

Sybase® Adaptive Server™ Enterprise 参考手册

第三卷：过程

Adaptive Server Enterprise 版本 12

文档 ID: 37417-01-1200-01

最后修订日期: 1999 年 10 月

主要作者: Enterprise Data Studios Publications

参与作者: Anneli Meyer、 Evelyn Wheeler

文档 ID: 37417-01-1200-01

本手册适用于 **Sybase** 数据库管理软件的 **Adaptive Server Enterprise** 版本 12 及所有后续版本, 除非在新版本或技术说明中另有指定。本文档的信息若有变动, 恕不另行通知。本手册中所述的软件按许可协议提供, 因此必须按照协议条款来使用或复制该软件。

文档订购

要订购附加文档, 美国和加拿大的客户可拨打客户服务部门电话 (800) 685-8225 或发传真至 (617) 229-9845。

其它国家的客户如持有美国许可协议, 也可通过上述传真号与客户服务部门联系。所有其他国际客户应与 **Sybase** 子公司或当地分销商联系。

升级产品仅在软件的定期发布日提供。

Copyright © 1989-2001 by Sybase, Inc. 保留所有权利。

未经 **Sybase, Inc.** 的事先书面授权, 本书的任何部分不能以任何形式、任何手段 (电子的、机械的、手工的、光学的或其它) 复制、传播或翻译。

Sybase 商标

Sybase、SYBASE 徽标、Adaptive Server、APT-FORMS、Certified SYBASE Professional、Certified SYBASE Professional 徽标、Column Design、ComponentPack、Data Workbench、First Impression、InfoMaker、ObjectCycle、PowerBuilder、PowerDesigner、Powersoft、Replication Server、S-Designor、SQL Advantage、SQL Debug、SQL SMART、Transact-SQL、Visual Components、VisualWriter 和 VQL 均为 Sybase, Inc. 的注册商标。

Adaptable Windowing Environment、Adaptive Component Architecture、Adaptive Server Enterprise Monitor、Adaptive Warehouse、ADA Workbench、AnswerBase、Application Manager、AppModeler、APT-Build、APT-Edit、APT-Execute、APT-Library、APT-Translator、APT Workbench、Backup Server、BayCam、Bit-Wise、ClearConnect、Client-Library、Client Services、CodeBank、Connection Manager、DataArchitect、Database Analyzer、DataExpress、Data Pipeline、DataServer、DataWindow、DB-Library、dbQueue、Developers Workbench、DirectConnect、Distribution Agent、Distribution Director、Embedded SQL、EMS、Enterprise Application Server、Enterprise Application Studio、Enterprise Client/Server、EnterpriseConnect、Enterprise Data Studio、Enterprise Manager、Enterprise SQL Server Manager、Enterprise Work Architecture、Enterprise Work Designer、Enterprise Work Modeler、EWA、Formula One、Gateway Manager、GeoPoint、ImpactNow、InformationConnect、InstaHelp、InternetBuilder、iScript、Jaguar CTS、jConnect for JDBC、KnowledgeBase、Logical Memory Manager、MainframeConnect、Maintenance Express、MAP、MDI Access Server、MDI Database Gateway、media.splash、MetaBridge、MetaWorks、MethodSet、

MySupport、Net-Gateway、NetImpact、Net-Library、Next Generation Learning、ObjectConnect、OmniConnect、OmniSQL Access Module、OmniSQL Toolkit、Open Client、Open ClientConnect、Open Client/Server、Open Client/Server Interfaces、Open Gateway、Open Server、Open ServerConnect、Open Solutions、Optima++、PB-Gen、PC APT-Execute、PC DB-Net、PC Net Library、Power++、Power AMC、PowerBuilt、PowerBuilt with PowerBuilder、PowerDynamo、PowerJ、PowerScript、PowerSite、PowerSocket、Powersoft Portfolio、PowerStudio、Power Through Knowledge、PowerWare Desktop、PowerWare Enterprise、ProcessAnalyst、Replication Agent、Replication Driver、Replication Server Manager、Report-Execute、Report Workbench、Resource Manager、RW-DisplayLib、RW-Library、SAFE、SDF、Secure SQL Server、Secure SQL Toolset、Security Guardian、SKILS、smart.partners、smart.parts、smart.script、SQL Code Checker、SQL Edit、SQL Edit/TPU、SQL Modeler、SQL Remote、SQL Server、SQL Server/CFT、SQL Server/DBM、SQL Server Manager、SQL Server SNMP SubAgent、SQL Station、SQL Toolset、Sybase Central、Sybase Client/Server Interfaces、Sybase Development Framework、Sybase Financial Server、Sybase Gateways、Sybase Learning Connection、Sybase MPP、Sybase SQL Desktop、Sybase SQL Lifecycle、Sybase SQL Workgroup、Sybase Synergy Program、Sybase Virtual Server Architecture、Sybase User Workbench、SybaseWare、SyberAssist、SyBooks、System 10、System 11、System XI 徽标、SystemTools、Tabular Data Stream、The Enterprise Client/Server Company、The Extensible Software Platform、The Future Is Wide Open、The Learning Connection、The Model for Client/Server Solutions、The Online Information Center、Translation Toolkit、Turning Imagination Into Reality、UltraLite、UNIBOM、Unilib、Uninull、Unisep、Unistring、URK Runtime Kit for UniCode、Viewer、VisualSpeller、VisualWriter、WarehouseArchitect、Warehouse Studio、Warehouse WORKS、Watcom、Watcom SQL、Watcom SQL Server、Web.PB、Web.SQL、WebSights、WebViewer、WorkGroup SQL Server、XA-Library、XA-Server 和 XP Server 均为 Sybase, Inc. 的商标。

本文档中使用的所有其它公司名和产品名均可能是相应公司的商标或注册商标。

有限权利

政府使用、复制或公开本软件受 DFARS 52.227-7013 中附属 (c)(1)(ii) 条款（针对 DOD）和 FAR 52.227-19(a)-(d) 中相应条款（针对民间组织）的限制。

Sybase, Inc., 6475 Christie Avenue, Emeryville, CA 94608.

目录

关于本手册

如何使用本手册	xvii
---------------	------

7. 系统过程

系统过程简介	7-9
系统过程的权限	7-9
执行系统过程	7-9
参数的值	7-10
系统过程消息	7-11
系统过程表	7-11
<i>sp_activeroles</i>	7-12
<i>sp_addalias</i>	7-14
<i>sp_addauditrecord</i>	7-16
<i>sp_addaudittable</i>	7-18
<i>sp_addengine</i>	7-20
<i>sp_addexeclass</i>	7-22
<i>sp_addextendedproc</i>	7-24
<i>sp_addexternlogin</i>	7-26
<i>sp_addgroup</i>	7-28
<i>sp_addlanguage</i>	7-29
<i>sp_addlogin</i>	7-32
<i>sp_addmessage</i>	7-35
<i>sp_addobjectdef</i>	7-37
<i>sp_add_qpgroup</i>	7-41
<i>sp_addremotelogin</i>	7-42
<i>sp_add_resource_limit</i>	7-45
<i>sp_addsegment</i>	7-50
<i>sp_addserver</i>	7-52
<i>sp_addthreshold</i>	7-55
<i>sp_add_time_range</i>	7-60
<i>sp_addtype</i>	7-63
<i>sp_addumpdevice</i>	7-67
<i>sp_adduser</i>	7-69
<i>sp_altermessage</i>	7-71
<i>sp_audit</i>	7-73

<i>sp_autoconnect</i>	7-79
<i>sp_bindcache</i>	7-81
<i>sp_bindefault</i>	7-84
<i>sp_bindexeclass</i>	7-86
<i>sp_bindmsg</i>	7-89
<i>sp_bindrule</i>	7-91
<i>sp_cacheconfig</i>	7-94
<i>sp_cachestrategy</i>	7-103
<i>sp_changedbowner</i>	7-106
<i>sp_changegroup</i>	7-108
<i>sp_checknames</i>	7-110
<i>sp_checkreswords</i>	7-112
<i>sp_checksouce</i>	7-124
<i>sp_chgattribute</i>	7-126
<i>sp_clearpsex</i>	7-129
<i>sp_clearstats</i>	7-131
<i>sp_cmp_all_qplans</i>	7-133
<i>sp_cmp_qplans</i>	7-136
<i>sp_commonkey</i>	7-138
<i>sp_companion</i>	7-140
<i>sp_configure</i>	7-143
<i>sp_copy_all_qplans</i>	7-147
<i>sp_copy_qplan</i>	7-149
<i>sp_countmetadata</i>	7-150
<i>sp_cursorinfo</i>	7-152
<i>sp_dboption</i>	7-155
<i>sp_dbrecovery_order</i>	7-162
<i>sp_dbremap</i>	7-164
<i>sp_defaultloc</i>	7-166
<i>sp_depends</i>	7-169
<i>sp_deviceattr</i>	7-172
<i>sp_diskdefault</i>	7-174
<i>sp_displayaudit</i>	7-176
<i>sp_displaylevel</i>	7-180
<i>sp_displaylogin</i>	7-182
<i>sp_displayroles</i>	7-184
<i>sp_dropalias</i>	7-186
<i>sp_drop_all_qplans</i>	7-187
<i>sp_dropdevice</i>	7-188

<i>sp_dropengine</i>	7-189
<i>sp_dropexeclass</i>	7-190
<i>sp_dropextendedproc</i>	7-191
<i>sp_dropexternlogin</i>	7-192
<i>sp_dropglockpromote</i>	7-194
<i>sp_dropgroup</i>	7-195
<i>sp_dropkey</i>	7-196
<i>sp_droplanguage</i>	7-198
<i>sp_droplogin</i>	7-199
<i>sp_dropmessage</i>	7-201
<i>sp_drop_qpgroup</i>	7-202
<i>sp_drop_qplan</i>	7-203
<i>sp_dropobjectdef</i>	7-204
<i>sp_dropremotelogin</i>	7-206
<i>sp_drop_resource_limit</i>	7-208
<i>sp_droprowlockpromote</i>	7-212
<i>sp_dropsegment</i>	7-214
<i>sp_dropserver</i>	7-216
<i>sp_dropthreshold</i>	7-218
<i>sp_drop_time_range</i>	7-219
<i>sp_droptype</i>	7-220
<i>sp_dropuser</i>	7-221
<i>sp_dumpoptimize</i>	7-223
<i>sp_estspace</i>	7-228
<i>sp_export_qpgroup</i>	7-232
<i>sp_extendsegment</i>	7-234
<i>sp_familylock</i>	7-236
<i>sp_find_qplan</i>	7-238
<i>sp_flushstats</i>	7-240
<i>sp_forceonline_db</i>	7-241
<i>sp_forceonline_object</i>	7-243
<i>sp_forceonline_page</i>	7-245
<i>sp_foreignkey</i>	7-247
<i>sp_freedll</i>	7-249
<i>sp_getmessage</i>	7-250
<i>sp_grantlogin</i>	7-252
<i>sp_ha_admin</i>	7-254
<i>sp_help</i>	7-256
<i>sp_helppartition</i>	7-263

<i>sp_helpcache</i>	7-266
<i>sp_helpconfig</i>	7-268
<i>sp_helpconstraint</i>	7-273
<i>sp_helpdb</i>	7-277
<i>sp_helpdevice</i>	7-280
<i>sp_helpextendedproc</i>	7-282
<i>sp_helpexternlogin</i>	7-284
<i>sp_helpgroup</i>	7-286
<i>sp_helpindex</i>	7-288
<i>sp_helpjava</i>	7-290
<i>sp_helpjoins</i>	7-293
<i>sp_helpkey</i>	7-295
<i>sp_helplanguage</i>	7-297
<i>sp_helplog</i>	7-299
<i>sp_helpobjectdef</i>	7-300
<i>sp_help_qpgroup</i>	7-302
<i>sp_help_qplan</i>	7-304
<i>sp_helpremotelogin</i>	7-306
<i>sp_help_resource_limit</i>	7-307
<i>sp_helpprotect</i>	7-310
<i>sp_helpsegment</i>	7-314
<i>sp_helpserver</i>	7-317
<i>sp_helpsort</i>	7-318
<i>sp_helptext</i>	7-320
<i>sp_helpthreshold</i>	7-322
<i>sp_helpuser</i>	7-323
<i>sp_hidetext</i>	7-325
<i>sp_import_qpgroup</i>	7-327
<i>sp_indsuspect</i>	7-329
<i>sp_listsuspect_db</i>	7-330
<i>sp_listsuspect_object</i>	7-331
<i>sp_listsuspect_page</i>	7-332
<i>sp_lock</i>	7-333
<i>sp_locklogin</i>	7-337
<i>sp_logdevice</i>	7-339
<i>sp_loginconfig</i>	7-341
<i>sp_logininfo</i>	7-343
<i>sp_logiosize</i>	7-345
<i>sp_modifylogin</i>	7-348

<i>sp_modify_resource_limit</i>	7-350
<i>sp_modify_time_range</i>	7-353
<i>sp_modifythreshold</i>	7-355
<i>sp_monitor</i>	7-359
<i>sp_monitorconfig</i>	7-362
<i>sp_object_stats</i>	7-366
<i>sp_passthru</i>	7-369
<i>sp_password</i>	7-371
<i>sp_placeobject</i>	7-373
<i>sp_plan_dbccdb</i>	7-375
<i>sp_poolconfig</i>	7-378
<i>sp_primarykey</i>	7-383
<i>sp_processmail</i>	7-385
<i>sp_procqmode</i>	7-388
<i>sp_procxmode</i>	7-390
<i>sp_recompile</i>	7-392
<i>sp_remap</i>	7-393
<i>sp_remoteoption</i>	7-395
<i>sp_remotesql</i>	7-398
<i>sp_rename</i>	7-401
<i>sp_renamedb</i>	7-403
<i>sp_rename_qpgroup</i>	7-406
<i>sp_reportstats</i>	7-407
<i>sp_revokelogin</i>	7-409
<i>sp_role</i>	7-411
<i>sp_sendmsg</i>	7-413
<i>sp_serveroption</i>	7-415
<i>sp_setlangalias</i>	7-418
<i>sp_setpglockpromote</i>	7-419
<i>sp_setpsex</i>	7-422
<i>sp_set_qplan</i>	7-424
<i>sp_setrowlockpromote</i>	7-426
<i>sp_setsuspect_granularity</i>	7-429
<i>sp_setsuspect_threshold</i>	7-432
<i>sp_showcontrolinfo</i>	7-434
<i>sp_showexeclass</i>	7-436
<i>sp_showplan</i>	7-438
<i>sp_showpsex</i>	7-440
<i>sp_spaceused</i>	7-442

<i>sp_syntax</i>	7-445
<i>sp_sysmon</i>	7-447
<i>sp_thresholdaction</i>	7-450
<i>sp_transactions</i>	7-452
<i>sp_unbindcache</i>	7-459
<i>sp_unbindcache_all</i>	7-461
<i>sp_unbindefault</i>	7-462
<i>sp_unbindexeclass</i>	7-464
<i>sp_unbindmsg</i>	7-466
<i>sp_unbindrule</i>	7-467
<i>sp_volchanged</i>	7-469
<i>sp_who</i>	7-473

8. 分类存储过程

分类存储过程简介	8-1
指定可选参数	8-2
模式匹配	8-2
系统过程表	8-3
ODBC 数据类型	8-3
<i>sp_column_privileges</i>	8-5
<i>sp_columns</i>	8-7
<i>sp_databases</i>	8-10
<i>sp_datatype_info</i>	8-12
<i>sp_fkeys</i>	8-14
<i>sp_pkeys</i>	8-16
<i>sp_server_info</i>	8-18
<i>sp_special_columns</i>	8-21
<i>sp_sproc_columns</i>	8-23
<i>sp_statistics</i>	8-25
<i>sp_stored_procedures</i>	8-27
<i>sp_table_privileges</i>	8-29
<i>sp_tables</i>	8-31

9. 系统扩展存储过程

简介	9-1
对系统 ESP 的权限	9-1
与系统 ESP 相关的 DLL	9-2
使用系统 ESP	9-2
<i>xp_cmdshell</i>	9-3
<i>xp_deletemail</i>	9-5
<i>xp_enumgroups</i>	9-6
<i>xp_findnextmsg</i>	9-7
<i>xp_logevent</i>	9-9
<i>xp_readmail</i>	9-10
<i>xp_sendmail</i>	9-13
<i>xp_startmail</i>	9-16
<i>xp_stopmail</i>	9-18

10. dbcc 存储过程

指定对象名和日期	10-2
指定对象名	10-2
指定日期	10-2
<i>sp_dbcc_alterws</i>	10-4
<i>sp_dbcc_configreport</i>	10-6
<i>sp_dbcc_createws</i>	10-8
<i>sp_dbcc_deletedb</i>	10-10
<i>sp_dbcc_deletehistory</i>	10-12
<i>sp_dbcc_differentialreport</i>	10-14
<i>sp_dbcc_evaluatedb</i>	10-16
<i>sp_dbcc_faultreport</i>	10-18
<i>sp_dbcc_fullreport</i>	10-20
<i>sp_dbcc_runcheck</i>	10-22
<i>sp_dbcc_statisticsreport</i>	10-24
<i>sp_dbcc_summaryreport</i>	10-27
<i>sp_dbcc_updateconfig</i>	10-31

如需索引，参见第四卷“表格和参考手册索引”。

图目录

图 7-1:	带有缺省缓存和用户定义缓存的数据缓存	7-96
图 7-2:	重新启动和 <code>sp_cacheconfig</code> 对缓存状态的影响	7-100
图 7-3:	带有缺省缓存和用户定义缓存的数据缓存	7-380

表目录

表 7-1:	系统过程	7-1
表 7-2:	objecttype 允许使用的值	7-38
表 7-3:	objecttype 用法摘要	7-38
表 7-4:	server_class 参数的允许值	7-52
表 7-5:	审计选项	7-73
表 7-6:	控制审计的配置参数	7-77
表 7-7:	新旧绑定规则的优先顺序	7-92
表 7-8:	Transact-SQL 命令的缓存使用情况	7-97
表 7-9:	sp_cacheconfig 输出	7-98
表 7-10:	sp_rename 和更改标识符	7-118
表 7-11:	更改标识符时直接更新系统表的替代方法	7-120
表 7-12:	更改标识符时要更新的系统表列	7-121
表 7-13:	更改标识符时的考虑事项	7-122
表 7-14:	sp_cmp_all_qplans 的报告模式	7-134
表 7-15:	sp_cmp_qplans 的返回状态值	7-137
表 7-16:	允许在事务中使用的 DDL 命令	7-158
表 7-17:	不允许在事务中使用的 DDL 命令	7-159
表 7-18:	defaulttype 允许使用的值	7-166
表 7-19:	sp_monitor 报告中的列	7-360
表 7-20:	sp_object_stats 的输出	7-367
表 7-21:	tempdb.syslkstats 表中的列	7-368
表 7-22:	sp_serveroption 选项	7-415
表 7-23:	sp_sysmon 报告各个部分	7-447
表 7-24:	sp_sysmon 中 applmon 参数的值	7-448
表 7-25:	commit_node 和 parent_node 的值	7-457
表 7-26:	在 UNIX 系统中更改磁带的卷	7-471
表 8-1:	分类存储过程	8-1
表 8-2:	ODBC 数据类型代码编号	8-3
表 8-3:	扩展数据类型代码编号	8-4
表 8-4:	sp_column_privileges 结果集	8-6
表 8-5:	sp_columns 结果集	8-8
表 8-6:	sp_databases 结果集	8-10
表 8-7:	sp_datatype_info 结果集	8-12
表 8-8:	sp_fkeys 结果集	8-15
表 8-9:	sp_pkeys 结果集	8-16
表 8-10:	sp_server_info 结果集	8-18
表 8-11:	sp_server_info 返回的强制结果	8-18
表 8-12:	sp_special_columns 结果集	8-22

表 8-13:	sp_sproc_columns 结果集	8-23
表 8-14:	sp_statistics 结果集	8-26
表 8-15:	sp_stored_procedures 结果集	8-27
表 8-16:	sp_table_privileges 结果集	8-29
表 8-17:	sp_tables 结果集	8-32
表 9-1:	系统扩展存储过程	9-1
表 10-1:	dbcc 存储过程	10-1
表 10-2:	类型名称和预期值	10-32

关于本手册

Adaptive Server Enterprise 参考手册 共有四卷，它是一本有关 Sybase® Adaptive Server™ Enterprise 和 Transact-SQL® 语言的指南。

第一卷 “*构件块*” 介绍 Transact-SQL 的 “*构件*”：数据类型、内部函数、表达式和标识符、SQLSTATE 错误，以及保留字。要成功地使用 Transact-SQL，您首先需要理解每个构件块的目的，并明确它们的使用会如何影响 Transact-SQL 语句的结果。

第二卷 “*命令*” 提供有关创建语句时所使用的 Transact-SQL 命令的参考信息。

第三卷 “*过程*” 提供有关系统过程，分类存储过程，扩展存储过程和 dbcc 存储过程的参考信息。所有过程都是使用 Transact-SQL 语句创建的。

第四卷 “*表格和参考手册索引*” 提供有关系统表的参考信息。系统表中存储了有关服务器、数据库、用户的信息以及其它信息。它还提供 dbccdb 和 dbccalt 数据库中表的信息。此卷中还有一个索引，它包括了全部四卷中的主题。

有关本手册所针对的读者、相关文档、其它信息来源、本手册中使用的约定以及帮助的信息，参见第一卷中的 “关于本手册”。

如何使用本手册

本手册包括：

- 第 7 章 “系统过程”，提供 Adaptive Server 系统过程的参考页。
- 第 8 章 “分类存储过程”，提供 Adaptive Server 分类存储过程的参考页。
- 第 9 章 “系统扩展存储过程”，提供 Adaptive Server 系统扩展存储过程的参考页。
- 第 10 章 “dbcc 存储过程”，提供 Adaptive Server dbcc 存储过程的参考页。

7

系统过程

本章说明各个系统过程，它们是 Sybase 提供的、用于更新或获取系统表报告的存储过程。表 7-1 列出了本章中要介绍的系统过程。

表 7-1：系统过程

过程	说明
sp_activeroles	显示所有授予用户登录的活动角色。
sp_addalias	允许 Adaptive Server 用户在数据库中以其他用户的名义操作。
sp_addauditrecord	允许用户将用户定义的审计记录（注释）输入到审计追踪中。
sp_addauditable	在安装审计后添加其它系统审计表。
sp_addengine	为现有引擎组添加引擎，如果该组不存在，则创建引擎组并添加引擎。
sp_addexecclass	创建或更新用户定义的执行类，该类可以绑定到客户端应用、登录和存储过程。
sp_addextendedproc	在 <i>master</i> 数据库中创建扩展存储过程 (ESP)。
sp_addexternlogin	创建备用的登录帐户和口令，以供在通过组件集成服务与远程服务器进行通信时使用。
sp_addgroup	将组添加到数据库。在授予和撤销权限时，组是作为集合名使用的。
sp_addlanguage	定义替代语言的月份和日的名称及其日期格式。
sp_addlogin	将新的用户帐户添加到 Adaptive Server。
sp_addmessage	将用户定义的消息添加到 <i>sysusermessages</i> ，以供存储过程 print 和 raiserror 调用以及 sp_bindmsg 使用。
sp_addobjectdef	指定本地表与外部存储位置之间的映射。
sp_add_qpgroup	添加抽象计划组。
sp_addremotelogin	通过在 <i>master.dbo.sysremotelogins</i> 中添加条目向新的远程服务器用户授权。
sp_add_resource_limit	创建登录或应用程序可用来执行查询、查询批处理或事务的服务器资源量的限制。
sp_addsegment	定义当前数据库中数据库设备上的段。
sp_addserver	定义远程服务器或者定义本地服务器的名称。
sp_addthreshold	创建阈值来监控数据库段的空闲空间。如果段上的可用空间小于指定级别，Adaptive Server 将执行相关的存储过程。
sp_add_time_range	将指定的时间范围添加到 Adaptive Server。

表 7-1: 系统过程 (续)

过程	说明
sp_addtype	创建用户定义的数据类型。
sp_addumpdevice	将转储设备添加到 Adaptive Server。
sp_adduser	将新的用户添加到当前数据库。
sp_altermessage	允许或禁止在 Adaptive Server 错误日志中记录特定的系统定义的消息或用户定义的消息。
sp_audit	允许系统安全员配置审计选项。
sp_autoconnect	为特定用户定义到远程服务器的直通连接, 使被命名的用户在登录时自动进入直通模式。
sp_bindcache	将数据库、表、索引、 <i>text</i> 对象或 <i>image</i> 对象绑定到数据缓存。
sp_bindefault	将用户定义的缺省值绑定到列或用户定义的数据类型。
sp_bindexeclass	使执行类与客户端应用、登录或存储过程相关联。
sp_bindmsg	将用户消息绑定到参照完整性约束或检查约束。
sp_bindrule	将规则绑定到列或用户定义的数据类型。
sp_cacheconfig	创建、配置、重新配置和删除数据缓存, 并提供有关数据缓存的信息。
sp_cachestrategy	对表、索引、 <i>text</i> 对象或 <i>image</i> 对象启用或禁用预取 (大型 I/O) 和 MRU 缓存替换策略。
sp_changedbowner	更改数据库的所有者。
sp_changegroup	更改用户的组。
sp_checknames	检查当前数据库中是否存在包含 7 位 ASCII 字符集以外字符的名称。
sp_checkreswords	检测并显示 Transact-SQL 保留字的标识符。检查服务器名、设备名、数据库名、段名、用户定义的数据类型、对象名、列名、用户名、登录名和远程登录名。
sp_checksourc	检查是否存在已编译对象的源文本。
sp_chgattribute	更改 max_rows_per_page 的值, 以用于表或索引的未来空间分配。
sp_clearpsex	清除 sp_setpsex 所设置的客户端应用、登录或存储过程的执行特性。
sp_clearstats	为所有服务器用户或指定用户开始新的会计周期。通过执行 sp_reportstats , 输出以前周期的统计信息。
sp_cmp_all_qplans	比较两个抽象计划组中所有的抽象计划。
sp_cmp_qplans	比较两个抽象计划。
sp_commonkey	定义两个表或视图间的公用键 (经常被连接的列)。

表 7-1: 系统过程 (续)

过程	说明
sp_companion	执行集群操作，例如在高可用性系统中将 Adaptive Server 配置为辅助协同服务器，将协同服务器从一种故障替换模式转换到其它模式。
sp_configure	显示或更改配置参数。
sp_copy_all_qplans	将一个抽象计划组的所有计划复制到另一个组。
sp_copy_qplan	将一个抽象计划复制到抽象计划组。
sp_countmetadata	显示 Adaptive Server 中索引、对象或数据库的个数。
sp_cursorinfo	报告有关特定游标或为会话激活的所有游标的信息。
sp_dboption	显示或更改数据库选项。
sp_dbremap	强制 Adaptive Server 识别 alter database 所作的更改。只有在 Adaptive Server 消息指示要这样做时，才运行该过程。
sp_depends	显示有关数据库对象依赖性的信息，这些对象包括依赖于所指定的表或视图的视图、触发器和过程，以及指定的视图、触发器或过程所依赖的表和视图。
sp_deviceattr	更改现有数据库设备文件的 dsync 设置。
sp_diskdefault	在用户没有指定数据库设备或通过 create database 或 alter database 命令指定 default 的情况下，指定数据库设备是否可以用于数据库存储。
sp_displayaudit	显示审计选项的状态。
sp_displaylevel	设置或显示哪些 Adaptive Server 配置参数会出现在 sp_configure 输出中。
sp_displaylogin	显示有关登录帐户的信息。
sp_displayroles	显示授予另一角色的所有角色，或者以表的格式显示整个角色层次树。
sp_dropalias	删除用 sp_addalias 所建立的别名用户名标识。
sp_drop_all_qplans	删除抽象计划组中所有的抽象计划。
sp_dropdevice	删除 Adaptive Server 数据库设备或转储设备。
sp_dropengine	从指定的引擎组中删除一个引擎，如果该引擎是组中的最后一个引擎，则删除该引擎组。
sp_dropexeclass	删除用户定义的执行类。
sp_dropextendedproc	从 master 数据库删除 ESP。
sp_dropglockpromote	从表或数据库中删除锁升级值。
sp_dropgroup	从数据库删除组。

表 7-1: 系统过程 (续)

过程	说明
sp_dropkey	从 <i>syskeys</i> 表删除使用 sp_primarykey 、 sp_foreignkey 或 sp_commonkey 定义的键。
sp_droplanguage	从服务器中删除替代语言并从 <i>master.dbo.syslanguages</i> 中删除它的行。
sp_droplogin	通过删除 <i>master.dbo.syslogins</i> 中用户的条目, 删除 Adaptive Server 用户登录。
sp_dropmessage	从 <i>sysusermessages</i> 删除用户定义的消息。
sp_drop_qpgroup	删除抽象计划组。
sp_drop_qplan	删除抽象计划。
sp_droptotelogin	删除远程用户登录。
sp_drop_resource_limit	从 Adaptive Server 中删除一个或多个资源限制。
sp_dropsegment	从数据库中删除某段或从特定数据库设备取消映射某段。
sp_dropserver	从已知服务器的列表删除服务器。
sp_droptreshold	从段中删除可用空间阈值。
sp_drop_time_range	从 Adaptive Server 删除用户定义的时间范围。
sp_droptype	删除用户定义的数据类型。
sp_dropuser	从当前数据库中删除用户。
sp_dumpoptimize	指定在转储数据库操作期间由 Backup Server 转储的数据的总量。
sp_estspace	估计表及其索引要求的空间大小和创建索引需要的时间。
sp_export_qpgroup	将指定用户和抽象计划组的所有计划导出到用户表。
sp_extendsegment	将段的范围扩展到其它数据库设备。
sp_familylock	报告有关系列 (协调进程及其工作进程) 持有的所有锁的信息, 而该系列是以并行方式执行语句的。
sp_find_qplan	在给定查询文本或计划文本的模式的情况下, 查找抽象计划。
sp_flushstats	将内存中存储的统计信息刷新到 <i>sysabstats</i> 系统表。
sp_forceonline_db	通过恢复提供对数据库中所有以前脱机的页的访问。
sp_forceonline_page	通过恢复提供对以前脱机的页的访问。
sp_foreignkey	在当前数据库中定义表或视图的外键。
sp_freedll	卸载先前装载到 XP Server 内存中以支持执行 ESP 的动态链接库 (DLL)。
sp_getmessage	从 <i>sysmessages</i> 和 <i>sysusermessages</i> 中检索存储的消息字符串, 供 print 和 raiserror 语句使用。

表 7-1: 系统过程 (续)

过程	说明
sp_grantlogin	(仅适用于 Windows NT) 当集成的安全方式或混合方式 (具有命名管道) 处于活动状态时, 为 Windows NT 用户和组指派 Adaptive Server 角色或 default 权限。
sp_ha_admin	在配置有高可用性系统中 Sybase 故障替换的 Adaptive Server 上执行管理任务。使用 <i>installhavss</i> 脚本 (在 Windows NT 上为 <i>insthasv</i>) 来安装 sp_ha_admin 。
sp_help	报告有关数据库对象 (在 <i>sysobjects</i> 中列出的任何对象) 的信息, 还报告有关 Adaptive Server 提供的或用户定义的数据类型的信息。
sp_helppartition	列出已分区的表中每个分区的首页和控制页。
sp_helpcache	显示绑定到数据缓存的对象信息或指定的缓存大小所要求的开销量信息。
sp_helppdb	报告有关特定数据库或有关所有数据库的信息。
sp_helpdevice	报告有关特定设备或有关所有 Adaptive Server 数据库设备和转储设备的信息。
sp_helpextendedproc	显示当前数据库中注册的 ESP 及其相关的 DLL 文件。
sp_helpexternlogin	(仅适用于组件集成服务) 报告有关外部登录名的信息。
sp_helpgroup	报告当前数据库中有关特定组或所有组的信息。
sp_helpindex	报告在表上创建的索引的信息。
sp_helpjava	显示有关数据库上安装的 Java 类和相关 JAR 的信息。
sp_helpjoins	列出两个表或视图中可能要连接的列。
sp_helpkey	报告有关特定的表或视图中主键、外键或公用键的信息, 或者有关当前数据库中所有键的信息。
sp_helplanguage	报告有关特定替代语言或有关所有语言的信息。
sp_helplog	报告含有事务日志首页的设备的名称。
sp_helpobjectdef	(仅适用于组件集成服务) 报告有关远程对象定义的信息。显示所有者、对象、类型和定义。
sp_help_qpgroup	报告抽象计划组的信息。
sp_help_qplan	报告有关抽象计划的信息。
sp_helpremotelogin	报告有关特定远程服务器登录的信息, 或有关所有远程服务器登录的信息。
sp_help_resource_limit	报告所有资源限制的信息, 包括对给定登录或应用程序的限制, 在给定时间或一星期中的某天生效的限制, 或具有给定范围或操作的限制。
sp_helpprotect	报告有关数据库对象、用户、组或角色具有的权限的信息。

表 7-1: 系统过程 (续)

过程	说明
sp_helpsegment	报告当前数据库中有关特定段或所有段的信息。
sp_helpserver	报告有关特定远程服务器的信息，或有关所有远程服务器的信息。
sp_helpsort	显示 Adaptive Server 的缺省排序顺序和字符集。
sp_helptext	输出系统过程、触发器、视图、缺省值、规则或完整性检查约束的文本。
sp_helpthreshold	报告与当前数据库中所有阈值或特定段中所有阈值相关联的段、可用空间值、状态和存储过程。
sp_helpuser	报告当前数据库中有关特定用户或有关所有用户的信息。
sp_import_qpgroup	将抽象计划从用户表导入抽象计划组。
sp_indsuspect	在更改排序顺序后紧接着进行的恢复过程中，为标记为可疑的索引检查用户表。
sp_listsuspect_db	列出具有脱机页的所有数据库，这些脱机页是由于恢复中所检测到的损坏而导致的。
sp_listsuspect_page	列出当前由于恢复中检测到的损坏而导致脱机的所有页。
sp_lock	报告有关当前持有锁的进程的信息。
sp_locklogin	锁定 Adaptive Server 帐户从而使用户无法登录，或显示所有锁定的帐户的列表。
sp_logdevice	将日志和数据在同一设备上的数据库事务日志移到单独的数据库设备上。
sp_loginconfig	(仅适用于 Windows NT) 显示一个或所有集成的安全参数的值。
sp_logininfo	(仅适用于 Windows NT) 显示使用 sp_grantlogin 授予 Windows NT 用户和组的所有角色。
sp_logiosize	当 Adaptive Server 为当前数据库的事务日志执行 I/O 时，将其使用的日志 I/O 大小更改为另一内存池。
sp_modifylogin	修改 Adaptive Server 登录帐户的缺省数据库、缺省语言、缺省激活角色或全名。
sp_modify_resource_limit	通过指定新的限制值 (和/或指定超过限制时所执行的操作) 来更改资源限制。
sp_modifythreshold	通过将阈值与其它阈值过程、可用空间级别或段名相关联，可以修改该阈值。 不能使用 sp_modifythreshold 来更改最后机会阈值的可用空间量或段名。
sp_modify_time_range	更改与指定的时间范围相关联的起始日 (星期)、开始时间、结束日 (星期) 和/或结束时间。
sp_monitor	显示有关 Adaptive Server 的统计信息。

表 7-1: 系统过程 (续)

过程	说明
sp_password	添加或更改 Adaptive Server 登录帐户的口令。
sp_placeobject	将表或索引的未来空间分配情况放置到特定段上。
sp_plan_dbccdb	为新的 <i>dbccdb</i> 和 <i>dbccalt</i> 数据库推荐合适的大小，为 <i>dbccdb</i> 和 <i>dbccalt</i> 列出合适的设备，并为目标数据库建议缓存大小和合适的工作进程数。
sp_poolconfig	创建、删除内存池或调整内存池大小，并提供有关数据缓存中内存池的信息。
sp_primarykey	定义表或视图的主键。
sp_processmail	(仅适用于 Windows NT) 读取、处理、发送和删除 Adaptive Server 收件箱中的消息。
sp_procmode	显示存储过程、视图或触发器的查询处理模式。
sp_procxmode	显示或更改与存储过程相关联的事务模式。
sp_recompile	导致使用指定表的每个存储过程和触发器在下次运行时重新进行编译。
sp_remap	重新映射高于 4.8 且低于 10.0 版本的存储过程、触发器、规则、缺省值或视图，使其与 10.0 版本或更高版本兼容。对于低于 11.0 版本的对象（这些对象在 11.0 版本升级过程中无法重新映射），使用 sp_remap 。
sp_remoteoption	显示或更改远程登录选项。
sp_rename	更改当前数据库中用户创建的对象名称或用户定义的数据类型的名称。
sp_renamedb	更改数据库的名称。 不能 重命名系统数据库或具有外部参照完整性约束的数据库。
sp_rename_qpgroup	重命名抽象计划组。
sp_reportstats	报告有关系统使用情况的统计信息。
sp_revokelogin	(仅适用于 Windows NT) 当集成的安全方式或混合方式（具有命名管道）处于活动状态时，撤消 Windows NT 用户和组的 Adaptive Server 角色和缺省权限。
sp_role	授予或撤消 Adaptive Server 登录帐户的系统角色。
sp_serveroption	显示或更改远程服务器选项。
sp_setlangalias	指派或更改替代语言的别名。
sp_setpglockpromote	设置或更改数据库、表或 Adaptive Server 的锁升级阈值。
sp_setpsexex	在运行过程中为活动的客户端应用、登录或存储过程设置自定义执行特性。
sp_set_qplan	更改现有计划的抽象计划的文本，而不更改相关查询。

表 7-1: 系统过程 (续)

过程	说明
sp_setsuspect_granularity	显示和设置恢复故障隔离模式。
sp_setsuspect_threshold	在恢复期间, 设置在指定数据库中 Adaptive Server 所允许的最大可疑页数, 超过该页数整个数据库将脱机。
sp_showcontrolinfo	显示有关引擎组赋值、绑定的客户端应用、登录和存储过程的信息。
sp_showexeclass	显示执行类特性以及与指定执行类相关联的任何引擎组中的引擎。
sp_showplan	显示当前 SQL 语句 (或同一批处理中的前一语句) 的任意用户连接的查询计划。以 showplan 格式显示查询计划。
sp_showpsex	显示在 Adaptive Server 上运行的所有进程的执行类、当前优先级和相关的父 (子) 进程。
sp_spaceused	显示估计的、由当前数据库中一个表或所有表使用的行数、数据页数以及空间大小。
sp_syntax	根据 Adaptive Server 上具有的产品和对应的 sp_syntax 脚本, 显示 Transact-SQL 语句、系统过程、实用程序和其它例程的语法。
sp_sysmon	显示性能信息。
sp_thresholdaction	将在日志段的可用页数小于最后机会阈值时自动执行 (除非阈值与其它过程关联)。 Sybase 不提供此过程。
sp_transactions	报告有关活动事务的信息。
sp_unbindcache	从数据缓存将数据库、表、索引、 text 对象或 image 对象解除绑定。
sp_unbindcache_all	对所有绑定到缓存的对象解除绑定。
sp_unbindefault	从列或用户定义的数据类型解除绑定已创建的缺省值。
sp_unbindexclass	从数据缓存将数据库、表、索引、 text 对象或 image 对象解除绑定。
sp_unbindmsg	从约束解除绑定用户定义的消息。
sp_unbindrule	从列或用户定义的数据类型解除绑定规则。
sp_volchanged	通知 Backup Server™: 操作员已在转储或装载期间执行了请求的卷处理。
sp_who	报告有关当前所有 Adaptive Server 用户和进程的信息, 或有关特定用户和进程的信息。

