

# ***The Complete Sybase ASE Quick Reference Guide***

***ASE versions 12.5, 15.0.3 & Cluster Edition  
5<sup>th</sup> edition***

These pages are taken from "*The Complete Sybase ASE Quick Reference Guide*", as a sample of the contents. Note that the original page size is 11 by 22 cm (4.3 by 8.6 inch).

The complete book can be ordered from <http://www.sypron.nl/qr> .

***Rob Verschoor***

---

***Sypron Publications***

# The Complete Sybase ASE Quick Reference Guide

ASE versions 12.5, 15.0.3 & Cluster Edition

5<sup>th</sup> edition

by Rob Verschoor

ISBN 90-806117-1-9

Published by:

Sypron B.V.  
Het Wolfseind 24  
3823 VS Amersfoort  
The Netherlands  
Internet [www.sypron.nl](http://www.sypron.nl)  
Email [sypron@sypron.nl](mailto:sypron@sypron.nl)

Printed in The Netherlands

First edition	:	January 2001	(ASE 11.9 & 12.0)
Second edition	:	March 2002	(ASE 11.9, 12.0 & 12.5)
Third edition	:	July 2004	(ASE 12.0, 12.5, 12.5.1 & 12.5.2)
Fourth edition	:	August 2006	(ASE 12.0, 12.5.4 & 15.0.1)
Fifth edition	:	April 2009	(ASE 12.5, 15.0.3 & Cluster Edition)

Sypron is a registered trademark of Sypron B.V.

Sybase, Transact-SQL, Adaptive Server Enterprise and Replication Server are registered trademarks of Sybase, Inc. Other product or brand names may be (registered) trademarks of their respective owners.

Copyright © 2001-2009 Sypron B.V. All rights reserved. No part of this publication may be reproduced in any form, or by any means, without the prior written permission of the publisher.

While this book has been prepared with care, neither the author, nor the publisher, nor Sybase Inc. (nor its subsidiaries), assume any responsibility for errors or omissions, nor do they accept any liabilities for damages resulting from the use of the information herein.

# Contents

Introduction .....	5
About this Quick Reference Guide .....	5
ASE versions covered in this edition .....	5
How complete is 'Complete' ? .....	5
How dynamic is 'Dynamic' ? .....	6
How to use this Quick Reference Guide .....	6
Syntax conventions .....	6
Errata .....	7
Electronic version of code samples .....	7
Undocumented commands: warning & disclaimer .....	7
<b>Developer Topics</b> .....	<b>8</b>
1. Datatypes .....	8
2. Identifiers .....	9
3. Quotes & Comments .....	9
4. Operators .....	9
5. Datatype conversion functions .....	10
6. Date & Time .....	10
7. String functions .....	12
8. System functions .....	14
9. Mathematical functions .....	19
10. Trigonometric functions .....	20
11. Login/user/role functions .....	20
12. Miscellaneous functions .....	21
13. Text & Image data .....	21
14. Aggregate functions .....	22
15. DML statements .....	23
16. Joins .....	26
17. Tables .....	26
18. Table Partitioning .....	28
19. Encrypted Columns (12.5.4, 15.0.2) .....	29
20. Indexes .....	32
21. Referential Integrity Constraints .....	33
22. Object storage properties .....	34
23. Views .....	35
24. Rules & Defaults .....	36
25. Stored Procedures .....	36
26. SQL Functions (15.0.2) .....	37
27. Triggers .....	38
28. Identity columns .....	39
29. Programming & flow control .....	41
30. Cursors .....	43
31. Example of cursor programming .....	44
32. Transactions .....	45
33. Example of transaction programming .....	46
34. Java in ASE (SQL) .....	46
35. XML (XPath/XQuery/SQLX) in ASE (12.5.1) .....	47
36. Web Services (15.0) .....	50
37. Miscellaneous stored procedures .....	52
38. Settings affecting query results .....	53
39. Settings affecting query plans .....	54
40. Advanced settings affecting query plans .....	55
41. Settings for displaying query plan information .....	56
42. Statement cache (12.5.2) .....	57
43. Miscellaneous settings .....	58
<b>DBA Topics</b> .....	<b>59</b>
44. Database devices & dump devices .....	59
45. Device mirroring .....	60
46. Creating & maintaining databases .....	61
47. Temporary databases (12.5.0.3) .....	62
48. Dumping & loading databases .....	63
49. Segments .....	66
50. Thresholds .....	67
51. Logins (ASE server 'users') .....	67
52. Database users .....	70

