

The Complete Sybase ASE Quick Reference Guide

***ASE versions 12.0, 12.5.4 & 15.0.1
4th 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 12.0, 12.5.4 & 15.0.1

4th edition

by Rob Verschoor

ISBN 90-806117-1-9

Published by:

Sypron B.V.
P.O. Box 10695
2501 HR Den Haag
The Netherlands
Internet www.sypron.nl
Email 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)

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 © 2001-2006 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
ASE server page size.....	5
How complete is 'Complete' ?	6
How dynamic is 'Dynamic' ?.....	6
How to use this Quick Reference Guide.....	6
Syntax conventions	6
Acronyms.....	7
Errata	7
Electronic version of code samples.....	7
Undocumented commands: warning & disclaimer.....	7
Developer Topics.....	8
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.....	18
10. Trigonometric functions	18
11. Login/user/role functions.....	19
12. Miscellaneous functions	20
13. Text & Image data	20
14. Aggregate functions.....	21
15. DML statements	21
16. Joins	24
17. Tables	25
18. Encrypted Columns (12.5.3a).....	28
19. Indexes	30
20. Referential Integrity Constraints.....	31
21. Object storage properties	32
22. Views	33
23. Rules & Defaults	33
24. Stored Procedures.....	34
25. Triggers.....	35
26. Identity columns.....	37
27. Programming & flow control	38
28. Cursors	40
29. Example of cursor programming.....	42
30. Transactions	42
31. Example of transaction programming.....	43
32. Locking	44
33. Java in ASE (SQL).....	46
34. XML (XPath/XQuery/SQLX) in ASE (12.5.1)	47
35. Web Services (15.0).....	49
36. Miscellaneous stored procedures.....	51
37. Settings affecting query results	52
38. Settings affecting query plans.....	53
39. Settings for displaying query plan information	55
40. Miscellaneous settings	56
DBA Topics.....	57
41. Database devices & dump devices	57
42. Device mirroring	58
43. Creating & maintaining databases	58
44. Temporary databases (12.5.0.3).....	60
45. Dumping & loading databases	61
46. Segments.....	63
47. Thresholds.....	64
48. Logins	64
49. Database users	67
50. Roles.....	68

51.	Permissions	69
52.	Row-level access control (12.5)	71
53.	Managing server configuration options	72
54.	Memory allocation in ASE 12.0	74
55.	Memory allocation in ASE 12.5+	74
56.	Named caches & buffer pools	74
57.	Update statistics	76
58.	sp_sysmon	77
59.	Reorg	78
60.	Languages & messages	78
61.	Character set & sort order	80
62.	Setting up sybsyntax, pubs3 & sybssystemprocs	80
63.	Configuring parallel processing	81
64.	Miscellaneous DBA commands	82
Advanced DBA Topics		84
65.	Setting up the 'dbccdb' database	84
66.	'dbccdb' stored procedures	84
67.	Supported DBCC commands	86
68.	Unsupported DBCC commands	88
69.	ASE Traceflags	93
70.	Configuring remote access with RPCs	94
71.	Configuring remote access with CIS	95
72.	Component Integration Services (CIS) features	95
73.	XP Server	99
74.	Lock promotion	99
75.	Auditing	100
76.	Database recovery	102
77.	Resource limits ('Resource Governor')	103
78.	Logical Process Manager ('Execution Classes')	104
79.	Abstract Query Plans	105
80.	Query metrics capture (15.0)	106
81.	MDA tables / monitoring tables (12.5.0-3)	107
82.	Advanced configuration parameters	107
83.	Replication Server	108
84.	ASE Replicator (12.5)	108
85.	Job Scheduler (12.5.1)	108
86.	Enterprise Java Beans (12.5 only)	108
87.	Shared memory dumps	109
Miscellaneous Topics		110
88.	The interfaces file	110
89.	Server programs	110
90.	Client programs	112
91.	ASE environment variables	115
92.	Running servers as Windows NT services	116
93.	Global variables: Session-specific	117
94.	Global variables: Server-wide, non-static	118
95.	Global variables: Server-wide, static	119
96.	Logical keys	120
97.	Catalog stored procedures	120
98.	Issues with BCP-in	121
99.	Minimally logged operations	121
100.	Monitor Server & Historical Server	121
101.	ASE limits	122
102.	ASE licensing	122
103.	Essential DBA tasks	124
104.	ASE resources on the Internet	124
Index		125

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).