系统过程简介

系统过程在安装时由 **installmaster** 创建。它们位于 **sybsystemprocs** 数据库，为系统管理员所拥有。

某些系统过程只能在特定数据库中运行，但许多过程可在任意数据库中运行。可以创建您自己的、可从任意数据库执行的系统过程。有关详细信息，参见 *系统管理指南*。

所有系统过程都在隔离级别 **1** 上执行。

所有系统过程都会报告返回状态。例如：

```
return status = 0
```

表示该过程已成功执行。本手册中的示例未包含该返回状态。

系统过程的权限

在 **sybsystemprocs** 数据库中设置系统过程的权限。

某些系统过程只能由数据库所有者来运行。这些过程确保执行过程的用户是从中执行这些过程的数据库的所有者。

其它系统过程（例如所有 **sp_help** 过程）可由任何被授予权限的用户来执行，但前提是该权限是 **sybsystemprocs** 中授予的权限。用户要么在所有数据库中都具有执行系统过程的权限，要么在任何数据库中都不具有这一权限。

未在 **sybsystemprocs..sysusers** 中列出的用户将被视为 **sybsystemprocs** 中的 “**guest**” 用户，并自动向其授予多种系统过程的权限。

若要拒绝向用户授予系统过程权限，系统管理员必须将该用户添加到 **sybsystemprocs..sysusers** 中，并编写适用于该过程的 **revoke** 语句。用户数据库的所有者不能从自己的数据库内部直接控制该系统过程的权限。

执行系统过程

如果从 **sybsystemprocs** 之外的数据库执行系统过程，该过程将对其执行位置的数据库中的系统表进行操作。例如，如果 **pubs2** 的数据库所有者在 **pubs2** 中运行 **sp_adduser**，则新的用户将添加到 **pubs2..sysusers**。

要在特定数据库中运行系统过程，可以执行以下操作之一：

- 使用 **use** 命令打开该数据库，然后执行该过程；或者
- 用数据库名限定过程名。

例如，用户定义的系统过程 *sp_foo*（执行 *db_name()* 系统函数）将返回数据库的名称，而该过程就是在这一数据库中执行的。当在 *pubs2* 数据库中执行该过程时，它会返回值 “pubs2”：

```
exec pubs2..sp_foo

-----
pubs2
(1 row affected, return status = 0)
```

当在 *sybsystemprocs* 中执行该过程时，它会返回值 “sybsystemprocs”：

```
exec sybsystemprocs..sp_foo

-----
sybsystemprocs
(1 row affected, return status = 0)
```

参数的值

如果系统过程的参数值中包含标点符号或嵌入的空白，或者参数值为保留字，则必须用单引号或双引号将其引起来。如果参数是被数据库名或所有者名限定的对象名，则用单引号或双引号将整个名称引起来。

► 注意

不要将分隔标识符用作系统过程参数；否则它们会导致意外的结果。

如果过程具有多个可选参数，则可以按下列形式提供参数：

```
@parametername = value
```

而不用提供所有参数。语法语句中的参数名与过程定义的参数名是相匹配的。

例如，*sp_addlogin* 的语法格式如下：

```
sp_addlogin login_name, password [, defdb
    [, deflanguage [, fullname]]]
```

要使用 *sp_addlogin* 为 “susan” 创建口令为 “wonderful” 的登录（该登录具有 “Susan B. Anthony” 的全名以及服务器的缺省数据库和语言），则可以使用：

```
sp_addlogin susan, wonderful,
    @fullname="Susan B. Anthony"
```

它提供的信息与以下指定了所有参数的命令所提供的信息相同：

```
sp_addlogin susan, wonderful, public_db,  
us_english, "Susan B. Anthony"
```

还可以使用 “null” 作为占位符：

```
sp_addlogin susan, wonderful, null, null,  
"Susan B. Anthony"
```

不要将 “null” 用引号引起来。

SQL 没有规定每一行中的单词数量或者必须折行的位置。如果在发出系统过程后发出命令，Adaptive Server 将尝试执行系统过程，然后再执行命令。例如，如果执行以下命令：

```
sp_help checkpoint
```

Adaptive Server 将返回 **sp_help** 的输出，然后运行 **checkpoint** 命令。

如果指定的参数数量超出系统过程预期的数量，Adaptive Server 将忽略多余的参数。

系统过程消息

系统过程返回信息性消息和错误消息，它们随本手册中的每个过程一起列出。系统过程错误消息的错误编号从 17000 开始。

有关过程中所含函数和命令的错误消息的信息，参见 *Troubleshooting and Error Messages Guide*。

系统过程表

系统过程使用 *master* 数据库中的几个系统过程表（如 *spt_values*、*spt_committab*、*spt_monitor* 和 *spt_limit_types*），以便将内部系统值（如状态位）转换为人们可以阅读的格式。

spt_values 永远不会被更新。要查看有关该表的使用方法，可以执行 **sp_helptext** 以便查看引用该表的一个系统过程的文本。

此外，某些系统过程可创建临时表，继而删除它们。

sp_activeroles

功能

显示所有活动角色。

语法

```
sp_activeroles [expand_down]
```

参数

expand_down — 显示您的角色所包含的所有活动角色的层次树。

示例

1. sp_activeroles

```
Role Name
-----
sa_role
sso_role
oper_role
replication_role
```

2. sp_activeroles expand_down

Role Name	Parent Role Name	Level
sa_role	NULL	1
doctor_role	NULL	1
oper_role	NULL	1

注释

- **sp_activeroles** 显示所有活动角色以及这些角色所包含的全部角色。
- 有关角色的详细信息，参见 *系统管理指南*。

权限

任何用户都可以执行 **sp_activeroles**。

使用的表

master.dbo.sysattributes、*master.dbo.syssrvroles*、*master.dbo.sysloginroles*

参见

命令	alter role、create role、drop role、grant、revoke、set
函数	mut_excl_roles、proc_role、role_contain、role_name
系统过程	sp_displayroles

sp_addalias

功能

允许 Adaptive Server 用户在数据库中以其他用户的名义操作。

语法

```
sp_addalias loginame, name_in_db
```

参数

loginame — 是需要在当前数据库中使用替代标识的用户的 *master.dbo.syslogins* 名称。

name_in_db — 是 *loginame* 别名所指的数据库用户名。该名称必须存在于当前数据库的 *master.dbo.syslogins* 和 *sysusers* 表中。

示例

1. sp_addalias victoria, albert

数据库 *sysusers* 表中有一个名为 “Albert” 的用户，而 *master.dbo.syslogins* 中有一个名为 “Victoria” 用户的登录。该命令允许 “Victoria” 在沿用名称 “Albert” 的情况下使用当前数据库。

注释

- 执行 **sp_addalias** 可以将用户映射到当前数据库中的另一个用户。该映射显示在 *sysalternates* 中，其中两个用户的 *suids*（系统用户 ID）连在一起。
- 一次只能使一个用户的别名指向一个数据库用户。
- 可以用 **sp_helpuser** 生成映射到指定用户的任何用户的报告，将指定用户的名称作为参数。
- 当用户试图使用数据库时，Adaptive Server 将检查 *sysusers* 以确认用户是否已经在那里列出。如果该用户没有列出，Adaptive Server 将检查 *sysalternates*。如果用户的 *suid* 在 *sysalternates* 中列出，并映射到数据库用户的 *suid*，Adaptive Server 将在使用数据库时把第一个用户作为第二个用户对待。

如果在 *loginame* 中指定的用户在数据库的 *sysusers* 表中，Adaptive Server 将不使用用户的别名标识，这是因为它将先检查 *sysusers* 并查找 *loginame*，然后才检查 *sysalternates*（该别名在其中列出）。

权限

只有数据库所有者或系统管理员才能执行 **sp_addalias**。

使用的表

master.dbo.syslogins、 *sysalternates*、 *sysobjects*、 *sysusers*

参见

命令	use
系统过程	sp_addlogin、 sp_adduser、 sp_dropalias、 sp_helpuser

sp_addauditrecord

功能

允许用户将用户定义的审计记录（注释）输入到审计追踪中。

语法

```
sp_addauditrecord [text [, db_name [, obj_name  
[, owner_name [, dbid [, objid]]]]]]
```

参数

text — 是要添加到当前审计表中的消息的文本。该文本将被插入到表的 *extrainfo* 域中。

db_name — 是记录中所引用的数据库的名称。该名称将被插入到当前审计表的 *dbname* 域中。

obj_name — 是记录中所引用的对象的名称。该名称将被插入到当前审计表的 *objname* 域中。

owner_name — 是记录中引用的对象的所有者。该名称将被插入到当前审计表的 *objowner* 域中。

dbid — 是 *db_name* 的数据库 ID 号。不要用引号将该整数引起来。
dbid 将被插入到当前审计表的 *dbid* 域中。

objid — 是 *obj_name* 的对象 ID 号。不要用引号将该整数值引起来。
objid 将被插入到当前审计表的 *objid* 域中。

示例

```
1. sp_addauditrecord "I gave A. Smith permission to  
view the payroll table in the corporate database.  
This permission was in effect from 3:10 to 3:30 pm  
on 9/22/92.", "corporate", "payroll", "dbo", 10,  
1004738270
```

将以下内容分别添加到当前审计表的各域：将 “I gave A. Smith permission to view the payroll table in the corporate database. This permission was in effect from 3:10 to 3:30 pm on 9/22/92.” 添加到 *extrainfo* 域；将 “corporate” 添加到 *dbname* 域；将 “payroll” 添加到 *objname* 域；将 “dbo” 添加到 *objowner* 域；将 “10” 添加到 *dbid* 域，以及将 “1004738270” 添加到 *objid* 域。

```
2. sp_addauditrecord @text="I am disabling auditing
   briefly while we reconfigure the system",
   @db_name="corporate"
```

将该记录添加到审计追踪。本示例使用带 @ 前缀的参数名，它允许将某些域留空。

注释

- Adaptive Server 将把所有审计记录写入当前审计表。当前审计表由 **current audit table** 配置参数的值决定，而后者是用 **sp_configure** 设置的。所安装的一个系统最多可以有八个系统审计表，分别用 *sysaudits_01*、*sysaudits_02* 直到 *sysaudits_08* 来命名。

► 注意

实际上，记录首先存储在内存中的审计队列中，而审计进程以后将这些记录从审计队列写入到当前审计表中。因此，不能指望立即将审计记录存储到审计表中。

- 在下列情况下，可以使用 **sp_addauditrecord**:
 - 已被授予 **sp_addauditrecord** 的执行权限。（无需特殊的角色。）
 - 已启用审计。（系统安全员已通过 **sp_configure** 启用了 **auditing** 配置参数。）
 - **sp_audit** 的 **adhoc** 选项被设置为 **on**。

权限

只有系统安全员才能执行 **sp_addauditrecord**。 *sybsecurity* 数据库所有者（他必须同时是系统安全员）可以将执行权限授予其他用户。

使用的表

sybsecurity.dbo.sysaudits_01...sysaudits_08

参见

系统过程	sp_audit
------	----------

sp_addaudittable

功能

在安装审计后添加其它系统审计表。

语法

```
sp_addaudittable devname
```

参数

devname — 是审计表的设备名。指定设备名或指定“缺省”。如果指定“缺省”，Adaptive Server 将在 *sybsecurity* 数据库所在的同一设备上创建审计表。否则，Adaptive Server 在指定的设备上创建表。

示例

1. sp_addaudittable auditdev2

在 *auditdev2* 上创建系统审计表。如果在执行该过程时，只存在一个系统审计表 (*sysaudits_01*)，Adaptive Server 将把新审计表命名为 *sysaudits_02* 并把它放在其自身的段上，该段名为 *aud_seg_02*，位于 *auditdev2* 上。

2. sp_addaudittable "default"

在 *sybsecurity* 数据库所在的同一设备上创建系统审计表。如果在执行该过程时，存在两个系统审计表 (*sysaudits_01* 和 *sysaudits_02*)，Adaptive Server 将把新的审计表命名为 *sysaudits_03* 并把它放在其自身的段上，该段名为 *aud_seg_03*，位于 *sybsecurity* 数据库所在的同一设备上。

注释

- 在运行 **sp_addaudittable** 时，必须已经安装了审计。按照以下过程添加系统审计表：
 - a. 用 **disk init** 创建审计表的设备。例如，对于 UNIX，可以运行以下命令：

```
disk init name = "auditdev2",  
physname = "/dev/rxyla",  
vdevno = 2, size = 5120
```
 - b. 用 **alter database** 命令将设备添加到 *sybsecurity* 数据库。例如，要将 *auditdev2* 添加到 *sybsecurity* 数据库，可以使用以下命令：

```
alter database sybsecurity on auditdev2
```
 - c. 执行 **sp_addaudittable** 来创建该表。

- **Adaptive Server** 将根据已经定义的审计表数来命名新系统审计表和新段。例如，如果在执行该过程之前已经定义了五个审计表，**Adaptive Server** 将把新审计表命名为 **sysaudits_06**，而把新段命名为 **aud_seg_06**。如果指定了“缺省”，**Adaptive Server** 将该段放在 **sybsecurity** 数据库所在的同一数据库上。否则，**Adaptive Server** 将该段放在指定的设备上。
- 最多允许有八个审计表。在已经有了八个审计表的情况下，如果试图执行 **sp_addauditable** 来添加新的审计表，**Adaptive Server** 将显示错误消息。
- 有关安装审计的方法，参见 **the installation documentation for your platform**。有关使用审计方法的讨论，参见 *系统管理指南*。

权限

只有同时具有系统管理员和系统安全员角色的用户才能执行 **sp_addauditable**。

使用的表

sybsecurity..sysobjects

参见

系统过程	sp_audit
------	----------

sp_addengine

功能

为现有引擎组添加引擎，如果该组不存在，则创建引擎组并添加引擎。

语法

```
sp_addengine engine_number, engine_group
```

参数

engine_number — 是要添加到组中的引擎数。合法的值介于 0 和配置的联机引擎数减一的数值之间。

engine_group — 是要添加引擎的引擎组的名称。如果 **engine_group** 不存在，**Adaptive Server** 则创建该组并向组中添加引擎。引擎组名称必须遵循标识符的规则。有关详细信息，参见第 3 章“表达式、标识符和通配符”。

示例

```
1. sp_addengine 2, DS_GROUP
```

如果没有名为 **DS_GROUP** 的引擎组，该语句将创建该组。如果 **DS_GROUP** 已经存在，该语句将在该组中添加第 2 个引擎。

注释

- 如果 **engine_group** 的值不存在，**sp_addengine** 将创建新引擎组。
- 引擎组 **ANYENGINE** 和 **LASTONLINE** 是预定义的。**ANYENGINE** 包含所有现有的引擎。**LASTONLINE** 指定引擎编号最高的引擎。系统管理员可以创建其它引擎组。不能修改预定义的引擎组。
- 只要使用 **sp_bindexclass** 将应用程序或登录绑定到与 **engine_group** 相关的执行类，相关的进程将在 **engine_number** 上开始运行。
- 在进行与引擎相关的赋值前，需要研究环境并考虑可用的非首选应用程序数和 **Adaptive Server** 引擎数。有关非首选应用程序的详细信息，参见 *Performance and Tuning Guide*。

权限

只有系统管理员才能执行 **sp_addengine**。

使用的表*master..sysattributes***参见**

系统过程	sp_addexclass、sp_bindexclass、 sp_clearpsexex、sp_dropengine、sp_setpsexex、 sp_showcontrolinfo、sp_showexclass、 sp_showpsexex、sp_unbindexclass
------	---

sp_addexeclass

功能

创建或更新用户定义的执行类，该类可以绑定到客户端应用程序、登录和存储过程上。

语法

```
sp_addexeclass classname, priority, timeslice,  
engine_group
```

参数

classname — 是新执行类的名称。

priority — 是优先级值，当该值与执行类关联后，可以利用它来运行客户端应用程序、登录或存储过程。合法的值为 **HIGH**、**LOW** 和 **MEDIUM**。

timeslice — 是指派给与该类相关的进程的时间单元。当前，Adaptive Server 将忽略该参数。

engine_group — 标识现有的引擎组，而与该类相关的进程可以在这些引擎上运行。

示例

```
1. sp_addexeclass "DS", "LOW", 0, "DS_GROUP"
```

该语句定义了新的名为 *DS* 的执行类（此执行类的 *priority* 值为 **LOW**），并将此执行类与引擎组 *DS_GROUP* 关联起来。

注释

- **sp_addexeclass** 创建或更新用户定义的执行类，您可以将该执行类绑定到客户端应用、登录和存储过程上。如果该类已经存在，将用用户提供的值更新类特性值。
- 如果不希望将执行对象限制在引擎组内，可以使用预定义的引擎组参数 **ANYENGINE**。
- 使用 **sp_addengine** 可以定义引擎组。使用 **sp_showexeclass** 可以显示执行类特性以及与该执行类相关的任何引擎组中的引擎。
sp_showcontrolinfo 可以列出现有引擎组。

权限

只有系统管理员才能执行 **sp_addexeclass**。

使用的表*master..sysattributes***参见**

系统过程	sp_addengine、sp_bindexeclass、 sp_clearpsexec、sp_dropengine、 sp_dropexeclass、sp_setpsexec、 sp_showcontrolinfo、sp_showexeclass、 sp_showpsexec、sp_unbindexeclass
------	---

sp_addextendedproc

功能

在 *master* 数据库中创建扩展存储过程 (ESP)。

语法

```
sp_addextendedproc esp_name, dll_name
```

参数

esp_name — 是扩展存储过程的名称。该名称必须与执行 ESP 的过程语言函数的名称相同。 *esp_name* 必须是有效的 Adaptive Server 标识符。

dll_name — 是动态链接库 (DLL) 文件的名称，该文件包含 *esp_name* 指定的函数。可以指定 *dll_name* 不带扩展名，也可以指定其带有特定平台的扩展名，如 Windows NT 上的 *.dll* 或 Sun Solaris 上的 *.so*。如果指定了扩展名，则必须用引号将 *dll_name* 引起来。

示例

```
1. sp_addextendedproc xp_echo, "sqlsrvdll.dll"
```

为名为 *xp_echo* 的函数注册 ESP，该函数在 *sqlsrvdll.dll* 文件中。生成的 ESP 数据库对象的名称也是 *xp_echo*。

注释

- 从 *master* 数据库执行 **sp_addextendedproc**。
- *esp_name* 区分大小写。它必须与 DLL 中的函数名匹配。
- *dll_name* 表示的 DLL 必须驻留在创建 ESP 的服务器计算机上，并且 DLL 目录必须在 *\$PATH* 中（对于 Windows NT）、*\$LD_LIBRARY_PATH* 中（对于 Digital UNIX）或在 *\$SSH_LIBRARY_PATH* 中（对于 HP）。如果文件没有找到，搜索机制还将搜索 *\$SYBASE/dll*（对于 Windows NT）和 *\$SYBASE/lib*（对于其它平台）。
- 在 Windows NT 上，ESP 函数不调用 C 运行期信号例程。这会导致 XP Server 失败，因为 Open Server™ 不支持 Windows NT 上的信号处理。

权限

只有系统管理员才能执行 **sp_addextendedproc**。

使用的表

master.dbo.syscomments、*sysobjects*

参见

命令	create procedure
系统过程	sp_dropextendedproc、sp_helpextendedproc

sp_addexternlogin

(仅适用于组件集成服务)

功能

创建备用的登录账号和口令，以供在通过组件集成服务与远程服务器进行通信时使用。

语法

```
sp_addexternlogin server, loginname, externname  
[, externpassword]
```

参数

server — 是远程服务器的名称，该服务器已经通过 **sp_addserver** 添加到本地服务器。

loginname — 是 Adaptive Server 登录账号的名称，可以为该账号创建替代的登录账号。

externname — 是远程服务器 *server* 上的账号名称。该账号在登录到 *server* 时使用。

externpassword — 是 *externname* 的口令。

示例

1. **sp_addexternlogin JOBSERV, sa, system, sys_pass**

允许本地服务器以用户 “sa” 的身份使用远程登录名 “system” 和口令 “sys_pass”，访问远程服务器 JOBSERV。

2. **sp_addexternlogin CIS1012, bobj, jordan, hitchpost**

当用户 “bobj” 登录到远程服务器 CIS1012 上时，他使用远程服务器登录名 “jordan” 和口令 “hitchpost” 进行连接。

注释

- **sp_addexternlogin** 在与远程服务器进行通信时，将指派要使用的替代登录名和口令。它以加密的形式将该口令存储在内部。
只有在安装并配置了组件集成服务的情况下，才能使用 **sp_addexternlogin**。
- 在运行 **sp_addexternlogin** 之前，用 **sp_addserver** 将远程服务器添加到 Adaptive Server。

- *externname* 和 *externpassword* 必须是 *server* 所运行节点处的有效用户和口令的组合。
- 具有自动口令到期功能的站点需要制定计划，以定期更新外部登录使用的口令。
- 使用 **sp_dropexternlogin** 可以删除外部登录的定义。
- 不能从事务内部使用 **sp_addexternlogin**。
- 只有 “sa” 账号和 *loginname* 账号所代表的用户才能对给定本地用户的远程访问进行修改。

权限

只有 *loginname*、系统管理员和系统安全员才能执行 **sp_addexternlogin**。

使用的表

master.dbo.syslogins、 *master.dbo.sysattributes*、 *master.dbo.sysservers*

参见

系统过程	sp_addserver 、 sp_dropexternlogin 、 sp_helpserver
------	---

sp_addgroup

功能

将组添加到数据库。在授予和撤消权限时，组是作为集合名使用的。

语法

```
sp_addgroup grpname
```

参数

grpname — 是组名。组名必须遵循标识符的规则。

示例

1. sp_addgroup accounting

在当前数据库中创建名为 *accounting* 的组。

注释

- **sp_addgroup** 向数据库的 *sysusers* 表中添加新组。每个组的用户 ID (*uid*) 都是 16384 或更大 (“public” 除外，它总是为 0)。
- 组名和用户名不能相同。
- 创建组后，需要用 **sp_adduser** 添加新用户。要将现有用户添加到组中，可以使用 **sp_changegroup**。
- 每个数据库都是用名为 “public” 的组创建的。每个用户都自动成为 “public” 的成员。每个用户都可以是另一个组的成员。

权限

只有数据库所有者、系统管理员或系统安全员才能执行 **sp_addgroup**。

使用的表

sysobjects、*sysusers*

参见

命令	grant、revoke
系统过程	sp_adduser、sp_changegroup、sp_dropgroup、sp_helpgroup

sp_addlanguage

功能

定义替代语言的月份和星期及其日期格式。

语法

```
sp_addlanguage language, alias, months, shortmons,  
days, datefmt, datefirst
```

参数

language — 是该语言的正式语言名称，只用 7 位 ASCII 字符输入。

alias — 将替换替代语言的正式名称。您可以输入 “null”，使别名与正式语言名称相同，也可以输入一个自己喜欢的名称。如果终端支持，您可以在别名中使用 8 位 ASCII 字符（如 “français”）。

months — 是 12 个月份的全名列表，其顺序从 January（一月）到 December（十二月），中间用逗号分隔（不允许有空格）。月份名的长度最多 20 个字符，并且可以包含 8 位 ASCII 字符。

shortmons — 是 12 个月份的简称列表，其顺序从一月到十二月，中间用逗号分隔（不允许有空格）。月份简写名称的长度最多为 9 个字符，并且可以包含 8 位 ASCII 字符。

days — 是一周七日的全名列表，其顺序从 Monday（星期一）到 Sunday（星期日），中间用逗号分隔（不允许有空格）。日名的长度最多为 30 个字符，并且可以包含 8 位 ASCII 字符。

datefmt — 是日期分量 *month/day/year* 的日期顺序，供输入 *datetime* 或 *smalldatetime* 数据时使用。有效的参数为：*mdy*、*dmy*、*ymd*、*ydm*、*myd* 或 *dym*。“*dmy*”表示日期采用日/月/年的顺序。

datefirst — 在日期计算中设置的第一个星期日期的编号。例如，Monday 为 1，Tuesday 为 2，依此类推。

示例

```
1. sp_addlanguage french, null,
    "janvier,fevrier,mars,avril,mai,juin,juillet,
    aout,septembre,octobre,novembre,decembre",
    "jan,fev,mars,avr,mai,juin,jui,aout,sept,oct,
    nov,dec",
    "lundi,mardi,mercredi,jeudi,vendredi,samedi,
    dimanche",
    dmy, 1
```

该存储过程将法语添加到可用的服务器语言中。“null”可以使别名与正式名称“french”相同。日期顺序是“dmy”，也就是日/月/年。“1”指定 **lundi** 为星期的第一天，而 **lundi** 是 **days** 列表的第一项。由于法语的星期和月份名称不使用大写（除非它们出现在句子的开头），所以在本示例中使用的是小写形式。

注释

- 通常使用 **langinstall** 实用程序或 **Adaptive Server** 安装程序，从一个 **Adaptive Server** 语言模块添加替代语言。语言模块提供该语言日期的名称和已翻译的错误消息。但是，如果语言模块没有随服务器提供，则使用 **sp_addlanguage** 可以定义日期的名称和格式。
- 使用 **sp_modifylogin** 可以更改用户的缺省语言。如果将用户的缺省语言设置为用 **sp_addlanguage** 添加的语言，但没有该语言的本地化文件，用户将在登录时收到信息性消息，指明客户端软件无法打开本地化文件。

系统表更改

- 通过在 **langid** 列中为每个替代语言插入唯一的数值，**sp_addlanguage** 在 **master.dbo.syslanguages** 中创建一个条目。**langid 0** 是为美式英语保留的。
- **language** 参数变为正式语言名称，存储在 **master.dbo.syslanguages** 的 **name** 列中。语言名必须是唯一的。使用 **sp_helplanguage** 可以显示 **Adaptive Server** 上可用的替代语言的列表。
- 如果在 **alias** 中输入 **NULL**，**sp_addlanguage** 将把 **master.dbo.syslanguages** 中的 **alias** 列设置为正式语言名称，但系统管理员可以用 **sp_setlangalias** 更改 **syslanguage.alias** 的值。
- **sp_addlanguage** 将 **master.dbo.syslanguages** 中的 **upgrade** 列设置为 **0**。

用 *sp_addlanguage* 添加的语言的日期

- 对于用语言模块添加的替代语言，Adaptive Server 将把日期值作为 *datetime* 数据类型发送到客户端，而客户端可以使用本地化文件以用户当前的语言显示日期。对于用 *sp_addlanguage* 添加的日期字符串，使用 *convert* 函数可以将日期转换为服务器的字符数据。

select convert(char, pubdate) from table

其中， *pubdate* 是 *datetime* 数据，而 *table* 是任意表。

- 当用户在日期值上执行数据输入并需要使用由 *sp_addlanguage* 创建的日期名时，客户端必须以字符数据的形式输入这些值并将它们以字符数据的形式发送到服务器。

权限

只有系统管理员才能执行 *sp_addlanguage*。

使用的表

master.dbo.syslanguages、 *sysobjects*

参见

命令	set
系统过程	sp_droplanguage、 sp_helplanguage、 sp_setlangalias、 sp_modifylogin

sp_addlogin

功能

将新的用户账号添加到 **Adaptive Server**；在创建时指定特定登录的口令有效期、最小口令长度和所允许的最大失败登录次数。

语法

```
sp_addlogin loginname, passwd [, defldb]
           [, deflanguage ] [, fullname ] [, passwdexp ]
           [, minpwrlen ] [, maxfailedlogins ]
```

参数

loginname — 是用户的登录名。登录名必须遵循标识符的规则。

passwd — 是用户的口令。口令的长度不得少于 6 个字符。如果指定的口令较短，**sp_addlogin** 将返回错误消息并退出。用引号将包含除字符 **A** 到 **Z**、**a** 到 **z** 或 **0** 到 **9** 以外的字符的口令引起来。对于以 **0** 到 **9** 开头的口令，也要用引号引起来。

defldb — 是用户登录到 **Adaptive Server** 时指派的缺省数据库的名称。如果未指定 **defldb**，将使用缺省的 **master**。

deflanguage — 是当用户登录到 **Adaptive Server** 时指派的缺省语言的正式名称。**Adaptive Server** 缺省语言是由 **default language id** 配置参数定义的，如果您未指定 **deflanguage**，将使用该缺省语言。

fullname — 是拥有登录账号的用户的全称。该名称可供归档和识别使用。

passwdexp — 指定口令的有效期，用天表示。它可以是 0 到 32767 之间的任意值，包括 0 和 32767。

minpwrlen — 指定登录要求的最小口令长度。这些值介于 0 到 30 个字符之间。

maxfailedlogins — 指定允许的失败登录尝试次数。它可以是 0 到 32767 之间的任意整数。

示例

1. `sp_addlogin albert, longer1, corporate`

为 “albert” 创建 Adaptive Server 登录，口令为 “longer1”，缺省数据库为 *corporate*。

2. `sp_addlogin claire, bleurouge, public_db, french`

为 “claire” 创建 Adaptive Server 登录。她的口令为 “bleurouge”，缺省数据库为 *public_db*，缺省语言为 French（法语）。

3. `sp_addlogin robertw, terrible2, public_db, null, "Robert Willis"`

为 “robertw” 创建 Adaptive Server 登录。他的口令为 “terrible2”，缺省数据库为 *public_db*，而他的全名为 “Robert Willis”。不要用引号将 `null` 引起来。

4. `sp_addlogin susan, wonderful, null, null, "Susan B. Anthony"`

为 “susan” 创建登录，口令为 “wonderful”，全名为 “susan B. Anthony”，使用服务器的缺省数据库和语言。不要用引号将 `null` 引起来。

5. `sp_addlogin susan, wonderful, @fullname="Susan B. Anthony"`

示例 4 中显示了创建登录的另一方法。

注释

- 为了易于管理，强烈推荐所有用户都使用其操作系统登录名称作为其 Adaptive Server 登录名称。这样便于将操作系统和 Adaptive Server 之间的审计数据关联起来。否则，需要记录操作系统和服务器登录名称之间的对应关系。
- 在用 `sp_addlogin` 为用户指派了缺省数据库后，数据库所有者或系统管理员必须通过执行 `sp_adduser` 或 `sp_addalias` 来提供对数据库的访问。
- 虽然用户任何时候都可以使用 `sp_modifylogin` 更改自己的缺省数据库，但如果没有获得数据库所有者授予的权限，将无法使用数据库。
- 用户可以在任何时候使用 `sp_password` 更改自己的口令。系统安全人员可以使用 `sp_password` 更改任何用户的口令。
- 用户可以使用 `sp_modifylogin` 更改自己的缺省语言。系统管理员可以使用 `sp_modifylogin` 更改任何用户的缺省语言。

- 用户可以使用 **sp_modifylogin** 更改自己的 *fullname*。系统管理员可以使用 **sp_modifylogin** 更改任何用户的 *fullname*。

权限

只有系统管理员或系统安全员才能执行 **sp_addlogin**。

使用的表

master.dbo.sysdatabases、 *master.dbo.syslogins*、 *sysobjects*

参见

系统过程	sp_addalias 、 sp_adduser 、 sp_droplogin 、 sp_locklogin 、 sp_modifylogin 、 sp_password 、 sp_role
------	--

sp_addmessage

功能

将用户定义的消息添加到 *sysusermessages*，以供存储过程 *print* 和 *raiserror* 调用以及 *sp_bindmsg* 使用。

语法

```
sp_addmessage message_num, message_text [, language  
[, with_log [, replace]]]
```

参数

message_num — 是要添加的消息的编号。用户定义的消息编号必须等于或大于 20000。

message_text — 是要添加消息的文本。其最大长度为 255 字节。

language — 要添加消息的语言。它必须是 *syslanguages* 表中的有效语言名称。如果该参数缺失，Adaptive Server 将假定消息的语言为 @@langid 指定的缺省会话语言。

with_log — 指定在启用日志的情况下，是否将消息记录在 Adaptive Server 错误日志以及 Windows NT 服务器的 Windows NT 事件日志中。如果 *with_log* 为 TRUE，将消息记录下来，而不考虑错误的严重性。如果 *with_log* 为 FALSE，消息可以记录下来，也可以不记录，具体情况取决于错误的严重性。如果未指定 *with_log* 的值，则缺省状态为 FALSE。

replace — 指定是否覆盖具有相同编号和 *language* 的现有消息。如果指定了 *replace*，将覆盖现有消息；如果省略 *replace*，则不会覆盖现有消息。如果未指定 *replace* 的值，将使用缺省值 FALSE。

示例

```
1. sp_addmessage 20001, "The table '%1!' is not owned  
by the user '%2!'."
```

将编号为 20001 的消息添加到 *sysusermessages*。

```
2. sp_addmessage 20002, "The procedure '%1!' is not  
owned by the user '%2!'." , NULL, TRUE, "replace"
```

将编号为 20002 的消息添加到 *sysusermessages*。如果启用了事件日志记录，该消息将记录在 Adaptive Server 错误日志以及 Windows NT 服务器的 Windows NT 事件日志中。如果缺省会话语言中存在编号为 20002 的消息，该消息将覆盖原有的消息。

注释

- **sp_addmessage** 不覆盖具有相同编号和 *langid* 的现有消息，除非您指定了 **@replace = "replace"**。
- **print**、**raiserror** 和 **sp_bindmsg** 可以识别在消息文本中输出的占位符。一个消息最多可以包含 **20** 个占位符数，它们的顺序可以是任意的。当消息文本发送到客户端时，可以用消息后面的参数的格式化内容替换这些占位符。

对这些占位符进行编号，这样当 **Adaptive Server** 将消息翻译为其它语法结构的语言时，就可以对参数重新排序。参数的占位符显示为 “%nn!”，即一个百分比符号 (%) 后面跟着一个从 **1** 到 **20** 的整数，再后面跟着一个惊叹号 (!)。该整数表示参数列表中字符串中的参数编号。“%1!” 表示原始版本的第一个参数，“%2!” 表示第二个参数，依此类推。

权限

任何用户都可以执行 **sp_addmessage**。

使用的表

master.dbo.syslanguages、*sysobjects*、*sysusermessages*

参见

命令	print、raiserror
系统过程	sp_altermessage、sp_dropmessage、 sp_getmessage

sp_addobjectdef

(仅适用于组件集成服务)

功能

指定本地表与外部存储位置之间的映射。

语法

```
sp_addobjectdef tablename, "objectdef" [, "objecttype"]
```

参数

tablename — 是在本地表中定义的对象名称。**tablename** 可以是以下形式之一：

- *dbname.owner.object*
- *dbname..object*
- *owner.object*
- *object*

dbname 和 **owner** 是可选的。**object** 是必选的。如果未指定 **owner**，将使用缺省值（当前用户名）。如果指定 **dbname**，它必须是当前用户名，您必须指定 **owner** 或用占位符以格式 *dbname..object* 标记所有者。用引号将所有多分量 **tablename** 值引起来。

objectdef — 是命名对象的外部存储位置的字符串。**objectdef** 上的 **objecttype** 可以是远程服务器可以访问的表、视图或只读远程过程调用 (RPC) 结果集。对于 **objectdef**，表、视图或 RPC 使用以下格式：

server_name.dbname.owner.object

server_name 和 **object** 是必选的。**dbname** 和 **owner** 是可选的，但如果没有提供两者，则在格式 *dbname..object* 中必须有一个占位符。

有关详细信息，参见 *Component Integration Services User's Guide* 中的 “Server Classes”。

objecttype — 是指定对象格式的值之一，而该对象是由 *objectdef* 命名的。表 7-2 对有效值进行了说明。用引号将 *objecttype* 值引起来。

表 7-2: objecttype 允许使用的值

值	说明
table	指示由 <i>objectdef</i> 命名的对象作为远程服务器可以访问的表。该值是 <i>objecttype</i> 的缺省值。
view	指示由 <i>objectdef</i> 命名的对象是远程服务器管理的视图并且作为表处理。
rpc	指示由 <i>objectdef</i> 命名的对象是远程服务器管理的 RPC。Adaptive Server 将 RPC 的结果集作为只读表处理。

表 7-3 总结了每个 *objecttype* 的使用方法：

表 7-3: objecttype 用法摘要

<i>objecttype</i>	创建表	创建现有表	写入表	读取表
table	是	是	是	是
view	否	是	是	是
rpc	否	是	否	是

示例

1. `sp_addobjectdef "finance.dbo.accounts",
"SYBASE.pubs.dbo.accounts", "table"`

将数据库 *finance* 中的本地表 *accounts* 映射到名为 SYBASE 的远程服务器中的远程对象 *pubs.dbo.accounts* 上。当前数据库必须是 *finance*。后续的 `create table` 将在 *pubs* 数据库中创建表。如果 *pubs.dbo.accounts* 是现有表，`create existing table` 语句将在表 *finance.dbo.accounts* 中填入有关远程表的信息。

2. `sp_addobjectdef stockcheck,
"NEWYORK.wallstreet.kelly.stockcheck", "rpc"`

将本地表 *stockcheck* 映射到远程服务器 NEWYORK 中名为 *stockcheck* 的 RPC 上。该服务器位于数据库 *wallstreet* 中，所有者为 “kelly”。RPC *stockcheck* 结果集被看作是一个只读表。通常，下一个操作是针对对象 *stockcheck* 的 `create existing table` 语句。

注释

- **sp_addobjectdef** 指定本地表和外部存储位置之间的映射。它标识在该位置的对象的格式。
只有在安装并配置了组件集成服务的情况下，才能使用 **sp_addobjectdef**。
- **sp_addobjectdef** 将替换 **sp_addtabledef** 命令。**sp_addtabledef** 允许在不作修改的情况下运行现有脚本。在内部，**sp_addtabledef** 将调用 **sp_addobjectdef**。
- 只有系统管理员才能提供作为表所有者的其他用户的名称。
- 当 *objecttype* 是 **table**、**view** 或 **rpc** 时，*objectdef* 参数将采取以下形式：
"server_name.database.owner.tablename"
 - *server_name* 表示已经由 **sp_addserver** 添加到 *sys.servers* 中的服务器。
 - *database* 不一定是必需的。某些服务器类不支持它。
 - 始终应该提供 *owner*，以避免歧义。如果不指定 *owner*，引用的远程对象可能会不同，这取决于外部登录是否与远程对象所有者相对应。
 - *tablename* 是远程服务器表的名称。
- 在发出任何 **create table** 或 **create existing table** 命令之前，需要使用 **sp_addobjectdef**。**create table** 仅对 *objecttype* 的值 **table** 和 **file** 有效。当使用 **create table** 或 **create existing table** 时，Adaptive Server 将检查 *sysattributes* 以确定是否为该对象指定了表映射。在 *objecttype* 值 **view** 和 **rpc** 之后使用 **create existing table** 语句。
- 创建表之后，以后所有对本地表名的引用（通过 **select**、**insert**、**delete** 和 **update**）都将映射到正确的位置。
- 有关 RMS 的信息，参见 *Component Integration Services User's Guide*。