53.	Roles .....	71
54.	Permissions .....	72
55.	Row-level access control .....	74
56.	Locking .....	75
57.	Managing server configuration options.....	77
58.	Memory allocation.....	78
59.	Named caches & buffer pools .....	79
60.	Update statistics.....	80
61.	sp_sysmon.....	82
62.	Reorg .....	82
63.	Languages & messages.....	83
64.	Character set & sort order.....	84
65.	Setting up subsystemprocs, pubs3 & sybsyntax.....	85
66.	Configuring parallel processing.....	85
67.	ASE Cluster Edition (CE).....	87
68.	Miscellaneous DBA commands.....	88
Advanced DBA Topics.....		91
69.	Setting up 'dbccdb' for 'dbcc checkstorage'.....	91
70.	'dbccdb' stored procedures .....	91
71.	Supported DBCC commands.....	93
72.	Unsupported DBCC commands.....	95
73.	ASE Traceflags.....	100
74.	Configuring remote access with RPCs.....	102
75.	Configuring remote access with CIS.....	103
76.	Component Integration Services (CIS) features.....	103
77.	XP Server .....	106
78.	Lock promotion .....	107
79.	Auditing.....	108
80.	Database recovery.....	110
81.	Resource limits ('Resource Governor').....	111
82.	Logical Process Manager ('Execution Classes').....	112
83.	Abstract Plans.....	113
84.	Query metrics capture (15.0).....	114
85.	MDA tables / monitoring tables (12.5.0.3).....	115
86.	Advanced configuration parameters.....	115
87.	Replication Server.....	116
88.	ASE Replicator.....	116
89.	Job Scheduler (12.5.1).....	116
90.	Shared memory dumps.....	117
91.	The interfaces file.....	118
92.	Server programs .....	118
93.	Client programs.....	120
94.	ASE environment variables.....	124
95.	Running servers as Windows NT services.....	125
96.	Global variables: Session-specific.....	125
97.	Global variables: Server-wide, non-static.....	127
98.	Global variables: Server-wide, static.....	128
99.	Logical keys .....	129
100.	Catalog stored procedures.....	129
101.	Issues with BCP-in.....	130
102.	Minimally logged operations.....	130
103.	Monitor Server & Historical Server.....	130
104.	ASE limits .....	131
105.	ASE licensing.....	131
106.	Essential DBA tasks.....	133
107.	ASE resources on the Internet.....	133
Index.....		134

## 9. Mathematical functions

### **abs** ( *number* )

Returns the absolute value of a given expression. Example: **select abs(-3)** returns 3.

### **ceiling** ( *number* )

Returns the smallest integer greater than or equal to *number*.

Example: **select ceiling(1.3)** returns 2.

### **floor** ( *number* )

Returns the largest integer that is less than or equal to *number*.

Example: **select floor(1.9)** returns 1.

### **exp** ( *number* )

Returns the exponential of *number*. Example: **select exp(1)** returns 2.718282 (use **convert()** to get the full precision of 2.7182818284590451).

### (15.0.2) **hashbytes** ( '*algorithm*', *expression* [, *more expressions*] [, *using options*] )

Generates a hash for the concatenated expressions, using the specified *algorithm*, which can be 'md5', 'sha', 'sha1' (all 'message digests'), 'ptn' (used for ASE semantic hash partitioning), or 'xor32' (used for the statement cache hash key), returning results of **varbinary(16)**, **varbinary(20)**, **varbinary(20)** and **unsigned int** (twice), respectively. *options* specifies pre-hashing byte ordering for platform-independence: **lsb** (little endian), **msb** (big endian), **unicode** (UTF-16), **unicode\_lsb**, **unicode\_msb** (combinations of the above). *options* does not apply for 'ptn'.

Examples: **select hashbytes ( 'md5', 'thisismysecret!' )**

**select hashbytes ( 'ptn', my\_col ) % 5 from my\_tab** -- for 5 hash ptns

### (15.0.2) **hash** ( *expression* [, '*algorithm*' ] )

Broadly similar to **hashbytes()** above, but takes only a single expression, and the result is not platform-independent. *algorithm* can only be 'md5' (=default when omitted), 2 (= md5), 'sha1' or 3 (= sha1) returning results of **varchar(32)**, **varbinary(16)**, **varchar(40)** and **varbinary(20)**, respectively.

### **log** ( *number* ) / **log10** ( *number* )

Returns the natural logarithm, or base 10 logarithm, of *number*.

### **pi** ( )

Returns the constant *pi*. **select pi()** returns 3.141593 (use **convert()** to get the full precision of 3.1415926535897931).

### **power** ( *value*, *power* )

Returns *value* to the power of *power*. Example: **select power( 2 , 8 )** returns 256.

### **rand** ( [ *integer* ] )

Returns a random float value between 0 and 1, optionally using *integer* as a seed. For example, **select rand()** might return 0.802937 (or something else...).

