table stores the results of the analysis performed by dbcc checkstorage. Counters are maintained for each database, table, index, partition, device, and invocation of dbcc.

Columns

The columns for dbcc_counters are:

Column name	Datatype	Description
dbid	smallint	Identifies the target database.
id	int	Identifies the table. The value is derived from sysindexes and sysobjects.
indid	smallint	Identifies the index. The value is derived from sysindexes.
partitionid	smallint	Identifies the defined object-page affinity. The value is derived from sysindexes and syspartitions.
devid	smallint	Identifies the disk device. The value is derived from sysdevices.
opid	smallint	Identifies the dbcc operation that was performed.
type_code	int	Matches the type_code column of a row in the dbcc_types table. Valid values are 5000 through 5024.
value	real	Matches the appropriate type_name for the given type_code as described in dbcc_types.

Primary key

Combination of dbid, id, indid, partitionid, devid, opid, and type_code

dbcc_fault_params

Description — The dbcc_fault_params table provides additional descriptive information for a fault entered in the dbcc_faults table.

Columns — The columns for dbcc_fault_params are:

Column name	Datatype	Description
dbid	smallint	Identifies the target database.
opid	smallint	Identifies the dbcc operation that was performed.
faultid	int	Identifies the fault ID.
type_code	int	Defines the interpretation of the value, which is provided by the "value" columns. Valid values are 1000–1009. They are described in dbcc_types.
intvalue	int	Specifies the integer value.
realvalue	real	Specifies the real value.
binaryvalue	varbinary(255)	Specifies the binary value.
stringvalue	varchar(255)	Specifies the string value.
datevalue	datetime	Specifies the date value.

Each "value" column (intvalue, realvalue, binaryvalue, stringvalue, and datevalue) can contain a null value. At least one must not be null. If more than one of these columns contains a value other than null, the columns provide different representations of the same value.

Primary key — Combination of dbid, opid, faultid, and type_code

dbcc_faults

Description — The dbcc_faults table provides a description of each fault detected by dbcc checkstorage.

Columns — The columns for dbcc_faults are:

Column name	Datatype	Description
dbid	smallint	Identifies the target database.
id	smallint	Identifies the table. The value is derived from sysindexes and sysobjects.
indid	smallint	Identifies the index. The value is derived from sysindexes.
partitionid	smallint	Identifies the partition. The value is derived from sysindexes and syspartitions. Counters are maintained for page ranges, so “partition” refers to the defined object-page affinity, rather than the actual object page chain.
devid	smallint	Identifies the disk device. The value is derived from sysdevices
opid	smallint	Identifies the dbcc operation that was performed.
faultid	int	Provides a unique sequence number assigned to each fault recorded for the operation.
type_code	int	Identifies the type of fault. Valid values are 100000–100032. They are described in Table 2-1 on page 106.
status	int	Classifies the fault. Valid values are: • 0 – Soft fault, possibly transient • 1– Hard fault • 2 – Soft fault that proved to be transient • 3 – Soft fault upgraded to a hard fault • 5 – Repaired hard fault • 7 – Repaired upgraded hard fault • 9 – Hard fault not rapirable • 11 – Soft fault upgraded to a hard fault and not repairable • 16 – Soft fault, object dropped (inaccessible) • 17 – Hard fault, object dropped (inaccessible) • 18 – Transient soft fault, object dropped (inaccessible) • 19 – Soft fault upgraded to a hard fault and object dropped (inaccessible) For more information, see the *System Administration Guide*.

Primary key — Combination of dbid, id, indid, partitionid, devid, opid, faultid, and type_code

dbcc_operation_log

Description — The dbcc_operation_log table records the use of the dbcc checkstorage operations.

Columns — The columns for dbcc_operaiton_log are:

Column Name	Datatype	Description
dbid	smallint	Identifies the target database
opid	smallint	Identifies the sequence number of the dbcc checkstorage operation. opid is an automatically incrementing number, unique for each dbid and finish pair.
optype	smallint	The following value is valid for optype: • 2 = checkstorage
suid	int	Identifies the user executing the command
start	datetime	Identifies when the operation started
finish	datetime	Identifies when the operation ended

Summary results are recorded in the dbcc_operation_results table.

Primary key — Combination of dbid, opid, and optype

dbcc_operation_results

Description — The dbcc_operation_results table provides additional descriptive information for an operation recorded in the dbcc_operation_log table.