权限

任何用户都可以执行 **sp_addobjectdef**。

使用的表

sysobjects、*sysattributes*、*sys.servers*

参见

命令	create existing table、create table、drop table
系统过程	sp_addlogin、sp_addserver、sp_defaultloc、 sp_dropobjectdef、sp_helpserver

sp_add_qpgroup

功能

添加抽象计划组。

语法

```
sp_add_qpgroup new_name
```

参数

new_name — 是新的抽象计划组的名称。组名必须是有效的标识符。

示例

```
1. sp_add_qpgroup dev_plans
```

创建新的名为 *dev_plans* 的抽象计划组。

注释

- 使用 **sp_add_qpgroup** 可以在捕获或创建抽象计划中添加抽象计划组。抽象计划组必须已经存在，才能在组中进行创建、保存或复制。
- **sp_add_qpgroup** 不能在事务中运行。

权限

只有系统管理员和数据库所有者才能执行 **sp_add_qpgroup**。

使用的表

sysattributes

参见

命令	set plan
系统过程	sp_help_qpgroup

sp_addremotelogin

功能

通过在 *master.dbo.sysremotelogins* 中添加条目向新的远程服务器用户授权。

语法

```
sp_addremotelogin remoteserver [, loginame  
                                [, remotename] ]
```

参数

remoteserver — 是远程登录申请的远程服务器的名称。本地服务器必须通过 *master.dbo.sysservers* 表中的条目了解该服务器，而该条目是用 *sp_addserver* 创建的。

► 注意

本手册页使用术语“本地服务器”表示正在执行从“远程服务器”运行的远程过程的服务器。

loginame — 是本地服务器上用户的登录名。*loginame* 必须已经存在于 *master.dbo.syslogins* 表中。

remotename — 是在登录到本地服务器时远程服务器所使用的名称。所有没有显式与本地 *loginame* 匹配的 *remotename* 都自动与某个本地名匹配。在示例 1 中，本地名是用于登录的远程名。在示例 2 中，本地名是“albert”。

示例

1. sp_addremotelogin GATEWAY

在 *sysremotelogins* 表中为远程服务器 GATEWAY 创建条目，供登录验证使用。当本地服务器和远程服务器的用户相同时，这是一种将远程名映射到本地名的简单方法。

在 *sysremotelogins* 的某一行中，本示例将为 *suid* 生成值 -1，为 *remoteusername* 生成值 NULL。

2. sp_addremotelogin GATEWAY, albert

创建条目，将所有来自远程服务器 GATEWAY 的登录映射到用户名 “albert”。Adaptive Server 将向 *sysremotelogins* 中添加一行，其中 Albert 的服务器用户 ID 在 *suid* 列中，*remoteusername* 为空值。

要使这些登录能够在本地服务器上运行 RPC，必须指定在登录到本地服务器时用于 RPC 连接的口令，或者在本地服务器上其状态必须是 “trusted”。要将这些登录定义为 “trusted”，可以使用 *sp_remotoption*。

3. sp_addremotelogin GATEWAY, ralph, pogo

将远程登录从远程服务器 GATEWAY 上的远程用户 “pogo” 映射到本地用户 “ralph”。Adaptive Server 将在 *sysremotelogins* 中添加一行，其中 Ralph 的服务器用户 ID 在 *suid* 列中，而 “pogo” 在 *remoteusername* 列中。

注释

- 本地服务器在收到远程登录时，将尝试用三种不同的方法将远程用户映射到本地用户：
 - 首先，本地服务器将在 *sysremotelogins* 中查找与远程服务器名和远程用户名匹配的行。如果本地服务器找到了匹配行，远程用户将使用该行的本地服务器用户 ID 进行登录。这适用于来自指定远程用户的映射。
 - 如果没有找到匹配映射，本地服务器将搜索具有空远程名且本地服务器用户 ID 不为 -1 的行。如果找到了这样的行，远程用户将被映射到该行中的本地服务器用户 ID。这适用于将远程服务器的任何远程用户映射到特定本地名的映射。
 - 最后，如果前面的尝试失败了，本地服务器将在 *sysremotelogins* 表中查找具有空远程名且本地服务器用户 ID 为 -1 的条目。如果找到了这样的行，本地服务器将使用远程服务器提供的远程名在 *syslogins* 表中查找本地服务器用户 ID。这适用于远程服务器和本地服务器的登录名相同的情况。
- 本地用户名在远程服务器上可能会有变化。
- 如果使用 *sp_addremotelogin* 将远程服务器的所有用户映射为相同的本地名，可使用 *sp_remotoption* 为这些用户指定 “trusted” 选项。例如，要使所有映射到 “albert” 的服务器 GOODSRV 的用户都为 “trusted”（被信任）状态，可以按照如下所示使用 *sp_remotoption*：

```
sp_remotoption GOODSRV, albert, NULL, trusted,
               true
```

没有被指定为 “trusted”（被信任）的登录不能在本地服务器上执行 RPC，除非在登录到远程服务器时为本地服务器指定了口令。在 Open Client™ Client-Library™ 中，用户可以使用 **ct_remote_pwd** 例程指定服务器到服务器的连接口令。**isql** 和 **bcp** 不允许用户指定 RPC 连接的口令。

如果用户使用 “统一登录” 登录到远程服务器，则安全性机制已经对这些登录进行了验证。这些登录在本地服务器上必须是 “被信任” 的，否则用户必须指定登录到远程服务器时服务器的口令。

- 有关为远程过程调用设置服务器和使用 “统一登录” 的详细信息，参见 *系统管理指南*。
- 每个远程登录条目都有一个状态。**trusted** 选项的缺省状态是 **false**（不被信任）。这意味着在远程登录开始使用该条目时，将检查口令。如果不希望检查口令，可以用 **sp_remotoption** 将 **trusted** 选项的状态更改为 **true**。

权限

只有系统管理员才能执行 **sp_addremotelogin**。

使用的表

*master.dbo.syslogins、 master.dbo.sysremotelogins、
master.dbo.sysservers、 sysobjects*

参见

系统过程	sp_addlogin、 sp_addserver、 sp_dropremotelogin、 sp_helpremotelogin、 sp_helprotect、 sp_helpserver、 sp_remotoption
实用程序	isql

sp_add_resource_limit

功能

创建对 Adaptive Server 登录和/或应用程序可用来执行查询、查询批处理或事务的服务器资源数量限制。

语法

```
sp_add_resource_limit name, appname, rangename,  
    limittype, limitvalue [, enforced [, action  
    [, scope ]]]
```

参数

- name* — 是要施加限制的 Adaptive Server 登录。必须指定 *name* 或 *appname*，或指定两者。要创建施加于特定应用程序所有用户的限制，可以将 *name* 指定为 NULL。
- appname* — 是要施加限制的应用程序的名称。必须指定 *name* 或 *appname*，或指定两者。要创建施加于 Adaptive Server 登录所使用的所有应用程序的限制，可以将 *appname* 指定为 null。要创建施加于特定应用程序的限制，可以指定客户端程序在登录包中传递给 Adaptive Server 的应用程序的名称。
- rangename* — 是施加限制的时间范围。在创建限制时，时间范围必须存在于 *master* 数据库的 *systimeranges* 系统表中。
- limittype* — 是限制的资源类型。它可以为以下类型之一：

限制类型	说明
row_count	限制查询可以返回的行数
elapsed_time	限制查询批处理或事务运行允许经历的时间，用秒表示
io_cost	限制处理查询的实际开销或优化程序的开销估计值

limitvalue — 是在 **Adaptive Server** 施加限制之前，登录或应用程序可以使用的服务器资源（I/O 开销、用秒表示的经历时间或行数）的最大值。该值必须是正的非零整数，且小于或等于 2^{31} 。下表给出了应该为每种限制类型指定什么值：

限制类型	限制值
row_count	在施加限制之前，查询可以返回最大的行数。
elapsed_time	在施加限制之前，查询批处理或事务运行允许经历的时间，用秒表示。
io_cost	从优化程序的开销公式中派生出来的无单位度量值。

enforced — 确定在查询执行之前或在查询执行过程中是否施加限制。下表列出了每种限制类型的有效值：

enforced 代码	说明	限制类型
1	当估计的执行 I/O 开销超过指定限制时进行操作。	io_cost
2	当实际行计数、经历的时间或执行 I/O 开销超过指定限制时进行操作。	row_count elapsed_time io_cost
3	当估计开销或实际开销超过指定限制时进行操作。	io_cost

如果指定 **enforced** 的值为 **3**，**Adaptive Server** 将对 **1** 和 **2** 执行逻辑“或”。例如，假定将 **enforced** 设置为 **3**。如果运行 **io_cost** 超过估计开销的查询，将执行指定的 **action**。如果查询在指定的估计开销限制之内但超过了实际开销，也将执行指定的 **action**。

如果不指定 **enforced** 值，**Adaptive Server** 将对 **row_count** 和 **elapsed_time** 施加限制 **2**，而对 **io_cost** 施加限制 **3**。换句话说，如果限制类型为 **io_cost**，在查询超过估计开销或实际开销的情况下，将执行指定操作。

action — 是超过限制时采取的操作。以下操作代码对所有限制类型都有效：

<i>action</i> 代码	说明
1	发出警告
2	中止查询批处理
3	中止事务
4	注销会话

如果不指定 *action* 值， Adaptive Server 将使用缺省值 2 （中止查询批处理）。

scope — 是限制的范围。指定以下适用于该限制类型的代码之一：

<i>scope</i> 代码	说明	限制类型
1	查询	io_cost row_count
2	查询批处理 （通过客户端发送到服务器的一个或多个 SQL 语句）	elapsed_time
4	事务	elapsed_time
6	查询批处理和事务	elapsed_time

如果不指定 *scope* 值， 限制将适用于限制类型的所有可能的范围。

示例

1. `sp_add_resource_limit NULL, payroll, early_morning, elapsed_time, 120, 2, 1, 2`

在 *early_morning* 时间范围内，创建适用于 *payroll* 应用程序所有用户的资源限制。如果执行查询批处理的时间超过 120 秒， Adaptive Server 将发出警告。
2. `sp_add_resource_limit joe_user, NULL, midday, row_count, 5000, 2, 3, 1`

在 *midday* 时间范围内，创建适用于 “joe_user” 所运行的所有即席查询和应用程序的资源限制。当查询返回的行数超过 5000 时， Adaptive Server 将中止事务。

3. `sp_add_resource_limit joe_user, NULL, midday, io_cost, 650, 1, 3, 1`

在 *midday* 时间范围内，创建适用于 “joe_user” 所运行的所有即席查询和应用程序的资源限制。当优化程序估计 I/O 开销超过 650 时，Adaptive Server 将中止事务。

注释

- 必须启用 `sp_configure "allow resource limits"` 以使资源限制生效。
- 只要其时间范围不重叠，对于给定的用户、应用程序、限制类型、范围和施加时间，都可以存在多个资源限制。
- 在登录时，当前处于活动状态的命名时间范围和登录和/或应用程序名称的 “at all times” 范围将绑定到用户的会话上。因此，如果用户独立于给定的应用程序登录到 Adaptive Server，将不会施加与该应用程序一起约束用户的资源限制。要保证对该用户上的约束，可以创建针对该用户且独立于任何应用程序的资源限制。
- 由于确定资源限制要使用用户的登录名或应用程序名，或者使用两者，所以 Adaptive Server 在扫描 *sysresourcelimits* 表以搜索适用于登录会话的限制时，遵守预定义的搜索优先级。下表说明了匹配的有序登录名和应用程序名对的优先级：

级	登录名	应用程序名
1	“joe_user”	payroll
2	NULL	payroll
3	“joe_user”	NULL

如果在给定的优先级别中发现了一个或多个匹配项，则不会搜索其它级别。这样可防止用于不同登录/应用程序组合的相同限制之间出现冲突。

如果在所有级别都没有发现匹配项，则不对会话施加限制。

- 在添加、删除或修改资源限制时，Adaptive Server 将在每个会话的下一个查询批处理开始时，为登录和/或应用程序重新捆绑该会话的限制。
- 在更改当前处于活动状态的时间范围时，Adaptive Server 将重新捆绑会话的限制。这种捆绑发生在下一个查询批处理开始时。

- 不能将特定登录、应用程序或登录/应用程序组合的限制与重叠的命名时间范围关联在一起。（使用同一时间范围的限制除外）。
例如，如果限制用户在上午 9:00 和下午 1:00 之间检索 50 行，则不能再创建如下针对同一用户的资源限制：在上午 10:00 和中午 12:00 之间检索 100 行。但是，可以创建资源层次，其方法是：为用户在上午 10:00 和中午 12:00 之间指派 100 行的限制，而为应用程序（如 isql）在上午 9:00 和下午 1:00 之间指派 50 行的限制。
- 有关资源限制的详细信息，参见 *系统管理指南*。

权限

只有系统管理员才能执行 `sp_add_resource_limit`。

使用的表

master..sysresourcelimits、 *master..systimeranges*

参见

系统过程	<code>sp_configure</code> 、 <code>sp_drop_resource_limit</code> 、 <code>sp_help_resource_limit</code> 、 <code>sp_modify_resource_limit</code>
实用程序	<code>isql</code>

sp_addsegment

功能

定义数据库中数据库设备上的段。

语法

```
sp_addsegment segname, dbname, devname
```

参数

segname — 是添加到数据库的 *syssegments* 表的新段的名称。每个数据库中的段名是唯一的。

dbname — 指定要在其中定义段的数据库的名称。*dbname* 必须是当前数据库的名称或与限定 **sp_addsegment** 的数据库名称相匹配。

devname — 是要在其中定位 *segname* 的数据库设备的名称。可以有多个段与一个数据库设备相关联。

示例

1. **sp_addsegment indexes, pubs2, dev1**

在名为 *dev1* 的数据库设备上为 *pubs2* 数据库创建名为 *indexes* 的段。

2. **disk init**

```
name = "pubs2_dev",  
physname = "/dev/pubs_2_dev",  
vdevno = 9, size = 5120  
  
go  
alter database pubs2 on pubs2_dev = 2  
go  
pubs2..sp_addsegment indexes, pubs2, dev1
```

在名为 *dev1* 的数据库设备上为 *pubs2* 数据库创建名为 *indexes* 的段。

注释

- **sp_addsegment** 为用 **disk init** 创建的数据库设备定义段名，并用 **alter database** 或 **create database** 命令将其指派给特定数据库。
- 在定义段以后，将其用于 **create table** 和 **create index** 命令以及 **sp_placeobject** 过程，从而把表或索引放置到段上。

在特定段上创建表或索引后，该表或索引的所有后续数据也位于该段上。

- 使用 **sp_extendsegment** 系统过程可将段的范围扩展到同一数据库所用的其它数据库设备。
- 如果在该数据库所用的设备上使用 **alter database** 扩展了该数据库，则映射到该设备的段也将被扩展。
- **system** 和 **default** 段将映射到 **create database** 或 **alter database** 命令所包括的每一个数据库设备。**logsegment** 也映射到每个设备，除非使用 **create database** 的 **log on** 扩展或使用 **sp_logdevice** 将其置于不同设备上。有关详细信息，参见 *系统管理指南*。
- 如果数据库的数据和日志同在一个设备上，则在该数据库中尝试使用 **sp_addsegment** 时， Adaptive Server 将返回错误消息。

权限

只有数据库所有者或系统管理员才能执行 **sp_addsegment**。

使用的表

master.dbo.sysdevices、 *master.dbo.sysusages*、 *sysobjects*、 *syssegments*

参见

命令	alter database、 create index、 create table、 disk init
系统过程	sp_dropsegment、 sp_extendsegment、 sp_helpdb、 sp_helpdevice、 sp_placeobject

sp_addserver

功能

定义远程服务器，或者定义本地服务器的名称。

语法

```
sp_addserver lname [, class [, pname]]
```

参数

lname — 是用于表示系统中的服务器的名称。如果 *lname* 的条目尚不存在，**sp_addserver** 将向 *sys.servers* 表添加一行。服务器的名称必须是唯一的，并且必须符合标识符的规则。

class — 标识所添加的服务器的种类。表 7-4 列出 *class* 参数所允许的值：

表 7-4: server_class 参数的允许值

class 参数值	说明
access_server	编码方式符合 DirectConnect™ 规范的服务器（只适用于组件集成服务）
db2	Net-Gateway™ 或 MDI™ Database Gateway 可访问的服务器（只用于组件集成服务）
direct_connect	功能与 access_server 相同（只用于组件集成服务）
generic	编码方式符合 Generic Access Module 规范的服务器（只适用于组件集成服务）
local	本地服务器（只能有一个），该服务器仅在启动或重新启动 Adaptive Server 后使用一次，用以标识本地服务器名称，以便可以将该名称显示在 Adaptive Server 输出的消息中
null	未定义种类的远程服务器
sql_server	其它 Adaptive Server 或 Omni 服务器（这是缺省值）

pname — 是名为 *lname* 的服务器在接口文件中的名称。这使您可以为其它可能需要进行通信的 Adaptive Server 或 Backup Server 建立本地别名。如果未指定 *pname*，将使用 *lname*。

示例

1. `sp_addserver GATEWAY`

在 `master.dbo.sysservers` 中为名为 GATEWAY 的远程服务器添加一个条目。 `pname` 也是 GATEWAY。

2. `sp_addserver GATEWAY, null, VIOLET`

在 `master.dbo.sysservers` 中为名为 GATEWAY 的远程服务器添加一个条目。 `pname` 为 VIOLET。如果 GATEWAY 已有一个 `sysservers` 条目，但具有不同的 `pname`，服务器 GATEWAY 的 `pname` 将更改为 VIOLET。

3. `sp_addserver PRODUCTION, local`

添加名为 PRODUCTION 的本地服务器的条目。

4. `sp_addserver SQLSRV10, sql_server, SS_MOSS`

为本地服务器所知的远程服务器添加一个条目 SQLSRV10。该远程服务器属于服务器类 `sql_server`。SQLSRV10 的 `network_name` 为 SS_MOSS。

5. `sp_addserver RDBAM_ALPHA, generic, rdbam_alpha`

为本地服务器所知的远程服务器添加一个名为 `RDBAM_ALPHA` 的条目。远程服务器 `RDBAM_ALPHA` 被写入 Generic Access Module 规范，该规范要求服务器类 `generic`。

注释

- `sysservers` 表标识本地服务器的名称及其选项，以及所有本地服务器可与之进行通信的远程服务器。
要在远程服务器上执行远程过程调用，`sysservers` 表中必须存在该远程服务器。
- 如果 `sysservers` 表中已存在服务器名称 `lname`，`sp_addserver` 将把远程服务器的 `srvnetname` 更改为 `pname` 所指定的名称。此时，`sp_addserver` 会报告它所更改的服务器、旧的网络名称以及新的网络名称。
- 服务器的安装或更新进程将在 `sysservers` 中添加用于 Backup Server 的条目。如果删除此条目，则无法备份数据库。
- Adaptive Server 要求 Backup Server 具有 `lname` SYB_BACKUP。如果不希望将它用作 Backup Server 的名称，或者有多个 Backup Server 在系统上运行，则使用 `sp_addserver` 修改服务器 SYB_BACKUP 的 `pname`，以便 Adaptive Server 可以与 Backup Server 进行通信以进行数据库转储和装载。
- 如果所指定的 `lname`、`pname` 和 `class` 已存在于 `sysservers`，`sp_addserver` 将输出错误消息，且不更新 `sysservers`。

- 使用 **sp_serveroption** 来设置或清除服务器选项。
- 有关使用组件集成服务的信息，参见 *Component Integration Services User's Guide*。

权限

只有系统安全员才能执行 **sp_addserver**。

使用的表

master.dbo.sysservers、*sysobjects*

参见

系统过程	sp_addremotelogin、sp_dropremotelogin、 sp_dropserver、sp_helpremotelogin、 sp_helpserver、sp_serveroption
------	---

sp_addthreshold

功能

创建阈值来监控数据库段的空间。如果段上的可用空间小于指定级别， Adaptive Server 将执行相关联的存储过程。

语法

```
sp_addthreshold dbname, segname, free_space, proc_name
```

参数

dbname — 是要为其添加阈值的数据库。它必须是当前数据库的名称。

segname — 是监控其可用空间的段。当指定 “default” 段时使用引号。

free_space — 是超过阈值时的可用空间页数。如果段上的可用空间低于此级别， Adaptive Server 将执行相关联的存储过程。

proc_name — 是当 **segname** 的可用空间量小于 **free_space** 时所执行的存储过程。此过程可在当前 Adaptive Server 或 Open Server 的任意数据库中找到。阈值不能执行远程 Adaptive Server 上的过程。

示例

```
1. sp_addthreshold mydb, segment1, 200, pr_warning
```

创建 *segment1* 的阈值。当 *segment1* 的可用空间小于 200 页时， Adaptive Server 将执行过程 *pr_warning*。

```
2. sp_addthreshold userdb, user_data, 100,  
   "o_server...mail_me"
```

为 *user_data* 段创建阈值。当 *user_data* 的可用空间低于 100 页时， Adaptive Server 将执行 Open Server *mail_me* 过程的远程过程调用。

```
3. pubs2..sp_addthreshold pubs2, indexes, 100,  
   pr_warning
```

为 *pubs2* 数据库的 *indexes* 段创建阈值。该命令可从任意数据库中发出。

注释

- 有关使用阈值的详细信息，参见 *系统管理指南*。

超过阈值

- 如果超过阈值，Adaptive Server 将执行相关的存储过程。
Adaptive Server 使用以下阈值过程的搜索路径：
 - 如果过程名中未指定数据库，Adaptive Server 将在超过阈值的数据库中进行搜索。
 - 如果在此数据库中未找到该过程，而该过程名称以 “sp_” 开头，则 Adaptive Server 将在 *sybsystemprocs* 数据库中搜索。

如果在这两个数据库中都没有找到该过程，Adaptive Server 将向错误日志发送错误消息。

- Adaptive Server 使用一个 **停滞值**，即全局变量 `@@thresh_hysteresis`，以确定阈值对可用空间变化的敏感程度。一旦阈值执行其过程，该阈值将失效。该阈值一直保持不活动的状态，直到段中可用空间量增加到 `@@thresh_hysteresis` 页，大于该阈值。这样可防止响应可用空间的微小波动而使阈值重复执行其过程。

最后机会阈值

- 缺省情况下，Adaptive Server 监控日志所处的段的可用空间，如果可用空间量小于进行成功事务日志转储所需的大小，将执行 `sp_thresholdaction`。可用空间量称为 **最后机会阈值**，它是由 Adaptive Server 计算的，用户无法更改。
- 在记录事务前，如果超过了最后机会阈值，Adaptive Server 将挂起该事务，直到释放日志空间。使用 `sp_dboption` 可为特定数据库更改此种行为。`sp_dboption "abort tran on log full", true` 将导致 Adaptive Server 回退所有当超过最后机会阈值时尚未记录的事务。

创建其它阈值

- 每个数据库最多可以有 256 个阈值，其中包括最后机会阈值。
- 添加的阈值必须至少是最接近阈值 `@@thresh_hysteresis` 的页数的两倍。

创建阈值过程

- 任何具有 **create procedure** 权限的用户都可以在数据库中创建阈值过程。通常系统管理员在 **sybsystemprocs** 数据库中创建 **sp_thresholdaction**，数据库所有者则在用户数据库中创建阈值过程。
- **sp_addthreshold** 不检查指定的过程是否存在。在创建阈值所执行的过程前先添加该阈值的可能性是存在的。
- **sp_addthreshold** 将检查以确保添加阈值过程的用户已被直接授予“sa_role”。所有在创建阈值过程中处于活动状态的系统角色将输入到 **systhresholds** 中，作为写入该过程的用户的有效角色。然而，触发阈值时，仅激活直接授予的系统角色。间接授予的系统角色和用户定义的角色则不被激活。
- Adaptive Server 向阈值过程传递四个参数：
 - **@dbname, varchar(30)**，用于标识数据库
 - **@segmentname, varchar(30)**，用于标识段
 - **@space_left, int**，指示与该阈值相关的可用页数
 - **@status, int**，其中包含一个值，对最后机会阈值该值为 1，对其它阈值该值为 0。

按位置传递这些参数，而非名称；阈值过程可以使用它们的其它名称，但必须按参数出现的顺序对其进行声明，而且数据类型必须正确。

- 无需为每个阈值创建不同的过程。要进行最低限度的维护，您可以在 **sybsystemprocs** 数据库中创建一个单个的供 Adaptive Server 中的所有阈值执行的阈值过程。
- 将 **print** 和 **raiserror** 语句包括在阈值过程中，以便向错误日志输出信息。

执行阈值过程

- 超过阈值时启动的任务将作为后台任务执行。这些任务不具有相关的终端或用户会话。如果在运行这些任务时执行 **sp_who**，**status** 列将显示“background”。
- Adaptive Server 在执行该阈值过程时具有用户添加该阈值时所具有的权限，但不具有此后已被删除的任何权限。
- 每个阈值过程在执行时都将使用一个用户连接。

更改或删除阈值

- 使用 **sp_helpthreshold** 可获得有关现有阈值的信息。
- 使用 **sp_modifythreshold** 可将阈值与新的阈值过程、可用空间值或段相关联。（您不能更改与最后机会阈值相关联的可用空间值或段名。）

用户一旦修改阈值，此用户就成为该阈值的所有者。如果超过该阈值，**Adaptive Server** 将执行该阈值，其执行权限是所有者修改该阈值时所具有的权限，但不具有此后已被删除的任何权限。

- 使用 **sp_droptreshold** 从段中删除阈值。

禁用可用空间会计

- 使用 **sp_dboption** 的 **no free space acctg** 选项可在非日志段上禁用可用空间会计。
- 您不能对日志段禁用可用空间统计。

◆ 警告!

如果禁用可用空间会计，系统过程将无法提供有关空间分配的准确信息。

为 11.0 之前版本的数据库创建最后机会阈值

- 从 10.0 以前版本进行升级时，数据库不会自动获得最后机会阈值。使用 **lct_admin** 系统函数可在升级到当前版本的早于 10.0 版本的数据库中创建最后机会阈值。
- 只有将其日志存储在不同段上的数据库才可具有最后机会阈值。使用 **sp_logdevice** 可将事务日志移动到不同设备上。

权限

只有数据库所有者或系统管理员能执行 **sp_addthreshold**。

使用的表

master.dbo.sysusages、sysobjects、syssegments、systhresholds

参见

命令	create procedure、dump transaction
函数	lct_admin
系统过程	sp_dboption、sp_droptreshold、 sp_helpthreshold、sp_modifythreshold、 sp_thresholdaction

sp_add_time_range

功能

将指定的时间范围添加到 Adaptive Server。

语法

```
sp_add_time_range name, startday, endday,  
starttime, endtime
```

参数

name — 是时间范围的名称。时间范围名称必须在 30 个字符以内。
该名称不能已存在于 *master* 数据库的 *systimeranges* 系统表中。

startday — 是时间范围的开始日期。它必须是存储在 *master* 数据库 *syslanguages* 系统表中的缺省服务器语言的完整星期名称。

endday — 时间范围的结束日期。它必须是存储在 *master* 数据库 *syslanguages* 系统表中的缺省服务器语言的完整日期名称。日期中 **endday** 可以早于或晚于 **startday**，也可以与 **startday** 相同。

starttime — 是时间范围开始的当日时间。按 24 小时制，在 “00:00”（午夜）到 “23:59” (11:59 p.m.) 的时间范围内，指定 **starttime**。
其形式如下：

"HH:MM"

endtime — 是时间范围结束的当日时间。按 24 小时制，在 “00:00”（午夜）到 “23:59” (11:59 p.m.) 的时间范围内，指定 **endtime**。
其形式如下：

"HH:MM"

► 注意

要创建一个全天跨度的时间范围，将开始时间和结束时间都指定为 “00:00”。

除非 **endtime** 是 “00:00”，否则当天中的 **endtime** 必须晚于 **starttime**。

示例

1. `sp_add_time_range business_hours, monday, Friday, "09:00", "17:00"`

创建 *business_hours* 时间范围，该范围在星期一到星期五的 9:00 a.m. 到 5:00 p.m. 都有效。

2. `sp_add_time_range before_hours, Monday, Friday, "00:00", "09:00"`

`sp_add_time_range after_hours, Monday, Friday, "18:00", "00:00"`

创建两个时间范围 *before_hours* 和 *after_hours*，二者的跨度是从星期一到星期五的所有非工作时间。*before_hours* 时间范围涵盖从星期一到星期五，午夜 12:00 到上午 9:00 的时间间隔。

after_hours 时间范围涵盖从星期一到星期五，下午 6:00 到午夜 12:00 的时间间隔。

3. `sp_add_time_range weekends, Saturday, Sunday, "00:00", "00:00"`

创建 *weekends* 时间范围，该范围从星期六的午夜 12:00 到星期日的午夜 12:00。

4. `sp_add_time_range Fri_thru_Mon, Friday, Monday, "09:00", "17:00"`

创建 *Fri_thru_Mon* 时间范围，该范围是星期五、星期六、星期日和星期一的上午 9:00 到下午 5:00。

5. `sp_add_time_range Wednesday_night, Wednesday, Wednesday, "17:00", "00:00"`

创建 *Wednesday_night* 时间范围，该范围从星期三的下午 5:00 到午夜 12:00。

注释

- Adaptive Server 有一个指定时间范围，即“全天候”时间范围。此时间范围包括所有时间，从星期中的第一天到最后一天，从 00:00 到 23:59。不能对其进行修改或将其删除。
- Adaptive Server 为每个指定的时间范围生成一个唯一 ID 号，并将其插入 *systimeranges* 系统表。
- 在 *systimeranges* 系统表中存储时间范围时，Adaptive Server 将其 *startday* 和 *endday* 值转换为整数。对于缺省语言为 *us_english* 的服务器而言，每星期的开始时间是星期一（第一天），结束时间是星期日（第七天）。
- 可以创建与一个或多个其它时间范围重叠的时间范围。

- 范围中的天是连续的，因此星期中的天可以从星期的结束延续到下星期的开始。换言之，正如同星期二和星期三是连续的日期一样，星期日和星期一也是连续的。
- 每个查询批次开始时，活动的时间范围捆绑到会话。在查询批处理期间，由于实际时间的改变而导致的服务器的活动时间范围的改变不会对会话产生影响。换言之，如果资源限制在给定的时间范围内对查询批加以限制，但该查询批是在此时间范围变为活动状态之前开始的，则资源限制对已在运行的查询批不会产生影响。
- 使用系统过程对时间范围的添加、修改和删除操作不会影响当前正在进行的会话的活动时间范围。
- 如果资源限制的范围是某一事务，且在事务运行时，服务器的活动时间范围发生了改变，则新的活动时间范围不会影响当前正在进行的事务。
- 把事务作为限制范围的资源限制的更改不影响当前正在进行的任何事务。
- 有关时间范围的详细信息，参见 *系统管理指南*。

权限

只有系统管理员可以执行 `sp_add_time_range`。

使用的表

master..systimeranges、*master..syslanguages*

参见

系统过程	<code>sp_add_resource_limit</code> 、 <code>sp_drop_time_range</code> 、 <code>sp_modify_time_range</code>
------	--

sp_addtype

功能

创建用户定义的数据类型。

语法

```
sp_addtype typename,  
           phystype [(length) | (precision [, scale])]  
           [, "identity" | nulltype]
```

参数

typename — 是用户定义的数据类型的名称。类型名称必须遵循标识符的规则，且在每个数据库中是唯一的。

phystype — 是物理的或 Adaptive Server 提供的数据类型，以此为基础来创建用户定义的数据类型。您可以指定除 *timestamp* 以外的任何 Adaptive Server 数据类型。

char、*varchar*、*nchar*、*nvarchar*、*binary* 和 *varbinary* 数据类型需要在小括号中指定 *length*。如果未提供，Adaptive Server 将使用 1 个字符的缺省长度。

float 数据类型需要在小括号中指定二进制的 *precision*。如果未提供，Adaptive Server 将使用适合于您的平台的缺省精度。

numeric 和 *decimal* 数据类型需要在小括号中指定十进制 *precision* 和 *scale*，以逗号分隔。如果未提供，Adaptive Server 使用的缺省精度为 18，标度为 0。

将包括标点符号（如小括号或逗号）的物理类型用单引号或双引号引起。

identity — 表示用户定义的数据类型具有 IDENTITY 属性。将 *identity* 关键字用单引号或双引号引起。只可为标度是 0 的 *numeric* 数据类型指定 IDENTITY 属性。

IDENTITY 列存储顺序号，例如 Adaptive Server 生成的发票号或职员号。IDENTITY 列的值唯一地标识表的每一行。

IDENTITY 列不可更新且不允许空值。

nulltype — 表示用户定义的数据类型如何处理空值条目。该参数可接受的值为 **null**、**NULL**、**nonull**、**NONULL**、**"not null"** 和 **"NOT NULL"**。所有包括空格的 **nulltype** 必须用单引号或双引号引起。

如果同时省略 **IDENTITY** 属性和 **nulltype**，Adaptive Server 创建的数据类型将使用为数据库定义的空值模式。缺省情况下，未指定 **nulltype** 的数据类型将创建为 **NOT NULL**（即不允许空值并且需要显式条目）。为符合 **SQL** 标准，应使用 **sp_dboption** 系统过程将 **allow nulls by default** 选项设置为 **true**。这将把数据库的空值模式更改为 **NULL**。

示例

1. sp_addtype ssn, "varchar(11)"

创建名为 **ssn** 的用户定义的数据类型，该类型用于保存社会保险号的列值。由于未指定 **nulltype** 参数，Adaptive Server 将使用数据库缺省的空值模式创建数据类型。注意，因为 **varchar(11)** 包含标点符号（小括号），所以将其用引号引起。

2. sp_addtype birthday, "datetime", null

创建名为 **birthday** 的用户定义的数据类型，该类型允许空值。

3. sp_addtype temp52, "numeric(5,2)"

创建名为 **temp52** 的用户定义的数据类型，该类型用于存储温度，数值最多为 5 位有效数字，其中小数点右边有 2 位。

4. sp_addtype "row_id", "numeric(10,0)", "identity"

创建名为 **row_id** 的用户定义的数据类型，该类型具有 **IDENTITY** 属性，用作唯一的行标识符。使用该数据类型创建的列存储系统生成的长度最多为 10 位的值。

5. sp_addtype systype, sysname

创建用户定义的数据类型，其基础类型为 **sysname**。虽然无法使用 **create table**、**alter table** 或 **create procedure** 语句中的 **sysname** 数据类型，但是可以使用基于 **sysname** 的用户定义的数据类型。

注释

- **sp_addtype** 创建用户定义的数据类型并将其添加到 **systypes** 系统表。一旦创建用户定义的数据类型，就可以将其用于 **create table** 和 **alter table** 语句，并为其绑定缺省设置和规则。
- 按照 Adaptive Server 提供的数据类型之一建立每个用户定义的数据类型，指定适当的长度或精度和标度。不能在 **create table** 或 **alter table** 语句中替换长度、精度或标度。
- 用户定义的数据类型名称在数据库中必须是唯一的，但用户定义的不同名称的数据类型可以具有相同的定义。

- 如果将 *nchar* 或 *nvarchar* 指定为 *phystype*，则使用新类型创建的列的最大长度就是 **sp_addtype** 中指定的长度乘以添加该类型时的 *@@ncharsize* 的值。
- 每个系统类型都具有**层次**，存储在 *systypes* 系统表中。用户定义的数据类型可与它所基于的物理类型拥有相同的数据类型层次。在混合方式的表达式中，所有类型都被转换为公用类型，即具有最低一级层次的类型。

使用以下查询可列出数据库中每个系统提供的类型和用户定义的类型层次：

```
select name, hierarchy
from systypes
order by hierarchy
```

具有 IDENTITY 属性的类型

- 如果将用户定义的数据类型定义为具有 IDENTITY 属性，则从该类型创建的所有列都是 IDENTITY 列。可以在 **create** 或 **alter table** 语句中指定 IDENTITY 或 NOT NULL，也可以都不指定。以下是三种从具有 IDENTITY 属性的用户定义的数据类型创建 IDENTITY 列的不同方式：

```
create table new_table (id_col IdentType)
create table new_table (id_col IdentType identity)
create table new_table (id_col IdentType not null)
```

- 如果使用 **create table** 或 **alter table** 语句创建列，可以替换 **sp_addtype** 系统过程所指定的空类型：
 - 指定为 NOT NULL 的类型可用于创建 NULL 或 IDENTITY 列。
 - 指定为 NULL 的类型可用于创建 NOT NULL 列，但不能创建 IDENTITY 列。

► 注意

如果尝试从 IDENTITY 类型创建空列，则 **create** 或 **alter table** 语句将失效。

权限

任何用户都可以执行 **sp_addtype**。

使用的表

master.dbo.spt_values、*master.dbo.sysdatabases*、*sysobjects*、*systypes*

参见

命令	create default、create rule、create table
数据类型	“用户定义的数据类型”
系统过程	sp_bindefault、sp_bindrule、sp_dboption、 sp_droptype、sp_rename、sp_unbindefault、 sp_unbindrule

sp_addumpdevice

功能

将转储设备添加到 Adaptive Server。

语法

```
sp_addumpdevice {"tape" | "disk"}, logicalname,  
                physicalname [, tapesize]
```

参数

"tape" — 表示磁带驱动器。将 **tape** 用引号引起。

"disk" — 表示磁盘或文件设备。将 **disk** 用引号引起。

logicalname — 是“逻辑”转储设备名称。它必须是有效的标识符。一旦将转储设备添加到 *sysdevices*，就可以在 **load** 和 **dump** 命令中指定其逻辑名。

physicalname — 是设备的物理名。您可以指定绝对路径名，也可指定相对路径名。转储和装载期间，**Backup Server** 通过搜索 **Adaptive Server** 的当前工作目录来解析相对路径名。将包含非字母字符的名称用引号引起。对于 **UNIX** 平台，应指定不可倒带的磁带设备名。

tapesize — 是磁带转储设备的容量（以兆字节计）。如果指定，**OpenVMS** 系统将忽略 *tapesize* 参数。其它平台要求此参数用于磁带设备，但不用于磁盘设备。*tapesize* 应至少为五个数据库页（每页需要 2048 个字节）。Sybase 建议指定的容量应比设备的额定容量稍低。

示例

```
1. sp_addumpdevice "tape", mytapedump, "/dev/nrmt8",  
   40
```

添加 40MB 的磁带设备。转储和装载命令可以按设备的物理名称 */dev/nrmt8*（或逻辑名称 *mytapedump*）来引用该设备。

```
2. sp_addumpdevice "disk", mydiskdump,  
   "/dev/rxyld/dump.dat"
```

添加名为 *mydiskdump* 的磁盘设备。指定绝对路径名或相对路径名以及文件名。

注释

- **sp_addumpdevice** 将把转储设备添加到 *master.dbo.sysdevices* 表。磁带设备被分配 *cntrltype* 3，磁盘设备则被分配 *cntrltype* 2。
- 要将操作系统文件用作转储设备，应为 *physicalname* 指定 **disk** 类型的设备以及绝对或相对路径名。省略 *tapesize* 参数。如果指定相对路径名，则在执行转储或装载命令时，将向当前 **Adaptive Server** 工作目录进行转储或从中装载转储。
- 所有权和权限问题会影响磁盘或文件转储设备的使用。**sp_addumpdevice** 向 *sysdevices* 表添加设备，但并不保证可以创建作为转储设备的文件或用户可以向特定设备进行转储。
- **dump database** 和 **dump transaction** 命令的 *with capacity = megabytes* 子句可以替换 **sp_addumpdevice** 所指定的 *tapesize*。在不能可靠地检测出磁带结尾标记的平台上，**Backup Server** 将在转储完指定的兆字节数后发出卷更改请求。
- 如果转储设备失败，使用 **sp_dropdevice** 可将其从 *sysdevices* 删除。进行设备替换后，使用 **sp_addumpdevice** 可将逻辑设备名与新的物理设备相关联。这样，每当转储设备失败时就不必更新备份脚本和阈值过程。
- 要将数据库设备添加到 *sysdevices*，请使用 **disk init** 命令。