### (15.0.1) **rand2** ( [ *integer* ] )

Similar to **rand()**, but in the select list, **rand2()** is evaluated for every row in the result set (**rand()** is usually evaluated only once for the entire result set).

### (12.5.1) **newid** ( [ *flag* ] )

Returns a unique 16-byte hexadecimal UUID/GUID value (by definition, UUIDs/GUIDs are always unique, even in different servers). *flag* specifies the result format: when 0 (=default), it's a **varchar(36)** string; when 1: like 0, but 4 dashes are included; when 0x0, as a **varbinary(16)** string. Other values for *flag* are invalid and return NULL.

NB: **newid()** is also supported in 12.5.0.3, but may return duplicate values in certain queries; it also accepts different parameters than in 12.5.1, so best upgrade to 12.5.1. Example: **select newid(1)** returned '519a220b-ae98-473e-be5d-fb726ecc1240' in my ASE server – you should never see this same value returned in your server.

### **round** ( *number*, *integer* )

Rounds the *number* so that it has *integer* significant digits. A positive integer determines the number of significant digits to the right of the decimal point; a negative integer, the number of significant digits to the left of the decimal point.

### **sign** ( *number* )

Returns the sign of *number*: positive (+1), zero (0), or negative (-1).

**create func[tion]** *sqlj\_function\_name* ( [ *sql\_parameter\_name sql\_datatype* [, ...more parameters...] ) **returns** *sql\_datatype* [**modifies sql data**] [ { **returns NULL | called** } **on NULL input** ] [ **[not] deterministic**] [**exportable**] **language java parameter style java external name** '*java\_method\_name* [ ( [ *java\_datatype* [, ...more java datatypes...]] ) ]'

Creates a Java function (for SQL functions, >p.37.), mapping to a static method in a Java class in the current database. When any actual parameter is **NULL**, and with **returns NULL on NULL input** specified, the function returns **NULL** without invoking the method; with **called on NULL input** (=default), the function is executed. **modifies sql data** and **[not] deterministic** are for SQLJ compatibility and can be omitted. With **exportable**, the function resides on a remote server and will be executed via OS.

Example: **create function my\_func(par1 int) returns java.lang.String language java parameter style java external name 'MyClass.MyMethod'**  
**select my\_func(col\_1) from my\_table**

**drop func[tion]** *sqlj\_function\_name* [, ...more functions... ]  
 Drops the specified Java function(s) from the current database.

**create proc[edure]** *sqlj\_procedure\_name* ( [ **in | out | inout** ] *sql\_param\_name sql\_datatype* [(12.5.2) = *default\_value*] [, ...more parameters...] ) [**dynamic result sets nr\_result\_sets**] [**modifies sql data**] [**[not] deterministic**] **language java parameter style java external name** '*java\_method\_name* [ ( [ *java\_datatype* [,...more java datatypes...]] ) ]'

Creates a Java stored procedure, mapping to a static method in a Java class in the current database. **in** (=default), **out** and **inout** specify an input-only, output-only and input-output parameter, respectively. **dynamic result sets** specifies the max number of SQL result sets generated; specifying **0** is identical to omitting this clause. **modifies sql data** and **[not] deterministic** are for SQLJ compatibility and can be omitted.

Example: **create procedure my\_proc(par1 int) dynamic result sets 1 language java parameter style java external name 'MyClass.MyOtherMethod'**

**drop proc[edure]** *procedure\_name* [, ...more procedures... ]  
 Drops the specified Java procedure(s) from the current database.

**set stringsize** *nr\_of\_characters*

Defines the maximum number of characters returned by the **toString()** Java method before truncation. **@@stringsize** contains the current setting; default=50.

To install a Java class named **MyClass** into an ASE database, follow these steps:

1. Outside ASE, create an uncompressed JAR file:  
**javac MyClass.java** (→ produces file **MyClass.class**)  
**jar [-cf0 MyJar.jar MyClass.class]** (→ produces file **MyJar.jar**)
2. (Option A): install the JAR file with **installjava** (Windows: **instjava**) (>p.123):  
**installjava -f MyJar.jar -S MYSERVER -U mylogin -P mypasswd -D my\_db**
2. (Option B, SQL): **install java [ update ] from file 'JAR\_or\_ZIP\_file\_pathname'** (undocumented) Installs the Java class in the specified JAR or ZIP file into the current database. With **update**, overwrites an already installed class.

**remove java** { **class** *class\_name* [, ...more classes...] | **package** *package\_name* [,...more packages...] | **jar** *jar\_name* [,...more jars...] [ **retain classes** ] }  
 Drops the specified classes, packages or JARs from the current database. For JARs, the related classes are dropped as well, unless **retain classes** is specified.  
 To extract a JAR and its Java classes, use **extractjava** (Windows:**extrjava**); >p.122.