Columns — The columns for dbcc_operation_results are:

Column Name	Datatype	Description
dbid	smallint	Identifies the target database
opid	smallint	Identifies the dbcc operation ID
optype	smallint	Identifies the dbcc operation type
type_code	int	Defines the dbcc operation type. Valid values are 1000–1007. They are described in Table 2-1 on page 106.
intvalue	int	Specifies the integer value
realvalue	real	Specifies the real value
binaryvalue	varbinary(255)	Specifies the binary value
stringvalue	varchar(255)	Specifies the string value
datevalue	datetime	Specifies the date value

Each "value" column (intvalue, realvalue, binaryvalue, stringvalue, and datevalue) may contain a null value. At least one is not null. If more than one of these columns contains a value other than null, the columns provide different representations of the same value.

Results of the dbcc checkstorage operations include the number of:

- Hard faults found
- Soft faults found
- Operations stopped due to a hard error

Primary key — Combination of dbid, opid, optype, and type_code

dbcc_types

Description

The dbcc_types table provides the definitions of the data types used by dbcc checkstorage. This table is not actually used by the dbcc stored procedures. It is provided to facilitate the use of the other tables in dbccdb, and to document the semantics of the data types. Type codes for operation configuration, analysis data reported, fault classification, and fault report parameters are included. If you create your own stored procedures for generating reports, the values listed in the type_name column can be used as report headings.

Columns

The columns for dbcc_types are as follows.

Note To allow for future additions to dbcc_types, some type_code numbers are not used at this time:

Table 2-1: dbcc Types

type_code	*type_name*	Description
1	max worker processes	*Optional* – Specifies the maximum number of worker processes that can be employed. This is also the maximum level of concurrent processing used. Minimum value is 1.
2	dbcc named cache	Specifies the size (in kilobytes) of the cache used by dbcc checkstorage and the name of that cache.
3	scan workspace	Specifies the ID and name of the workspace to be used by the database scan.
4	text workspace	Specifies the ID and name of the workspace to be used for text columns.
5	operation sequence number	Specifies the number that identifies the dbcc operation that was started most recently.
6	database name	Specifies the name of the database in sysdatabases.
7	OAM count threshold	Specifies the percentage by which the OAM counts must vary before they can be considered to be an error.
8	IO error abort	Specifies the number of I/O errors allowed on a disk before dbcc stops checking the pages on that disk.
9	linkage error abort	Specifies the number of linkage errors allowed before dbcc stops checking the page chains of an object. Some kinds of page chain corruptions might require a check to be stopped with fewer linkage error than other kinds of page chain corruptions.
1000	hard fault count	Specifies the number of persistent inconsistencies (hard faults) found during the consistency check.
1001	soft fault count	Specifies the number of suspect conditions (soft faults) found during the consistency check.

type_code	type_name	Description
1002	checks aborted count	Specifies the number of linkage checks that were stopped during the consistency check.
1007	text column count	Specifies the number of non-null text/image column values found during the consistency check.
5000	bytes data	Specifies (in bytes) the amount of user data stored in the partition being checked.
5001	bytes used	Specifies (in bytes) the amount of storage used to record the data in the partition being checked. The difference between bytes used and bytes data shows the amount of overhead needed to store or index the data.
5002	pages used	Specifies the number of pages linked to the object being checked that are actually used to hold the object.
5003	pages reserved	Specifies the number of pages that are reserved for the object being checked, but that are not allocated for use by that object. The difference between (8 * extents used) and (pages used + pages reserved) shows the total uncommitted deallocations and pages incorrectly allocated.
5004	pages overhead	Specifies the number of pages used for the overhead functions such as OAM pages or index statistics.
5005	extents used	Specifies the number of extents allocated to the object in the partition being checked. For object 99 (allocation pages), this value is the number of extents that are not allocated to a valid object. Object 99 contains the storage that is not allocated to other objects.
5006	count	Specifies the number of component items (rows or keys) found on any page in the part of the object being checked.
5007	max count	Specifies the maximum number of component items found on any page in the part of the object being checked.
5008	max size	Specifies the maximum size of any component item found on any page in the part of the object being checked.
5009	max level	Specifies the maximum number of levels in an index. This datatype is not applicable to tables.
5010	pages misallocated	Specifies the number of pages that are allocated to the object, but are not initialized correctly. It is a fault counter.
5011	io errors	Specifies the number of I/O errors encountered. This datatype is a fault counter.
5012	page format errors	Specifies the number of page format errors reported. This datatype is a fault counter.
5013	pages not allocated	Specifies the number of pages linked to the object through its chain, but not allocated. This datatype is a fault counter.
5014	pages not referenced	Specifies the number of pages allocated to the object, but not reached through its chains. This datatype is a fault counter.
5015	overflow pages	Specifies the number of overflow pages encountered. This datatype is only applicable to clustered indexes.