权限

只有系统管理员才能执行 **sp_addumpdevice**。

使用的表

master.dbo.sysdevices、*sysobjects*

参见

命令	disk init、dump database、dump transaction、load database、load transaction
系统过程	sp_dropdevice、sp_helpdevice

sp_adduser

功能

将新的用户添加到当前数据库。

语法

```
sp_adduser loginame [, name_in_db [, grpname]]
```

参数

loginame — 是 *master.dbo.syslogins* 中的用户名。

name_in_db — 是当前数据库中新的用户名。

grpname — 将用户添加到数据库中的现有组。

示例

1. sp_adduser margaret

将 “margaret” 添加到数据库。她的数据库用户名与她的 Adaptive Server 登录名相同，属于缺省组 “public”。

2. sp_adduser haroldq, harold, fort_mudge

将 “haroldq” 添加到数据库。当 “haroldq” 使用当前数据库时，他的名称为 “harold”。他属于 *fort_mudge* 组以及缺省组 “public”。

注释

- 数据库所有者执行 **sp_adduser** 以将用户名添加到当前数据库的 *sysusers* 表，使该用户可以以其自己的名称访问当前数据库。
- 指定 *name_in_db* 参数为用户指定用在数据库中的新名称，该名称不同于该用户在 Adaptive Server 中的登录名。为用户指派不同用户名的功能旨在为您带来方便。它不是 **sp_addalias** 所提供的别名，因为它不映射到其他用户身份和特权。
- 组名和用户名不能相同。
- 除了是缺省组 “public” 以外，用户只可以是另外一个组的成员。每个用户都是缺省组 “public” 的成员。使用 **sp_changegroup** 可更改用户的组。
- 要访问数据库，用户必须列在 *sysusers* 中（用 **sp_adduser** 命令添加），或者必须映射到 *sysalternates* 中的其他用户（使用命令 **sp_addalias**），或者 *sysusers* 中必须有 “guest” 条目。

权限

只有数据库所有者、系统管理员或系统安全员才能执行 **sp_adduser**。

使用的表

master.dbo.syslogins、 *master.dbo.syssrvroles*、 *sysalternates*、
sysobjects、 *sysusers*

参见

命令	grant、 revoke、 use
系统过程	sp_addalias、 sp_addgroup、 sp_changegroup、 sp_dropalias、 sp_dropgroup、 sp_helpuser

sp_altermessage

功能

允许或禁止在 Adaptive Server 错误日志中记录系统定义的消息或用户定义的消息。

语法

```
sp_altermessage message_id, parameter, parameter_value
```

参数

message_id — 要变更的消息的消息号。它是在 *sysmessages* 或 *sysusermessages* 系统表的 *error* 列中记录消息时的编号。

parameter — 是要变更的消息参数。最大长度是 255 字节。唯一有效的参数是 *with_log*。

parameter_value — 是在 *parameter* 中指定的参数的新值。最大长度是 255 字节。值是 *true* 和 *false*。

示例

```
1. sp_altermessage 2000, 'with_log', 'TRUE'
```

指定 *sysmessages* 中的消息号 2000 应该被记录在 Adaptive Server 的错误日志和 Windows NT 事件日志（如果日志记录功能已启用）中。

注释

- 如果 *parameter_value* 为 *true*，则始终记录指定消息。如果是 *false*，将使用缺省的记录行为；根据错误的严重性以及其它因素，可能记录也可能不记录消息。将 *parameter_value* 设置为 *false* 所产生的行为与未调用 *sp_altermessage* 所产生的行为相同。
- 在 Windows NT 服务器上，*sp_altermessage* 还启用和禁用 Windows NT 事件日志中的日志记录功能。

权限

只有数据库所有者或系统管理员才能执行 *sp_altermessage*。

使用的表

sysmessages、*sysusermessages*

参见

系统过程	sp_addmessage、sp_dropmessage
------	------------------------------

sp_audit

功能

允许系统安全人员配置审计选项。

语法

```
sp_audit option, login_name, object_name [,setting]
```

参数

option — 是要设置的审计选项的名称。表 7-5 列出了有效的审计选项。

表 7-5: 审计选项

选项	说明
adhoc	允许用户使用 sp_addauditrecord 在审计追踪中添加他们自己的用户定义审计记录
all	审计由特定用户或具有特定角色的用户执行的所有操作 注意: 审计所有操作时，不会影响用户添加即席审计记录的能力。
alter	审计 alter table 或 alter database 命令的执行
bcp	审计 bcp in 实用程序的执行
bind	审计 sp_bindefault 、 sp_bindmsg 和 sp_bindrule 系统过程的执行
cmdtext	审计由特定用户或具有特定角色的用户执行的所有操作。
create	审计数据库对象的创建
dbaccess	审计从其它数据库对当前数据库的访问
dbcc	审计任何 dbcc 命令的执行
delete	审计表或视图中行的删除
disk	审计 disk init 、 disk refit 、 disk reinit 、 disk mirror 、 disk unmirror 和 disk remirror 的执行
drop	审计数据库对象的删除
dump	审计 dump database 或 dump transaction 命令的执行
errors	审计致命或非致命错误
exec_procedure	审计存储过程的执行

表 7-5: 审计选项 (续)

选项	说明
exec_trigger	审计触发器的执行
func_dbaccess	审计通过 Transact-SQL 函数对数据库的访问
func_obj_access	审计通过 Transact-SQL 函数对数据库对象的访问
grant	审计 grant 命令的执行
insert	审计表或视图中行的插入
load	审计 load database 或 load transaction 命令的执行
login	审计所有登录到 Adaptive Server 的尝试
logout	审计所有从 Adaptive Server 注销的尝试
reference	审计表之间的引用
revoke	审计 revoke 命令的执行
rpc	审计远程过程调用的执行
security	审计以下的安全性相关事件： <ul style="list-style-type: none"> • 启动或关闭服务器 • 激活角色或使角色失效 • 发出下列任何命令： <ul style="list-style-type: none"> connect kill online database set proxy set session authorization sp_configure • 使用下列任何函数： <ul style="list-style-type: none"> valid_user proc_role (从系统过程中) • 重新生成 SSO 口令
select	审计 select 命令的执行
setuser	审计 setuser 命令的执行
table_access	审计特定用户对任何表的访问
truncate	审计 truncate table 命令的执行

表 7-5: 审计选项 (续)

选项	说明
unbind	审计 <code>sp_unbindrule</code> 、 <code>sp_unbindmsg</code> 和 <code>sp_unbinddefault</code> 系统过程的执行
update	审计表或视图中行的更新
view_access	审计特定用户对任何视图的访问

login_name — 是要审计的特定登录的名称。要审计所有登录，可为 *option* 参数指定 **all**。如果指定了 **all**，即可将 *login_name* 参数设置为特定的系统角色，以审计由具有该系统角色的用户执行的所有操作。不能为 *login_name* 指定用户定义的角色。有关对每个 *option* 值都有效的 *login_name* 值的详细信息，参见 *系统管理指南*。

object_name — 是要审计的对象的名称。有效值取决于指定的 *option* 值，其中包括：

- 对象名；如果您不拥有该对象，则需包括所有者的名称。例如，要审计 Joe 拥有的名为 *inventory* 的表，应为 *object_name* 指定 *joe.inventory*。
- 对于所有对象为 **all**。
- **default table**、**default view**、**default procedure** 或 **default trigger**，用于审计对任何新表、视图、过程或触发器的访问。

当为 *option* 参数指定 **delete**、**insert**、**select** 或 **update** 时，**default table** 和 **default view** 是 *object_name* 的有效值。当指定 **exec_procedure** 选项时，**default procedure** 为有效值。当指定 **exec_trigger** 选项时，**default trigger** 为有效值。

有关对每个 *option* 值都有效的 *object_name* 值的详细信息，参见 *系统管理指南*。

setting — 是审计的级别。如果不指定 *setting* 的值， Adaptive Server 将显示该选项的当前审计设置。下表说明了 *setting* 参数的有效值：

<i>setting</i> 值	说明
on	激活对指定选项的审计。无论该事件是否通过权限检查， Adaptive Server 都将为该选项控制的事件生成审计记录。
off	使指定选项的审计失效。
pass	激活对通过权限检查的事件的审计。
fail	使未通过权限检查的事件的审计失效。

如果为选项指定了 **pass** 后，又为同一选项指定 **fail** （或者相反），其结果相当于指定 **on**。无论事件是否通过权限检查， Adaptive Server 都会生成审计记录。**on** 或 **off** 的设置将应用于所有审计选项。**pass** 和 **fail** 的设置将应用于除 **errors** 和 **adhoc** 之外的所有选项。对于这些例外的选项，只能应用 **on** 或 **off**。所有选项的初始（缺省）值都是 **off**。

示例

1. `sp_audit "security", "all", "all", "on"`
开始审计安全性相关事件。对成功和失败的事件都会进行审计。
2. `sp_audit "security", "all", "all"`
显示 `security` 审计选项的设置。
3. `sp_audit "create", "all", master, "on"`
开始审计 `master` 数据库中对象的创建，包括 `create database`。
4. `sp_audit "create", "all", db1, "on"`
开始审计 `db1` 数据库中所有对象的创建。
5. `sp_audit "all", "sa_role", "all", "fail"`
开始审计系统管理员的所有失败操作。
6. `sp_audit "update", "all", "default table", "on"`
开始审计对当前数据库中将来表的所有更新。例如，如果当前数据库是 `utility`，在 `utility` 中创建的所有新表的更新都将得到审计。这并不影响对现有表的审计。

注释

- **sp_audit** 确定启用审计后将要审计的内容。直到使用 **sp_configure** 将 **auditing** 参数设置为 **on** 后，才实际进行审计。此时，所有用 **sp_audit** 配置的审计选项都将生效。有关详细信息，参见 **sp_configure**。
- 如果您不是指定对象的所有者，则要用所有者的名称限定 **object_name** 参数值，格式如下：
"ownername.objname"
- 不能对 **tempdb** 数据库中的以下选项激活缺省审计：
 - **delete**
 - **insert**
 - **select**
 - **update**
 - **exec_procedure**
 - **exec_trigger**
- 表 7-6 列出了控制审计的配置参数。

表 7-6: 控制审计的配置参数

配置参数	作用
auditing	启用或禁用对服务器的审计。
audit_queue_size	确定审计队列的大小。
current_audit_table	设置当前的审计表。 Adaptive Server 将在该表中写入所有审计记录。
suspend_auditing_when_full	在审计设备已满时控制审计进程的行为。

auditing、**current_audit_table** 和 **suspend_auditing_when_full** 配置参数是动态参数，它们将立即生效。由于 **audit_queue_size** 影响内存分配，所以该参数是静态的，它要在重新启动 **Adaptive Server** 后才会生效。

- 有关对 **Adaptive Server** 进行审计配置的详细信息，参见 *系统管理指南* 中的 **sp_configure**。

权限

只有系统安全员才能执行 **sp_audit**。

使用的表

master..sysdatabases、*sysobjects*、*sybsecurity..sysauditoptions*、*sybsecurity..sysaudits_01...sybsecurity..sysaudits_08*

参见

系统过程	sp_addauditrecord、sp_configure
实用程序命令	bcp

sp_autoconnect

(仅适用于组件集成服务)

功能

为特定用户定义与远程服务器的直通连接，使该用户在登录时自动进入直通模式。

语法

```
sp_autoconnect server, {true|false}  
[, loginname]
```

参数

server — 是建立自动直通模式连接的服务器的名称。**server** 必须是已通过 **sp_addserver** 添加的远程服务器的名称。该服务器不能是本地服务器。

true|false — 确定对 **server** 启用还是禁用自动直通模式连接。**true** 启用自动连接。**false** 禁用自动连接。

loginname — 指定需要自动连接的用户名称。如果未提供 **loginname**，自动连接状态就将修改为用于当前用户。

示例

1. sp_autoconnect SYBASE, true

当前用户在下次登录时将自动连接到服务器 SYBASE。该用户的连接被置于直通模式。

2. sp_autoconnect SYBASE, false, steve

对用户 “steve” 禁用自动连接功能。

注释

- **sp_autoconnect** 为特定用户定义到远程服务器的直通连接，使该用户在登录时自动进入直通模式。
- 执行 **sp_autoconnect** 之前，系统管理员必须向该登录授予 **connect to** 权限。
- 只有在安装并配置了组件集成服务后，才能使用 **sp_autoconnect**。
- 不要更改 “sa” 登录帐户的自动连接状态。
- 对于当前已建立连接的用户，对自动连接状态的更改不会立即生效。他们必须先断开与本地服务器的连接，然后重新连接，才能使更改生效。

- 使用 **disconnect** 可退出直通模式。

权限

只有系统管理员才能执行 **sp_autoconnect**。

使用的表

syssservers、*syslogins*

参见

命令	connect to...disconnect、grant
系统过程	sp_addlogin、sp_passthru、sp_remotesql、sp_addserver

sp_bindcache

功能

将数据库、表、索引、 *text* 对象或 *image* 对象绑定到数据缓存。

语法

```
sp_bindcache cachename, dbname  
[, [ownername.]tablename  
[, indexname | "text only"]]
```

参数

cachename — 是活动数据缓存的名称。

dbname — 是要绑定到缓存的数据库名称，或是包含要绑定到缓存的表、索引、 *text* 或 *image* 对象的数据库的名称。

ownername — 是表所有者的名称。如果表由 “dbo” 所有，表所有者名称就是可选的。

tablename — 是要绑定到缓存的表的名称，或是其索引、 *text* 对象或 *image* 对象要绑定到缓存的表的名称。

indexname — 是要绑定到缓存的索引的名称。

text only — 将 *text* 或 *image* 对象绑定到缓存。使用该参数时，不能同时给出索引名。

示例

1. `sp_bindcache pub_cache, pubs2, titles`

将 *titles* 表绑定到名为 *pub_cache* 的缓存。

2. `sp_bindcache pub_ix_cache, pubs2, titles,
title_id_cix`

将集群索引 *titles.title_id_cix* 绑定到 *pub_ix_cache*。

3. `sp_bindcache tempdb_cache, tempdb`

将 *tempdb* 绑定到 *tempdb_cache*。

4. `sp_bindcache logcache, pubs2, syslogs`

将 *pubs2* 事务日志 *syslogs* 绑定到名为 *logcache* 的缓存。

5. `sp_bindcache pub_cache, pubs2, au_pix, "text only"`

将 *au_pix* 表的 *image* 链绑定到名为 *pub_cache* 的缓存。

注释

- 一个数据库或数据库对象只能被绑定到一个缓存上。可将一个数据库绑定到一个缓存，然后将该数据库中的各个表、索引、**text** 对象或 **image** 对象绑定到其它缓存。如果数据库没有其它绑定，该数据库绑定是数据库中所有对象的缺省绑定。表或索引的数据缓存层次如下：
 - 如果对象被绑定到缓存，则使用对象绑定。
 - 如果对象未被绑定到缓存，但是对象的数据库被绑定到缓存，则使用数据库绑定。
 - 如果对象及其数据库都未绑定到缓存，则使用缺省的数据缓存。
- 在执行 **sp_bindcache** 之前，缓存和要绑定到该缓存上的对象或数据库必须已存在。将对象绑定到缓存之前，用 **sp_cacheconfig** 创建缓存并重新启动 **Adaptive Server**。
- 缓存绑定立即生效，不需要重新启动服务器。当把对象绑定到数据缓存时：
 - 内存中当前对象的任何页都被清除。
 - 当在查询中使用对象时，它的页将被读取到绑定的缓存中。
- 可将索引绑定到其引用表之外的缓存。如果将集群索引绑定到缓存，绑定仅影响该索引的根页和中间页，而不影响数据页（它们被定义为索引的叶页）。
- 要绑定数据库，必须使用 **master** 数据库。要绑定表、索引、**text** 对象或 **image** 对象，必须使用存储这些对象的数据库。
- 要绑定数据库中的任何系统表，必须使用数据库且该数据库必须处于单用户模式。使用以下命令：

```
sp_dboption db_name, "single user", true
```

有关详细信息，参见 **sp_dboption**。

- 将对象或数据库绑定到不同的缓存前，不必将它们解除绑定。如果对已绑定的对象发出 **sp_bindcache**，则将删除旧绑定并创建新绑定。
- 将表或其索引绑定到缓存时，**sp_bindcache** 需要获取排它表锁，以便使发生绑定时不能读取任何页。如果用户在表上持有锁，那么对该对象发出 **sp_bindcache** 后，执行绑定的任务将休眠，直到这些锁被释放。
- 将对象绑定或解除绑定时，所有引用该对象的存储过程将在下次执行时被重新编译。当更改数据库的绑定时，所有引用绑定数据库中对象的存储过程将在下次执行时被重新编译。

- 当删除表、索引或数据库时，将会删除所有关联的缓存绑定。如果重新创建了表、索引或数据库，则必须再次使用 **sp_bindcache** 才能将其绑定到缓存。
- 如果数据库或数据库对象被绑定到缓存，那么在删除缓存后，缓存绑定将被标记为无效，但仍存储在 *sysattributes* 系统表中。当重新启动 **Adaptive Server** 时，将在错误日志中输出警告。如果创建了同名的缓存，绑定将在重新启动 **Adaptive Server** 时生效。
- 以下过程提供了相应对象的绑定信息：**sp_helpdb** 提供数据库绑定信息，**sp_help** 提供表绑定信息，**sp_helpindex** 提供索引绑定信息。**sp_helpcache** 提供所有要绑定到特定缓存的对象的信息。
- 使用 **sp_spaceused** 可查看表和索引的当前大小，使用 **sp_estspace** 可估计表的增长大小。使用 **sp_cacheconfig** 可查看有关缓存大小和状态的信息，并配置和重配置缓存。

限制

- 不能将 *master* 数据库、*master* 中的系统表和 *master* 中系统表的索引绑定到缓存。但可以将 *master* 中的非系统表及其索引绑定到缓存。
- 下列情况下，不能将数据库或对象绑定到缓存：
 - 隔离级别 0 读取在表上是活动的
 - 执行绑定的任务当前在表上有一个打开的游标
- 如果缓存的类型是 **log only**，*syslogs* 表就只能绑定到该缓存。使用 **sp_cacheconfig** 可查看缓存的类型。

权限

只有系统管理员才能执行 **sp_bindcache**。

使用的表

master.sysattributes、*master.sysdatabases*、*sysindexes*、*sysobjects*

参见

系统过程	sp_cacheconfig 、 sp_configure 、 sp_help 、 sp_helpcache 、 sp_helpdb 、 sp_helpindex 、 sp_poolconfig 、 sp_unbindcache 、 sp_unbindcache_all
------	--

sp_bindefault

功能

将用户定义的缺省值绑定到列或用户定义的数据类型。

语法

```
sp_bindefault defname, objname [, futureonly]
```

参数

defname — 是缺省值的名称，该缺省值用 **create default** 语句创建并绑定到特定列或用户定义的数据类型。

objname — 是要绑定缺省值的表和列或用户定义数据类型的名称。如果 **objname** 参数的形式不是 “*table.column*”，它将被认为是用户定义的数据类型。如果对象名包含嵌入空白或标点符号，或是保留字，则应该用引号将其引起。

除非指定 **futureonly**，否则用户定义数据类型的现有列将继承缺省 **defname**。

futureonly — 禁止用户定义数据类型的现有列获取新的缺省值。将缺省值绑定到用户定义的数据类型时，该参数是可选的。它从不用于将缺省值绑定到列。

示例

1. sp_bindefault today, "employees.startdate"

假定已用 **create default** 在当前数据库中定义了名为 *today* 的缺省值，该命令将缺省值绑定到 *employees* 表的 *startdate* 列。除非提供了其它值，否则添加到 *employees* 表的每一新行在 *startdate* 列上的值都是 *today* 缺省值。

2. sp_bindefault def_ssn, ssn

假定存在名为 *def_ssn* 的缺省值和名为 *ssn* 的用户定义数据类型，该命令将 *def_ssn* 绑定到 *ssn*。当创建表时，分配有用户定义数据类型 *ssn* 的所有列都将继承该缺省值。除非指定了 **futureonly**（禁止用户定义数据类型的现有列继承缺省值），或者以前已更改列的缺省值（在此情况下，将保留更改后的缺省值），否则类型 *ssn* 的现有列也将继承缺省值 *def_ssn*。

3. sp_bindefault def_ssn, ssn, futureonly

将缺省值 *def_ssn* 绑定到用户定义的数据类型 *ssn*。由于包含了 **futureonly** 参数，类型为 *ssn* 的所有现有列都不受影响。

注释

- 可通过两种方法创建列缺省值：在 **create table** 或 **alter table** 语句中将缺省值声明为列约束，或者使用 **create default** 语句创建缺省值并使用 **sp_bindefault** 将该缺省值绑定到某一列上。使用 **create default** 可将该缺省值绑定到数据库中的多个列。
- 不能将缺省值绑定到 Adaptive Server 提供的数据类型。
- 不能将缺省值绑定到系统表。
- 绑定到具有 **IDENTITY** 属性的列或用户定义数据类型的缺省值对于列值没有任何影响。每次将一行插入到表中时，Adaptive Server 都会为 **IDENTITY** 列分配一个顺序号。
- 如果将缺省值绑定到列，则应以 “*table.column*” 的形式提供 *objname* 参数。任何其它格式都将被认为是用户定义数据类型的名称。
- 如果列上已存在缺省值，那么在绑定新的缺省值之前必须将其删除。使用 **sp_unbindefault** 可删除用 **sp_bindefault** 创建的缺省值。要删除用 **create table** 或 **alter table** 创建的缺省值，可使用 **alter table** 将缺省值替换为 **NULL**。
- 除非指定 **futureonly**，否则用户定义数据类型的现有列将继承新的缺省值。用户定义数据类型的新列始终继承新缺省值。将缺省值绑定到用户定义的数据类型后，将替换已绑定该类型列上的缺省值；要恢复列绑定，先解除列缺省值的绑定，再重新绑定列缺省值。
- 使用缺省值的语句不能与它们的 **sp_bindefault** 语句位于同一个批处理中。

权限

只有对象所有者才能执行 **sp_bindefault**。

使用的表

syscolumns、*sysobjects*、*sysprocedures*、*systypes*

参见

命令	create default、create table、drop default
系统过程	sp_unbindefault

sp_bindexeclass

功能

使执行类与客户端应用、登录或存储过程相关联。

语法

```
sp_bindexeclass "object_name", "object_type", "scope",  
                "classname"
```

参数

object_name — 是要与执行类 **classname** 相关联的客户端应用、登录或存储过程的名称。

object_type — 标识 **object_name** 的类型。**ap** 用于应用，**lg** 用于登录，**pr** 用于存储过程。

scope — 是客户端应用或登录的名称，对于 **ap** 和 **lg** 可以是 NULL。它是对象的存储过程所有者的名称（用户名）。当具有 **object_name** 的对象与应用或登录相互作用时，**classname** 特性将应用于您设置的范围。

classname — 指定要与 **object_name** 相关联的类的类型。这些值为：

- **EC1**、**EC2** 或 **EC3**
- 用户定义执行类的名称
- **ANYENGINE**

示例

1. sp_bindexeclass 'isql', 'ap', NULL, 'EC3'

此语句指定，对于调用 **isql** 的任何登录或应用程序过程，Transact-SQL 应用程序将用 **EC3** 特性来执行（因为 **scope** 的值是 NULL），除非登录或应用程序绑定到更高的执行类。

2. sp_bindexeclass 'sa', 'lg', 'isql', 'EC1'

此语句指定当具有系统管理员角色的登录执行 Transact-SQL 应用程序时，登录过程将用 **EC1** 特性来执行。如果已执行第一个示例中的语句，那么调用 **isql** 的任何其它登录或客户端应用将用 **EC3** 特性来执行。

3. sp_bindexeclass 'my_proc', 'PR', 'kundu', 'EC3'

此语句为用户 **kundu** 拥有的 **my_proc** 存储过程分配了 **EC3** 特性。

注释

- **sp_bindexeclass** 使执行类与客户端应用、登录或存储过程相关联。执行类用 **sp_addexeclass** 来创建。
- 当 *scope* 是 NULL 时，*object_name* 没有范围。*classname* 的执行特性将应用于它的所有交互。例如，如果 *object_name* 是应用程序名，特性将应用于任何调用该应用程序的登录过程。如果 *object_name* 是登录名，特性将应用于由登录过程调用的任何应用程序的特定登录过程。
- 将存储过程绑定到执行类时，必须使用存储过程所有者的名称（用户名）作为 *scope* 参数。当同一数据库中具有存储过程的多个调用时，这就可以缩小该存储过程的标识范围。
- 由于优先顺序和范围规则，被绑定的执行类对于称作 *object_name* 的对象可能有效也可能无效。对象可能会自动将其自身绑定到另一个执行类，这取决于其它绑定规范、优先顺序和范围规则。如果其它所有绑定都不适用，对象将绑定到缺省执行类 **EC2**。
- 当尝试将活动进程绑定到无联机引擎的引擎组时，绑定失败。
- Adaptive Server 在 *sysattributes* 表中创建行，该表在存储绑定数据的行中包含对象 ID 和用户 ID。
- 绑定存储过程之前，该存储过程必须已经存在。
- 存储过程绑定必须在存储过程所在的数据库中进行。因此，当绑定系统过程时，应从 *sybsystemprocs* 数据库内部执行 **sp_bindexeclass**。
- 将类绑定到存储过程时，仅使用执行类的“优先级特性”。
- 将存储过程绑定到执行类时，必须提供存储过程所有者的名称，作为 *scope* 参数。当数据库中存在多个同名（但所有者不同）的存储过程时，这有助于唯一地标识存储过程。

权限

只有系统管理员才能执行 **sp_bindexeclass**。

使用的表

sysattributes、*syslogins*

参见

系统过程	sp_addengine、sp_addexeclass、 sp_clearpsexec、sp_dropengine、 sp_dropexeclass、sp_setpsexec、 sp_showcontrolinfo、sp_showexeclass、 sp_showpsexec、sp_unbindexeclass
实用程序	isql

sp_bindmsg

功能

将用户消息绑定到参照完整性约束或检查约束。

语法

```
sp_bindmsg constrname, msgid
```

参数

constrname — 是要将消息绑定到的完整性约束的名称。使用 **create table** 命令的 **constraint** 子句或 **alter table** 命令的 **add constraint** 子句可创建和命名约束。

msgid — 是要绑定到完整性约束的用户消息的编号。调用 **sp_bindmsg** 之前，该消息必须存在于本地数据库的 *sysusermessages* 表中。

示例

```
1. sp_bindmsg positive_balance, 20100
```

将编号为 20100 的用户消息绑定到 *positive_balance* 约束。

注释

- **sp_bindmsg** 向 *sysconstraints* 表的约束行添加消息号，从而将用户消息绑定到完整性约束。
- 只能将一条消息绑定到一项约束。要更改绑定到约束的消息，绑定一条新的消息即可。新消息的编号将替换 *sysconstraints* 表中旧消息的编号。
- 不能将消息绑定到唯一约束，因为唯一约束在 *sysconstraints* 中没有约束行（唯一约束是唯一索引）。
- 使用 **sp_addmessage** 过程可将用户消息插入到 *sysusermessages* 表中。
- **sp_getmessage** 过程从 *sysusermessages* 表中检索消息文本。
- **sp_help tablename** 显示在 *tablename* 上声明的所有约束名称。

权限

只有对象所有者才能执行 **sp_bindmsg**。

使用的表

sysconstraints、*sysobjects*、*sysusermessages*

参见

命令	alter table、create table
系统过程	sp_addmessage、sp_getmessage、 sp_unbindmsg

sp_bindrule

功能

将规则绑定到列或用户定义的数据类型。

语法

```
sp_bindrule rulename, objname [, futureonly]
```

参数

rulename — 是规则的名称。用 **create rule** 语句创建规则，并用 **sp_bindrule** 将规则绑定到特定列或用户定义的数据类型。

objname — 是要将规则绑定到的表和列或用户定义数据类型的名称。如果 **objname** 的形式不是 “**table.column**”，它将被认为是用户定义的数据类型。如果对象名包含嵌入空白或标点符号，或是保留字，则应用引号将其引起。

futureonly — 禁止用户定义数据类型的现有列继承新的规则。将规则绑定到用户定义的数据类型时，该参数是可选的。如果将规则绑定到列，该参数无意义。

示例

1. **sp_bindrule today, "employees.startdate"**

假定已用 **create default** 在当前数据库中创建了名为 **today** 的规则，该命令将规则绑定到 **employees** 表的 **startdate** 列。当向 **employees** 添加行时，将对照规则 **today** 检查 **startdate** 列的数据。

2. **sp_bindrule rule_ssn, ssn**

假定存在名为 **rule_ssn** 的规则和名为 **ssn** 的用户定义数据类型，该命令将 **rule_ssn** 绑定到 **ssn**。在 **create table** 语句中，类型为 **ssn** 的列继承规则 **rule_ssn**。除非以前已更改过 **ssn** 的规则（在这种情况下，更改后的规则仅在以后维护），否则类型为 **ssn** 的现有列也会继承规则 **rule_ssn**。

3. **sp_bindrule rule_ssn, ssn, futureonly**

将规则 **rule_ssn** 绑定到用户定义的数据类型 **ssn**，但类型为 **ssn** 的现有列不受影响。**futureonly** 禁止类型为 **ssn** 的现有列继承该规则。

注释

- 使用 **create rule** 语句创建规则。然后执行 **sp_bindrule** 将该规则绑定到当前数据库中的列或用户定义的数据类型。
- 当试图执行 **insert** 时（而不是执行 **sp_bindrule** 时），将强制使用规则。可将字符规则绑定到数据类型为精确数值或近似数值的列，尽管这样的 **insert** 并不合法。
- 不能使用 **sp_bindrule** 为 **create table** 语句中的列绑定检查约束。
- 不能将规则绑定到 Adaptive Server 提供的数据类型，或绑定到 *text* 或 *image* 列。
- 不能将规则绑定到系统表。
- 如果将规则绑定到列，*objname* 语句的形式必须是 “*table.column*”。任何其它格式都将被认为是用户定义数据类型的名称。
- 使用规则的语句不能与它们的 **sp_bindrule** 语句位于同一个批处理中。
- 可以将规则绑定到列或用户定义的数据类型，而无需解除绑定现有的规则。绑定到列的规则总是比绑定到用户定义数据类型的规则具有更高的优先级。将规则绑定到列后，将替换该列的用户定义数据类型已绑定的规则。但是，如果将规则绑定到数据类型，则不会替换该用户定义数据类型的列已绑定的规则。将规则绑定到已存在规则的列和用户定义数据类型时，优先顺序如表 7-7 所示：

表 7-7：新旧绑定规则的优先顺序

新规则的绑定位置：	旧规则的绑定位置：	
	用户定义的数据类型	列
用户定义的数据类型	替换旧规则	没有变化
列	替换旧规则	替换旧规则

- 除非以前更改过列的规则，或者可选的第三个参数值是 **futureonly**，否则用户定义数据类型的现有列将继承新规则。用户定义数据类型的新列始终继承新规则。

权限

只有对象所有者才能执行 **sp_bindrule**。

使用的表

syscolumns、*sysconstraints*、*sysobjects*、*sysprocedures*、*systypes*

参见

命令	create rule、drop rule
系统过程	sp_unbindrule

sp_cacheconfig

功能

创建、配置、重新配置和删除数据缓存，并提供有关数据缓存的信息。

语法

```
sp_cacheconfig [cachename [ , "cache_size[P|K|M|G]" ]
               [,logonly | mixed ] [,strict | relaxed ] ]
               [, "cache_partition=[1|2|4|8|16|32|64]" ]
```

参数

cachename — 是将要创建或配置的数据缓存的名称。缓存名称必须是唯一的，其长度最多可以是 30 个字符。缓存名称不必是有效的 Adaptive Server 标识符，即，它可以包含空格以及其它特殊字符。

cache_size — 是将要创建的数据缓存的大小（或者，如果缓存已存在，则是新的数据缓存大小）。一个缓存的最小大小为 512K。在指定大小单位时，可以用 **P** 表示页，用 **K** 表示千字节，用 **M** 表示兆字节，或用 **G** 表示千兆字节。缺省单位是 **K**。对于兆字节和千兆字节，可以指定浮点值。

logonly | mixed — 指定缓存的类型。

strict | relaxed — 指定缓存替换策略。

cache_partition — 指定要在缓存中创建的分区数。缓存中的每个池必须至少为 512K。

示例

1. sp_cacheconfig pub_cache, "10M"

创建具有 10MB 空间的数据缓存 *pub_cache*。所有空间都在缺省的 2K 内存池中。

2. sp_cacheconfig pub_cache

报告 *pub_cache* 的当前配置以及缓存中的所有内存池。

3. sp_cacheconfig pub_cache, "0"

在下次启动 Adaptive Server 时删除 *pub_cache*。

4. sp_cacheconfig pub_log_cache, "2000K", logonly

仅通过一个步骤即创建 *pub_log_cache* 并将其类型设置为 **logonly**。


```
5. sp_cacheconfig pub_log_cache, "2000K"  
   sp_cacheconfig pub_log_cache, logonly
```

第一个命令创建缓存 *pub_log_cache*，其缺省类型为 **mixed**。第二个命令将其状态更改为 **logonly**。最终配置与示例 4 中的配置相同。

```
6. sp_cacheconfig 'newcache', '50M', mixed, strict,  
   "cache_partition=2"
```

创建一个缓存，并设置大小、类型、替换策略和缓存分区数。

注释

- 创建数据缓存后，Adaptive Server 的单个缺省数据缓存将被划分为多个小的缓存。然后，可以在数据缓存内使用 **sp_poolconfig** 配置内存池，使 Adaptive Server 能够执行大型的 I/O 操作。您可以使用 **sp_bindcache** 将表、索引、数据库以及 *text* 或 *image* 链绑定到特定的缓存。
- 当首次创建某个数据缓存时：
 - 所有空间都将分配给 2K 的内存池中。
 - 缺省类型为 **mixed**。
- 图 7-1 显示了一个数据缓存，它配置有两个用户定义的数据缓存以及以下内存池：
 - 缺省数据缓存，包括一个 2K 内存池和一个 16K 内存池
 - 用户缓存，包括一个 2K 内存池和一个 16K 内存池
 - 日志缓存，包括一个 2K 内存池和一个 4K 内存池

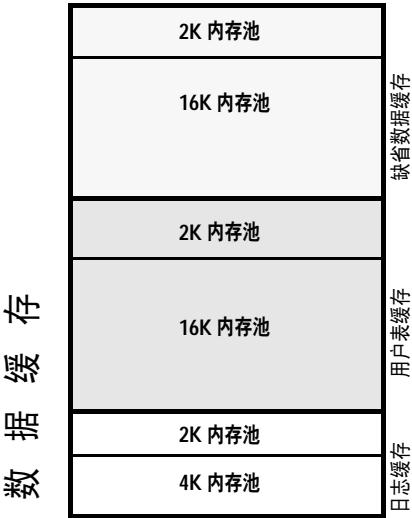


图 7-1：带有缺省缓存和用户定义缓存的数据缓存

- 创建、删除和更改替换策略或分区数之后，必须重新启动 **Adaptive Server** 才能使配置生效。在缓存启用（即服务器重新启动）之前，不能配置内存池或将对象绑定到缓存。
对数据缓存作出的其它更改无需重新启动即可生效，这些更改包括更改类型，用 **sp_poolconfig** 创建、删除内存池或调整内存池大小，更改内存池的清洗百分比，以及将对象绑定和解除绑定。
- 缺省的数据缓存必须始终具有类型 **default**，而任何其它缓存都不能具有类型 **default**。
- 在类型为 **logonly** 或具有宽松 LRU 替换策略的缓存中，**Adaptive Server** 管家任务不执行任何缓冲清洗。

- 以下命令只执行 2K 的 I/O 操作：**disk init**、某些 **dbcc** 命令和 **drop table**。**dbcc checkdb** 和 **dbcc checktable** 命令可以对表执行大型的 I/O 操作，但对索引只执行 2K 的 I/O 操作。表 7-8 显示了在绑定/不绑定数据库或对象时的缓存使用情况。

表 7-8: Transact-SQL 命令的缓存使用情况

命令	数据库已绑定	表或索引已绑定	数据库或对象未绑定
create index	绑定的缓存	N/A	缺省数据缓存
disk init	N/A	N/A	缺省数据缓存
dbcc checkdb	绑定的缓存	N/A	缺省数据缓存
dbcc checktable、 indexalloc、 tablealloc	绑定的缓存	绑定的缓存	缺省数据缓存
drop table	绑定的缓存	绑定的缓存	缺省数据缓存

- 恢复只使用缺省数据缓存的 2K 内存池。对于必须回退或前进的所有事务，它们的所有页都会读入该内存池中并在池中进行更改。请确保缺省的 2K 内存池对于这些事务来说足够大。
- 当使用不带任何参数的 **sp_cacheconfig** 时，它将报告有关服务器上所有缓存的信息。如果仅指定了一个缓存名称，它就只报告指定缓存的信息。如果使用了缓存名称的一部分，它将报告所有匹配“*fragment%*”的名称的信息。

所有报告都包括一个报告缓存信息的信息区，并且为每个缓存提供了一个单独的数据区，以报告有关该缓存中内存池的信息。

以下输出显示了这样的配置：

- 缺省的数据缓存，包括两个内存池：一个 2K 内存池和一个 16K 内存池。缺省数据缓存有 2 个分区。
- *pubs_cache*，包括两个内存池：2K 和 16K
- *pubs_log*，其类型设置为 *logonly*，缓存替换策略设置为 *relaxed*，它包括一个 2K 内存池和一个 4K 内存池

Cache Name	Status	Type	Config Value	Run Value
default data cache	Active	Default	0.00 Mb	26.09 Mb
pubs_cache	Active	Mixed	10.00 Mb	10.00 Mb
pubs_log	Active	Log Only	2.40 Mb	2.40 Mb
Total			12.40 Mb	38.49 Mb
=====				
Cache: default data cache, Status: Active, Type: Default				
Config Size: 0.00 Mb, Run Size: 26.09 Mb				
Config Replacement: strict LRU, Run Replacement: strict LRU				
Config Partition: 2, Run Partition: 2				
IO Size	Wash Size	Config Size	Run Size	APF Percent

2 Kb	3704 Kb	0.00 Mb	18.09 Mb	10
16 Kb	1632 Kb	8.00 Mb	8.00 Mb	10
=====				
Cache: pubs_cache, Status: Active, Type: Mixed				
Config Size: 10.00 Mb, Run Size: 10.00 Mb				
Config Replacement: strict LRU, Run Replacement: strict LRU				
Config Partition: 1, Run Partition: 1				
IO Size	Wash Size	Config Size	Run Size	APF Percent