In pre-15.0.3, a debugger for Java-in-ASE is in **\$\$SYBASE/\$SYBASE\_ASE/debugger /Debug.jar**. To start, run **java sybase.vm.Debug** (from the command line).

In 15.0.3, any Java-compliant debugger can be used, like **jdb** (requires installing the JDK). See the ASE manual *Java in Adaptive Server Enterprise* for details.

## 35. XML (XPath/XQuery/SQLX) in ASE (12.5.1)

In 12.5.1, native XML processing was introduced in ASE. Compared with the pre-12.5.1 Java-based XML features (not covered in this book), the native XML engine offers more functionality (ANSI SQLX; XPath/XQuery queries), better performance, and easier setup/configuration. In 12.5.1, ASE supports XPath queries (>p.50).

For full details about the XML/XPath/SQLX functionality, see the manual *XML Services in Adaptive Server Enterprise*.

**sp\_configure 'enable xml', { 0 | 1 }** (dynamic)

Enables (1) or disables (0) XML processing features. To enable, in pre-15 only, the **ASE\_XML** option must be licensed.

**dbcc traceon** [ ( *traceflag* [, *traceflag*... ] ) ]

**dbcc traceoff** [ ( *traceflag* [, *traceflag*... ] ) ]

Enable / disable traceflags. If nothing is specified in **dbcc traceoff**, all previously enabled flags are disabled. In 12.5.4 and 15.0.2, **set switch** can be used instead.

**dbcc tune** ( *option*, *value\_1*, *value\_2*... )

Modify certain run-time configuration settings. These settings are not persistent (i.e. not maintained after an ASE restart) unless specified otherwise. Possible options are:

- ('**ascinserts**', *value*, *table\_name*) - only for APLs tables with a composite, clustered index, when set to **1** (default=**0**) reduces page splits for inserts in ascending key order. This setting is persistent. In 15.0.2, this property can also be set with **sp\_chgattribute**, and displayed with **sp\_help**.
- ('**cleanup**', *value*) - if *value* = **0**, memory cleanup checking is enabled (=default); if *value* = **1**, disabled.
- ('**cpuaffinity**', *firstcpu*, { '**on**' | '**off**' }) - enables/disables CPU affinity, provided this is supported by the platform; starting with engine 0, engines are bound to CPU *firstcpu*. Run ('**cpuaffinity**', **-1**) to write current setting to server errorlog.
- ('**des\_bind**', *db\_id*, *table\_name*) - Binds a table to an object descriptor so that it won't be pushed out of the metadata cache. Use for hot tables when observing Object Manager Spinlock contention. Specify '**des\_unbind**' to remove.
- ('**des\_greedyalloc**', *db\_id*, *table\_name*, { '**on**' | '**off**' }) - enables (**on**) or disables (**off**, =default) 'greedy allocation' for the table (to reduce larch contention).
- ('**deviochar**', *vdevno*, *value*) - sets max. outstanding I/Os by housekeeper for device *vdevno* to *value* (**1..255**; default=**3**); if *vdevno* = **-1**; applies to all devices.
- (**pre-12.5.1**) ('**doneinproc**', { **0** | **1** }) - enables (**1**, =default) or disables (**0**) most (but not all) *done-in-proc* network packets. To disable all *done-in-proc* packets, enable traceflag 292 and run **dbcc tune('doneinproc', 0)** (note: disabling all *done-in-proc* packets may cause certain client apps to fail). In 12.5.1, this command is replaced by the config option '**send doneinproc tokens**'.
- (**pre-12.5.1**) ('**maxwritedes**', *value*) - sets max. no. of outstanding disk writes; default=**10**. In 12.5.1, replaced by **sp\_configure 'i/o batch size'** (dynamic).

## 72. Unsupported DBCC commands

**WARNING!** These (arbitrarily selected) DBCC commands represent unsupported functionality which may have unexpected side effects, may cause irreversible damage to your databases and/or lead to loss of data. Use entirely at your own risk. Do **not** contact Sybase Technical Support for assistance!

Many **dbcc** commands require that the role **sybase\_ts\_role** be enabled for the executing session. This can be done as follows (assuming the login is 'sa'):

```
grant role sybase_ts_role to sa
set role sybase_ts_role on
```

**dbcc help** ( *command* )

Displays syntax info for the specified **dbcc** command.

(12.5.0.3) **dbcc addtempdb** ( *db\_id* | *db\_name* )

Adds a temporary database to the global list of temporary databases.