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...).

(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**.

newid() is also supported in 12.5.0.3, but may return duplicate values in certain queries; it also accepts different parameters. Best upgrade to 12.5.1 when using **newid()**. 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).

(12.5.0.3) square (number)

Returns the square of the specified value. Example: **select square(-3)** returns 9.

sqrt (number)

Returns the square root of the specified value, which must be ≥ 0 .

Example: **select sqrt(36)** returns 6.

10. Trigonometric functions

acos (number) / asin (number)

Returns the angle (in radians) having a cosine or sine of *number*.

Example: **select degrees(acos(0.5))** returns 60.000000.

atan (number)

Returns the angle (in radians) having a tangent of *number*.

Example: **select degrees(atan(1.0))** returns 45.000000.

atan2 (number1, number2)

Returns the angle (in radians) having a tangent of (*number1* / *number2*). *number1* typically represents the angle's sine, and *number2* its cosine.

cos (angle) / cot (angle) / sin (angle) / tan (angle)

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

(12.5) drop proc[edure] procedure_name [, ...more procedures...]

Drops the specified Java procedure(s) from the current database.

(12.5) 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 using the **installjava** tool (NT: **instjava**) (→ p.114):
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** (NT: **extrjava**); → p.113.

A GUI debugger for Java-in-ASE is in `$$SYBASE/$SYBASE_ASE/debugger/Debug.jar` (must be in the CLASSPATH). To start, run: **java sybase.vm.Debug** (from the command line). See the ASE manual *Java in Adaptive Server Enterprise* for details.

34. 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.49). 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.

select ...rest of statement...

for xml [(15.0) schema | all] [(15.0) returns datatype] [option 'sqlx_options...']

For a **select** statement only, specifying **for xml** converts the SQL result set to a SQLX-XML document. **returns** can only be used when the **select** is a subquery.

In 15.0, **for xml schema** generates the XML schema describing the result set generated by **for xml**, but not the result set data itself; **for xml all** generates both schema and data. For *sqlx_options*, see below.

Example: **select * from my_table where col1 > 0 order by col2 for xml option 'columnstyle=attribute statement=yes'**

xmlextract ('xpath_query(/text())', xml_data [option 'options...'] returns datatype)
 Executes the XPath query against the XML data. By default, the result is returned as an XML document; when **(/text())** is added to the XPath query, the outermost XML tags are removed and the result is returned as a scalar value. By default, the result datatype is **text**, unless specified otherwise (as an ASE datatype or **java.lang.String**) with **returns**. *options* can be **xmlerror** and **ncr** (see *sqlx_options* below).

Examples:

select xmlextract('/t', '<doc><t>Hello</t></doc>') returns: '<t>Hello</t>'

select xmlextract('/t/text()', '<doc><t>Hello</t></doc>') returns: 'Hello'

select xmlextract('/t/text()', '<doc><t>1234</t></doc>') returns int)
 returns: 1234 (as an integer)

select xmlextract('/t', xml_col option 'xmlerror=message' returns int)
from your_tab ...rest of query...

xmlparse (xml_data [option 'options...'])

Parses an XML document, returning it as an **image** value. Parsed XML data can be

dbcc cacheremove (db_id | db_name, object_id | object_name)

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

dbcc cis (subcommand)

Displays CIS-related information. Subcommands are:

- **'remcon'** displays all CIS remote connections
- **'showcaps'** [, 'server_name'] - displays the capabilities of the remote server
- **'srvdes'** [, server_id] - without argument, displays all SRVDES structures. With *server_id*, syncs the in-memory SRVDES with **master..sys.servers**
- **(pre-12.5) 'rusage'** - displays CIS shared memory usage

dbcc connection_hangup (remote_server_name)

Closes a site handler connection to the specified remote server (like a Backup Server).

dbcc corrupt (table_name | table_id, index_id, error_nr)

Creates a corruption of the specified *error_nr* in the specified table or index. The table must reside in, and this command must be issued from, a database named **victimdb**. *error_nr* is an error number as used in **dbccdb** (i.e. 100000 and higher).