2 Kb	1228 Kb	0.00 Mb	6.00 Mb	10
16 Kb	816 Kb	4.00 Mb	4.00 Mb	10
=====				
Cache: pubs_log, Status: Active, Type: Log Only				
Config Size: 2.40 Mb, Run Size: 2.40 Mb				
Config Replacement: relaxed LRU, Run Replacement: relaxed LRU				
Config Partition: 1, Run Partition: 1				
IO Size	Wash Size	Config Size	Run Size	APF Percent

2 Kb	206 Kb	0.00 Mb	1.01 Mb	10
16 Kb	272 Kb	1.40 Mb	1.39 Mb	10

表 7-9 列出了输出中各列的含义：

表 7-9: sp_cacheconfig 输出

列	含义
Cache Name	缓存的名称。
Status	下面中的一个： <ul style="list-style-type: none">“Active”“Pend/Act”“Pend/Del” 这些状态将在表后解释。

表 7-9: sp_cacheconfig 输出 (续)

列	含义
Type	用户定义的缓存为 “Mixed” 或 “Log Only”，缺省数据缓存为 “Default”。
I/O Size	内存池的 I/O 大小。在显示缓存配置的行上，该列为空白。
Wash Size	内存池的清洗区大小。当数据页进入缓存的清洗区时，它们将被写入磁盘。在显示缓存配置的行上，该列为空白。
Config Value 或 Config Size	下次重新启动 Adaptive Server 后，缓存或内存池将具有的大小。这些值在下次重新启动 Adaptive Server 后生效。如果值为 0，则表示尚未显式配置大小，将使用缺省值。
Run Value 或 Run Size	当前在 Adaptive Server 上使用的缓存或内存池的大小。
Config/Run Replacement	将在下次重新启动后用于缓存的缓存策略（严格或宽松），以及当前的替换策略。只有当策略在上次重新启动后已被更改时，它们才会有所不同。
Config/Run Partition	将在下次重新启动后用于缓存的缓存分区数，以及当前的分区数。在上次重新启动后，如果已经使用 sp_cacheconfig 更改了分区数，它们就会有所不同。
APF Percent	在可容纳缓冲区的内存池中，已被异步预取指令读取但尚未使用的缓冲区的百分比。
Total	数据缓存的大小总和（如果报告中包括所有缓存），或特定缓存的当前大小（如果已指定缓存名称）。

状态 “Pend” 是 pending（待执行）的缩写。它通常与 “Act”（即 Active，活动的）或 “Del”（即 Delete，删除）一起使用。“Pend” 表示配置动作已发生，但必须重新启动服务器才能使更改生效。

当首次创建一个新缓存但尚未重新启动 Adaptive Server 时，状态为 “Pend/Act”，表示缓存刚经过配置，将在重新启动后处于活动状态。如果将缓存的大小设置为 0 以将其删除，状态将从

“Active” 更改为 “Pend/Del”，表示缓存仍然存在并继续工作，但它将在下次重新启动时被删除。

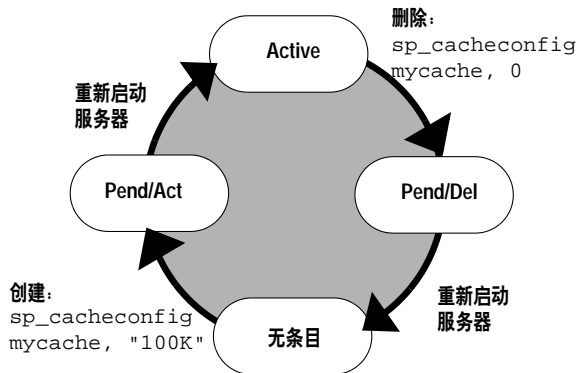


图 7-2: 重新启动和 sp_cacheconfig 对缓存状态的影响

- 您也可以通过编辑配置文件来配置缓存和内存池。有关详细信息，参见 *系统管理指南*。

数据缓存存储空间

- 当首次安装 **Adaptive Server** 时，所有数据缓存存储空间都分配到名为 **缺省数据缓存** 的缓存的 **2K** 内存池中。缺省数据缓存用于所有未用 **sp_bindcache** 显式绑定到数据缓存或其数据库未绑定到缓存的对象。
- 当创建数据缓存时，所分配的存储空间将来自于缺省数据缓存。缓存的存储空间用 **total memory** 配置参数，从分配给 **Adaptive Server** 的存储空间中进行分配。要增加可用于缓存的空间量，可增加 **total memory**，或减小其它使用存储空间配置的设置。如需减少 **total memory** 的大小，缺省数据缓存中必须有可用的空间。

不能将缺省数据缓存的大小减少到低于 **512K**。在大多数情况下，缺省缓存应远远大于最小值。该缓存可用于所有对象（包括未绑定到其它缓存的系统表），并且是唯一在恢复过程中使用的缓存。有关详细信息，参见 *系统管理指南* 中的第 15 章“配置数据高速缓存”。

- 数据缓存要求将小部分的开销用于缓存的管理结构。所有缓存开销都取自于缺省数据缓存。要查看特定缓存大小所需的开销量，可使用 **sp_helpcache** 并提供以下大小：

```
sp_helpcache "200M"
```

```
10.38Mb of overhead memory will be needed to manage
a cache of size 200M
```

更改现有缓存

- 要更改现有缓存的大小，应指定缓存的名称和新的尺寸。
 - 如果增加现有缓存的大小，所有新增空间都将放置在 **2K** 内存池中。
 - 如果减少现有缓存的大小，必须在 **2K** 内存池中提供所有的空间。您可能需要使用 **sp_poolconfig** 将空间从其它内存池移动到 **2K** 池中。
- 如果有数据库或任何非日志对象绑定到缓存，则不能将其类型更改为 **logonly**。

使用缓存分区

- 利用缓存分区，可减少对缓存旋转锁的争用，而不必创建单独的缓存并将数据库对象绑定到这些缓存。有关监控缓存旋转锁争用的详细信息，参见 *Performance and Tuning Guide* 中的第 32 章 “Memory Use and Performance”。
- 使用配置参数 **global cache partition number**，可以设置所有缓存的缺省缓存分区数。参见 *系统管理指南* 中的第 15 章 “配置数据高速缓存”。

删除缓存

- 要删除数据缓存，可将其大小更改为 **0**，如示例 3 所示。将缓存的大小设置为 **0** 时，会给该缓存标上删除标记，但它要到下次重新启动服务器时才被删除。该缓存仍保持活动状态，所有绑定到该缓存的对象将继续使用该缓存。

不能删除缺省数据缓存。

- 如果删除已绑定对象的缓存，当下次重新启动 **Adaptive Server** 时，该缓存的所有对象绑定将标记为无效。有一条消息会在重新启动时输出到错误日志中，它提供了数据库 ID、对象 ID 和索引 ID：

```
00:95/11/05 18:20:39.42 server  Cache binding for
database '6', object '8', index '0' is being marked
invalid in Sysattributes.
```

如果您在随后创建了同名的缓存，绑定就将在激活该缓存时被标记为有效。

权限

只有系统管理员才能执行 **sp_cacheconfig** 来更改缓存配置。任何用户都可以执行 **sp_cacheconfig** 来查看缓存配置。

使用的表

master..sysconfigures、 *masters..syscurconfigs*

参见

系统过程	sp_bindcache、 sp_configure、 sp_help、 sp_helpcache、 sp_helpdb、 sp_helpindex、 sp_poolconfig、 sp_unbindcache、 sp_unbindcache_all
------	--

sp_cachestrategy

功能

对表、索引、 *text* 对象或 *image* 对象启用或禁用预取（大型 I/O）和 MRU 缓存替换策略。

语法

```
sp_cachestrategy dbname, [ownername.]tablename
    [, indexname | "text only" | "table only"
    [, { prefetch | mru }, { "on" | "off"}]]
```

参数

- dbname* — 是存储对象的数据库的名称。
- ownername* — 是表所有者的名称。如果表由 “dbo” 所有，表所有者名称就是可选的。
- tablename* — 是表的名称。
- indexname* — 是该表上索引的名称。
- text only* — 更改 *text* 或 *image* 对象的缓存策略。
- table only* — 更改表的缓存策略。
- prefetch* | *mru* — 即 *prefetch* 或 *mru* 命令，指定要更改的设置。
- on* | *off* — 指定 "on" 或 "off" 设置（用引号引起）。

示例

- sp_cachestrategy pubs2, titles**

object name	index name	large IO	MRU
dbo.titles	titleidind	ON	ON

显示有关 *titles* 表缓存策略的信息。
- sp_cachestrategy pubs2, titles, titleind**
显示有关 *titleind* 索引缓存策略的信息。
- sp_cachestrategy pubs2, titles, titleind, prefetch, "off"**
对 *titles* 表的 *titleind* 索引禁用预取。

```
4. sp_cachestrategy pubs2, authors, "table only", mru,  
   "on"
```

对 *authors* 表重新启用 MRU 替换策略。

```
5. sp_cachestrategy pubs2, blurbs, "text only",  
   prefetch, "on"
```

对 *blurbs* 表的文本页重新启用预取。

注释

- 如果为表或索引所用的缓存配置了大型 I/O 内存池，优化程序就可以选择通过一次执行多至 8 个数据页的大型 I/O 操作来预取数据或索引页。此 **prefetch** 策略可用于表的数据页或非集群索引的叶级页。缺省情况下，将对所有表、索引和 *text* 或 *image* 对象启用预取。将 **prefetch** 选项设置为 "off" 后，会对指定对象禁用预取。
- 为了对任何大小的 I/O 进行表扫描和索引扫描，优化程序可以选择使用 **MRU 替换策略** 来读取和放弃缓存中的缓冲区。缺省情况下，此策略对所有对象启用。将 **mru** 设置为 "off" 后，会禁用此策略。如果对某个对象禁用 **mru**，所有页都将读入到缓存中的 **MRU/LRU** 链。在被其它 I/O 刷新之前，它们会一直保留在缓存中。有关缓存策略的详细信息，参见 *Performance and Tuning Guide*。
- 仅可为当前数据库中的对象更改缓存策略。
- 如果在不指定缓存策略和设置的情况下使用 **sp_cachestrategy**，它将报告对象的当前设置，如示例 1 所示。
- 要查看服务器上所有数据缓存的大小、状态和 I/O 大小，可以使用 **sp_cacheconfig**。
- 如果已将表或索引读入只允许 2K I/O 的缓存，那么设置 **prefetch "on"** 对这些表或索引不会有任何影响。无论可用 I/O 大小是多少，**mru** 策略都可以在所有缓存中使用。

替换

- 如果对表或索引启用了预取，则可以用 **set prefetch "off"** 替换对会话的预取。如果对某个对象禁用了预取，则不能替换该设置。
- **select**、**delete** 和 **update** 命令的 **prefetch**、**lru** 和 **mru** 选项将建议各个语句的 I/O 大小和缓存策略。如果对表或索引启用了预取或 **MRU** 策略，就可以为 **prefetch** 指定 2K I/O 并指定 **lru** 策略，以替换查询的预取或 **MRU** 策略。例如，以下命令对 *titles* 表强制 **LRU** 策略、2K I/O 和表扫描：

```
select avg(advance)
from titles (index titles prefetch 2 lru)
```

如果您请求了预取大小，但没有为具有请求大小的 I/O 配置对象缓存，那么优化程序就会选择最佳的可用 I/O 大小。

- 如果用 **sp_cachestrategy** 为对象启用了预取，那么在 **select**、**update** 或 **delete** 命令中指定 2K 的 **prefetch** 时，将替换先前的 **set prefetch "on"** 语句。当在 **select**、**update** 或 **delete** 命令中指定较大的 I/O 大小时，将不会替换 **set prefetch "off"** 命令。

权限

只有系统管理员或对象所有者才能执行 **sp_cachestrategy**。

使用的表

master..sysattributes、*master..sysdatabases*、*sysattributes*、*sysindexes*、*sysobjects*

参见

命令	delete、select、set、update
存储过程	sp_cacheconfig、sp_poolconfig

sp_changedbowner

功能

更改用户数据库的所有者。

语法

```
sp_changedbowner loginame [, true ]
```

参数

loginame — 是当前数据库的新所有者的登录名。

true — 将别名及其权限移交给新的数据库所有者。其值为 “true” 和 “TRUE”。

示例

```
1. sp_changedbowner albert
```

使用用户 “albert” 成为当前数据库的所有者。

注释

- 新所有者不得是已知的用户或别名（即，新所有者不得已经列在 *sysusers* 或 *sysalternates* 中）。用单个参数 *loginame* 执行 **sp_changedbowner** 时，可将数据库所有权移交给 *loginame*，同时删除充当旧 “dbo” 的用户的别名。
- 在执行 **sp_changedbowner** 之后，新的所有者就成为数据库中的数据库所有者。
- **sp_changedbowner** 不能移交系统数据库的所有权。
- 新所有者必须已经在 Adaptive Server 中有一个登录名，但不得有数据库中的数据库用户名或别名。要将数据库所有权分配给这样的用户，应在执行 **sp_changedbowner** 之前先删除用户名或别名条目。
- 要向新所有者授予权限，系统管理员必须将权限授予数据库所有者，因为该用户在数据库中不再有任何其它名称。

权限

只有系统管理员可以执行 **sp_changedbowner**。

使用的表

master..syslogins、*sysalternates*、*sysobjects*、*sysusers*

参见

命令	create database
系统过程	sp_addlogin、sp_dropalias、sp_dropuser、 sp_helpdb

sp_changegroup

功能

更改用户的组。

语法

```
sp_changegroup grpname, username
```

参数

grpname — 是组名。该组必须已经存在于当前数据库中。如果将“public”用作 *grpname*，则应该将它用引号引起来，因为“public”是关键字。

username — 是要添加到组中的用户的名称。该用户必须已经存在于当前数据库中。

示例

1. **sp_changegroup fort_mudge, albert**

用户“albert”现在是“fort_mudge”组的成员。“albert”以前属于哪个组并不重要。

2. **sp_changegroup "public", albert**

将“albert”从他所属的组中删除，而不使其成为新组的成员（所有用户始终是“public”的成员）。

注释

- 通过执行 **sp_changegroup**，可以将指定用户添加到指定组。该用户从其先前所属的组中删除，添加到由 *grpname* 指定的组中。
- 在用 **sp_adduser** 为新数据库用户提供数据库访问权的同时，可以将这些用户添加到组中。
- 在授予和撤消权限时，组将用作集合名称。所有用户始终是缺省组“public”的成员，他们还可以是另外一个组（且仅一个组）的成员。
- 要将某个用户从组中删除而不使其成为新组的成员，可使用 **sp_changegroup** 将用户的组更改为“public”，如示例 2 所示。
- 当某个用户从一个组移动到另一个组时，该用户将失去在旧组拥有的所有权限，而获得向新组授予的权限。

权限

只有数据库所有者、系统管理员或系统安全员才能执行 **sp_changegroup**。

使用的表

master..sysrvroles、 *syscolumns*、 *sysobjects*、 *sysprotects*、 *sysusers*

参见

命令	grant、 revoke
系统过程	sp_addgroup、 sp_adduser、 sp_dropgroup、 sp_helpgroup

sp_checknames

功能

在当前数据库中检查是否有名称包含 7 位 ASCII 字符集以外字符。

语法

sp_checknames

参数

无。

示例

1. sp_checknames

```
Looking for non 7-bit ASCII characters in the system tables
of database:
"master"
```

```
=====
Table.Column name: "syslogins.password"
```

```
The following logins have passwords that contain non 7-bit
ASCII characters. If you wish to change them use "sp_password";
Remember, only the sa and the login itself may examine or change
the syslogins.password column:
```

```
suid    name
-----
1 sa
2 probe
3 bogususer
```

注释

- **sp_checknames** 检查当前数据库中所有对象、列、索引、用户名、组名和其它元素的名称，验证它们是否包含 7 位 ASCII 字符集以外的字符。它将报告非法的名称，并提供使这些名称与 7 位 ASCII 字符集兼容的指导。
- 当从 SQL Server 版本 4.0.x 或 4.2.x 升级后以及使用非 7 位 ASCII 的缺省字符集后，应该在服务器的每个数据库中运行 **sp_checknames**。
- 按照 **sp_checknames** 报告中的指导来更正所有非 ASCII 名称。

权限

任何用户都可以执行 **sp_checknames**。

使用的表

在任何数据库中执行 **sp_checknames** 时，该命令将使用以下表：

dbo.syscolumns、 *dbo.sysindexes*、 *dbo.sysobjects*、 *dbo.syssegments*、
dbo.systypes、 *dbo.sysusers*

在 *master* 数据库中执行 **sp_checknames** 时，该命令将使用以下表：

master..sysdatabases、 *master..sysdevices*、 *master..syslogins*、
master..sysremotelogins、 *master..sys.servers*

参见

命令	update
系统过程	sp_password、 sp_rename、 sp_renamedb

sp_checkreswords

功能

检测并显示属于 **Transact-SQL** 保留字的标识符。检查服务器名、设备名、数据库名、段名、用户定义的数据类型、对象名、列名、用户名、登录名和远程登录名。

语法

```
sp_checkreswords [user_name_param]
```

参数

user_name_param — 是当前数据库中用户的名称。如果提供 *user_name_param*, **sp_checkreswords** 将只检查指定用户拥有的对象。

示例

1. sp_checkreswords (在 master 数据库中执行)

Reserved Words Used as Database Object Names for Database master

Upgrade renames sysobjects.schema to sysobjects.schemacnt.

Owner	

dbo	
Table	Reserved Word Column Names
-----	-----
authorization	cascade
Object Type	Reserved Word Object Names
-----	-----
rule	constraint
stored procedure	check
user table	arith_overflow
user table	authorization
-----	-----
-----	-----

Owner	

lemur	
Table	Reserved Word Column Names

```

-----
key                                close
Table                             Reserved Word Index Names
-----
key                                isolation
Object Type                       Reserved Word Object Names
-----
default                           isolation
rule                              level
stored procedure                  mirror
user table                        key

Reserved Word Datatype Names
-----
identity

-----
-----

Database-wide Objects
-----

Reserved Word User Names
-----
at
identity

Reserved Word Login Names
-----
at
identity

Reserved Word as Database Names
-----
work

Reserved Word as Language Names
-----
national

```

Reserved Word as Server Names

mirror
primary

Reserved Word ServerNetNames

mirror
primary

2. **sp_checkreswords** (在用户数据库中执行)

Reserved Words Used as Database Object Names for Database user_db

Upgrade renames sysobjects schema to sysobjects.schemactnt.

Owner

tamarin

Table

Reserved Word Column Names

cursor	current
endtran	current
key	identity
key	varying
schema	primary
schema	references
schema	role
schema	some
schema	user
schema	work

Table

Reserved Word Index Names

key	double
-----	--------

Object Type

Reserved Word Object Names

default	escape
rule	fetch
stored procedure	foreign
user table	cursor
user table	key
user table	schema

view

endtran

Database-wide Objects

Found no reserved words used as names for database-wide objects.

注释

- **sp_checkreswords** 报告属于保留字的现有对象名称。如果使用的不是分隔标识符，**Transact-SQL** 则不允许将任何命令语法中的字词用作标识符。保留字是 **SQL** 语法的构成元素。当用作命令一部分时，它们具有特殊的含义。例如，在 **10.0** 之前版本的 **SQL Server** 中，可以创建一个名为 **work** 的表，然后用以下查询选择该表的数据：

```
select * from work
```

在 **SQL Server** 版本 **10.0** 中，**work** 是 **commit work** 命令的一部分，它是新增的保留字。如果在版本 **10.0** 或更高版本中发出相同的 **select** 语句，则将导致语法错误。**sp_checkreswords** 能够查找会导致这种问题的标识符。

- 对于用 **set quoted_identifier** 选项创建的标识符，**sp_checkreswords** 也会查找用作标识符的保留字。
- 请在升级到新版本的 **Adaptive Server** 之前（或在升级之后立即）使用 **sp_checkreswords**。有关在执行升级前安装和运行此过程的信息，参见针对于所用平台的安装文档。

在 **master** 数据库和每个用户数据库中运行 **sp_checkreswords**。此外，如果已经将用户或对象添加到 **model** 和 **sybsystemprocs** 中，还应在这些数据库中运行此过程。

- 返回状态将指出已找到的项目数。
- 如果提供了用户名，**sp_checkreswords** 将检查所有可由用户拥有的对象，如表、索引、视图、过程、触发器、规则、缺省值和用户定义的数据类型。它会报告所有属于保留字的标识符。
- 如果当前数据库不是 **master** 数据库且没有提供用户名，**sp_checkreswords** 将检查上述的所有对象，并在报告中分别为每个用户名提供一个信息项。它还在 **sysusers** 和 **syssegments** 中检查是否有用户名和段名属于保留字。如果已经添加对象、用户或用户定义的数据类型，则只需检查 **model** 和 **sybsystemprocs**。

- 如果当前数据库是 *master* 数据库且没有提供用户名，**sp_checkreswords** 将执行上述所有检查，同时还检查 *sysdatabases*、*syslogins*、*syscharsets*、*sysservers*、*sysremotelogins*、*sysdevices* 和 *syslanguages*，以验证是否有保留字用作数据库、本地或远程登录、本地或远程服务器、字符集和语言的名称。

处理所报告的保留字实例

- 如果 **sp_checkreswords** 报告有保留字用作标识符，则可以执行以下两项操作之一：
 - 使用 **sp_rename**、**sp_renamedb** 或更新系统表，以更改标识符的名称。
 - 使用 **set quoted_identifier on**（如果保留字是表名、视图名或列名）。如果您的大多数应用程序都使用存储过程，则可以删除这些过程并用 **set quoted_identifier on** 将其重新创建，然后用引号引起所有标识符。所有用户都将能够运行这些过程，而不必为他们的会话使用 **set quoted_identifier on**。您可以使用 **set quoted_identifier on**，创建为表或列提供替代名称的视图并将您的应用程序更改为引用这些视图。

下例提供新保留字 “key”、“level” 和 “work” 的替代名称：

```
create view keyview
as
select lvl = "level", wrk = "work"
from "key"
```

set 命令的语法是：

```
set quoted_identifier on
```

- 如果既未更改标识符，也未使用分隔标识符，那么任何将保留字用作标识符的查询都会报告错误（通常是语法错误）。例如：

```
select level, work from key
```

```
Msg 156, Level 15, State 1:
Server 'rosie', Line 1:
Incorrect syntax near the keyword 'level'.
```

► 注意

带引号的标识符选项是 SQL92 选项，许多支持 Adaptive Server 功能的客户端产品都可能不支持该选项。例如，不能在名称为保留字的表上使用 **bcp**。在选择带引号的标识符选项之前，应使用所有将用来访问 Adaptive Server 的工具对各种对象执行测试。使用 **set quoted_identifier on**，创建一个表名是保留字，各个列名也是保留字的表。如果客户端产品生成 SQL 代码，它就必须将标识符用双引号引起来（如果它们是保留字），并将字符常量用单引号引起来。

- 如果过程、触发器和视图所依赖的对象已经更改了名称，它们可能会在更名后继续工作，但这些对象将在重新编译查询计划时停止工作。进行重新编译的原因有多种，并且不会通知用户。要避免意外的功能损失，可在更改所依赖对象的名称后立即更改过程、触发器和视图中对象的名称。
- 无论是更改对象名还是使用分隔标识符，都必须更改所有包括保留字的存储过程、视图、触发器和应用程序。如果更改对象名，则必须更改标识符；如果使用分隔标识符，则必须添加 **set quoted_identifier** 选项和引号。
- 如果没有将过程、触发器、视图、规则和缺省值的文本保存在操作系统文件中，就可以使用 **defncopy** 将定义从服务器复制到文件中。参见针对于所用平台的 *实用程序手册* 中的 **defncopy**。

更改标识符

- 在更改 **sp_checkreswords** 所报告项目的名称时，如果有过程、触发器、视图和应用程序引用了使用保留字的对象，则必须更改所有这些对象中的名称。
- 在更改标识符名称之前应转储您的数据库。更改标识符名称之后，可运行 **dbcc** 来确定是否已经不存在问题，然后再次转储该数据库。
- 如果您正在活动的生产数据库上更改标识符：
 - 应在系统最为空闲时执行更改，以便尽量少地干扰用户。
 - 应通过查找所有使用特定标识符的 **Open Client DB-Library™** 程序、窗口应用程序、存储过程、触发器和脚本来进行仔细的准备。这样，就可以在源代码中进行所需的编辑，然后尽可能快地更改标识符并替换过程及代码。
- 过程 **sp_depends** 有助于查找使用表名和视图名的过程、视图和触发器。

使用 *sp_rename* 更改标识符

- 系统过程 **sp_rename** 用于重命名表、索引、视图、过程、触发器、规则、缺省值、用户定义的数据类型以及列。使用 **sp_renamedb** 可将数据库重命名。
- 表 7-10 显示了可以用 **sp_rename** 更改的标识符类型，并列出了必须在服务器和应用程序中作出的其它更改。

表 7-10: *sp_rename* 和更改标识符

标识符	务必要
表名	<ul style="list-style-type: none">• 删除所有引用该表的过程、触发器和视图，并用新名称重新创建这些对象。使用 sp_depends 查找依赖于该表的对象。• 更改所有引用该表的应用程序或 SQL 源脚本，以使用新的表名。• 更改用表名执行表级检查的 dbcc 脚本。
索引名	<ul style="list-style-type: none">• 删除所有创建或删除该索引的存储过程，并用新名称创建这些过程。• 更改所有创建或删除该索引的应用程序或 SQL 源脚本。• 更改用索引名执行索引级检查的 dbcc 脚本。
视图名	<ul style="list-style-type: none">• 删除所有引用该视图的过程、触发器和视图，并用新名称重新创建这些对象。使用 sp_depends 查找依赖于该视图的对象。• 更改所有引用该视图的应用程序或 SQL 源脚本，以使用新的视图名。
过程名称	<ul style="list-style-type: none">• 删除所有引用该过程的过程和触发器，并用新的过程名重新创建这些对象。• 更改所有执行该过程的应用程序或 SQL 源脚本，以使用新的名称。• 如果其它服务器远程调用该过程，则必须更改远程服务器上的应用程序，以使用新的过程名。
触发器名称	<ul style="list-style-type: none">• 更改所有创建该触发器的 SQL 源脚本。
规则名称	<ul style="list-style-type: none">• 更改所有创建该规则的 SQL 源脚本。
缺省名称	<ul style="list-style-type: none">• 更改所有创建该缺省名称的 SQL 源脚本。
用户定义数据类型 的名称	<ul style="list-style-type: none">• 删除所有通过用户定义的数据类型来创建表的过程，并用新名称重新创建这些过程。• 更改所有通过用户定义的数据类型来创建表的应用程序。

表 7-10: sp_rename 和更改标识符 (续)

标识符	务必要
列名	<ul style="list-style-type: none">删除所有引用该列的过程、触发器和视图，并用新名称重新创建这些对象。<code>sp_depends</code> 无法找到列名称引用。以下查询显示了引用“key”列的过程、触发器和视图的名称： <pre>select distinct sysobjects.name from sysobjects, syscomments where sysobjects.id = syscomments.id and syscomments.text like "%key%"</pre>更改所有通过名称引用该列的应用程序和 SQL 源脚本。

以下命令将视图 *isolation* 的名称更改为 *isolated*:

```
sp_rename "isolation", isolated
```

以下命令将更改视图 *isolated* 中某一列的名称:

```
sp_rename "isolated.key", keyname
```

- 使用 `sp_depends` 可以列出所有引用将重命名的视图、过程或表的视图、过程和触发器。要在将对象重命名后使用 `sp_depends`，应提供新的名称。例如:

```
sp_depends new_name
```

用 *sp_renamedb* 重命名数据库

- 要更改数据库的名称，可使用 `sp_renamedb`。该数据库必须处于单用户模式。删除并重新创建所有显式引用该数据库名的过程、触发器和视图。有关详细信息，参见 `sp_renamedb`。

更改其它标识符

- 要更改用户名、设备名、远程服务器名、远程服务器用户名、段名、字符集名和语言名，首先应确定是否可以删除该对象或用户并将其再次添加或创建。如果不能这样做，则使用以下命令:

```
sp_configure "allow updates to system tables", 1
```

它允许对系统表直接更新。只有系统安全员才能设置 `allow updates to system tables` 配置参数。

在直接更新系统表时产生的错误可能会在 **Adaptive Server** 中造成严重的错误。要确定是否可以删除对象或用户并将其重新创建，参见表 7-11。

第 1-122 页的表 7-13: *更改标识符时的考虑事项* 显示了对这一组标识符的可能依赖性。无论您选择的升级方法是删除和重新创建对象、使用分隔标识符还是对系统表直接更新，都应在该表中查找可能的依赖性。

表 7-11: 更改标识符时直接更新系统表的替代方法

标识符类型	用于避免直接更新系统表的建议操作
用户名和登录名	要更改无对象的用户的名称，首先在每个数据库中使用 <code>sp_helprotect username</code> 记录该用户的权限。然后，从所有数据库中删除该用户 (<code>sp_dropuser</code>)，并删除登录 (<code>sp_droplogin</code>)。最后，添加新的登录名 (<code>sp_addlogin</code>)，在数据库中添加新的用户名 (<code>sp_adduser</code>)，并用 <code>grant</code> 恢复用户的权限。
设备名	如果设备已经完全分配，则无需在 <code>create database</code> 命令中使用它的名称，这样就可使该设备名保持不变。
远程服务器名	如果远程服务器中没有大量的远程登录名，则应删除远程服务器 (<code>sp_dropserver</code>)，然后用新名称添加该服务器 (<code>sp_addserver</code>)。
远程服务器登录	用 <code>sp_dropremotelogin</code> 删除远程登录，使用 <code>sp_addremotelogin</code> 并以新名称添加该登录，然后用 <code>grant</code> 恢复该用户执行过程的权限。
段名	在段上创建对象之后，很少会使用段名。
字符集名和语言名	只有当系统管理员用 <code>sp_addlanguage</code> 创建替代语言后，语言和字符集才会将保留字用作标识符。用 <code>sp_droplanguage</code> 删除该语言，然后用新名称将其添加。

◆ 警告!

直接更新系统表会非常危险。它可能导致严重的错误，使 Adaptive Server 无法运行或使数据库中的对象无法访问。要直接更新系统表，应该在您心平气和、精力集中，并且服务器上没有任何（或只有很少的）生产活动时进行。如果可能，使用表 7-11 所述的替代方法。

- 下例显示了一个“安全”的用户名更新过程，在所有数据修改命令之前是一个 `begin transaction` 命令：

系统安全员执行以下命令：

```
sp_configure "allow updates to system tables", 1
```

然后，您可以执行以下命令：

```
begin transaction
update sysusers
set name = "workerbee"
where name = "work"
```

此时运行查询，检查该命令是否只影响要更改的行。只有更改 *syslogins* 中的 *language* 名称才会影响多个行的标识符。

- 如果查询只影响正确的行，可使用 **commit transaction**。
- 如果查询影响多个行或错误的行，可使用 **rollback transaction**，确定问题的根源，然后正确地执行该命令。

完成执行后，系统安全员将用以下命令关闭 **allow updates to system tables** 配置参数：

```
sp_configure "allow updates to system tables", 0
```

◆ 警告！

在每个用户定义的事务中，只应更新单个数据库中的系统表。不要在发出 **begin transaction** 命令后更新多个数据库中的表。这种操作会使恢复变得极为困难。

表 7-12 显示了更改保留字时应更新的系统表和列。前面带有 “*master.dbo*” 的表只出现在 *master* 数据库中。所有其它表都出现在 *master* 和用户数据库中。在进行更新之前，应确保使用的是正确的数据库。您可以用以下命令检查数据库名称是否正确：

```
select db_name()
```

表 7-12: 更改标识符时要更新的系统表列

标识符类型	要更新的表	列名
用户名	<i>sysusers</i>	<i>name</i>
登录名	<i>master.dbo.syslogins</i>	<i>name</i>
段名	<i>syssegments</i>	<i>name</i>
设备名	<i>sysdevices</i>	<i>name</i>
远程服务器名	<i>syssservers</i>	<i>srvname</i>
远程服务器网络名	<i>syssservers</i>	<i>srvnetname</i>
字符集名	<i>master.dbo.syscharsets</i>	<i>name</i>
语言名	<i>master.dbo.syslanguages</i> <i>master.dbo.syslogins</i>	<i>name</i> <i>language</i>

表 7-13 显示了需要在服务器和应用程序中进行的其它更改：

表 7-13：更改标识符时的考虑事项

标识符	务必要
登录名	更改该用户在其每个数据库中的用户名。
用户名	删除、编辑并重新创建所有对该用户拥有的对象使用限定 (<i>owner_name.object_name</i>) 引用的过程、触发器和视图。更改所有使用限定对象名的应用程序和 SQL 源脚本，以使用新的用户名。您不必删除这些对象本身，因为 <i>sysusers</i> 已按照存储用户 ID（而不是用户名）的列链接到 <i>sysobjects</i> 。
设备名	更改所有引用该设备名的 SQL 源脚本或应用程序，以使用新的设备名。
远程服务器名	更改远程服务器上的名称。如果 <i>sp_checkreswords</i> 报告的是本地服务器的名称，则必须先重新启动服务器才能发出或接收远程过程调用。
远程服务器网络名	更改接口文件中的服务器名。
远程服务器登录名	更改远程服务器上的名称。
段名	删除并重新创建所有在该段名上创建表或索引的过程。更改所有在段上创建对象的应用程序，以使用新的段名。
字符集名	无。
语言名	更改 <i>master.dbo.syslanguages</i> 和 <i>master.dbo.syslogins</i> 。对 <i>syslogins</i> 的更新可能会涉及多个行。此外，务必要更改本地化文件的名称。

使用分隔标识符

- 可以将分隔标识符用于表名、列名和视图名。但不能将分隔标识符用于其它对象名。
- 如果选择使用分隔标识符，则应使用 **set quoted_identifier on**，然后删除并重新创建所有使用该标识符的过程、触发器和视图。编辑这些对象的文本，并将保留字用双引号引起来，将所有字符串用单引号引起来。

下例显示为使用分隔标识符而对查询作出的更改。该示例更新名为 *work* 的表，该表包括名为 *key* 和 *level* 的列。下面列出的是 10.0 之前版本的查询（它将字符文字用双引号引起来），以及为使用分隔标识符对该查询进行编辑后的版本：

```
/* pre-release 10.0 version of query */
update work set level = "novice"
  where key = "19-732"
```

```
/* 10.0 or later version of query, using
** the quoted identifier option
*/
update "work" set "level" = 'novice'
      where "key" = '19-732'
```

- 对于所有将保留字用作标识符的应用程序，必须作出如下更改：
 - 应用程序必须启用带引号标识符选项。
 - 保留字的所有用户都必须用双引号引起来。
 - 应用程序在启用带引号标识符选项时使用的所有字符文字必须用单引号引起来。否则，Adaptive Server 会把它们解释为对象名。

例如，以下查询会导致错误消息：

```
set quoted_identifier on
select * from titles where title_id like "BU%"
```

以下是正确的查询：

```
select * from titles where title_id like 'BU%'
```

- 在启用分隔标识符时创建的存储过程必须在关闭该选项时运行。（**allow updates to system tables** 选项也是如此。）也就是说，可以打开带引号标识符模式，删除存储过程，编辑该过程以在用作标识符的保留字两边插入引号，然后重新创建该过程。所有用户无需使用 **set quoted_identifier** 即可执行该过程。

权限

只有系统管理员才能执行 **sp_checkreswords**。

使用的表

master..syscharsets、master..sysdatabases、master..sysdevices、master..syslanguages、master..syslogins、master..sysremotelogins、master..sysservers、master..sysmessages、syscolumns、sysindexes、sysobjects、syssegments、systypes、sysusers

参见

命令	set
系统过程	sp_configure、sp_depends、sp_rename、sp_renamedb
实用程序命令	defncopy

sp_checksource

功能

检查是否存在已编译对象的源文本。

语法

```
sp_checksource [objname [, tabname [, username]]]
```

参数

objname — 是要检查其源文本是否存在的已编译对象。

tabname — 是要检查在其上定义的所有检查约束、缺省值和触发器是否存在的表或视图的名称。

username — 即用户名，该用户拥有要检查源文本是否存在的已编译对象。

示例

1. sp_checksource

检查是否存在当前数据库中所有已编译对象的源文本。

2. sp_checksource titleview

检查是否存在 *titleview* 视图的源文本。

3. sp_checksource title_vu, @username = Mary

检查是否存在 Mary 拥有的 *title_vu* 视图的源文本。

4. sp_checksource list_phone_proc

检查是否存在自定义存储过程 *list_phone_proc* 的源文本。

5. sp_checksource @tabname = "my_tab"

检查是否存在所有定义于 *my_tab* 表上的检查约束、触发器和声明缺省值的源文本。

6. sp_checksource @objname = "my_vu", @tabname = "my_tab"

检查是否存在视图 *my_vu* 的源文本和所有定义于 *my_tab* 表上的检查约束、触发器和缺省值的源文本。

7. sp_checksource @username = "Tom"

检查是否存在 Tom 拥有的所有已编译对象的源文本。

注释

- **sp_checksource** 检查是否存在指定已编译对象的源文本。如果存在指定对象的源文本，**sp_checksource** 将返回 0。如果不存在指定对象的源文本，**sp_checksource** 将返回 1。
- 如果未提供任何参数，**sp_checksource** 将检查是否存在当前数据库中所有已编译对象的源文本。
- 要使用不带任何参数的 **sp_checksource**，您必须是数据库所有者或系统管理员。

权限

只有数据库所有者或系统管理员才能执行 **sp_checksource** 来检查是否存在其他用户拥有的已编译对象的源文本。任何用户都可以执行 **sp_checksource** 来检查是否存在自己的已编译对象的源文本。

使用的表

syscolumns、*syscomments*、*sysconstraints*、*sysobjects*、*sysprocedures*

参见

系统过程	sp_hidetext
------	-------------

sp_chgattribute

功能

为在以后分配表或索引的空间而更改 **max_rows_per_page**、**fillfactor**、**reservepagegap** 或 **exp_row_size** 值；设置表的 **concurrency_opt_threshold**。

语法

```
sp_chgattribute objname, {"max_rows_per_page" |  
    "fillfactor" | "reservepagegap" | "exp_row_size"  
    concurrency_opt_threshold }, optvalue
```

参数

objname — 是要为其更改特性的表或索引的名称。

max_rows_per_page — 指定行大小。此选项用于包含可变长度列的表。

fillfactor — 指定在运行 **reorg rebuild** 命令或 **alter table** 命令来更改锁定方案而导致重新创建索引或复制表页时，Adaptive Server 对每页的填充程度。**fillfactor** 百分比仅在重新创建索引时才起作用。有效值为 0-100。

reservepagegap — 指定扩展 I/O 分配操作期间填充页与要保留的空白页之比。对于每个指定的 **num_pages**，保留一个空白页以用于表的未来扩展。有效值为 0-255。缺省值为 0。

exp_row_size — 为 DOL 锁定表中的行保留指定数量的空间。使用此选项可减少转移的行数，而在更新期间转移行会耗费大量资源。有效值是 0、1 和表长度最大值和最小值之间的任一值。0 表示应用全服务器范围的设置，1 表示完全填充数据页上的行。

concurrency_opt_threshold — 指定表大小（以页表示），对 DOL 锁定表的访问应该从该大小值开始优化，以减少 I/O（而不是为了并发）。如果表小于由 **concurrency_opt_threshold** 指定的页数，则通常使用可用的索引来为并发进行查询优化；如果表大于由 **concurrency_opt_threshold** 指定的页数，则为 I/O 进行查询优化。有效值为 -1 到 32767。如果将该值设置为 0，则禁用并发优化。使用 -1 可对大于 32767 页的表强制并发优化。缺省值为 15 页。

optvalue — 是新的值。有效值和缺省值取决于指定的参数。

示例

1. `sp_chgattribute authors, "max_rows_per_page", 1`

为 *authors* 表将 `max_rows_per_page` 设置为 1，以便于以后的空间分配。

2. `sp_chgattribute "titles.titleidind",
"max_rows_per_page", 4`

为 *titleidind* 索引将 `max_rows_per_page` 设置为 4，以便于以后的空间分配。

3. `sp_chgattribute "titles.title_ix", "fillfactor", 90`

为 *title_ix* 中的页将 `fillfactor` 指定为 90%。

4. `sp_chgattribute authors, "exp_row_size", 120`

为 *authors* 表将 `exp_row_size` 设置为 120，以便于以后的空间分配。

5. `sp_chgattribute "titles.titleidind",
"reservepagegap", 16`

为 *titleidind* 索引将 `reservepagegap` 设置为 16，以便于以后的空间分配。

6. `sp_chgattribute "titles",
concurrency_opt_threshold, 0`

关闭 *titles* 表的并发优化。

注释

- `sp_chgattribute` 更改 `max_rows_per_page`、`fillfactor`、`reservepagegap` 或 `exp_row_size` 的值，以便在以后对表或索引进行空间分配或数据修改。它不会影响现有数据页的空间分配。只能为当前数据库中的对象更改这些值。
- 如果将 `max_rows_per_page` 设置为 0，则指示 Adaptive Server 添满数据或索引页且不限制行数（这是 Adaptive Server 在未设置 `max_rows_per_page` 时的缺省行为）。
- 如果 `optvalue` 的值过小，可能会导致页面拆分。当需要在页中添加新的数据或索引行，但页中没有足够空间可用于新行时，就会发生页面拆分。通常，现有页上的数据会相当平均地在新分配的页和现有页之间拆分。

- 如需求得非集群索引的最大值，可将页大小减去 32，然后将所得值除以索引键大小。以下语句计算非集群索引 *titleind* 的 **max_rows_per_page** 最大值：

```
select
    (select @@pagesize - 32) / minlen
    from sysindexes where name = "titleind"
-----
                288
```

如果为 *optvalue* 指定的值过大，Adaptive Server 将返回错误消息，指明允许的最大值。
- 如果为 **max_rows_per_page**、**fillfactor**、**reservepagegap** 或 **exp_row_size** 指定了不正确的值，**sp_chgattribute** 将返回错误消息，指明有效的值。
- 有关 **max_rows_per_page**、**fillfactor**、**reservepagegap**、**exp_row_size** 和 **concurrency_opt_threshold** 的详细信息，参见 *Performance and Tuning Guide*。

权限

只有对象所有者才能执行 **sp_chgattribute**。

使用的表

sysindexes、*sysobjects*、*systabstats*

参见

命令	alter table、create index、create table
系统过程	sp_helpindex

sp_clearpsexex

功能

清除已用 **sp_setpsexex** 设置的 Adaptive Server 会话的执行特性。

语法

```
sp_clearpsexex spid, exeattr
```

参数

spid — 是要清除执行特性的会话的进程 ID。

exeattr — 标识要清除的执行特性。 *exeattr* 的值为 "priority" 和 "enginegroup"。

示例

```
1. sp_clearpsexex 12, 'enginegroup'
```

删除进程 12 的引擎组条目。

注释

- **sp_clearpsexex** 清除已用 **sp_setpsexex** 设置的会话的执行特性。有关详细信息，参见 *Performance and Tuning Guide*。
- 如果执行特性未在会话有效期间清除，它们就会在该会话异常退出或终止时被清除。
- 如果关联的引擎组中没有联机引擎， **sp_clearpsexex** 将会失败。
- 当删除引擎组条目后，会话将在类定义或缺省类所确定的引擎组上执行。
- 使用 **sp_who** 可列出进程 ID (*spid*)。

权限

只有系统管理员才能执行 **sp_clearpsexex** 来清除所有用户的优先级特性。任何用户都可以执行 **sp_clearpsexex** 来清除该用户拥有的任务的优先级特性。

使用的表*sysattributes*、*sysprocesses***参见**

系统过程	sp_addengine、sp_addexclass、 sp_bindexclass、sp_dropengine、 sp_dropexclass、sp_setpsex、 sp_showcontrolinfo、sp_showexclass、 sp_showpsex、sp_unbindexclass
------	--

sp_clearstats

功能

为所有服务器用户或指定用户开始新的会计周期。通过执行 **sp_reportstats**，可输出以前周期的统计信息。

语法

```
sp_clearstats [loginame]
```

参数

loginame — 是用户的登录名。

示例

1. sp_clearstats

Name	Since	CPU	Percent CPU	I/O	Percent I/O
-----	-----	-----	-----	-----	-----
probe	Jun 19 1990	0	0%	0	0%
julie	Jun 19 1990	10000	24.9962%	5000	24.325%
jason	Jun 19 1990	10002	25.0013%	5321	25.8866%
ken	Jun 19 1990	10001	24.9987%	5123	24.9234%
kathy	Jun 19 1990	10003	25.0038%	5111	24.865%

(5 rows affected)

Total CPU	Total I/O
-----	-----
40006	20555

5 login accounts cleared.