**dbcc allocdump** ( *db\_id* | *db\_name*, *alloc\_page\_nr* )

Lists extents in an allocation unit, identified by the allocation page number.

**dbcc bytes** ( *address*, *length* ) / **dbcc bytes** ( *address*, *printopt*, *structure\_name* )

Dumps *length* bytes, starting at *address* (e.g. a physical page address), in hex & ASCII. **dbcc bytes(0,0,'showlist')** displays all structures that can be specified as masks in **dbcc bytes(address, printopt, structure\_name)**. Specify **-1** for *printopt*.

**dbcc cachremove**( *db\_id* | *db\_name*, *object\_id* | *object\_name* )

Deallocates the object descriptor (DES) for the specified table.

(12.5.3) **dbcc dbccachremove** ( *db\_id* | *db\_name* )

Clears the DBTABLE information for the specified database, for use when commands fail with a 'keep count' reported as > 0 although no users are active in the database.

(15.0.3) **dbcc cachedataremove** ( *db\_id* | *db\_name* [, *object\_id* | *object\_name* [, *ix\_id* | *ix\_name* [, *ptn\_id* | *ptn\_name* ]]] )

Removes all pages for the specified database/object/partition from the data cache.

## Miscellaneous Topics

### 91. The interfaces file

The **interfaces** file is an essential part of the Sybase client-server environment. For client applications, it must contain the network address for every server the client application connects to. To start a server, it must be able to find its own network address in the **interfaces** file. By default, the **interfaces** file is **\$\$SYBASE/interfaces** (on Windows: **%SYBASE%\INI\SQL.INI**). When named or located differently, it must be specified with a command-line parameter for the client or server program (see the following sections).

An LDAP server can replace the **interfaces** file for looking up the server network address by client applications; no license option is needed for this. For user authentication by an LDAP server instead of **syslogins**, license option **ASE\_DIRS** is needed. Best use the **dsedit** or **dscp** utilities to edit the **interfaces** file. Note that the **interfaces** file (for Unix) and the **SQL.INI** file (for Windows) use incompatible formatting.

**dscp** (Unix only) - an ASCII-interface utility to view/edit the **interfaces** file. **dscp** has its own command set. At the prompt, type **help** for on-line help.

**dsedit** - a GUI utility to view/edit the **interfaces** file.

### 92. Server programs

For all programs, the option **-v** displays the software version.

**backupserver** (Windows: **bcksvr**) - Backup Server

[**-C nr\_connections**] max. # Backup Server connections (default=30)  
 [**-S server\_name**] server name (default=**\$\$DSDLISTEN**, otherwise **SYB\_BACKUP**)  
 [**-I interfaces\_file**] **interfaces** file pathname (default=**\$\$SYBASE/interfaces**)  
 [**-e errorlog\_file**] Backup Server errorlog (default=**backup.log**)  
 [**-M sybmultbuf**] pathname of the **sybmultbuf** binary file  
 [**-N net\_connections**] max. # of Backup Server network connections (default=25)  
 [**-L language**] language used by Backup Server  
 [**-J character\_set**] character set used by Backup Server  
 [**-P active\_threads**] max. # active stripes (for multiple dump/load sessions)  
 [**-c tape\_config\_file**] tape configuration file (default=**\$\$SYBASE/backup\_tape.cfg**)  
 [**-V0 | -V1 | -V2 | -V3**] level of detail for error logging (lower = more detail).  
 [**-T traceflag**] boot-time traceflags (multiple options are allowed)  
 [**-m max\_MB**] max. amount of memory (MB) to be used by Backup Server (default= (number of active stripes)\*1MB)  
 [**-p packet\_size**] TDS packet size (in bytes; default=2048) requested by a local Backup Server from a remote Backup Server for a remote dump/load. Both servers must allow the requested value.

(pre-12.5) **buildmaster** (Windows: **bidmstr**) - Buildmaster (for (re)building a new master device). In 12.5, **buildmaster** is removed, and replaced by **dataserver -z -b**.

**dataserver** (Windows: **sqlsvr**) - the ASE server itself. The options marked '(build)' are for building a new ASE server.