This command is intended to help testing software for detecting database corruptions.

(12.5.3) dbcc dbcacheremove (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.

dbcc dbinfo (db_name)

Displays the DBINFO structure for the specified database.

dbcc dbrecover (db_id | db_name)

Performs recovery for the specified database (without restarting the ASE server); works only when bit 64 in **sysdatabases.status** is set.

dbcc dbrepair (db_id | db_name, option [, table_name, index_id])

Performs various maintenance actions. Possible *option* values:

- **'dropdb'** - drops a database marked suspect (when **drop database** can't); bits 256 and 64 must be set in **sysdatabases.status**.
- **'findstranded'** - displays data extents located ('stranded') on the logsegment'
- **(12.5) 'fixlogfreespace'** - recalculates amount of free space in **syslogs**.
- **'ltnignore'** - like **dbcc settrunc(ltn, ignore)**, but works for an offline database: can be issued from outside the target database.
- **'repairindex'** - rebuilds a system table index (dangerous!!); best use **sp_fixindex** instead: **sp_fixindex db_name, table_name, index_id**; the database should be in single-user mode before running this.

dbcc dbtable (db_id | db_name)

Displays the DBTABLE structure for the specified database (includes 'keep count').

(12.5.4) dbcc delete_dol_page (db_id, object_id, page_nr [, 'noblock'])

Deallocates a data page (typically, a corrupt page) from a DOL table. Without **'noblock'**, takes an exclusive-table lock.

dbcc des [(db_id | db_name [, object_name | object_id])]

Displays the DES structure for the specified object, or for all objects.

dbcc extentcheck (db_id, table_id, index_id, sort_bit)

Displays all extents of the specified table or index with the 'sort bit' specified (0 or 1).

dbcc extentdump (db_id, page_nr)

Displays the extent on which the specified page is located.

dbcc extenzap (db_id, object_id, index_id, sort_bit)

Delete extents for the specified object and index with the 'sort bit' specified (0 or 1). To delete an object completely, run once with *sort_bit* = 0 and once with *sort_bit* = 1.

dbcc findnotfull extents (db_id, object_id, index_id, sort_bit)

Finds extents for the table or index which contain unused pages; *sort_bit* is 0 or 1.

dbcc fix_al [(db_name)]

Fixes allocation errors within the specified database (default = current database).

Miscellaneous Topics

88. The interfaces file

Assuming LDAP is not used (=default), 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 should connect 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 located in **\$\$SYBASE/interfaces** (on NT: **%SYBASE%\IN\SQL.INI**). It may also be named or located differently, which must then be specified with a command-line parameter for the client or server program (see the following sections). 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 NT) 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.

89. Server programs

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

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

[-C nr_connections] max. # Backup Server connections (default=30)
[-S server_name] server name (default=**\$\$DSLISTEN**, 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 (NT: **bldmastr**) - Buildmaster (for (re)building a new master device). In 12.5, **buildmaster** is removed; its functionality is merged into **dataserver**.

[-d master_device] master device pathname
[-c controller_nr] controller number; always specify 0 (=default)
[-s size] size of master device in 2Kb pages
[-m] rewrites only **master** database (doesn't rebuild master device)
[-q] doesn't clear unallocated pages in **master & model** databases
[-x] rewrites only **model** database (overrides **-q**)

dataserver (NT: **sqlsvr**) - ASE server. In 12.5, this is also used for building a new ASE server (see the options marked '(build)' below).

[-d master_device] master device pathname
[-r master_mirror] master device mirror pathname (when master is mirrored)
[-s server_name] server name (default= **\$\$DSLISTEN**, 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)
[-h] (12.5) 'help' function; displays all possible parameters
[-K keytab_file] keytab filename (when using DCE)
[-k principal_name] (12.5.4) the server's principal name (used with Kerberos)

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**.