为所有用户开始新的会计周期。

2. sp_clearstats kathy

Name	Since	CPU	Percent CPU	I/O	Percent I/O
-----	-----	-----	-----	-----	-----
KATHY	Jul 24 1990	498	49.8998%	483924	9.1829%

(1 row affected)

Total CPU	Total I/O
-----	-----
998	98392

1 login account cleared.

为用户 “kathy” 开始新的会计周期。

注释

- **sp_clearstats** 用于创建会计周期，只应在周期结束时运行。
- 由于 **sp_clearstats** 将清除会计统计信息，所以在运行该过程之前必须记录这些统计信息。
- **sp_clearstats** 更新 *syslogins* 的 *accdate* 域，并清除 *syslogins* 的 *totcpu* 和 *totio* 域。

权限

只有系统管理员才能执行 **sp_clearstats**。

使用的表

master..syslogins、*sysobjects*

参见

系统过程	sp_reportstats
------	-----------------------

sp_cmp_all_qplans

功能

比较两个抽象计划组中所有的抽象计划。

语法

```
sp_cmp_all_qplans group1, group2 [, mode]
```

参数

group1, group2 — 是两个抽象计划组的名称。

mode — 是以下显示选项之一: **counts**、**brief**、**same**、**diff**、**first**、**second**、**offending** 和 **full**。缺省模式为 **counts**。

示例

1. sp_cmp_all_qplans dev_plans, prod_plans

If the two query plans groups are large, this might take some time.

Query plans that are the same

```
count
-----
49
```

Different query plans that have the same association key

```
count
-----
1
```

Query plans present only in group 'dev_plans' :

```
count
-----
1
```

Query plans present only in group 'prod_plans' :

```
count
-----
0
```

生成两个抽象计划组的缺省报告。

2. sp_cmp_all_qplans dev_plans, prod_plans, brief

使用 **brief** 模式生成报告。

注释

- 使用 **sp_cmp_all_qplans** 可检查两个计划组中抽象计划的差异。
- **sp_cmp_all_qplans** 匹配每个组中具有相同用户 ID 和查询文本的计划对。计划的分类如下：
 - 相同的计划
 - 在两个组中具有相同关联键但具有不同抽象计划的计划。关联键是组 ID、用户 ID 和查询文本。
 - 存在于一个组中但不存在于另一个组中的计划

表 7-14 显示了各种报告模式以及为各种模式报告的信息的类型。

表 7-14: sp_cmp_all_qplans 的报告模式

模式	报告的信息
counts	以下计划的计数：相同的计划、具有相同关联键但属于不同组的计划，以及存在于一个组但不存在于另一个组中的计划。这是缺省的报告模式。
brief	counts 所提供的信息、各个组中计划不同但关联键相同的抽象计划 ID，以及在一个组中但不在另一个组中的计划 ID。
same	所有计数以及所有查询和计划匹配的抽象计划的 ID、查询和计划。
diff	所有计数以及所有查询和计划不同的抽象计划的 ID、查询和计划。
first	所有计数以及所有在第一个计划组中但不在第二个计划组中的抽象计划的 ID、查询和计划。
second	所有计数以及所有在第二个计划组中但不在第一个计划组中的抽象计划的 ID、查询和计划。
offending	所有计数以及所有关联键不同或不同时存在于两个组中的抽象计划的 ID、查询和计划。它是 diff、first 和 second 模式的组合。
full	所有计数以及所有抽象计划的 ID、查询和计划。它是 same 和 offending 模式的组合。

- 要比较两个单独的抽象计划，可使用 **sp_cmp_qplans**。要查看抽象计划组的名称，可使用 **sp_help_qpgroup**。
- 当系统管理员或数据库所有者运行 **sp_cmp_all_qplans** 时，它将报告两个组中的所有计划。当其他用户执行 **sp_cmp_all_qplans** 时，它只报告具有该用户 ID 的计划。

权限

任何用户都可以执行 **sp_cmp_all_qplans**。

使用的表

sysattributes、 *sysqueryplans*

参见

系统过程	sp_cmp_qplans
------	----------------------

sp_cmp_qplans

功能

比较两个抽象计划。

语法

```
sp_cmp_qplans id1, id2
```

参数

id1, id2 一是两个抽象计划的 ID。

示例

```
1. sp_cmp_qplans 411252620, 1383780087
```

```
The queries are the same.  
The query plans are the same.
```

```
2. sp_cmp_qplans 2091258605, 647777465
```

```
The queries are the same.  
The query plans are different.
```

注释

- **sp_cmp_qplans** 比较查询、抽象计划和两个抽象计划的散列键，并报告查询是否相同以及计划是否相同。它将为查询输出以下消息之一：
 - 查询相同。
 - 查询不同。
 - 查询不同但具有相同的散列键。它将为抽象计划输出以下消息之一：
 - 查询计划相同。
 - 查询计划不同。

- **sp_cmp_qplans** 也会输出返回状态，以显示比较的结果。状态值 1、2 和 10 是附加值。表 7-15 中列出了这些状态值。

表 7-15: **sp_cmp_qplans** 的返回状态值

返回值	含义
0	查询文本和抽象计划相同。
+1	查询和散列键不同。
+2	查询不同但散列键相同。
+10	抽象计划不同。
100	某一个或两个计划 ID 不存在。

- 要查找计划的 ID，可使用 **sp_help_qpgroup** 或 **sp_find_qplan**。计划 ID 也可以由 **create plan** 返回并包含在 **showplan** 输出中。

权限

任何用户都可以执行 **sp_cmp_qplans** 来比较其拥有的计划。只有系统管理员或数据库所有者才能比较其它用户拥有的计划。

使用的表

sysqueryplans

参见

系统过程	sp_cmp_all_qplans、sp_help_qpgroup
------	-----------------------------------

sp_commonkey

功能

定义两个表或视图之间的公用键（经常被连接的列）。

语法

```
sp_commonkey tabaname, tabbname, col1a, col1b  
[, col2a, col2b, ..., col8a, col8b]
```

参数

tabaname — 是要连接的第一个表或视图的名称。

tabbname — 是要连接的第二个表或视图的名称。

col1a — 是在表或视图 *tabaname* 中构成公用键的第一个列的名称。至少应指定一对列（第一个表或视图中的一个列和第二个表或视图中的一个列）。

col1b — 是表或视图 *tabbname* 中与表或视图 *tabaname* 中的 *col1a* 连接的对应列的名称。

示例

```
1. sp_commonkey titles, titleauthor, title_id,  
   title_id
```

定义 *titles.titleid* 和 *titleauthor.titleid* 的公用键。

```
2. sp_commonkey projects, departments, empid, empid
```

假定有两个表 *projects* 和 *departments*，每个表都包含名为 *empid* 的列。此语句定义这两个列上的常用连接。

注释

- 创建公用键的目的是为了使数据库设计中的隐式逻辑关系变为显式关系。此信息可以由应用程序使用。**sp_commonkey** 并不强制参照完整性约束；可使用 **create table** 或 **alter table** 命令的 **primary key** 和 **foreign key** 子句来强制键关系。
- 执行 **sp_commonkey** 后，可将键添加到 **syskeys** 系统表中。要显示已定义的公用键的报告，可使用 **sp_helpkey**。
- 要定义两个表或视图之间的公用键，您必须至少是其中一个的所有者。

- 第一个表或视图中的列数必须与第二个表或视图中的列数相同。在每个表或视图中，参与公用键的列最多可以有 **8** 列。公用列的数据类型也必须一致。对于可指定长度的列，长度可以不同。公用列的空值类型无需一致。
- 安装进程在系统表的适当列上运行 **sp_commonkey**。

权限

只有 *tabaname* 或 *tabbname* 的所有者才能执行 **sp_commonkey**。

使用的表

syscolumns、*syskeys*、*sysobjects*

参见

命令	alter table、create table、create trigger
系统过程	sp_dropkey、sp_foreignkey、sp_helpjoins、sp_helpkey、sp_primarykey

sp_companion

功能

执行集群操作，例如在高可用性系统中将 **Adaptive Server** 配置为辅助协同服务器，以及将协同服务器从一种故障替换模式转换到其它模式。**sp_companion** 从辅助协同服务器中运行。

语法

```
sp_companion
  [server_name
  {,configure
    [{,with_proxydb | NULL}]
    [,srvlogin]
    [,server_password]
    [,cluster_login]
    [,cluspassword]]
  | drop
  | suspend
  | resume
  | prepare_failback
  | do_advisory}
    {, all
    | help
    | group attribute_name
    | base attribute_name}
```

参数

server_name — 是执行集群操作的 **Adaptive Server** 的名称。

configure — 将 **server_name** 所指定的服务器配置为故障替换配置中的主协同服务器。

drop — 将协同服务器从故障替换配置中永久删除。完成该命令后，服务器即处于单服务器模式。

suspend — 将协同服务器从故障替换配置中暂时删除。完成该命令后，协同服务器即处于挂起模式。

resume — 撤消 **suspend** 命令并恢复协同服务器之间的正常协同模式。

prepare_failback — 使辅助协同服务器准备释放主协同服务器的资源，以便进行故障恢复。

do_advisory — 为在故障替换模式中成功执行主协同服务器的功能，验证辅助协同服务器是否兼容。

all — 使 **do_advisory** 检查所有参数。

help — 显示有关 **do_advisory** 参数的信息和语法。

group attribute — 将 **do_advisory** 限制为仅检查组特性。

base attribute — 将 **do_advisory** 限制为仅检查基特性。

with_proxydb — 如果在协同服务器的初始配置中包括此参数，则将在辅助协同服务器上为除系统数据库外的所有数据库（以及随后添加的所有数据库）创建代理数据库。缺省情况下，**with_proxydb** 被禁用。

srvlogin — 是用户访问协同服务器的登录。缺省情况下，**srvlogin** 的值为 “sa”。

srvpassword — 是用户访问协同服务器的口令。缺省情况下，**srvpassword** 的值为空。

cluster_login — 是用户登录到集群中的登录。缺省情况下，**cluster_login** 的值为 sa。

cluspassword — 是登录到集群中时必须提供的用户口令。缺省情况下，**cluspassword** 的值为空。

示例

1. **sp_companion "MONEY1", configure**

将 Adaptive Server MONEY1 配置为主协同服务器。

2. **sp_companion "MONEY1", configure, with_proxydb, "sa", "sapsswd"**

将 Adaptive Server MONEY1 配置为主协同服务器并在辅助协同服务器上创建代理数据库。

3. **sp_companion "PERSONEL1", "drop"**

从故障替换配置中删除 Adaptive Server PERSONEL1。完成该命令后，主协同服务器和辅助协同服务器都将处于单服务器模式。

4. **sp_companion "MONEY1", "resume"**

恢复协同服务器（在此例中为 MONEY1）的正常协同模式。

5. **sp_companion "PERSONEL1", "prepare_failback"**

使主协同服务器（在此例中为 PERSONEL1）准备转换到正常的协同模式并恢复对已执行故障替换的 Adaptive Server 的控制。

6. `sp_companion "PERSONEL1", do_advisory, "all"`

检查并确保对 PERSONEL1 协同服务器的集群操作将成功执行。此例中的 `do_advisory` 使用了 `all` 参数，所以它将检查 PERSONEL1 的所有 `do_advisory` 特性，以确保它们将不会阻碍集群操作的成功执行，并确保辅助协同服务器能够在完成故障替换后成功地执行主协同服务器的操作。

7. `sp_companion "PERSONEL1, do_advisory, "CIS"`

检查并确保协同服务器上的所有 CIS（组件集成服务）特性都与本地服务器不兼容。

注释

- `sp_companion` 执行集群操作，例如在高可用性系统中将 Adaptive Server 配置为辅助协同服务器。`sp_companion` 还将协同服务器从一种故障替换模式转换到其它模式（例如，从故障替换模式回到正常模式）。`sp_companion` 从辅助协同服务器中运行。
- `sp_companion` 用 `installhasvss`（在 Windows NT 上用 `insthasv`）而不是 `installmaster` 脚本来安装。`installhasvss` 位于 `$$SYBASE/ASE-12_0/scripts`。
- `sp_companion` 自动禁用 Sybase 的镜像。Sybase 建议您使用第三方镜像软件来保护您的数据不受到磁盘故障的损坏。

有关详细信息，参见 *Using Sybase Failover in A High Availability System*。在运行 `do_advisory` 命令之前，务必要阅读本手册中有关配置的章节以及 `do_advisory` 一章。

权限

只有具有 `ha_role` 的用户才能发出 `sp_companion`。

sp_configure

功能

显示或更改配置参数。

语法

```
sp_configure [configname [, configvalue] | group_name
| non_unique_parameter_fragment]

sp_configure "configuration file", 0, {"write" |
"read" | "verify" | "restore"} "file_name"
```

参数

语法	作用
sp_configure	按组显示配置参数、其当前值、缺省值、最近设置的值以及该设置所使用的内存量。只显示其显示级别不高于用户显示级别的参数。
sp_configure configname	显示当前值、缺省值、最近更改的值以及设置所有符合 <i>parameter</i> 的参数所使用的内存量。
sp_configure configname, configvalue	将 <i>configname</i> 重新设置为 <i>configvalue</i> ，并显示当前值、缺省值、配置值和 <i>configname</i> 所使用的内存量。
sp_configure configname, 0, "default"	将 <i>configname</i> 重新设置为它的缺省值，并显示当前值、缺省值、配置值和 <i>configname</i> 所使用的内存量。
sp_configure group_name	显示 <i>group_name</i> 中的所有配置参数、其当前值、缺省值、最近设置的值（如果适用）以及该设置所使用的内存量。
sp_configure non_unique_parameter_fragment	显示所有符合 <i>non_unique_parameter_fragment</i> 的参数名，其当前值、缺省值、配置值以及所使用的内存量。
sp_configure "configuration file", 0, "write", "file_name"	从当前配置中创建 <i>file_name</i> 。如果 <i>file_name</i> 已经存在，则向错误日志写入一条消息；现有文件会根据 <i>file_name.001</i> 、 <i>file_name.002</i> 等约定来重命名。如果已经更改静态参数但没有重新启动服务器， <i>"write"</i> 就会提供该参数的当前运行值。
sp_configure "configuration file", 0, "read", "file_name"	对包含在 <i>file_name</i> 中的值执行校验检查，并且将通过校验的值读入服务器。如果 <i>file_name</i> 遗漏了任何参数，则使用这些参数的当前运行值。
sp_configure "configuration file", 0, "verify", "file_name"	对 <i>file_name</i> 中的值执行校验检查。
sp_configure "configuration file", 0, "restore", "file_name"	用 <i>sysconfigures</i> 中的值创建 <i>file_name</i> 。在配置文件的所有副本都已丢失，而需要生成一个新的副本时，这会很有帮助。

示例

1. sp_configure

按组显示所有配置参数、其当前值、缺省值、最近设置的值（如果适用）以及该设置所使用的内存量。

2. sp_configure "identity"

Configuration option is not unique.

Parameter Name	Default	Memory Used	Config Value	Run Value
identity burning set factor	5000	0	5000	5000
identity grab size	1	0	1	1
size of auto identity column	10	0	10	10

显示所有包括 “identity” 一词的配置参数。

3. sp_configure "recovery interval in minutes", 3

Parameter Name	Default	Memory Used	Config Value	Run Value
recovery interval in minutes	5	0	3	3

Configuration option changed. The SQL Server need not be rebooted since the option is dynamic.

将系统的 **recovery interval in minutes** 设置为 3 分钟。

4. sp_configure "number of device", 0, "default"

将 **number of devices** 的值重新设置为 Adaptive Server 缺省值。

注释

- 任何用户都可以执行 **sp_configure** 来显示有关参数及其当前值的信息（但不能修改参数）。系统管理员可以执行 **sp_configure** 来更改大多数配置参数的值。只有系统安全员才能执行某些参数。这些参数列在本节的“权限”中。
- 当执行 **sp_configure** 来修改动态参数时：
 - 将更新配置值和运行值。
 - 将更新配置文件。
 - 更改会立即生效。
- 当执行 **sp_configure** 来修改静态参数时：
 - 将更新配置值。
 - 将更新配置文件。
 - 只有当重新启动 Adaptive Server 时更改才会生效。

- 如果发出不带任何参数的 **sp_configure**，则将以报告形式按组显示所有配置参数、其当前值、缺省值、最近设置的值（如果适用）以及该设置所使用的内存量。
 - 报告中的 **default** 列显示随 **Adaptive Server** 提供的值。如果没有明确重新配置参数，该参数将保留其缺省值。
 - **memory used** 列显示参数在其当前值所使用的内存量。某些相关参数占用的内存来自同一内存池。例如，**stack size** 和 **stack guard size** 占用的内存已被计入 **number of user connections** 占用的内存。如果将这些参数使用的内存一一相加，则总值会大于实际使用的内存量。在 **memory used** 列中，与其它参数“共享”内存的参数带有井号 (#)。
 - **config_value** 列显示最近用 **sp_configure** 将配置参数设置到的值。
 - **run_value** 列显示 **Adaptive Server** 当前使用的值。该值将在用 **sp_configure** 修改参数值后发生变化。对于静态参数，它在重新启动 **Adaptive Server** 后才会发生变化。该值存储在 **syscurconfigs.value** 中。

► 注意

如果服务器使用的是不区分大小写的排序顺序，不带参数的 **sp_configure** 会按字母顺序返回所有配置参数和组的列表而不分组显示。

- 每个配置参数都具有一个关联的显示级别。一共有三个显示级别：
 - “**basic**” 级别显示最基本的参数。它适用于最一般的服务器调优。
 - “**intermediate**” 级别在显示“基本”参数的同时显示较为复杂的参数。该级别适用于中等复杂级别的服务器调优。
 - “**comprehensive**” 级别显示所有参数，包括最复杂的参数。该级别适用于对服务器进行高精度调优的用户。

缺省的显示级别为“**comprehensive**”。如果将其设置为其它显示级别，则可以使用配置参数的子集，从而减少 **sp_configure** 所显示的信息量。

显示当前显示级别的语法是：

```
sp_displaylevel
```

- 有关各个配置参数的详细信息，参见 *系统管理指南*。

权限

任何用户都可以执行 **sp_configure** 来显示有关参数及其当前值的信息。

只有系统管理员和系统安全员才能执行 **sp_configure** 来修改配置参数。
只有系统安全员才能执行 **sp_configure** 来修改以下参数的值：

- allow procedure grouping**
- allow select on syscomments.text**
- allow updates**
- audit queue size**
- auditing**
- current audit table**
- remote access**
- suspend auditing when full**
- systemwide password expiration**

系统管理员可以修改所有其它参数。

使用的表

*master..sysdevices、 master..sysconfigures、 master..syscurconfigs、
master..sysdatabases、 master..sysdevices、 master..sysindexes、
master..syslanguages、 master..sysmessages、 master..sysobjects、
master..sysservers*

参见

命令	set
系统过程	sp_addlanguage、 sp_audit、 sp_dboption、 sp_displaylevel、 sp_droplanguage、 sp_helpconfig、 sp_modifylogin、 sp_monitorconfig

sp_copy_all_qplans

功能

将一个抽象计划组的所有计划复制到另一个组。

语法

```
sp_copy_all_qplans src_group, dest_group
```

参数

src_group — 是源抽象计划组的名称。

dest_group — 是要将计划复制到的抽象计划组的名称。

示例

```
1. sp_copy_all_qplans dev_plans, ap_stdin
```

将 *dev_plans* 组中的所有抽象计划复制到 *ap_stdin* 组。

注释

- 在复制计划之前，目标组必须已经存在。它可能包括计划。
- sp_copy_all_qplans** 为源组中的每个计划调用 **sp_copy_qplan**。每个计划都当作单独的事务来复制，所以任何使 **sp_copy_all_qplans** 无法完成的问题都不会影响已复制的计划。
- 当 **sp_copy_qplan** 无法复制特定的抽象计划时，它将输出相应的消息。当运行 **sp_copy_all_qplans** 时，您也会看到这些消息。
- 如果目标组中计划的查询文本与源组中的查询文本完全匹配并且用户 ID 相同，则不复制该计划，同时向用户发送一条提供该计划 ID 的消息，但复制进程会继续处理源组中的下一个计划。
- 如果复制大量的抽象计划，可能会需要相当长的时间，同时还需要数据库 *system* 段上的空间以及用来记录数据库更改的空间。使用 **sp_spaceused** 可检查 *sysqueryplans* 的大小以及 *system* 和 *logsegment* 的 **sp_helpsegment** 大小，以检查可用的空间。

权限

任何用户都可以执行 **sp_copy_all_qplans** 来复制其拥有的抽象计划。只有系统管理员或数据库所有者才能复制其它用户拥有的计划。

使用的表

sysattributes、*sysqueryplans*

参见

系统过程	sp_copy_qplan、 sp_help_qpgroup
------	--------------------------------

sp_copy_qplan

功能

将一个抽象计划复制到抽象计划组。

语法

sp_copy_qplan *src_id*, *dest_group*

参数

src_id — 是要复制的抽象计划的 ID。

dest_group — 是目标抽象计划组的名称。

示例

1. sp_copy_qplan 2140534659, ap_stdin

注释

- 在复制抽象计划之前，目标组必须已经存在。由于计划已由计划 ID 唯一标识，所以无需指定源组。
- 当复制计划时，会生成一个新的计划 ID。即使系统管理员或数据库所有者复制计划，该计划仍保留其创建者的用户 ID。要分配不同的用户 ID，系统管理员或数据库所有者可使用 **sp_export_qpgroup** 和 **sp_import_qpgroup**。
- 如果目标组中计划的查询文本与源组中的查询文本以及用户 ID 完全匹配，则不复制该计划，同时向用户发送一条提供计划 ID 的消息。
- 要复制抽象计划组中的所有计划，可使用 **sp_copy_all_qplans**。

权限

任何用户都可以执行 **sp_copy_qplan** 来复制其拥有的计划。只有系统管理员或数据库所有者才能复制其它用户拥有的计划。

使用的表

sysattributes、*sysqueryplans*

参见

系统过程	sp_copy_all_qplans 、 sp_help_qpgroup 、 sp_help_qplan 、 sp_import_qpgroup
------	--

sp_countmetadata

功能

显示 Adaptive Server 中索引、对象或数据库的数目。

语法

```
sp_countmetadata "configname" [, dbname]
```

参数

configname — 可以是 "open indexes"、"open objects" 或 "open databases"。

dbname — 是要对其运行 **sp_countmetadata** 的数据库的名称。如果未提供任何数据库名称，**sp_countmetadata** 将提供所有数据库的总计数。

示例

1. sp_countmetadata "open objects"

```
There are 283 user objects in all database(s),  
requiring 117.180 Kbytes of memory. The 'open  
objects' configuration parameter is currently set  
to a run value of 500.
```

报告 Adaptive Server 中用户对象的数目。该值用于设置在数据库中允许的对象数，以及用于附加对象和临时表的空间。例如：

```
sp_configure "number of open objects", 310
```

2. sp_countmetadata "open indexes", pubs2

```
There are 21 user indexes in pubs2 database(s),  
requiring 8.613 kbytes of memory. The 'open  
indexes' configuration parameter is currently set  
to 600.
```

报告 Adaptive Server 中索引的数目。

注释

- **sp_countmetadata** 显示 Adaptive Server 中索引、对象或数据库的数目，包括系统数据库（如 *model* 和 *tempdb*）的数目。
- 应避免在 Adaptive Server 的高峰期运行 **sp_countmetadata**。否则会使 *sysindexes*、*sysobjects* 和 *sysdatabases* 系统表上出现争用。
- 如果需要有关特定数据库的信息，可对指定数据库运行 **sp_countmetadata**。但在为索引、对象或数据库配置缓存时，应在不使用 *database_name* 选项的情况下运行 **sp_countmetadata**。

- 平台不同，**sp_countmetadata** 返回的内存信息也会有所不同。例如，Adaptive Server for Windows NT 和 Sun Solaris 上的相同数据库会得到不同的 **sp_countmetadata** 结果。但用户索引数、对象数或数据库数方面的信息应该一致。
- **sp_countmetadata** 在计算中不包括临时表。要计算临时表，可将 **open objects** 值加 5%，将 **open indexes** 值加 10%。
- 如果为 *configname* 指定不唯一的 "open indexes"、"open objects" 或 "open databases" 片断，**sp_countmetadata** 将返回带有配置值和当前值的匹配配置参数名的列表。例如：

sp_countmetadata "open"

Configuration option is not unique.

option_name	config_value	run_value
-----	-----	-----
current change w/ open cursors	1	1
number of open databases	12	12
number of open indexes	500	500
number of open objects	500	500
open index hash spinlock ratio	100	100
open index spinlock ratio	100	100
open object spinlock ratio	100	100

权限

只有系统管理员或数据库所有者才能执行 **sp_countmetadata**。

使用的表

master..sysdatabases、*sysindexes*、*sysobjects*

参见

系统过程	sp_configure、sp_helpconfig、sp_monitorconfig
------	---

sp_cursorinfo

功能

报告有关特定游标或所有为会话激活的游标的信息。

语法

```
sp_cursorinfo [{cursor_level | null}] [, cursor_name]
```

参数

cursor_level | null — 是 Adaptive Server 为游标返回信息的级别。可以为 *cursor_level* 指定以下级别：

级别	游标类型
<i>N</i>	任何在特定过程嵌套级别的存储过程内声明的游标。可以为这种游标级别指定任何正数。
0	任何在存储过程外声明的游标。
-1	任一上述级别的游标。可以为该级别指定任何负数。

如果需要了解任何游标级别上具有特定 *cursor_name* 的游标，可为该参数指定 null。

cursor_name — 是游标的特定名称。Adaptive Server 报告指定 *cursor_level* 上使用该名称的所有活动游标的信息。如果省略该参数，Adaptive Server 将报告该级别上所有游标的信息。

示例

1. sp_cursorinfo 0, authors_crsr

Cursor name 'authors_crsr' is declared at nesting level '0'.
The cursor id is 327681
The cursor has been successfully opened 1 times.
The cursor was compiled at isolation level 0.
The cursor is not open.
The cursor will remain open when a transaction is committed or rolled back.
The number of rows returned for each FETCH is 1.
The cursor is read only.
There are 3 columns returned by this cursor.
The result columns are:
Name = 'au_id', Table = 'authors', Type = ID,
Length = 11 (read only)
Name = 'au_lname', Table = 'authors', Type = VARCHAR,
Length = 40 (read only)
Name = 'au_fname', Table = 'authors', Type = VARCHAR,
Length = 20 (read only)

显示级别 0 上名为 *authors_crsr* 的游标的信息。

2. sp_cursorinfo null, author_sales

Cursor name 'author_sales' is declared on procedure 'au_sales'.
Cursor name 'author_sales' is declared at nesting level '1'.
The cursor id is 327682
The cursor has been successfully opened 1 times.
The cursor was compiled at isolation level 1.
The cursor is currently scanning at a nonzero isolation level.
The cursor is positioned after the last row.
The cursor will be closed when a transaction is committed or rolled back.
The number of rows returned for each FETCH is 1.
The cursor is updatable.
There are 3 columns returned by this cursor.
The result columns are:
Name = 'title_id', Table = 'titleauthor', Type = ID,
Length = 11 (updatable)
Name = 'title', Table = 'titles', Type = VARCHAR,
Length = 80 (updatable)
Name = 'total_sales', Table = 'titles', Type = INT (updatable)

显示用户在所有级别上声明的任何 *author_sales* 游标的信息。

注释

- 如果未指定 *cursor_level* 或 *cursor_name*, Adaptive Server 将显示所有活动游标的信息。活动游标是您声明并由 Adaptive Server 分配的游标。
- Adaptive Server 报告各个游标的以下信息：
 - 游标名、嵌套级别、游标 ID 和过程名（如果在存储过程中声明）。
 - 游标已打开的次数。
 - 它编译和当前扫描（如果已打开）所在的隔离级别（0、1 或 3）。
 - 游标是打开还是关闭。如果游标打开，则显示当前游标位置和读取的行数。
 - 是否将关闭已打开的游标（如果删除游标的当前位置）。
 - 游标是否将保持打开或关闭状态（如果游标的当前事务已提交或回退）。
 - 为该游标的每次读取返回的行数。
 - 该游标是可更新还是只读。
 - 游标返回的列数。对于每一列，都将显示列名、表名或表达式结果以及该列是否可以更新。

根据游标的状态，sp_cursorinfo 的输出会有所不同。除列出的信息之外，sp_cursorinfo 还显示该游标的 showplan 输出。有关 showplan 的详细信息，参见 *Performance and Tuning Guide*。

权限

任何用户都可以执行 sp_cursorinfo。

使用的表

sysobjects

参见

命令	declare cursor、 set
----	---------------------

sp_dboption

功能

显示或更改数据库选项。

语法

```
sp_dboption [dbname, optname, {true | false}]
```

参数

dbname — 是要在其中设置选项的数据库的名称。必须使用 *master* 来执行带参数的 **sp_dboption**（即更改数据库选项）。但是，您不能更改 *master* 数据库的选项。

optname — 是要设置的选项的名称。Adaptive Server 可以理解属于选项名称的一部分的任何唯一字符串。如果选项名是关键字或者含有嵌入空白或标点符号，则用引号将选项名引起来。

{true | false} — **true** 打开选项，**false** 关闭选项。

示例

1. sp_dboption

```
Settable database options
database_options
-----
abort tran on log full
allow nulls by default
auto identity
dbo use only
ddl in tran
identity in nonunique index
no chkpt on recovery
no free space acctg
read only
select into/bulkcopy/pllsort
single user
trunc log on chkpt
trunc. log on chkpt.
unique auto_identity index
```

显示数据库选项列表。

```
2. use pubs2
go
master..sp_dboption pubs2, "read", true
go
checkpoint
go
```

使数据库 *pubs2* 变为只读。**read** 字符串将 **read only** 选项从所有可用的数据库选项中唯一标识出来。注意使用引号将关键字 **read** 引起来。

```
3. pubs2..sp_dboption pubs2, "read", false
go
checkpoint
go
```

使数据库 *pubs2* 再次变为可写的。

```
4. use pubs2
go
master..sp_dboption pubs2, "select into", true
go
checkpoint
go
```

允许对 *pubs2* 数据库中的表进行 **select into**、**bcp** 和并行排序操作。**select into** 字符串将 **select into/ bulkcopy** 选项从所有可用的数据库选项中唯一标识出来。注意由于有嵌入的空格，所以需要引号将选项引起来。

```
5. use mydb
go
master..sp_dboption mydb, "auto identity", true
go
checkpoint
go
```

自动在 *mydb* 中所创建的新表中定义 10 位数字的 **IDENTITY** 列。如果在创建新表时没有指定主键、**unique** 约束或 **IDENTITY** 列，则将为每个这样的新表定义 **IDENTITY** 列 **SYB_IDENTITY_COL**。

```
6. use master
go
sp_dboption mydb, "identity in nonunique index",
true
go
use mydb
go
checkpoint
go
```

自动在 *mydb* 表的索引键中包括 **IDENTITY** 列，只要这些表已经拥有了 **IDENTITY** 列。所有在表上创建的索引都将是内部唯一的。

```
7. use master
go
sp_dboption pubs2, "unique auto_identity index",
true
go
use pubs2
go
checkpoint
go
```

自动为 *pubs2* 数据库中的新表包括带有唯一非集群索引的 **IDENTITY** 列。

注释

- *master* 数据库选项设置不能被更改。
- 要显示数据库选项列表，请在 *master* 数据库中不带任何参数地执行 **sp_dboption**。
- 要得到关于特定数据库中数据库选项设置的报告，执行 **sp_helpdb**。
- 数据库所有者或系统管理员可以通过在 *model* 上执行 **sp_dboption**，为所有的新数据库设置或取消设置特定的数据库选项。
- 执行完 **sp_dboption** 之后，直到在为其更改选项的数据库中发出了 **checkpoint** 命令，所作的更改才会生效。

数据库选项

- 当指定数据库的日志段中超过了最后机会阈值时，**abort tran on log full** 选项决定正在进行的事务将被如何处理。其缺省值是 **false**，表示事务被挂起并只有在空间释放后才被激活。如果将此设置改变为 **true**，所有需要写入事务日志的用户查询均被注销，直到日志中的空间被释放。
- 将 **allow nulls by default** 选项设置为 **true**，可以将列的缺省值从 **not null** 更改为 **null**，以符合 **SQL** 标准。列的 **Transact-SQL** 缺省值是 **not null**，表示列值不能为空，除非在 **create table** 或 **alter table** 列定义中指定了 **null**。**allow nulls by default true** 则与之相反。

- **auto identity** 选项设置为 **true (on)** 时，如果在创建新表时未指定 **primary** 键、**unique** 约束或 **IDENTITY** 列，则将为每个新表定义一个 10 位数字的 **IDENTITY** 列。当使用 **select *** 语句选择所有列时，该列不可见。要检索它，必须在选择列表中明确提供此列的名称 **SYB_IDENTITY_COL**。
要设置自动 **IDENTITY** 列的精度，可使用 **size of auto identity column** 配置参数。
尽管可以在 **tempdb** 中将 **auto identity** 设置为 **true**，但其并不被承认和使用，且在该数据库中创建的临时表也不会自动包括 **IDENTITY** 列。
要得到某一含有 **IDENTITY** 列的表中索引的报告，请执行 **sp_helpindex**。
- **dbo use only** 选项被设置为 **true (on)** 时，只有数据库所有者可以使用此数据库。
- 当 **ddl in tran option** 设置为 **true (on)** 时，就可以在事务中使用某些数据定义语言命令。如果在某一数据库中 **ddl in tran** 为 **true**，则 **create table**、**grant** 和 **alter table** 等命令就允许用在该数据库的事务中。如果在 **model** 数据库中 **ddl in tran** 为 **true**，则在 **model** 中设置了 **ddl in tran** 之后所创建的所有数据库都允许在事务中使用这些命令。

◆ 警告!