[**-h**] (help function) displays all possible parameters  
 [**-d master\_device**] master device pathname  
 [**-r master\_mirror**] master device mirror pathname (when master is mirrored)  
 [**-s server\_name**] server name (default= **\$\$DSDLISTEN**, otherwise **SYBASE**)  
 [**-c config\_file**] server config. file (default= **\$\$SYBASE/server\_name.cfg**)  
 [**-e errorlog\_file**] ASE server errorlog (default=**errorlog**)  
 [**-m**] boot server in standalone (single-user) mode  
 [**-M sharedmem\_dir**] shared memory directory  
 [**-i interfaces\_file\_dir**] directory containing **interfaces** file (default=**\$\$SYBASE**)  
 [**-T traceflag**] boot-time traceflags (multiple **-T** options can be specified)  
 [**-a keytab\_file**] CAPs directive filename  
 [**-G logserver\_name**] specifies a server name for event logging  
 [**-g**] disables event logging  
 [**-H**] specified when using the High Availability feature (ASE\_HA)  
 [**-K keytab\_file**] keytab filename (when using DCE)  
 [**-k principal\_name**] (12.5.4,15.0.2) server's principal name (used with Kerberos)  
 [**-P**] (Windows) start ASE server with high priority(default=medium)  
 [**-p login\_name**] generates a new random password (printed on the console) for the specified login (which must have been granted **ssorole**)  
 [**-y**] prompts for SSL certificate password (use with **sp\_ssladmin**)

# Index

## Legend

- Keywords followed by **' , sp\_ '** are stored procedures where the **sp** prefix has been chopped off to make a better reference; for example, **addserver sp\_** refers to **sp\_addserver**. This is done for other commands as well; for example **checkdb, dbcc** refers to **dbcc checkdb**, and **showplan, set** to **set showplan**.
- Keywords followed by **(config)** are server configuration options, settable with **sp\_configure**. Keywords followed by **(DB option)** are database options, settable with **sp\_dboption**.