type_code	*type_name*	**Description**
5016	page gaps	Specifies the number of pages not linked to the next page in ascending sequence. This number indicates the amount of table fragmentation.
5017	page extent crosses	Specifies the number of pages that are linked to pages outside of their own extent. As the number of page extent crosses increases relative to pages used or extents used, the effectiveness of large I/O buffers decreases.
5018	page extent gaps	Specifies the number of page extent crosses where the subsequent extent is not the next extent in ascending sequence. Maximal I/O performance on a full scan is achieved when the number of page extent gaps is minimized. A seek or full disk rotation is likely for each gap.
5019	ws buffer crosses	Specifies the number of pages that are linked outside of their workspace buffer cache during the dbcc checkstorage operation. This information can be used to size the cache, which provides high performance without wasting resources.
5020	deleted rows	Deleted rows, the number of deleted rows in the object
5021	forwarded rows	Forwarded rows, number of forwarded rows in the object
5022	empty pages	The number of pages allocated but not containing data
5023	pages with garbage	Pages with garbage, number of pages that could benefit from garbage collection
5024	non-contiguous free space	Non-contiguous free space, number of bytes of non-contiguous free space
10000	page id	Specifies the location in the database of the page that was being checked when the fault was detected. All localized faults include this parameter.
10001	page header	Specifies the hexadecimal representation of the header of the page that was being checked when the fault was detected. This information is useful for evaluating soft faults and for determining if the page has been updated since it was checked. The server truncates trailing zeros.
10002	text column id	Specifies an 8-byte hexadecimal value that gives the page, row, and column of the reference to a text chain that had a fault. The server truncates trailing zeros.
10003	object id	Specifies a 9-byte hexadecimal value that provides the object id (table), the partition id (partition of the table) if applicable, and the index id (index) of the page or allocation being checked. For example, if a page is expected to belong to table T1 because it is reached from T1's chain, but is actually allocated to table T2, the object id for T1 is recorded, and the object id expected for T2 is recorded. The server truncates trailing zeros.

type_code	type_name	Description
10007	page id expected	Specifies the page ID that is expected for the linked page when there is a discrepancy between the page ID that is expected and the page ID that is actually encountered. For example, if you follow the chain from P1 to P2 when going forward, then, when going backward, P1 is expected to come after P2. The value of page id expected is P1, and the value of page id is P2. When the actual value of P3 is encountered, it is recorded as page id actual.
10008	page id actual	When there is a discrepancy between the page ID that is encountered and the expected page ID, this value specifies the actual page ID that is encountered. (See also, type_code 10007.) For example, if you follow the chain from P1 to P2 when going forward, then, when going backward, P1 is expected to come after P2. The value of page id expected is P1, and the value of page id is P2. When the actual value of P3 is encountered, it is recorded as page id actual.
10009	object id expected	Specifies a 9-byte hexadecimal value that provides the expected object id (table), the partition id (partition of the table) if applicable, and the index id (index) of the page or allocation being checked. For example, if a page is expected to belong to table T1 because it is reached from T1's chain, but is actually allocated to table T2, the object id for T1 is recorded, and the object id expected for T2 is recorded. The server truncates trailing zeros.
10010	data-only locked data page header	Indicates the 44 byte page header for the page where the fault is located
10011	data-only locked b-tree leaf page header	Indicates the 44 byte page header for the page where the fault is located
10012	data-only locked b-tree header	Indicates the 44 byte page header for the page where the fault is located
100000	IO error	Indicates that part of the identified page could not be fetched from the device. This is usually caused by a failure of the operating system or the hardware.
100001	page id error	Indicates that the identifying ID (page number) recorded on the page is not valid. This might be the result of a page being written to or read from the wrong disk location, corruption of a page either before or as it is being written, or allocation of a page without subsequent initialization of that page.
100002	page free offset error	Indicates that the end of data on a page is not valid. This event affects insertions and updates on this page. It might affect some access to the data on this page.