数据定义语言命令持有系统表（如 **sysobjects**）锁。应避免在事务中使用它们；如果必须使用的话，要使事务尽量简短。

在事务中对 **tempdb** 使用数据定义语言命令，可能会使系统速度减慢以至于挂起。应始终将 **tempdb** 中的 **ddl in tran** 设置为 **false**。

- 表 7-16 列出了当 **ddl in tran** 选项设置为 **true** 时，可以在用户定义的事务中使用的命令：

表 7-16: 允许在事务中使用的 DDL 命令

alter table （允 许除 partition 和 unpartition 之外的子句）	create default create index create procedure create rule create schema create table create trigger create view	drop default drop index drop procedure drop rule drop table drop trigger drop view	grant revoke
---	---	--	-----------------

- 表 7-17 列出了在任何情况下，都不能在用户定义的事务中使用的命令：

表 7-17: 不允许在事务中使用的 DDL 命令

alter database	dump database	select into
alter table...partition	dump transaction	truncate table
alter table...unpartition	drop database	update statistics
create database	load transaction	
disk init	load database	

此外，创建临时表或更改 *master* 数据库的系统过程不能用在用户定义的事务中。

- **identity in nonunique index** 选项自动在表的索引键中包含一个 **IDENTITY** 列，以使在表上创建的所有索引都唯一。此数据库选项使逻辑上非唯一的索引成为内部唯一的，并使这些索引可用于处理对可更新游标和隔离级别 0 的读取。

表必须已经具有 **IDENTITY** 列，**identity in nonunique index** 选项才能工作。**IDENTITY** 列可用 **create table** 语句或在创建表之前将 **auto identity** 数据库选项设置为 **true** 来创建。

如果计划在具有非唯一索引的表上使用游标和隔离级别 0 读取，可使用 **identity in nonunique index**。唯一索引可确保下一次在游标处执行 **fetch** 时，游标定位于正确的行。如果打算在具有唯一的索引和任一隔离级别的表上使用游标，最好使用 **unique auto_identity index** 选项。

要得到某一含有 **IDENTITY** 列的表中索引的报告，执行 **sp_helpindex**。

- **no free space acctg** 选项禁止对非日志段进行可用空间会计和阈值操作。由于无需为这些段重新计算可用空间值，此选项可加速恢复过程。
- 当 **no chkpt on recovery** 选项被设置为 **true (on)** 时，将保留数据库的最新副本。在这些情况下，存在一个“主”数据库和一个“辅助”数据库。最初，主数据库被转储并被装载到辅助数据库中。然后，每隔一段时间，主数据库的事务日志被转储并被装载到辅助数据库中。

如果此选项设置为 **false (off)**（缺省条件），当重新启动 **Adaptive Server** 时，在数据库恢复之后会向数据库中添加一个检查点记录。此检查点可以确保恢复机制不会被不必要地再次运行，并可更改顺序号，从而使从主数据库中对事务日志进行的后续装载失败。

如果将辅助数据库的此选项设置为 **true (on)**，可以防止从恢复进程中取得检查点，从而保证主数据库中的后继事务日志转储能够被装载到其中。

- **read only** 选项表示用户可以从数据库检索数据，但是不能修改任何数据。
- 将 **select into/bulkcopy/pllsort** 选项设置为 **true (on)**，可以启用如下操作：**writetext**；对永久表执行 **select into**；向没有索引或触发器的表中进行快速批量复制；使用 **bcp** 或批量复制库例程以及并行排序。事务日志转储不能恢复这些进行最低程度记录的操作，因此禁止对转储设备执行 **dump transaction**。在完成了不记日志的操作之后，将 **select into/bulk copy/pllsort** 设置为 **false (off)** 并发出 **dump database**。

在用 **select into**、批量复制或并行排序对数据库作出了不记日志的更改之后，发出 **dump transaction** 命令会产生一条错误消息，告知您应使用 **dump database**。（**writetext** 命令没有这样的保护。）

由于 **tempdb** 从不恢复，因此要对临时表执行 **select into** 时无需将 **select into/bulkcopy/pllsort** 选项设置为 **true**。不必为了在带有索引的表上运行 **bcp** 而将该选项设置为 **true**，这是因为带有索引的表总是采用较慢形式的批量复制进行复制，而且会记录日志。

- 当 **single user** 设置为 **true** 时，一次只有一个用户可以访问该数据库（单用户模式）。

不能在存储过程的用户数据库中或用户已打开的数据库中将 **single user** 设置为 **true**。不能把 **tempdb** 的 **single user** 设置为 **true**。

- **trunc log on chkpt** 选项表示如果事务日志有多于 50 行的已提交事务，则每当 **checkpoint** 检查进程（通常每分钟多于一次）时事务日志都会被截断（提交的事务将被删除）。但是，当数据库所有者手工运行 **checkpoint** 时，该日志**不会被截断**。在进行开发工作时，为阻止日志的增加可能需要将此选项打开。

当 **trunc log on chkpt** 选项打开时，由于不能使用截断的事务日志转储从介质故障中恢复，所以禁止对转储设备执行 **dump transaction**。如果发出 **dump transaction** 语句，则会产生错误消息，指示您使用 **dump database**。

- 当 **unique auto_identity index** 选项设置为 **true** 时，它向新表添加具有唯一非集群索引的 **IDENTITY** 列。缺省情况下，**IDENTITY** 列为 10 位数字的 **numeric** 数据类型，但可以使用 **size of auto identity column** 配置参数改变此缺省值。同 **auto identity** 一样，当您用 **select *** 语句选择了所有的列时，**IDENTITY** 列是不可见的。要检索它，必须在选择列表中明确提供此列的名称 **SYB_IDENTITY_COL**。

如果需要使用游标或具有非唯一索引的隔离级别 0 读取，使用 **identity in nonunique index** 选项。

尽管可以在 *tempdb* 中将 **unique auto_identity index** 设置为 **true**，但此选项并不被承认或使用，并且在这里创建的临时表不会自动包括具有唯一索引的 **IDENTITY** 列。

- 有关数据库选项的详细信息，参见 *系统管理指南*。

权限

只有系统管理员或数据库所有者可以执行带有参数的 **sp_dboption** 来更改数据库选项。别名为“数据库所有者”的用户不能执行 **sp_dboption** 来更改数据库选项。任何用户都可以执行不带参数的 **sp_dboption** 来查看数据库选项。

使用的表

master.dbo.sysdatabases、*master.dbo.sysmessages*、
master.dbo.sysprocesses、*sysobjects*

参见

命令	checkpoint、select
系统过程	sp_configure、sp_helpdb、sp_helpindex、sp_helpjoins
实用程序命令	bcp

sp_dbrecovery_order

功能

指定用户数据库恢复的顺序，并列出一个数据库或所有数据库的用户定义的恢复顺序。

语法

```
sp_dbrecovery_order  
    [database_name [, rec_order [, force]]]
```

参数

database_name — 为其指派恢复顺序的数据库名称，或要列出其用户定义的恢复顺序的数据库的名称。

rec_order — 恢复数据库的顺序。若 **rec_order** 为 **-1**，将从用户定义的恢复序列中删除指定的数据库。

force — 允许用户将数据库插入现有的恢复序列中，而不将该数据库放在序列末尾。

示例

1. sp_dbrecovery_order pubs2, 1

使 *pubs2* 数据库成为在系统失败后恢复的第一个用户数据库。

2. sp_dbrecovery_order pubs3, 3, force

将 *pubs3* 数据库插入到用户定义的恢复序列中的第三位。如果另一数据库原先在第三位，它将被移动到第四位，后面的所有数据库依次向后移动。

3. sp_dbrecovery_order pubs2, -1

将 *pubs2* 数据库从用户定义的恢复序列中删除。这样，具有用户定义的恢复顺序的所有数据库都恢复后才恢复 *pubs2*。

4. sp_dbrecovery_order

列出具有恢复顺序（通过 **sp_dbrecovery_order** 指派）的所有数据库的当前恢复顺序。

注释

- 只有在 *master* 数据库中，才能使用 **sp_dbrecovery_order** 来输入或修改用户指定的恢复顺序。可从任何数据库列出用户定义的数据库恢复顺序。
- 要更改数据库的用户定义的恢复位置，使用 **sp_dbrecovery_order** 将数据库从恢复序列中删除，然后使用 **sp_dbrecovery_order** 将其插入新位置。
- 系统数据库总是在用户数据库之前恢复。系统数据库和它们的恢复顺序是：
 - *master*
 - *model*
 - *tempdb*
 - *sybsystemdb*
 - *sybsecurity*
 - *sybsystemprocs*
- 如果没有通过 **sp_dbrecovery_order** 指派数据库的恢复顺序，则先恢复系统数据库，然后按用户数据库的数据库 ID 顺序对其进行恢复。
- 如果指定了 *database_name*，但没有给出 *rec_order*，**sp_dbrecovery_order** 将显示指定数据库的用户定义的恢复位置。
- 如果没有指定 *database_name*，**sp_dbrecovery_order** 将列出具有用户定义的恢复顺序的所有数据库的恢复顺序。
- 通过 **sp_dbrecovery_order** 指派的恢复顺序必须是连续的，从 1 开始，值之间不存在间隔。对指派了恢复顺序的第一个数据库，必须给 *rec_order* 赋值为 1。如果有三个数据库的恢复顺序分别被指派为 1、2 和 3，则不能将下一个数据库的恢复顺序指派为 5。

权限

只有系统管理员才可以执行 **sp_dbrecovery_order**。

使用的表

master..sysattributes

sp_dbremap

功能

强制 Adaptive Server 承认 **alter database** 所作的更改。只有在 Adaptive Server 消息指示要这样做时，才运行该过程。

语法

```
sp_dbremap dbname
```

参数

dbname — **alter database** 命令在其中被中断的数据库的名称。

示例

```
1. sp_dbremap sample_db
```

alter database 命令更改数据库 *sample_db*。该命令作出对 Adaptive Server 可见的更改。

注释

- 如果在处于转储过程中的数据库上发布的 **alter database** 被中断了，Adaptive Server 将显示一条消息指示用户执行 **sp_dbremap**。
- 在数据库或事务转储过程中对 *sysusages* 的任何更改，在转储完成之前都不会被复制到活动内存，以确保数据库映射在转储过程中不被更改。运行 **alter database** 将立即对磁盘上的系统表作出更改。在转储完成之前，不能更改内存中的分配。这就是 **alter database** 暂停的原因。

当执行 **sp_dbremap** 时，在转储过程完成之前必须等待。

- 如果指示您运行 **sp_dbremap**，但是您却没有这样做，则在下一次重新启动前，Adaptive Server 不能使用利用 **alter database** 所分配的空间。

权限

只有系统管理员或数据库所有者才能执行 **sp_dbremap**。

使用的表

master.db.sysdatabases、*sysobjects*

参见

命令	alter database、dump database、dump transaction
-----------	--

sp_defaultloc

(仅适用于组件集成服务)

功能

为本地数据库中的对象定义一个缺省存储位置。

语法

```
sp_defaultloc dbname, {"defaultloc"| NULL}  
[, "defaulttype"]
```

参数

dbname — 要映射到远程存储位置上的数据库的名称。该数据库必须已用 **create database** 语句进行了定义。不能将系统数据库映射到远程位置。

defaultloc — 是数据库要映射到的远程存储位置。要指示服务器删除某一数据库现有的缺省映射，将此参数设为空值。 *defaultloc* 的值必须以句点 (.) 结束，如下所示：

```
server.dbname.owner.
```

defaulttype — 用于规定 *object_loc* 所指定的对象名称的格式。表 7-18 对有效值进行了说明。用引号将 *defaulttype* 值引起来。

表 7-18: defaulttype 允许使用的值

值	说明
table	表明由 <i>object_loc</i> 指定的对象是远程服务器可以访问的表。该值是 <i>defaulttype</i> 的缺省值。
view	表明由 <i>object_loc</i> 指定的对象是由远程服务器进行管理并被当作表来处理的视图。
rpc	表明由 <i>object_loc</i> 指定的对象是由远程服务器管理的 RPC；将 RPC 的结果集作为只读表来处理。

示例

1. **sp_defaultloc pubs, "SYBASE.pubs.dbo.", "table"**
create table pubs.dbo.book1 (bridges char(15))

sp_defaultloc 在名为 SYBASE 的远程服务器中定义远程存储位置 *pubs.dbo*。它将数据库 *pubs* 映射到远程位置。“**create table book1**”语句将在远程位置创建一个名为 *book1* 的表。用于 *bookN* 的 **create existing table** 语句要求在远程位置已经存在了 *pubs.dbo.bookN*，而表 *bookN* 的信息要存储到本地表 *bookN* 中。

2. **sp_defaultloc pubs, NULL**

删除数据库 *pubs* 到远程位置的映射。

3. **sp_defaultloc ticktape, "wallst.nasdaq.dbo.", "rpc"**
create existing table sybase (bestbuy integer)

标识出远程存储位置 *wallst.nasdaq.dbo*，其中“*wallst*”是为 *server_name* 提供的值，“*nasdaq*”是为 *database* 提供的值，而“*dbo*”则是为 *owner* 提供的值。RPC *sybase* 必须已经存在于远程位置。“**create existing table sybase**”语句将把有关 RPC *sybase* 的结果集的信息存储到本地表 *ticktape* 中。RPC *sybase* 结果集被视为只读表。不支持对 RPC 进行插入、更新和删除。

注释

- **sp_defaultloc** 为本地数据库中的表定义一个缺省存储位置。它将数据库中的表名映射到远程位置。它允许用户为整个数据库建立一个缺省值，而不是在每个 **create table** 和 **create existing table** 命令之前发出 **sp_addobjectdef** 命令。
- 当 *defaulttype* 是 *table*、*view* 或 *rpc* 时，*defaulttype* 参数格式如下：
server_name.dbname.owner.
 - 注意 *defaultloc* 规范必须以句点 (.) 结束。
 - *server_name* 代表已经由 **sp_addserver** 添加到 *sys.servers* 中的服务器。*server_name* 参数是必需的。
 - *dbname* 不一定是必需的。某些服务器类不支持它。
 - 应该始终提供 *owner* 以避免歧义。如果没有提供，那么实际引用的远程对象可能会不同，这取决于外部登录是否与远程对象所有者相对应。
- 在任一 **create table** 或 **create existing table** 语句之前发布 **sp_defaultloc**。当使用了任一种语句时，服务器将使用 *sysattributes* 表来确定是否已经为将要创建或定义的对象指定了任何表映射。如果已经指定了映射，**create table** 语句将在 *object_loc* 所指定的位置创建该表。**create existing table** 语句将关于现有的远程对象的信息存储到本地表中。

- 如果在 *defaulttype* view 上发布了 **sp_defaultloc**，然后又发布了 **create table**，组件集成服务将在远程服务器上创建一个新表，而不是一个视图。
- 更改数据库的缺省位置不会影响先前已经映射到不同缺省位置的表。
- 在数据库中创建表之后，以后所有对 *dbname* 中的表的引用（通过 **select**、**insert**、**delete** 和 **update**）都将映射到正确的位置。

权限

任何用户都可以执行 **sp_defaultloc**。

使用的表

master.dbo.sys.servers、*master.dbo.spt_values*、*master.dbo.sysattributes*

参见

命令	create existing table、create table
系统过程	sp_addobjectdef、sp_addserver、sp_help、sp_helpserver

sp_depends

功能

显示有关数据库对象依赖性的信息，这些对象包括：数据库中依赖于指定的表或视图的视图、触发器和过程，以及数据库中指定的视图、触发器或过程所依赖的表和视图。

语法

`sp_depends objname`

参数

objname — 是要对其进行依赖性检查的表、视图、存储过程或触发器的名称。不能指定数据库名。如果对象所有者不是运行该命令的用户并且不是数据库所有者，请使用所有者的名称。

示例

1. sp_depends sysobjects

列出依赖于表 *sysobjects* 的数据库对象。

2. sp_depends titleview

Things that the object references in the current database.

object	type	updated	selected
-----	-----	-----	-----
dbo.authors	user table	no	no
dbo.titleauthor	user table	no	no
dbo.titles	user table	no	no

Things inside the current database that reference the object.

object	type
-----	-----
dbo.tview2	view

列出了依赖于 *titleview* 视图的数据库对象，以及 *titleview* 视图所依赖的数据库对象。

3. sp_depends "mary.titles"

列出了依赖于用户 “Mary” 所拥有的 *titles* 表的数据库对象。由于句点是一种特殊的字符，所以需要引号。

注释

- 执行 **sp_depends** 将列出当前数据库中依赖于 **objname** 的对象以及 **objname** 所依赖的所有对象。例如，视图依赖于一个或多个表，并且可以拥有依赖于它们的过程或其它视图。引用了另一对象的对象就会依赖于那个对象。对当前数据库之外的对象的引用不包括在报告中。
- 通过查看 **sysdepends** 表，**sp_depends** 过程可以确定依赖性。
如果不按顺序地创建对象（例如，如果使用视图的过程在创建该视图之前被创建了），那么在 **sysdepends** 中就不存在用于依赖性的行，而且 **sp_depends** 不会报告这些依赖性。
- 如果所报告的对象是存储过程或触发器的话，**sp_depends** 报告中的 **updated** 和 **selected** 列是有意义的。**updated** 列的值指示存储过程是否会更新该对象。**selected** 列指出该对象是否用于读游标或数据修改语句。
- **sp_depends** 要遵循查找对象的这些 Adaptive Server 规则：
 - 如果用户没有指定所有者的姓名，并且执行该命令的用户拥有一个有指定名称的对象，则使用这个对象。
 - 如果用户没有指定所有者的姓名，并且用户不拥有一个具有这一名称的对象，但是数据库所有者却拥有这样的对象，则使用数据库所有者的对象。
 - 如果用户和数据库所有者都不拥有具有这一名称的对象，即使在数据库中存在一个具有此对象名但属于其他所有者的对象，该命令仍然会报告有一个错误条件。
 - 如果用户和数据库所有者都拥有具有指定名称的对象，并且用户想访问数据库所有者的对象，就必须指定名称，如 **dbo.objectname** 中所示。
- 由数据库用户、而不是由执行命令的用户和数据库所有者所拥有的对象，必须用所有者的名称来限定，如例 3 中所示。

权限

任何用户都可以执行 **sp_depends**。

使用的表

master.dbo.spt_values、 *master.dbo.sysmessages*、 *sysdepends*、
sysobjects、 *sysusers*

参见

命令	create procedure、 create table、 create view、 execute
系统过程	sp_help

sp_deviceattr

功能

更改现有数据库设备文件的 **dsync** 设置。

语法

```
sp_deviceattr logicalname, optname, optvalue
```

参数

logicalname — 是现有数据库设备的名称。该设备既可以存储在操作系统文件上，也可以存储在原始分区上，但是对于原始分区而言 **dsync** 设置将被忽略。

optname — 是要更改的设置的名称。当前，**optname** 可以接受的值只有 **dsync**。

optvalue — 既可以为 “true” 也可以为 “false”。

示例

```
1. sp_deviceattr file_device1, dsync, true
```

把名为 “file_device1” 设备的 **dsync** 设置为 “on”。

注释

- 对于存储在 UNIX 文件上的数据库设备而言，**dsync** 决定了对设备的更新是直接存储在存储介质上进行的，还是由 UNIX 文件系统进行了缓冲。

当 **dsync** 打开时，对数据库设备的写入可以直接在物理存储介质上进行，并且 **Adaptive Server** 能在系统出现故障时恢复设备上的数据。

当 **dsync** 关闭时，UNIX 文件系统可以将对数据库设备的写入放在缓冲中。即使物理介质尚未被修改，UNIX 文件系统也可能将更新过程标记为已完成。在系统出现故障时，无法确保物理介质上已作出了数据更新的请求，并且 **Adaptive Server** 可能无法恢复数据库。

- 在使用 **sp_deviceattr** 更改了 **dsync** 设置之后，在更改生效前必须重新启动 **Adaptive Server**。
- 对于主设备文件，**dsync** 总是打开的。不能用 **sp_deviceattr** 更改主设备文件的 **dsync** 设置。

- 只有当出现系统故障后不需要恢复设备上的数据库时，才可以将 **dsync** 值关闭。例如，对于只存储 *tempdb* 数据库的设备，可以考虑关闭 **dsync**。
- Adaptive Server 会忽略存储在原始分区上的设备的 **dsync** 设置 — 不管 **dsync** 设置如何，对那些设备的更新都不会进行缓冲。
- **dsync** 不能在 Windows NT 平台上使用。

权限

执行 **sp_deviceattr** 的用户必须拥有更新 *sysdevices* 表的权限。

使用的表

sysdevices

参见

系统过程	sp_helpdevice
------	---------------

sp_diskdefault

功能

如果用户没有指定数据库设备，或是用 **create database** 或 **alter database** 命令指定了 **default**，则指定数据库设备是否可以用于数据存储。

语法

```
sp_diskdefault logicalname, {defaulton | defaultoff}
```

参数

logicalname — 是 `master.dbo.sysdevices.name` 中所给出的设备的逻辑名。该设备必须是一个数据库设备，而不能是转储设备。

defaulton | **defaultoff** — **defaulton** 指定该数据库设备为缺省的数据库设备；**defaultoff** 指出所指定的数据库设备不是缺省的数据库设备。

在用 **disk init** 将数据库设备添加到系统之后，请使用 **defaulton**。请使用 **defaultoff** 来更改主设备（在第一次安装 Adaptive Server 时，被指定为缺省设备的设备）的缺省状态。

示例

1. **sp_diskdefault master, defaultoff**

对于数据库的缺省存储而言，主设备不能再被 **create database** 或 **alter database** 使用。

注释

- 缺省的数据库设备是指在用户没有指定数据库设备名或指定了关键字 **default** 的情况下，被 **create database** 或 **alter database** 用于数据存储的设备。
- 可以有多个缺省设备。它们将按照在 `master.dbo.sysdevices` 表中出现的顺序（即，字母顺序）被使用。当第一个缺省设备被填满后，将使用第二个缺省设备，等等。
- 在第一次安装 Adaptive Server 时，主设备是唯一的缺省数据库设备。

► **注意**

一旦设备被初始化用来存储用户数据库，请使用 **sp_diskdefault** 关闭主设备的缺省状态。这可以阻止用户在主设备上意外地创建数据库，并简化 *master* 数据库的恢复。

- 要找出那些数据库设备是缺省的数据库设备，请执行 **sp_helpdevice**。

权限

只有系统管理员才可以执行 **sp_diskdefault**。

使用的表

master.dbo.sysdevices、*sysobjects*

参见

命令	alter database、create database、disk init
系统过程	sp_helpdevice

sp_displayaudit

功能

显示审计选项的状态。

语法

```
sp_displayaudit ["procedure" | "object" | "login" |  
"database" | "global" | "default_object" |  
"default_procedure" [, "name"]]
```

参数

procedure — 显示所指定的存储过程或触发器的审计选项的状态。如果没有指定 *name* 值，**sp_displayaudit** 将显示当前数据库中所有过程和触发器的活动审计选项。

object — 显示所指定的表或视图的审计选项的状态。如果没有指定 *name* 值，**sp_displayaudit** 将显示当前数据库中所有表和视图的活动审计选项。

login — 显示所指定的用户登录的审计选项的状态。如果没有指定 *name* 值，**sp_displayaudit** 将显示 *master* 数据库中所有登录的活动审计选项。

database — 显示所指定的数据库的审计选项的状态。如果没有指定 *name* 值，**sp_displayaudit** 将显示服务器上的所有数据库的活动审计选项。

global — 显示所指定的全局审计选项的状态。如果没有指定 *name* 值，**sp_displayaudit** 将显示当前数据库中所有过程和触发器的活动审计选项。

default_object — 显示缺省的审计选项，这些审计选项将用于在所指定的数据库上创建的任何新表或视图。如果没有指定 *name* 值，**sp_displayaudit** 将为所有具有活动的缺省审计设置的数据库显示缺省审计选项。

default_procedure — 显示缺省的审计选项，这些审计选项将用于在所指定的数据库上创建的任何新过程或触发器。如果没有指定 *name* 值，**sp_displayaudit** 将为所有具有活动的缺省审计设置的数据库显示缺省审计选项。

name — 是所指定的参数的信息，如下表所说明的一样：

参数	<i>name</i> 值
procedure	过程或触发器名
object	表或视图名
login	用户登录
database	数据库名
global	全局审计选项
default_object	数据库名
default_procedure	数据库名

示例

1. `sp_displayaudit`

```
Procedure/Trigger      Audit Option  Value Database
-----
dbo.sp_altermessages  exec_procedure on      sybsystemprocs
dbo.sp_help            exec_procedure on      sybsystemprocs
dbo.sp_who             exec_procedure on      sybsystemprocs
No databases currently have default sproc/trigger auditing
enabled.
No objects currently have auditing enabled.
No databases currently have default table/view auditing enabled.
No logins currently have auditing enabled.
No databases currently have auditing enabled.
```

Option Name	Value
adhoc	off
dbcc	off
disk	off
errors	off
login	off
logout	off
navigator_role	off
oper_role	off
replication_role	off
rpc	off
sa_role	off
security	off
sso_role	off

当未指定参数时，将显示每个类别的状态和所有的审计选项。

2. sp_displayaudit "procedure"

Procedure/Trigger	Audit Option	Value	Database
-----	-----	----	-----
dbo.sp_altermessage	exec_procedure	on	sybsystemprocs
dbo.sp_help	exec_procedure	on	sybsystemprocs
dbo.sp_who	exec_procedure	on	sybsystemprocs

当未指定过程名时，将显示所有过程审计选项的状态。

3. sp_displayaudit "procedure", "sp_who"

Procedure/Trigger	Audit Option	Value	Database
-----	-----	----	-----
dbo.sp_who	exec_procedure	on	sybsystemprocs

当指定了过程的名称时，将只显示该过程的状态。

4. sp_displayaudit "global"

Option Name	Value
-----	-----
adhoc	off
dbcc	off
disk	off
errors	off
login	off
logout	off
navigator_role	off
oper_role	off
replication_role	off
rpc	off
sa_role	off
security	off
sso_role	off

当未指定全局审计选项时，将显示所有全局审计选项的状态。

注释

- **sp_displayaudit** 显示审计选项的状态。
- 下表显示了每一个参数的有效审计选项：

对象类型参数	有效审计选项
procedure	exec_procedure、 exec_trigger
object	delete、 func_obj_access、 insert、 reference、 select、 update
login	all、 cmdtext、 table_access、 view_access
database	alter、 bcp、 bind、 create、 dbaccess、 drop、 dump、 func_dbaccess、 grant、 load、 revoke、 setuser、 truncate、 unbind
global	adhoc、 dbcc、 disk、 errors、 login、 logout、 navigator_role、 oper_role、 replication_role、 rpc、 sa_role、 security、 sso_role
default_object	delete、 func_obj_access、 insert、 reference、 select、 update
default_procedure	exec_procedure、 exec_trigger

- 不能指定名称的值，除非首先指定了对象类型参数。
- 有关设置审计的信息，参见 *系统管理指南*。

权限

只有系统安全员才能执行 **sp_displayaudit**。

使用的表

sysauditoptions、 *sysdatabases*、 *syslogins*、 *sysobjects*

参见

系统过程	sp_audit
实用程序命令	bcp

sp_displaylevel

功能

设置或显示哪些 Adaptive Server 配置参数会出现在 **sp_configure** 输出中。

语法

```
sp_displaylevel [loginame [, level]]
```

参数

loginame — 是要为之设置或显示显示级别的用户 Adaptive Server 登录。

level — 将显示级别设置为下列之一：

“**basic**” 显示级别只显示最基本的配置参数。这适于非常一般的服务器调优。

“**intermediate**” 显示级别既显示基本级别的参数，也显示稍微复杂些的配置参数。此级别适用于中等复杂度的服务器调优。

“**comprehensive**” 显示级别显示所有的配置参数，包括最复杂的那些参数。这一级别适于非常细节化的服务器调优。

示例

1. **sp_displaylevel**

```
The current display level for login 'sa' is  
'comprehensive'.
```

显示调用 **sp_displaylevel** 的用户的当前显示级别。

2. **sp_displaylevel jerry**

```
The current display level for login 'jerry' is  
'intermediate'.
```

显示用户 “jerry” 的当前显示级别。

3. **sp_displaylevel jerry, comprehensive**

```
The display level for login 'jerry' has been  
changed to 'comprehensive'.
```

为用户 “jerry” 将显示级别设置为 “comprehensive”。

权限

只有系统管理员可以执行 **sp_displaylevel** 来为另一用户设置显示级别。任何用户都可以执行 **sp_displaylevel** 来设置和显示他或她自己的显示级别。

使用的表

master..sysattributes

参见

系统过程	sp_configure
------	--------------

sp_displaylogin

功能

显示有关登录帐号的信息。如果指定了该选项，将同时显示该登录帐号之上或之下的层次树信息。

语法

```
sp_displaylogin [loginame [, expand_up | expand_down]]
```

参数

loginame — 是您要得到其信息的用户登录帐号（如果它不是您自己的）。只有系统安全员或系统管理员才能获得其他人的登录帐号的信息。

expand_up — 指定 Adaptive Server 显示含有 *loginame* 角色的角色层次中的所有角色。

expand_down — 指定 Adaptive Server 显示 *loginame* 角色所包含的角色层次中的所有角色。

示例

1. sp_displaylogin

```
Suid: 1
Loginame: sa
Fullname:
Default Database: master
Default Language:
Configured Authorization:
    sa_role (default ON)
    sso_role (default ON)
    oper_role (default ON)
Locked: NO
Date of Last Password Change: Nov 16 1994 10:08AM
```

显示有关您的服务器登录帐号的信息。

2. sp_displaylogin susanne

```
Suid: 12
Loginame: susanne
Fullname:
Default Database: pubs2
Default Language:
Configured Authorization:
    supervisor (default OFF)
Locked: NO
Date of Last Password Change: May 12 1997 11:09AM
```

显示有关登录帐号“susanne”的信息。根据执行 **sp_displaylogin** 的用户角色，所显示的信息将有所不同。

3. sp_displaylogin pillai, expand_up

显示登录帐号为“pillai”角色的所有角色的信息。根据执行 **sp_displaylogin** 的用户角色，所显示的信息将有所不同。

注释

- **sp_displaylogin** 显示配置的角色，因此即使用 **set** 命令使某一角色成为不活动的，仍会显示它。
- 当使用 **sp_displaylogin** 来获得有关您自己的帐号的信息时，不必使用 *loginame* 参数。**sp_displaylogin** 显示您的服务器用户 ID、登录名、全名、所有授予您的角色、口令更改的最后日期、缺省语言以及您的帐号是否被锁定。
- 如果是系统安全员或系统管理员，则可以使用 *loginame* 参数访问有关任何帐号的信息。

权限

只有系统管理员或系统安全员可以执行带有 *loginame* 和 **expand** 参数的 **sp_displaylogin**，来获得其它用户的登录帐号的有关信息。任何用户都可以执行 **sp_displaylogin** 来获得有关他或她自己的登录帐号的信息。

使用的表

master.dbo.sysloginroles、*master.dbo.syslogins*、*master.dbo.syssrvroles*、*sysobjects*

参见

存储过程	sp_activeroles 、 sp_displayroles 、 sp_helprotect 、 sp_modifylogin
------	---

sp_displayroles

功能

显示授权给另一角色的所有角色，或者以表的格式显示整个角色层次树。

语法

```
sp_displayroles [grantee_name [, mode]]
```

参数

grantee_name — 是要获得其角色的有关信息的用户登录名，或者是要获得其信息的角色名。

mode — 是 **expand_up** 或 **expand_down**。 **expand_up** 显示父层次的角色层次树，而 **expand_down** 则显示了子层次的角色层次树。

示例

1. **sp_displayroles**

```
Role Name
-----
supervisor_role
```

显示授予发布该命令的用户的所有角色。

2. **sp_displayroles "supervisor_role"**

```
Role Name
-----
clerk
```

显示授予 *supervisor_role* 的所有角色。

3. **sp_displayroles susanne, expand_down**

Role Name	Parent Role Name	Level
supervisor_role	NULL	1
clerk_role	supervisor_role	2

显示授予登录 “susanne” 的活动角色，以及在层次中位于其下的角色。

4. **sp_displayroles "intern_role", expand_up**

显示授予 *intern_role* 的活动角色，以及在层次中位于其上的角色。

注释

- 当指定了可选的 **expand_up** 或 **expand_down** 参数时，将显示所有直接授予的角色，这些角色都是为所指定的角色所包含的或者含有所指定的角色。

权限

只有系统管理员或系统安全员可以执行 **sp_displayroles** 来显示由任何其它用户激活的角色的信息。任何用户都可以执行 **sp_displayroles** 来查看他或她自己的活动角色。

使用的表

master.dbo.sysattributes、*master.dbo.syssrvroles*、*master.dbo.syslogins*、*master.dbo.sysloginroles*

参见

命令	alter role、create role、drop role、grant、revoke、set
系统过程	sp_activeroles、sp_displaylogin、sp_helprotect、sp_modifylogin

sp_dropalias

功能

删除用 **sp_addalias** 所建立的别名用户名标识。

语法

sp_dropalias loginame

参数

loginame 一是 *master.dbo.syslogins* 中的用户名，该名称是另一用户的别名。

示例

1. sp_dropalias victoria

假定 “victoria” 在当前数据库中是别名（例如，是数据库所有者的别名），该语句将 “victoria” 作为一个用户别名从数据库中删除。

注释

- 执行 **sp_dropalias** 过程将从 *sysalternates* 表中删除用户的替代 *suid* 映射。
- 删除了用户的别名后，用户不再能够访问为其创建过别名的数据库。

权限

只有数据库所有者或系统管理员可以执行 **sp_dropalias**。

使用的表

sysalternates、*sysobjects*

参见

命令	use
系统过程	sp_addalias、sp_adduser、sp_changedbowner、sp_droplogin、sp_dropuser、sp_helpuser

sp_drop_all_qplans

功能

删除抽象计划组中所有的抽象计划。

语法

```
sp_drop_all_qplans name
```

参数

name — 是要从中删除所有计划的抽象计划组的名称。

示例

```
1. sp_drop_all_qplans dev_test
```

注释

- 要删除各个计划，可以使用 **sp_drp_qplan**。
- 要查看当前数据库中的抽象计划组的名称，可以使用 **sp_help_qpgroup**。
- **sp_drop_all_qplans** 自动删除组中属于指定用户的所有计划，或者是组中的所有计划（如果该命令是由系统管理员或数据库所有者来执行）。

权限

任何用户都可以执行 **sp_drop_all_qplans** 来删除其拥有的计划。只有系统管理员或数据库所有者才能删除其他用户所拥有的计划。

使用的表

```
sysattributes、 sysqueryplans
```

参见

系统过程	sp_drop_qplan、 sp_help_qpgroup
------	--------------------------------

sp_dropdevice

功能

删除 Adaptive Server 数据库设备或转储设备。

语法

```
sp_dropdevice logicalname
```

参数

logicalname — 是 *master.dbo.sysdevices.name* 中所列出的设备的名称。

示例

```
1. sp_dropdevice tape5
```

从 Adaptive Server 中删除名为 *tape5* 的设备。

```
2. sp_dropdevice fredsddata
```

从 Adaptive Server 中删除名为 *fredsddata* 的数据库设备。此时该设备一定不能被任何数据库使用。

注释

- **sp_dropdevice** 过程从 Adaptive Server 中删除某一设备，并从 *master.dbo.sysdevices* 中删除该设备条目。
- **sp_dropdevice** 不会删除被当作数据库设备而删除的文件；它使该文件不能为 Adaptive Server 所访问。在使用了 **sp_dropdevice** 之后，请使用操作系统命令来删除文件。

权限

只有系统管理员才能执行 **sp_dropdevice**。

使用的表

master.dbo.sysdatabases、 *master.dbo.sysdevices*、 *master.dbo.sysusages*、 *sysobjects*

参见

命令	drop database
系统过程	sp_addumpdevice、 sp_helpdb、 sp_helpdevice

sp_dropengine

功能

从指定的引擎组中删除一个引擎；如果该引擎是组中的最后一个引擎，则删除该引擎组。

语法

```
sp_dropengine engine_number, engine_group
```

参数

engine_number — 是要从组中删除的引擎数。该值介于零和配置的联机引擎数减一之间，最大值可等于配置的联机引擎数减一。

engine_group — 是要从中删除引擎的引擎组的名称。

示例

```
1. sp_dropengine 2, DS_GROUP
```

该语句从名为 *DS_GROUP* 的组中删除引擎 *2*。如果它是组中的最后一个引擎，还将删除该组。

注释

- `sp_dropengine` 只能从 *master* 数据库中调用。
- 如果 *engine_number* 是 *engine_group* 中的最后一个引擎，Adaptive Server 还会删除 *engine_group*。
- 指定的 *engine_number* 必须位于 *engine_group* 中。

权限

只有系统管理员才能执行 `sp_dropengine`。

使用的表

sysattributes

参见

系统过程	sp_addengine、sp_addexeclass、sp_bindexeclass、sp_clearpsex、sp_dropexeclass、sp_setpsex、sp_showcontrolinfo、sp_showexeclass、sp_showpsex
------	--

sp_dropexeclass

功能

删除用户定义的执行类。

语法

```
sp_dropexeclass classname
```

参数

classname — 是要删除的用户定义的执行类名称。

示例

```
1. sp_dropexeclass 'DECISION'
```

该语句删除了用户定义的执行类 *DECISION*。

注释

- 执行类帮助定义 Adaptive Server 处理任务时的执行优先次序。有关执行类和执行特性的详细信息，参见 *Performance and Tuning Guide*。
- 不要将 *classname* 绑定到任何客户端应用、登录或存储过程。首先使用 **sp_unbindexeclass** 对执行类解除绑定，然后使用 **sp_dropexeclass** 删除该执行类。
- 不能删除系统定义的执行类。

权限

只有系统管理员才能执行 **sp_dropexeclass**。

使用的表

sysattributes

参见

系统过程	sp_addengine、sp_addexeclass、 sp_bindexeclass、sp_clearpsex、 sp_dropengine、sp_setpsex、 sp_showcontrolinfo、sp_showexeclass、 sp_showpsex、sp_unbindexeclass
------	--

sp_dropextendedproc

功能

删除扩展存储过程 (ESP)。

语法

sp_dropextendedproc *esp_name*

参数

esp_name — 是要删除的扩展存储过程的名称。

示例

1. **sp_dropextendedproc xp_echo**

删除 *xp_echo*。

注释

- **sp_dropextendedproc** 必须从 *master* 数据库中执行。
- *esp_name* 区分大小写。它必须精确地匹配创建 ESP 所用的名称。

权限

只有系统管理员才能执行 **sp_dropextendedproc**。

使用的表

master.dbo.syscomments、*sysobjects*

参见

命令	drop procedure
系统过程	sp_addextendedproc、sp_freelibrary、 sp_helpextendedproc

sp_dropexternlogin

(仅适用于组件集成服务)