## @

@@active\_instances ..... 128

@@authmech ..... 14, 125

@@bootcount ..... 128

@@boottime ..... 128

@@bulkarraysize ..... 126

@@bulkbatchsize ..... 126

@@char\_convert ..... 126

@@cis\_rpc\_handling ..... 126

@@cis\_version ..... 103, 128

@@client\_csexpansion ..... 126

@@client\_csid ..... 126

@@client\_csname ..... 126

@@clusterboottime ..... 128

@@clustercoordid ..... 128

@@clustermode ..... 128

@@clustername ..... 128

@@connections ..... 127

@@cpu\_busy ..... 128

@@curluid ..... 126

@@cursor\_rows ..... 44, 126

@@datefirst ..... 12, 126

@@dbts ..... 127

@@error ..... 126

@@errorlog ..... 128

@@fetch\_status ..... 44, 127

@@identity ..... 40, 126

@@idle ..... 128

@@instanceid ..... 128

@@instancename ..... 128

@@io\_busy ..... 128

@@isolation ..... 77, 126

@@langauge ..... 126

@@langid ..... 126

@@language ..... 126

@@lastkpgendate ..... 128

@@lastlogindate ..... 126

@@lock\_timeout ..... 76, 126

@@logintrigger ..... 69, 128

@@max\_connections ..... 128

@@max\_precision ..... 128

@@maxcharlen ..... 126

@@maxpagesize ..... 128

@@monitors\_active ..... 128

@@ncharsize ..... 126

@@nestlevel ..... 37, 39, 126

@@nextkpgendate ..... 128

@@nodeid ..... 128

@@optgoal ..... 54, 126

@@options ..... 126

@@opttimeoutlimit ..... 56, 126

@@pack\_received ..... 128

@@pack\_sent ..... 128

@@packet\_errors ..... 128

@@pagesize ..... 129

@@parallel\_degree ..... 86, 127

@@procid ..... 37, 127

    in a trigger ..... 39

@@qpmode ..... 54, 127

@@quorum\_physname ..... 129

@@recovery\_state ..... 110, 128

@@repartition\_degree ..... 56, 127

@@resource\_granularity ..... 56, 127

@@rowcount ..... 127

    in a cursor ..... 44

    in a trigger ..... 39

@@scan\_parallel\_degree ..... 86, 127

@@servername ..... 129

@@setrowcount ..... 54, 127

@@spid ..... 127

@@sqlstatus ..... 44, 127

@@ssl\_ciphersuite ..... 127

@@stringsize ..... 47, 127

@@sys\_tempdbid ..... 63

@@system\_busy ..... 128

@@system\_view ..... 87

@@tempdbid ..... 62, 127

@@textcolid ..... 22, 127

@@textdbid ..... 22, 127

@@textobjid ..... 22, 127

@@textptr ..... 22, 127

@@textptr\_parameters ..... 127

@@textsize ..... 127

@@textts ..... 22, 127

@@thresh\_hysteresis ..... 67, 129

@@timeticks ..... 115, 129

@@total\_errors ..... 128

@@total\_read ..... 128

@@total\_write ..... 128

@@tranchained ..... 127

@@trancount ..... 45, 127

@@transactional\_rpc ..... 127

@@transtate ..... 45, 127

@@unicharsize ..... 127

@@user\_busy ..... 128

@@version ..... 129

@@version\_as\_integer ..... 129

@@version\_number ..... 129

## A

abort tran on log full (DB option) ..... 62

abs() ..... 19

abstract plan (show) ..... 56

abstract plan cache (config) ..... 113

abstract plan dump (config) ..... 113

abstract plan load (config) ..... 113

abstract plan replace (config) ..... 113

Abstract Plans ..... 113

Abstract Query Plan.. See Abstract Plans

helpthreshold, sp_.....	67	installpcidb.....	46		
helpuser, sp_.....	71	installpubs3, intpbs3.....	85		
hextobigint().....	10	installsecurity, instsecur.....	108		
hextoint().....	10	installws.....	50		
hidetext, sp_.....	53	instance, in ASE CE.....	87		
histogram tuning factor (config).....	81	instance_id().....	16		
Historical Server.....	130	instance_name().....	16		
histserver.....	119	instjava (tool).....	47, 123		
histsvr.....	119	interfaces file.....	118		
HK CHORES thread.....	116	International Sybase User Group.....	133		
HK GC thread.....	83, 116	intohex().....	10		
HK WASH thread.....	116	io_cost (resource limit).....	111		
holdlock.....	76	IP address of a client.....	17, 52		
host_id().....	15	is_quiesced().....	16, 66		
host_name().....	15	is_sec_service_on().....	16		
housekeeper free write percent(conf).....	116	isdate().....	12		
HOUSEKEEPER thread.....	83, 116	isnull().....	21		
hs_shutdown.....	131	isnumeric().....	13		
<b>I</b>				isolation level (transactions).....	17
i/o batch size (config).....	115	isolation level (transactions).....	76		
I/O fencing.....	87	isql.....	122		
i/o polling process count (config).....	115	isql subcommands.....	122		
Identifiers.....	9	istraceon_dbcc.....	97		
identity.....	27, 39	ISUG.....	133		
identity burning set factor (config).....	40	<b>J</b>			
identity columns.....	39	Java in ASE.....	46		
identity grab size (config).....	40	jisql.....	123		
identity in nonunique index(DB opt.).....	41, 62	job scheduler.....	116		
identity reservation size (config).....	40	join transitive closure (jtc).....	55		
identity().....	15, 40	join, ANSI.....	26		
identity_burn_max (sp_chgattribute).....	40	join, outer.....	26		
identity_burn_max().....	15, 39	Joins.....	26		
identity_gap.....		jreconfig, sp_.....	46		
create table.....	39	js_admin_role.....	71		
select...into.....	40	js_client_role.....	71		
sp_chgattribute.....	40	js_user_role.....	71		
identity_insert, set.....	40	jsagent.....	116		
identity_update, set.....	40	jtc, set.....	55		
idle migration timeout (config).....	87	<b>K</b>			
if.....	42	keep count.....	95		
ignore_dup_key.....	32	key copies, of encryption key.....	30		
ignore_dup_row.....	32	keycustodian_role.....	71		
image/text functions.....	21	kill.....	88		
import_qpgroup, sp_.....	114	<b>L</b>			
in (operator).....	9	langinst.....	123		
in (subquery operator).....	24	langinstall.....	123		
in row (Java).....	27, 46	language cursor.....	43		
index, forcing.....	24	language, set.....	84		
index, function.....	32	Languages & Messages.....	83		
index_col().....	15, 33	lc_id().....	16		
index_colorder().....	15, 33	lc_name().....	16		
index_name().....	15, 33	lct_admin().....	16		
indexalloc, dbcc.....	94	LDAP.....	118		
Indexes.....	32	left().....	13		
indsuspect, sp_.....	85	len().....	13		
inline view.....	See derived table	license information (config).....	133		
ins_syn_sql.....	85	License options/keys, ASE.....	131		
insert...select.....	26	license_enabled().....	16, 132		
insert...values.....	25	like (operator).....	9		
inserted (trigger table).....	38	Limits (of ASE).....	131		
inssynsq.....	85	line continuation character (\).....	9		
install java (command).....	47	list partitioning.....	29		
INSTALL_ALL_PATCH.....	125	list_appcontext().....	16, 74		
installdbextend.....	67				
installjava (tool).....	47, 123				
installjsdb.....	116				
installmaster, instrmstr.....	85				
installmontables.....	115				

System functions ..... 14  
 system local tempdb ..... 63  
 System roles ..... 71  
 system\_view, set ..... 87  
 systemwide password expir.(config) **68,72**