@

@@authmech	117	@@procid	35, 118
@@bootcount	119	in a trigger	36
@@boottime	119	@@recovery_state	102, 119
@@bulkarraysize	117	@@repartition_degree	54, 118
@@bulkbatchsize	117	@@resource_granularity	54, 118
@@char_convert	117	@@rowcount	118
@@cis_rpc_handling	117	in a cursor	41
@@cis_version	96, 119	in a trigger	36
@@client_csexpansion	117	@@scan_parallel_degree	81, 118
@@client_csid	117	@@servername	120
@@client_csname	117	@@setrowcount	53, 118
@@connections	118	@@spid	118
@@cpu_busy	119	@@sqlstatus	41, 118
@@curloid	117	@@ssl_cipher_suites	118
@@cursor_rows	41, 117	@@stringsize	47, 118
@@datefirst	12, 117	@@tempdbid	60, 118
@@dbs	119	@@textcollid	21, 118
@@error	117	@@textdbid	21, 118
@@errorlog	119	@@textobjid	21, 118
@@fetch_status	41, 117	@@textptr	21, 118
@@guestuserid	119	@@textptr_parameters	118
@@heapmersize	119	@@textsize	118
@@identity	38, 117	@@texts	21, 118
@@idle	119	@@thresh_hysteresis	64, 120
@@invaliduserid	119	@@timeticks	107, 120
@@io_busy	119	@@total_errors	119
@@isolation	46, 117	@@total_read	119
@@language	117	@@total_write	119
@@langid	117	@@tranchained	118
@@language	117	@@trancount	43, 118
@@lock_timeout	45, 117	@@transactional_rpc	118
@@logintrigger	66, 119	@@transtate	43, 118
@@max_connections	119	@@unicharsize	118
@@max_precision	119	@@version	120
@@maxcharlen	117	@@version_as_integer	120
@@maxgroupid	119	@@version_number	120
@@maxpagesize	120		
@@maxsuid	119		
@@maxuserid	119		
@@mingroupid	119		
@@minsuid	119		
@@minuserid	119		
@@monitors_active	119		
@@ncharsize	117		
@@nestlevel	35, 37, 117		
@@nodeid	119		
@@optgoal	53, 117		
@@options	117		
@@opttimeoutlimit	54, 118		
@@pack_received	119		
@@pack_sent	119		
@@packet_errors	119		
@@pagesize	120		
@@parallel_degree	81, 118		
@@probesuid	119		

A

abort tran on log full (DB option)	59
abs()	18
abstract plan (show)	55
abstract plan cache (config)	105
abstract plan dump (config)	105
abstract plan load (config)	105
abstract plan replace (config)	105
Abstract Query Plans	105
access rules	34, 71
acos()	18
Acronyms	7
activeroles, sp_	69
add_qpgroup, sp_	106
add_resource_limit, sp_	103
add_time_range, sp_	103
addalias, sp_	67
addauditrecord, sp_	100
addaudittable, sp_	100
addengine, sp_	104