type_code	type_name	Description
100003	page object id error	Indicates that the page appears to be allocated to some other table than the one expected. If this is a persistent fault, it might be the consequence of either: • An incorrect page allocation, which might only result in the effective loss of this page to subsequent allocation, or • A corrupted page chain, which might prevent access to the data in the corrupted chain
100004	timestamp error	Indicates that the page has a timestamp that is later than the database timestamp. This error can result in failure to recover when changes are made to this page.
100005	wrong dbid error	Indicates that the database ID dbid is stored on the database allocation pages. When this ID is incorrect, the allocation page is corrupt and all the indicated allocations are suspect.
100006	wrong object error	Indicates that the page allocation is inconsistent. The page appears to belong to one table or index, but it is recorded as being allocated to some other table or index in the allocation page. This error differs from page object id error in that the allocation is inconsistent, but the consequences are similar.
100007	extent id error	Indicates that an allocation was found for a table or index that is unknown to dbcc checkstorage. Typically, this results in the inability to use the allocated storage.
100008	fixed format error	Indicates that the page incorrectly indicates that it contains only rows of a single fixed length. dbcc checkstorage reports this error. dbcc checktable does not report it, but does repair it.
100009	row format error	Indicates that at least one row on the page is incorrectly formatted. This error might cause loss of access to some or all the data on this page.
100010	row offset error	Indicates that at least one row on the page is not located at the expected page offset. This error might cause loss of access to some or all of the data on this page.
100011	text pointer error	Indicates that the location of the table row that points to the corrupted text or image data. This information might be useful for correcting the problem.
100012	wrong type error	Indicates that the page has the wrong format. For example, a data page was found in an index or a text/image column.
100013	non-OAM error	This error is a special case of wrong type error. It is not reported as a separate condition in the current release.
100014	reused page error	Indicates that a page is reached by more than one chain and that the chains belong to different objects. This error indicates illegal sharing of a page through corrupt page chain linkages. Access to data in either or both tables might be affected.

type_code	type_name	Description
100015	page loop error	Indicates that a page is reached a second time while following the page chain for an object, which indicates a loop in the page chain. A loop can result in a session hanging indefinitely while accessing data in that object.
100016	OAM ring error	Indicates that a page is allocated but not reached by the page chains for the object.Typically, this results in the inability to use the allocated storage.
100017	OAM ring error	Indicates that the OAM page ring linkages are corrupted. This might not affect access to the data for this object, but it might affect insertions, deletions, and updates to that data.
100018	missing OAM error	Indicates that dbcc checkstorage found an allocation for the object that was not recorded in the OAM. This error indicates a corruption that might affect future allocations of storage, but probably does not affect access to the presently stored data.
100019	extra OAM error	Indicates that an allocation for this object was recorded in the OAM, but it was not verified in the allocation page. This error indicates a corruption that might affect future allocations of storage, but probably does not affect access to the presently stored data.
100020	check aborted error	Indicates that dbcc checkstorage stopped checking the table or index. To prevent multiple fault reports, the check operation on a single chain might be stopped without reporting this error. When an object contains several page chains, failure of the check operation for one chain does not prevent the continuation of the check operation on the other chains unless a fault threshold is exceeded.
100021	chain end error	Indicates that the end of the chain is corrupted. As a soft fault, it might indicate only that the chain was extended or truncated by more than a few pages during the dbcc checkstorage operation.
100022	chain start error	Indicates that the start of a chain is corrupted or is not at the expected location. If this is a persistent fault, access to data stored in the object is probably affected.
100023	used count error	Indicates an inconsistency between the count of the pages used that is recorded in the OAM page and the count of the pages used that is determined by examining the allocation pages.
100024	unused count error	Indicates an inconsistency between the count of the pages reserved but unused that is recorded in the OAM page and the count of the pages reserved but unused that is determined by examining the allocation pages.
100025	row count error	Indicates an inconsistency between the row count recorded in the OAM page and the row count determined by dbcc checkstorage.
100026	serialloc error	Indicates a violation of the serial allocation rules applied to log allocations.

type_code	*type_name*	Description
100027	text root error	Indicates a violation of the format of the root page of a text or image index. This check is similar to the root page checks performed by dbcc textalloc.
100028	page misplaced	Indicates that pages of this object were not found where they were expected to be from examination of the system tables. This usually indicates that sp_placeobject was used sometime in the past. In the dbcc_counters table, all misplaced pages are counted together, rather than being reported by device and partition.
100029	page header error	Indicates an internal inconsistency in the page's header other than the kind described by the other type codes. The severity of this error depends on the type of page and the inconsistency found.
100030	page format error	Indicates an internal inconsistency in the page's body other than the kind described by the other type codes. The severity of this error depends on the type of page and the inconsistency found.
100031	page not allocated	Indicates that dbcc checkstorage reached an unallocated page by following a page chain. This condition might affect access to data stored in this object.
100032	page linkage error	Indicates that dbcc checkstorage detected a fault with either the next or previous linkage of an interior page of a chain. If this is a persistent fault, access to data stored in the object is probably affected.
100033	non-contiguous free-space error	Indicates an invalid or inconsistent value for the non-contiguous free space on the page
100034	insert free space error	Indicates an invalid or inconsistent value for the contiguous free space on the page
100035	spacebits mismatch	Indicates an inconsistency in the page fullness indicator
100036	deleted row count error	Indicates an invalid or inconsistent value for the deleted row count on the page
100037	forwarded rows error	Indicates an inconsistency between the forwarded rows indicator and the number of forwarded rows on the page
100038	page header type error	Indicates that a Page header format indicator set incorrectly

Index

Symbols

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W