**T**

tab\_suspectptn, sp\_ ..... 29  
 table count, set ..... 55  
 table partitioning ..... 28  
 table\_privileges, sp\_ ..... 130  
 tablealloc, dbcc ..... 94  
 Tables ..... 26, 28  
 tables, sp\_ ..... 130  
 tan() ..... 20  
 tcp no delay (config) ..... 115  
 TCP/IP address of a client ..... 52  
 tempdb, sp\_ ..... 63  
 tempdb\_id() ..... 18, **62**  
 tempdb\_markdrop, sp\_ ..... 63  
 tempdb\_space (resource limit) ..... 111  
 Temporary databases ..... 62  
     local, system, global (CE) ..... 63  
 text prefetch size (config) ..... 22  
 text/image functions ..... 21  
 textalloc, dbcc ..... 94  
 textptr() ..... 21  
 textptr\_parameters, set ..... **22, 106**  
 textsize, set ..... 22  
 textvalid() ..... 22  
 thresholdaction, sp\_ ..... 67  
 Thresholds ..... 67  
 thresholds, dbcc ..... 99  
 Time & Date ..... 10  
 time range ..... 114  
 time slice (config) ..... 115  
 time zone information ..... 11  
 timeouts (server option) ..... 102  
 timeslice error ..... 115  
 timestamp ..... **8, 21, 128**  
 to\_unichar() ..... 14  
 top N, select ..... 23  
 total data cache size (config) ..... 78, 79  
 total logical memory (config) ..... 78  
 total memory (config) ..... 79  
 total physical memory (config) ..... 78  
 tracefile, set ..... 58  
 Traceflags ..... 100  
 traceflags, dbcc ..... 99  
 traceoff, dbcc ..... 95  
 traceon, dbcc ..... 95  
 tran\_dumpable\_status() ..... **18, 64**  
 transaction isolation level ..... 17, 76  
 transaction isolation level, set ..... 76  
 transactional\_rpc, set ..... 106  
 Transactions ..... 45  
 transactions, sp\_ ..... 45  
 trigger  
     for-trigger (on a table) ..... 38  
     global login trigger ..... 69  
     instead-of-trigger (on a view) ..... 38  
     login trigger (login specific) ..... 69  
 triggers, set ..... 38  
 Trigonometric functions ..... 20  
 trunc log on chkpt (DB option) ..... 62  
 truncate table ..... 26  
 tsequal() ..... 21  
 tune, dbcc ..... 95

**U**

UAF (Unified Agent Framework) ..... 120  
 uafstartup.sh, uafshutdown.sh ..... 120  
 uhighsurr() ..... 14  
 ulowsurr() ..... 14  
 unbindcache, sp\_ ..... 79  
 unbindcache\_all, sp\_ ..... 79  
 unbindexclass, sp\_ ..... 112  
 unbindmsg, sp\_ ..... **34, 84**  
 unbindrule, sp\_ ..... 36  
 Undocumented commands, warning & disclaimer ..... 7  
 union ..... 25  
 Unions in views ..... 35  
 unique (constraint) ..... 33  
 unique (index) ..... 32  
 unique auto\_identity index (DB opt.) **41, 62**  
 unlocking a login  
     (or a role) with dataserver -u ..... 119  
     'sa', with traceflag 4044 ..... 101  
     with sp\_locklogin ..... 70  
 unlogged operations ..... 130  
 unmount database ..... 66  
 unsigned datatypes ..... 8  
 Unsupported DBCC commands ..... 95  
 update ..... 25  
 update (cursors) ..... 43  
 update all statistics ..... 80  
 update index statistics ..... 80  
 update partition statistics ..... 81  
 update statistics ..... 80  
 update table statistics ..... 81  
 update where current of ..... 43  
 update() ..... 38  
 updateconfig, sp\_dbcc ..... 93  
 upgrade\_object, dbcc ..... 100  
 upper() ..... 14  
 uscalar() ..... 14  
 use ..... 43  
 used\_pages() ..... 18  
 used\_pgs() ..... 18  
 usedextents, dbcc ..... 99  
 user ..... **20, 70**  
 user log cache size (config) ..... 116  
 user log cache spinlock ratio (config) ..... 116  
 user\_id() ..... **21, 70**  
 user\_name() ..... **21, 70**  
 user\_stats, dbcc ..... 100  
 user-defined functions ..... 37  
     Java functions ..... 47  
 user-defined roles ..... 71  
 users (database users) ..... 70  
 UTC time, getutcdatetime() ..... 11  
 UUID - newid() ..... 19

**V**

valid\_name() ..... 18  
 valid\_user() ..... 21  
 var() ..... 23  
 var\_pop() ..... 23  
 var\_samp() ..... 23  
 variable assignment ..... 23  
 variance() ..... 23  
 variance, aggregate ..... 23  
 varp() ..... 23  
 version ID of ASE (@@version) ..... 129  
 version, sp\_ ..... 85  
 Views ..... 35