select...into (encryption)	29	set proxy	67
select...into existing table	22, 97	set quoted_identifier	9, 56
self_recursion, set	36	set raw_object_serialization	99
semantic partitioning	26	set remote_indexes	99
semi-dynamic config options	6	set repartition_degree	54
send doneinproc tokens (config) ..	88, 107	set replication	108
server cursor	40	set resource_granularity	54
server page size	5	set role	69
Server programs	110	set rowcount	53
server_info, sp_	120	set scan_parallel_degree	81
serverlimits, dbcc	92	set self_recursion	36
serveroption, sp_ (CIS)	96	set session authorization	67
serveroption, sp_ (EJB)	109	set showplan	55
serveroption, sp_ (RPCs)	95	set sort_merge	55
session authorization, set	67	set sort_resources	82
set ansi_permissions	52	set statement_cache	54
set ansinull	52	set statistics_io	55
set arithabort arith_overflow	52	set statistics_plancost	55
set arithabort numeric_truncation ..	52	set statistics_simulate	55
set arithignore	52	set statistics_subquerycache	55
set background	56	set statistics_time	55
set bulk array size	98	set string_truncation	53
set bulk batch size	98	set string_size	47
set chained	43, 114	set switch	93
set char_convert	80	set table_count	54
set ciphertext	29	set textptr_parameters	21, 99
set cis_rpc_handling	94, 98	set textsize	20
set clientapplname	56	set transaction_isolation_level	45
set clienthostname	56	set transactional_rpc	98
set clientname	56	set triggers	36
set close on endtran	41	set_appcontext()	17, 71
set command_status_reporting	87	set_identity_burn_max, dbcc	92
set cursor rows	41	set_qpian, sp_	106
set datefirst	12	set_langalias, sp_	79
set dateformat	12	set_pglockpromote, sp_	100
set delayed_commit	56, 60	set_psex, sp_	104
set export_options	56	set_rowlockpromote, sp_	100
set fipsflagger	56, 113	set_suspect_granularity, sp_	103
set flushmessage	56	set_suspect_threshold, sp_	103
set fmtonly	52	set_trunc, dbcc	87
set forceplan	53	setuser	67
set hash_join	53	shared	45
set identity_insert	37	Shared memory dumps	109
set identity_update	37	shared memory starting address(conf)	108
set index	53	shmdumpconfig, sp_	109
set jtc	54	shmdumpsize, sp_	109
set language	79	show switch	93
set literal_autoparam	54	show_options, sp_	117
set lock_nowait	45	show_role()	19, 69
set lock_wait	45	show_sec_services()	17
set merge_join	53	showcontrolinfo, sp_	105
set metrics_capture	106	showexeclass, sp_	104
set nl_join	53	showplan, set	55
set nocount	56	showplan, sp_	52
set nodata	53	showplan_in_xml()	17, 55
set noexec	53	showpsex, sp_	104
set option show	55	showrecovery, dbcc	92
set parallel_degree	81	shutdown	82
set parallel_query	53	sign()	18
set parseonly	53	simulated statistics	55, 77, 114
set plan_dump	105	sin()	18
set plan_exists_check	105	single user (DB option)	60
set plan_for_XML	55	site handler RPCs	94
set plan_load	105	size of auto identity column	38
set plan_optgoal	53	size of global fixed heap (config) ..	46, 74
set plan_opttimeoutlimit	54	size of process object heap (conf.) ..	46, 74
set plan_replace	105	size of shared class heap (config) ..	46, 74
set prefetch	53	size of unilif cache (config)	73
set proc_output_params	56	Small Business Edition	123
set proc_return_status	56	sms_shutdown	121
set process_limit_action	81	sms_status	121

Undocumented commands, warning & disclaimer 7

union 24

Unions in views 33

unique (constraint) 31

unique (index) 30

unique auto_identity index(DB opt.) **38**, 60

unlocking a login
 (or a role) with dataserver -u 111

'sa', with traceflag 4044 94

with sp_locklogin 67

unlogged operations 121

unmount database 63

unsigned datatypes 8

Unsupported DBCC commands 88

update 24

update (cursors) 41

update all statistics 76

update index statistics 76

update partition statistics 76

update statistics 76

update table statistics 76

update where current of 41

update() 36

updateconfig, sp_dbcc_ 86

upgrade_object, dbcc 92

upper() 14

uscalar() 14

use 40

used_pages() 17

used_pgs() 17

usedextents, dbcc 92

user 19, 67

user_id() 19, 67

user_name() 19, 67

user_stats, dbcc 92

User-defined roles 68

Users 67

UTC time, getutcdate() 14

UUID - newid() 48

V

valid_name() 17

valid_user() 19

variable assignment 22

version ID of ASE (@@version) 120

version, sp_ 81

Views 83

volchanged, sp_ 63

W

waitfor delay 40

waitfor errorexit 40

waitfor mirrorexist **40**, 58

waitfor processexist 40

waitfor time 40

Web Services **49**, 123

webservices, sp_ 50

webservices_role **50**, 68

where (clause) 22

who, sp_ 52

writetext 20

X

xa_bqual() 17

xa_gtrid() 17

xls, dbcc 92

XML 47

xml (select...for xml) 23

xmlextract() 47

xmlparse() 47

xmlrepresentation() 48

xmltest (predicate) 48

xmlvalidate() 48

XP Server 99

xp_cmdshell 99

xp_cmdshell context (config) 99

xp_deletemail 99

xp_enumgroups 99

xp_findnextmsg 99

xp_logevent 99

xp_readmail 99

xp_sendmail 99

xp_startmail 99

xp_stopmail 99

XPath 47, **49**

xpserver 111

XQL 49

XQuery 49

Y

year() 11

Z

zero, divide by
 nullif() 39

traceflag 3610 93

zeroes, padding with leading 14