功能

删除以前使用 **sp_addexternlogin** 定义的远程登录定义。

语法

```
sp_dropexternlogin remote_server [, login_name]
```

参数

remote_server — 是本地服务器从中删除帐户访问的远程服务器的名称。通过 *master.dbo.sysservers* 表中的条目，**remote_server** 可以为本地服务器所知。

login_name — 是本地服务器所知的登录帐户。如果未指定 **login_name**，则使用当前帐户。**login_name** 必须存在于 *master.dbo.syslogins* 表中。

示例

1. **sp_dropexternlogin JOBSERV, sa**

从 **login_name** “sa” 删除对远程服务器 JOBSERV 的外部登录定义。

2. **sp_dropexternlogin CIS1012, bobj**

从 “bobj” 删除对远程服务器 CIS1012 的外部登录定义。只有 “bobj” 帐户和 “sa” 帐户可以添加或修改 “bobj” 的远程登录。

注释

- **sp_dropexternlogin** 删除以前通过 **sp_addexternlogin** 定义到本地服务器的远程登录定义。
- 不能从事务中执行 **sp_dropexternlogin**。
- 必须使用 **sp_addserver** 将 **remote_server** 定义到本地服务器。
- 要添加和删除本地服务器用户，可以使用系统过程 **sp_addlogin** 和 **sp_droplogin**。

权限

只有 **login_name** 或系统管理员才能执行 **sp_dropexternlogin**。

使用的表

master.dbo.sysattributes、 *master.dbo.sysservers*

参见

系统过程	sp_addlogin、 sp_addexternlogin、 sp_addobjectdef、 sp_addserver、 sp_droplogin、 sp_dropobjectdef、 sp_helpdb、 sp_helpserver
------	---

sp_dropglockpromote

功能

从表或数据库中删除锁升级值。

语法

```
sp_dropglockpromote {"database" | "table"}, objname
```

参数

database | table — 指定是否从数据库或表中删除锁升级阈值。由于这些是 Transact-SQL 关键字，所以需要使用引号。

objname — 是要从中删除锁升级阈值的表或数据库的名称。

示例

```
1. sp_dropglockpromote "table", titles
```

从 *titles* 删除锁升级值。 *titles* 的锁升级现在使用数据库或全服务器范围的值。

注释

- 使用 **sp_dropglockpromote** 来删除用 **sp_setpglockpromote** 设置的锁升级值。
- 删除数据库的锁升级阈值后，没有配置锁升级阈值的表将使用全服务器范围的值。
- 删除表的值后， **Adaptive Server** 将使用数据库的锁升级阈值（如果配置的话），或全服务器范围的值（如果没有配置的话）。
- 全服务器范围的值可以使用 **sp_setpglockpromote** 来更改，但是不能被删除。

权限

只有系统管理员才能执行 **sp_dropglockpromote**。

使用的表

master.dbo.sysattributes、 *sysobjects*

参见

系统过程	sp_configure、 sp_setpglockpromote
------	-----------------------------------

sp_dropgroup

功能

从数据库中删除组。

语法

sp_dropgroup *grpname*

参数

grpname — 是当前数据库中的组的名称。

示例

1. **sp_changegroup accounting, martha**
sp_changegroup "public", george
sp_dropgroup purchasing

“purchasing”组已经与“accounting”组合并了。这些命令在删除“purchasing”组之前，将该组成员“martha”和“george”移到其它组中。由于“public”是保留字，所以组名“public”用引号引了起来。

注释

- 如果执行 **sp_dropgroup**，将从数据库的 *sysusers* 表中删除组名。
- 不能删除带有成员的组。在删除该组之前，必须为每一个成员执行 **sp_changegroup**。

权限

只有数据库所有者、系统管理员或系统安全员才能执行 **sp_dropgroup**。

使用的表

master.dbo.syssrvroles、*sysobjects*、*sysprotects*、*sysusers*

参见

命令	grant、revoke、use
系统过程	sp_addgroup、sp_adduser、sp_changegroup、sp_dropuser、sp_helpgroup

sp_dropkey

功能

从 *syskeys* 表中删除使用 *sp_primarykey*、*sp_foreignkey* 或 *sp_commonkey* 定义的键。

语法

```
sp_dropkey keytype, tablename [, deptabname]
```

参数

keytype — 是要删除的键的类型。 *keytype* 必须是 **primary**、**foreign** 或 **common**。

tablename — 是键表或视图的名称，其中含有要删除的键。

deptabname — 指定关系中的第二个表的名称（如果 *keytype* 是 **foreign** 或 **common** 的话）。如果 *keytype* 是 **primary**，则不需要这一参数，这是因为 **primary** 键没有相关表。如果 *keytype* 是 **foreign**，则这就是主键表的名称。如果 *keytype* 是 **common**，请按照它与 *sp_helpkey* 一起出现的顺序给出这两个表名。

示例

1. **sp_dropkey primary, employees**

删除 *employees* 表的主键。任何依赖于 *employees* 的主键的外键也将被删除。

2. **sp_dropkey common, employees, projects**

删除 *employees* 和 *projects* 表之间的公用键。

3. **sp_dropkey foreign, titleauthor, titles**

删除 *titleauthor* 和 *titles* 表之间的外键。

注释

- 如果执行 *sp_dropkey*，将从 *syskeys* 中删除指定的键。只有表的所有者才能从该表中删除键。
- 创建键是为了使数据库设计中隐式的逻辑关系变为显式的。这一信息只能由应用程序使用。
- 删除某个主键将自动删除与之相关联的任何外键。而删除某一外键不会对该表中所指定的主键产生影响。

- 执行 **sp_commonkey**、**sp_primarykey** 或 **sp_foreignkey** 将向 *syskeys* 系统表添加键。要显示有关已经定义的键的报告，可以执行 **sp_helpkey**。

权限

只有 *tablename* 的所有者才能执行 **sp_dropkey**。

使用的表

syskeys、*sysobjects*

参见

系统过程	sp_commonkey 、 sp_foreignkey 、 sp_helpkey 、 sp_primarykey
------	---

sp_droplanguage

功能

从服务器中删除替代语言并从 *master.dbo.syslanguages* 中删除相关行。

语法

```
sp_droplanguage language [, dropmessages]
```

参数

- language* — 是要删除的语言的正式名称。
- dropmessages* — 删除用 *language* 表示的所有 Adaptive Server 系统消息。在没有删除系统消息的情况下，不能删除与之相关联的语言。

示例

- sp_droplanguage french**
如果没有相关联的消息，可用此命令从可用的替代语言中删除法语。
- sp_droplanguage french, dropmessages**
如果有相关联的消息，可用此命令从可用的替代语言中删除法语。

注释

- 执行 **sp_droplanguage** 可以通过从 *master.dbo.syslanguages* 表中删除某个语言条目，而将该语言从替代语言列表中删除。
- 如果要删除有系统消息的语言，除非提供了 **dropmessages** 参数，否则请求将失败。

权限

只有系统管理员才能执行 **sp_droplanguage**。

使用的表

master.dbo.syslanguages、*master.dbo.sysmessages*、*sysobjects*

参见

系统过程	sp_addlanguage、sp_helplanguage
------	--------------------------------

sp_droplogin

功能

通过从 *master.dbo.syslogins* 中删除用户条目，可以删除 Adaptive Server 用户登录。

语法

```
sp_droplogin loginame
```

参数

loginame — 是 *master.dbo.syslogins* 中所列的用户的名称。

示例

```
1. sp_droplogin victoria
```

从 Adaptive Server 删除 “victoria” 登录。

注释

- 如果执行 **sp_droplogin**，将从 Adaptive Server 中删除一个用户登录，并从 *master.dbo.syslogins* 中删除该用户的条目。
- 如果删除的登录所用服务器用户 ID 危及到责任，Adaptive Server 会重新使用它。应避免彻底删除帐户，请尽量使用 **sp_locklogin** 来锁定所有不再使用的帐户。
如果需要删除登录，一定要审计这些事件（使用 **sp_audit**）从而对它们进行记录。
- **sp_droplogin** 删除所有与所删除的登录相关联的资源限制。
- 如果要删除的登录是服务器上任意数据库中的用户，**sp_droplogin** 将会失败。使用 **sp_dropuser** 删除某个数据库的用户。如果此用户在该数据库中拥有任何对象，则不能从数据库删除此用户。
- 如果要删除的登录是系统安全员，**sp_droplogin** 将验证是否存在另外一个已解锁的系统安全员帐户。如果不存在，**sp_droplogin** 将失败。类似地，**sp_droplogin** 将确保至少存在一个已解锁的系统管理员帐户。

权限

只有系统管理员或系统安全员才能执行 **sp_droplogin**。

使用的表

*master.dbo.sysloginroles、 master.dbo.syslogins、
master.dbo.sysprocesses、 sysobjects*

参见

系统过程	sp_addlogin、 sp_audit、 sp_changedbowner、 sp_dropalias、 sp_dropuser、 sp_helpuser、 sp_locklogin
------	---

sp_dropmessage

功能

从 *sysusermessages* 删除用户定义的消息。

语法

sp_dropmessage message_num [, language]

参数

message_num — 要删除的消息的编号。消息编号的值必须大于或等于 20000。

language — 是要删除的消息的语言。

示例

1. sp_dropmessage 20002, french

从 *sysusermessages* 删除编号为 20002 的消息的法语版本。

注释

- language* 参数是可选的。如果包含该参数，则只有用指定语言表示的、具有指定的 *message_num* 的消息才会被删除。如果未指定 *language*，则所有具有指定的 *message_num* 的消息都会被删除。

权限

只有数据库所有者、系统管理员或要删除的消息的创建者才能执行 **sp_dropmessage**。

使用的表

master.dbo.syslanguages、*sysobjects*、*sysusermessages*

参见

系统过程	sp_addmessage、sp_getmessage
------	-----------------------------

sp_drop_qpgroup

功能

删除抽象计划组。

语法

```
sp_drop_qpgroup group
```

参数

group — 是要删除的抽象计划组的名称。

示例

```
1. sp_drop_qpgroup dev_test
```

注释

- 不能删除缺省组 *ap_stdin* 和 *ap_stdout*。
- 不能删除含有计划的组。要删除组中的所有计划，可以使用 **sp_drop_all_qplans**。要查看组的列表及其含有的计划数，可以使用 **sp_help_qpgroup**。
- **sp_drop_qpgroup** 不能在事务中运行。

权限

只有系统管理员或数据库所有者才能执行 **sp_drop_qpgroup**。

使用的表

sysattributes、*sysqueryplans*

参见

系统过程	sp_drop_all_qplans、sp_help_qpgroup
------	------------------------------------

sp_drop_qplan

功能

删除抽象计划。

语法

```
sp_drop_qplan id
```

参数

id — 是要删除的抽象计划的 ID。

示例

```
1. sp_drop_qplan 1760009301
```

将删除具有指定 ID 的抽象计划。

注释

- 要查找某一计划的 ID，可以使用 `sp_help_qpgroup`、`sp_help_qplan` 或 `sp_find_qplan`。计划 ID 也可以由 `create plan` 返回并包含在 `showplan` 输出中。
- 要删除组中的所有抽象计划，可以使用 `sp_drop_all_qplans`。

权限

任何用户都可以执行 `sp_drop_qplan` 来删除其拥有的计划。只有系统管理员或数据库所有者才能删除其他用户所拥有的计划。

使用的表

sysqueryplans

参见

命令	create plan
系统过程	sp_drop_all_qplans、sp_find_qplan、 sp_help_qpgroup、sp_help_qplan

sp_dropobjectdef

(仅适用于组件集成服务)

功能

删除为本地对象提供的外部存储映射。

语法

```
sp_dropobjectdef "object_name"
```

参数

object_name 以 *dbname.owner.object* 形式表示，其中：

dbname 是数据库的名称，该数据库含有要删除其存储位置的對象。*dbname* 是可选的；如果有，它必须是当前数据库名，并且需要 *owner* 或占位符。

owner 是所有者的名称，他是要删除其存储位置的對象的所有者。*owner* 是可选的；如果指定了 *dbname*，就需要有该项。

object 是要为其删除外部存储映射的本地表的名称。

示例

1. **sp_dropobjectdef "personnel.dbo.colleges"**

从 *sysattributes* 中删除某个条目，该条目为服务器已知的表（*personnel* 数据库中的 *colleges* 表）提供外部存储映射。

2. **sp_dropobjectdef "andrea.fishbone"**

从 *sysattributes* 中删除某个条目，该条目为 *andrea.fishbone* 对象提供外部存储映射，其中：*andrea* 是 *owner*，本地表名是 *fishbone*。

注释

- **sp_dropobjectdef** 删除为本地对象提供的外部存储映射。它替代了 **sp_droptabledef**。
- 在用 **drop table** 删除了远程表之后，请使用 **sp_dropobjectdef**。
- 如果映射信息是用 **sp_addobjectdef** 添加的，则删除表将不会删除 *sysattributes* 表中的映射信息。必须使用 **sp_dropobjectdef** 显式地删除它。

- *object_name* 可以是以下形式之一：
 - *object*
 - *owner.object*
 - *dbname..object*
 - *dbname.owner.object*

权限

只有数据库所有者或系统管理员才能执行 **sp_dropobjectdef**。只有系统管理员才能执行 **sp_dropobjectdef** 来删除另一用户的对象的映射信息。

使用的表

sysobjects、*sysattributes*、*sysusers*

参见

命令	create existing table、create table、drop table
系统过程	sp_addobjectdef

sp_dropremotelogin

功能

删除远程用户登录。

语法

```
sp_dropremotelogin remoteserver [, loginame  
    [, remotename] ]
```

参数

remoteserver — 是具有要删除的远程登录的服务器的名称。

loginame — 是与 **sysremotelogins** 表中远程服务器相关联的本地服务器的用户名。

remotename — 是在从远程服务器登录时，被映射为 **loginame** 的远程用户名。

示例

1. sp_dropremotelogin GATEWAY

删除名为 GATEWAY 的远程服务器的条目。

2. sp_dropremotelogin GATEWAY, churchy

删除某个条目，该条目将远程服务器 GATEWAY 的远程登录映射到名为 “churchy” 的本地用户。

3. sp_dropremotelogin GATEWAY, churchy, pogo

删除远程服务器 GATEWAY 上远程用户 “pogo” 的登录，该登录被映射到名为 “churchy” 的本地用户。

注释

- 如果执行 **sp_dropremotelogin**，将从远程服务器中删除一个用户登录，并从 **master.dbo.sysremotelogins** 中删除该用户的条目。
- 有关远程登录的详细论述，参见 **sp_addremotelogin**。
- 要添加或删除本地服务器用户，可以分别使用系统过程 **sp_addlogin** 和 **sp_droplogin**。

权限

只有系统管理员才能执行 **sp_dropremotelogin**。

使用的表

master.dbo.sysremotelogins、 *master.dbo.sysservers*、 *sysobjects*

参见

系统过程	sp_addlogin、 sp_addremotelogin、 sp_addserver、 sp_droplogin、 sp_helpremotelogin、 sp_helpserver
------	---

sp_drop_resource_limit

功能

从 Adaptive Server 中删除一个或多个资源限制。

语法

```
sp_drop_resource_limit {name, appname }
                        [, rangename, limittype, enforced, action, scope]
```

参数

- name* — 是适用限制的 Adaptive Server 登录。要删除适用于特定应用程序的所有用户的资源限制，可以将 *appname* 和 *name* 指定为 NULL。
- appname* — 是适用限制的应用程序。要删除适用于指定登录所使用的所有应用程序的资源限制，可以将登录名和 *appname* 指定为 NULL。要删除适用于特定应用程序的限制，可以指定客户端程序在登录包中传递给 Adaptive Server 的应用程序名。
- rangename* — 是实施限制的时间范围。这必须是 *systimeranges* 系统中存储的现有时间范围或者为 NULL，以便在不考虑 *rangename* 的情况下，删除对指定的 *name*、*appname*、*limittype*、*action* 和 *scope* 的所有资源限制。
- limittype* — 是受限制的资源类型。它必须为以下类型之一：

限制类型	说明
row_count	只删除限定查询可以返回的行数的限制。
elapsed_time	只删除限定查询批处理或事务可以运行的秒数的限制。
io_cost	只删除限定实际的或估计的查询处理开销的限制。
NULL	删除拥有指定的 <i>name</i> 、 <i>appname</i> 、 <i>rangename</i> 、施加时间、 <i>action</i> 和 <i>scope</i> 的所有资源限制，而不考虑 <i>limittype</i> 。

enforced — 确定在查询执行之前或在查询执行过程中是否进行限制。
下表列出了每种限制类型的有效值：

强制代码	说明	限制类型
1	只删除当估计的执行开销超过指定限制时，对所采取操作的限制。	io_cost
2	只删除当实际的行数、经历时间或执行开销超过指定限制时，对所采取操作的限制。	row_count elapsed_time io_cost
3	只删除当估计的开销 (1) 或实际的开销 (2) 超过指定限制时，对所采取操作的限制。	io_cost
NULL	删除拥有指定的 <i>name</i> 、 <i>appname</i> 、 <i>rangename</i> 、 <i>limittype</i> 和 <i>scope</i> 的所有资源限制，而不考虑什么时候执行 <i>action</i> 。	

action — 是超过限制时所采取的操作。它必须为以下类型之一：

操作代码	说明
1	只删除发出警告的限制。
2	只删除中止查询批处理的限制。
3	只删除中止事务的限制。
4	只删除注销会话的限制。
NULL	删除拥有指定的 <i>name</i> 、 <i>appname</i> 、 <i>rangename</i> 、 <i>limittype</i> 、执行时间和 <i>scope</i> 的所有资源限制，而不考虑其采取的 <i>action</i> 。

scope — 是限制的范围。它必须为以下类型之一：

范围代码	说明
1	只删除应用于查询的限制。
2	只删除应用于查询批处理的限制。
4	只删除应用于事务的限制。
6	只删除应用于查询批处理和事务的限制。
NULL	删除拥有指定的 <i>name</i> 、 <i>appname</i> 、 <i>rangename</i> 、 <i>limittype</i> 、执行时间和 <i>action</i> 的所有资源限制，而不考虑其 <i>scope</i> 。

示例

```
1. sp_drop_resource_limit joe, payroll,  
   friday_afternoon, io_cost, 2, 4, 1
```

只要在导致了额外执行时间 I/O 开销的 *friday_afternoon* 时间范围内，*joe* 使用 *payroll* 应用程序运行了某个查询，就会删除有关注销会话的资源限制。

► 注意

如果没有资源限制匹配这些选择标准，*sp_drop_resource_limit* 将正常返回，不带任何错误。

```
2. sp_drop_resource_limit joe, payroll
```

删除有关对 *joe* 使用 *payroll* 应用程序的所有限制。

```
3. sp_drop_resource_limit joe
```

删除应用于用户 “*joe*” 的所有限制。

```
4. sp_drop_resource_limit NULL, payroll
```

删除应用于 *payroll* 应用程序的所有资源限制。

```
5. sp_drop_resource_limit NULL, payroll, NULL, NULL,  
   NULL, 4, NULL
```

删除对 *payroll* 应用程序的所有资源限制，该应用程序的操作是要注销会话。

注释

- 请使用 *sp_help_resource_limit* 系统过程来确定哪些资源限制适用于给定的用户、应用程序或当天的时间。
- 使用 *sp_droplogin* 删除 Adaptive Server 登录时，与此登录相关的所有资源限制也将被删除。
- 如果删除资源限制，将导致在每个会话（该会话用于要重新绑定的登录和/或应用程序）的下一个查询批处理开始时，对该会话进行限制。
- 有关资源限制的详细信息，参见 *系统管理指南*。

权限

只有系统管理员才能执行 *sp_drop_resource_limit*。

使用的表

master..sysresourcelimits、 *master..systimeranges*、
master..spt_limit_types

参见

系统过程	sp_add_resource_limit、 sp_help_resource_limit、 sp_modify_resource_limit
------	---

sp_droprowlockpromote

功能

从数据库或表中删除行锁升级阈值。

语法

```
sp_droprowlockpromote {"database" | "table"}, objname
```

参数

database | table — 指定是否从数据库或表中删除行锁升级阈值。

objname — 是要从中删除行锁升级阈值的数据库或表的名称。

示例

```
1. sp_droprowlockpromote "table", "sales"
```

从 *sales* 表删除行锁升级值。 *sales* 的锁升级现在将使用数据库或全服务器范围的值。

注释

- 请使用 **sp_droprowlockpromote** 来删除由 **sp_setrowlockpromote** 设置的行锁升级值。
- 删除数据库的行锁升级阈值后，对于没有配置行锁升级阈值的数据库行锁定表，将使用全服务器范围的值。使用 **sp_configure** 来检查行锁升级配置参数的值。
- 删除表的行锁升级阈值后，**Adaptive Server** 使用数据库的行锁升级阈值（如果配置了），或全服务器范围的值（如果没有为数据库设置阈值）。
- 要改变数据库的锁升级阈值，您必须是正在使用 *master* 数据库。要改变数据库中表的锁升级阈值，则正在使用的数据库必须是表驻留的数据库。
- 可以使用 **sp_setrowlockpromote** 来更改全服务器范围的值。这更改了行锁升级配置参数中的值，所以对于 **sp_droprowlockpromote** 而言没有相应的服务器选项。

权限

只有系统管理员可以执行 **sp_droprowlockpromote**。

使用的表

master.dbo.sysattributes、 *sysobjects*

参见

系统过程	sp_setrowlockpromote
------	----------------------

sp_dropsegment

功能

从数据库删除段或从特定数据库设备取消映射段。

语法

```
sp_dropsegment segname, dbname [, device]
```

参数

segname — 是要删除的段的名称。

dbname — 是要从中删除段的数据库的名称。

device — 是要从中删除段 *segname* 的数据库设备的名称。除非是正在从数据库设备中删除系统段 *system*、*default* 或 *logsegment*，否则这一参数是可选的。

示例

1. sp_dropsegment indexes, pubs2

该命令可从 *pubs2* 数据库删除 *indexes* 段。

2. sp_dropsegment indexes, pubs2, dev1

该命令可从数据库设备 *dev1* 取消对段 *indexes* 的映射。

注释

- 如果在指定的数据库中没有任何表或索引引用段，则可以删除该段。
- 如果没有提供可选参数 *device*，则从指定的数据库中删除该段。如果提供了一个 *device* 名，则该段将不再被映射到指定的数据库设备，但是不会删除该段。
- 如果删除段，将删除与该段相关联的所有阈值。
- 当从一个或多个设备取消映射段时，**Adaptive Server** 将删除超出该段整个空间的任何阈值。当从一个或多个设备取消映射 *logsegment* 段时，**Adaptive Server** 将重新计算最后机会阈值。
- **sp_placeobject** 逐段地更改将来用于表或索引的空间分配，并从原始的段中删除引用。使用 **sp_placeobject** 之后，可以用 **sp_dropsegment** 删除原始的段名。
- 对于系统段 *system*、*default* 和 *logsegment*，必须指定要从中删除段的设备名。

权限

只有数据库所有者或系统管理员才能执行 **sp_dropsegment**。

使用的表

master.dbo.spt_values、*sysdatabases*、*sysdevices*、*sysindexes*、*sysobjects*、*syssegments*、*systhresholds*、*sysusages*

参见

系统过程	sp_addsegment、sp_addthreshold、 sp_helpsegment、sp_helpthreshold、 sp_placeobject
------	--

sp_dropserver

功能

从已知服务器的列表中删除某个服务器，或者删除同一操作中的远程登录和外部登录。

语法

```
sp_dropserver server [, droplogins]
```

参数

server — 是要删除的服务器的名称。

droplogins — 指出 **server** 的任何远程登录也应该被删除。

示例

1. sp_dropserver GATEWAY

该命令可删除远程服务器 GATEWAY。

2. sp_dropserver RDBAM_ALPHA, droplogins

删除远程服务器 RDBAM_ALPHA 的条目，并删除该服务器的所有远程登录和外部登录。

注释

- 如果执行 **sp_dropserver**，将通过从 *master.dbo.sysservers* 表中删除其条目，从而将某个服务器从已知服务器列表中删除。
- 对于在 *master.dbo.sysremotelogins* 表中具有相关条目的服务器而言，在它上面运行 **sp_dropserver** 将会产生一条错误消息，指出在删除该服务器之前必须删除远程用户。要在删除服务器时删除服务器的所有远程登录，可以使用 **droplogins**。
- 对于在 *sysattributes* 表中具有相关条目的服务器而言，未执行 **droplogins** 就对其运行 **sp_dropserver** 会导致一个错误。在删除该服务器之前，必须删除远程登录和外部登录。
- 在配置了组件集成服务之后，将会检查 *sysattributes*（用于检查服务器的外部登录和对服务器的缺省映射）。

权限

只有系统安全员才能执行 **sp_dropserver**。

使用的表

master.dbo.sysremotelogins、 *master.dbo.sysservers*、 *sysobjects*

参见

系统过程	sp_addserver、 sp_dropremotelogin、 sp_helpremotelogin、 sp_helpserver
------	--

sp_droptreshold

功能

从段中删除可用空间阈值。

语法

```
sp_droptreshold dbname, segname, free_space
```

参数

- dbname* — 是从中删除阈值的数据库。它必须是当前数据库的名称。
- segname* — 是其可用空间由阈值监控的段。当指定 “default” 段时，请使用引号。
- free_space* — 是达到阈值时的可用页数。

示例

```
1. sp_droptreshold mydb, segment1, 200
```

从 *mydb* 的 *segment1* 中删除阈值。必须指定数据库、段和可用空间大小来标识该阈值。

注释

- 不能从日志段删除最后机会阈值。
- 可以将 *sp_dboption* 的 *no free space acctg* 选项作为 *sp_droptreshold* 的替代项来使用。该选项将在非日志段上禁用可用空间统计。您不能对日志段禁用可用空间统计。

权限

只有数据库所有者或系统管理员才能执行 *sp_droptreshold*。

使用的表

sysobjects、*syssegments*、*systhresholds*

参见

系统过程	<i>sp_addthreshold</i> 、 <i>sp_dboption</i> 、 <i>sp_helpthreshold</i> 、 <i>sp_thresholdaction</i>
------	---

sp_drop_time_range

功能

从 Adaptive Server 删除用户定义的时间范围。

语法

sp_drop_time_range name

参数

name — 是要删除的时间范围的名称。

示例

1. sp_drop_time_range evenings

删除 “evenings” 时间范围。

注释

- 不能删除 “at all times” 时间范围。
- 如果存在对某个时间范围的资源限制，则不能删除该时间范围。
- 对于当前正在运行的会话，删除某个时间范围不会影响它的活动时间范围。
- 有关时间范围的详细信息，参见 *系统管理指南*。

权限

只有系统管理员才能执行 **sp_drop_time_range**。

使用的表

master..systimeranges

参见

系统过程	sp_add_resource_limit、sp_add_time_range、 sp_modify_time_range
------	--

sp_droptype

功能

删除用户定义的数据类型。

语法

```
sp_droptype typename
```

参数

typename — 是所拥有的用户定义的数据类型的名称。

示例

```
1. sp_droptype birthday
```

删除名为 *birthday* 的用户定义的数据类型。

注释

- 如果执行 **sp_droptype**，将从 *systypes* 中删除用户定义的数据类型。
- 如果表或其它数据库对象引用了用户定义的数据类型，则不能删除该数据类型。

权限

只有数据库所有者或数据类型所有者才能执行 **sp_droptype**。

使用的表

syscolumns、*sysobjects*、*systypes*、*sysusers*

参见

数据类型	“系统和用户定义的数据类型”
系统过程	sp_addtype、sp_rename

sp_dropuser

功能

从当前数据库删除用户。

语法

```
sp_dropuser name_in_db
```

参数

name_in_db — 是当前数据库的 *sysusers* 表中的用户名。

示例

1. sp_dropuser albert

从当前数据库中删除用户 “albert”。用户 “albert” 将不能再使用该数据库。

注释

- **sp_dropuser** 通过从 *sysusers* 中删除用户行，从而从当前数据库中删除用户。
- 不能删除拥有数据库中的对象的用户。
- 不能删除曾授予其他用户权限的用户。
- 不能从数据库中删除数据库所有者。
- 如果其他用户的别名为要删除的用户，其别名也会被删除。他们将不能再访问数据库。
- 如果用户拥有一个绑定到该数据库中的某个执行类的存储过程，则不能从数据库中删除该用户。参见 **sp_bindexec**。

权限

只有数据库所有者、系统管理员或系统安全员才能执行 **sp_dropuser**。

使用的表

master.dbo.spt_values、*sysalternates*、*syscolumns*、*sysobjects*、*sysprotects*、*syssegments*、*systhresholds*、*systypes*、*sysusers*

参见

命令	grant、revoke、use
系统过程	sp_addalias、sp_adduser、sp_bindexclass、 sp_droplogin

sp_dumpoptimize

功能

指定在转储数据库操作期间由 Backup Server 转储的数据的总量。

语法

```
sp_dumpoptimize [ 'archive_space =  
    {maximum | minimum | default }' ]  
sp_dumpoptimize [ 'reserved_threshold =  
    {nnn | default }' ]  
sp_dumpoptimize [ 'allocation_threshold =  
    {nnn | default }' ]
```

参数

archive_space — 指定您要转储的数据库总量。

maximum — 在未确定分配哪些页的情况下转储整个数据库。档案图像占用的总空间量等于该数据库的大小。使用此选项的作用与使用 **reserved_threshold=0** 和 **allocation_threshold=0** 选项的作用相同。

minimum — 只转储分配的页，这将产生可能的最小的档案图像。当转储到其吞吐量远小于数据库设备吞吐量（例如 QIC 磁带驱动器）的档案设备时，该选项是有用的。使用此选项的作用与使用 **reserved_threshold=100** 和 **allocation_threshold=100** 选项的作用相同。

default — 指定应使用缺省值。

当和 **archive_space** 一起使用时，该选项通过 **reserved_threshold** 和 **allocation_threshold** 选项设置将数据库转储到其缺省值。使用该选项将 Backup Server 重新设置为缺省配置。

当和 **reserved_threshold** 一起使用时，**default** 指定 85%。

当和 **allocation_threshold** 一起使用时，**default** 指定 40%。

reserved_threshold — 如果数据库磁盘中保留的页的百分比等于或大于 **nnn**，则转储该磁盘中属于此数据库的所有页。例如，如果您将 **nnn** 指定为 60，并且数据库磁盘所具有的保留页的百分比大于或等于 60%，则转储整个磁盘，并且不确定分配该磁盘中的哪些页。此选项的缺省值是 85%。

nnn — 介于 0 和 100 之间的整数值，该值代表阈值。它用于确定要转储的数据量。

当和 **reserved_threshold** 一起使用时，如果磁盘中保留的页的百分比大于指定的值，则转储数据库磁盘中该数据库的所有页。

当和 **allocation_threshold** 一起使用时，如果分配单元中已分配页的百分比大于为 **allocation_threshold** 指定的百分比，则转储分配单元中的所有页。

allocation_threshold — 如果分配单元中所分配页的百分比大于或等于 **nnn**，则转储该分配单元中的所有页。例如，如果您将 **nnn** 指定为 70，并且分配单元中所分配页的百分比大于或等于 70%，则转储整个分配单元，并且不确定是否分配该分配单元中的页。如果 **reserved_threshold** 设置导致转储整个磁盘，则为该磁盘忽略 **allocation_threshold** 设置。此选项的缺省值是 40%。

示例

1. sp_dumpoptimize 'archive_space=maximum'

Backup Server: 4.172.1.1: The value of 'reserved pages threshold' has been set to 0%.

Backup Server: 4.172.1.2: The value of 'allocated pages threshold' has been set to 0%.

这将导致转储整个数据库。

2. sp_dumpoptimize 'archive_space=minimum'

Backup Server: 4.172.1.1: The value of 'reserved pages threshold' has been set to 100%.

Backup Server: 4.172.1.2: The value of 'allocated pages threshold' has been set to 100%.

这将导致只转储已分配的页，因而导致最小的档案图像。

3. sp_dumpoptimize 'archive_space=default'

Backup Server: 4.172.1.1: The value of 'reserved pages threshold' has been set to 85%.

Backup Server: 4.172.1.2: The value of 'allocated pages threshold' has been set to 40%.

这导致保留的阈值被设置为 85%，分配阈值被设置为 40%。

4. `sp_dumpoptimize 'reserved_threshold=60'`

Backup Server: 4.172.1.3: The value of 'reserved pages threshold' has been set to 60%.

转储其保留的页的百分比大于或等于 **60%** 的数据库中的这些磁盘，而不读取该磁盘上的分配页。对于其余磁盘，读取分配页，并且使用 `allocation_threshold` 的最后设定值。如果在启动 Backup Server 后未设置 `allocation_threshold`，则使用 `allocation_threshold` 的缺省值 **40%**。

5. `sp_dumpoptimize 'reserved_threshold=default'`

Backup Server: 4.172.1.3: The value of 'reserved pages threshold' has been set to 85%.

这将导致保留的阈值被设置为 **85%**。它不影响分配页阈值。

6. `sp_dumpoptimize 'allocation_threshold=80'`

Backup Server: 4.172.1.4: The value of 'allocated pages threshold' has been set to 80%.

为其保留的页的百分比小于 `reserved_threshold` 的最后设定值的那些磁盘读取分配页；如果分配单元中已有 **80%** 或更多的页被分配，则转储整个分配单元。

7. `sp_dumpoptimize 'allocation_threshold=default'`

Backup Server: 4.172.1.4: The value of 'allocated pages threshold' has been set to 40%.

这将导致分配页阈值被设置为缺省的 **40%**。它不影响保留的页阈值。

8. `sp_dumpoptimize 'reserved_threshold=60', 'allocation_threshold=30'`

Backup Server: 4.172.1.3: The value of 'reserved pages threshold' has been set to 60%.

Backup Server: 4.172.1.4: The value of 'allocated pages threshold' has been set to 30%.

转储其保留的页的百分比大于或等于 **60%** 的数据库中的这些磁盘，而不读取该磁盘上的分配页。对于其余磁盘，将读取分配页；如果分配单元中已有 **30%** 或更多的页被分配，则转储整个分配单元。

9. `sp_dumpoptimize`

Backup Server: 4.171.1.1: The current value of 'reserved pages threshold' is 60%.

Backup Server: 4.171.1.2: The current value of 'allocated pages threshold' is 30%.

这显示阈值的当前值。

注释

- 当您用 **sp_dumpoptimize** 设置值时，这些值将立即生效，而无需重新启动 **Backup Server**。但是，只有重新启动 **Backup Server** 后，更改才会生效。当 **Backup Server** 重新启动时，将使用缺省值。
- 如果您多次发出 **sp_dumpoptimize**，则以后的转储使用由最后的实例指定的阈值。例如，如果您首先设置 **reserved_threshold** 值，并且以后发出 **archive_space=maximum**，则该值将覆盖您以前为 **reserved_threshold** 设置的值。
- 通过在每次数据库转储前更改 **sp_dumpoptimize** 值，不同的数据库转储能够使用不同的阈值。
- 数据库不同，最佳的阈值也会有所不同。因此，转储的性能既取决于 I/O 配置，也取决于数据库中使用的空间总量。通过使用不同的值并选择导致转储时间最短的值，以便就各种转储过程进行实验，**DBA** 能够确定一个数据库的合适配置。
- 您可以将 **sp_dumpoptimize** 用于本地转储和远程转储。
- **sp_dumpoptimize** 不影响事务日志转储或装载的性能。因此，在执行 **dump transaction**、**load database** 或 **load transaction** 操作前，不需要发出该命令。
- 如果发出 **sp_dumpoptimize** 时没有提供任何参数，则在客户端上显示阈值的当前值。
- 如果将档案设备吞吐量配置为大于或等于所有数据库磁盘的累积吞吐量，则使用 **archive_space=maximum** 可以进行更快的转储。但是，如果将档案设备吞吐量配置为小于所有数据库磁盘的累积吞吐量，则使用该选项将导致较慢的转储。
- 可以将选项名和此过程的值简化为用来标识它们的唯一的子字符串。例如，**ar = ma** 足以唯一标识 **archive_space=maximum** 选项。
- 在选项字符串中，在等号 (=) 周围可以有零个或更多个空格。
- 选项名及其值是区分大小写的。
- 有关所有分配页的详细信息，参见 *系统管理指南* 中的第 25 章“检查数据库一致性”中的“了解页和对象分配概念”。

权限

只有系统管理员、数据库所有者或具有操作员角色的用户才能执行 **sp_dumpoptimize**。

参见

命令	dump database、dump transaction、load database、load transaction
-----------	--

sp_estspace

功能

估计表及其索引要求的空间大小和创建索引需要的时间。

语法

```
sp_estspace table_name, no_of_rows [, fill_factor  
[, cols_to_max [, textbin_len [, iosec]]]]
```

参数

table_name — 是表名。该表名必须已经存在于当前数据库中。

no_of_rows — 是估计的表将包含的行数。

fill_factor — 是索引的填充因子。缺省值为空值，它意味着 Adaptive Server 将使用其缺省填充因子。

cols_to_max — 是用逗号分隔的可变长度列的列表，您需要使用该列表的最大长度，而不是平均长度。缺省值是可变长度列的平均声明长度。

textbin_len — 是所有 **text** 和 **image** 列每行的长度。缺省值为 0。只有在表存储 **text** 或 **image** 数据时，才需要提供值。**text** 和 **image** 列存储在单独的数据页集中，与表的其它数据的存储位置不同。实际的表行存储指向 **text** 或 **image** 值的指针。**sp_estspace** 另外提供一行有关行的 **text** 或 **image** 页的大小的信息。

iosec — 是本机上的每秒磁盘 I/O 次数。缺省值是每秒 30 次 I/O。

示例

1. `sp_estspace titles, 10000, 50, "title,notes", 0, 25`

name	type	idx_level	Pages	Kbytes
titles	data	0	3364	6728
titles	text/image	0	0	0
titleidind	clustered	0	21	43
titleidind	clustered	1	1	2
titleind	nonclustered	0	1001	2002
titleind	nonclustered	1	54	107
titleind	nonclustered	2	4	8
titleind	nonclustered	3	1	2

Total_Mbytes

8.68

name	type	total_pages	time_mins
titleidind	clustered	3386	13
titleind	nonclustered	1060	5
titles	data	0	2

计算 *titles* 表及其索引的空间要求和创建索引需要的时间。行数为 10,000，填充因子为 50%，使用列的最大大小计算两个可变长度列，并且磁盘 I/O 速度为每秒 25 次 I/O。

2. `declare @i int`
`select @i = avg(datalength(pic)) from au_pix`
`exec sp_estspace au_pix, 1000, null, null, 16, @i`

au_pix has no indexes

name	type	idx_level	Pages	Kbytes
au_pix	data	0	31	63
au_pix	text/image	0	21000	42000

Total_Mbytes

41.08

使用 *au_pix* 表中的现有 *image* 数据的平均长度计算具有 1000 行的表的大小。还可以将该大小作为常量提供。

3. sp_estspace titles, 50000

name	type	idx_level	Pages	Kbytes
titles	data	0	4912	9824
titleidind	clustered	0	31	61
titleidind	clustered	1	1	2
titleind	nonclustered	0	1390	2780
titleind	nonclustered	1	42	84
titleind	nonclustered	2	2	4
titleind	nonclustered	3	1	2

Total_Mbytes

12.46

name	type	total_pages	time_mins
titleidind	clustered	4943	19
titleind	nonclustered	1435	8

计算具有 50,000 行的 *titles* 表的大小，而所有其它值都使用缺省值。

注释

- 要估计表及其索引要求的空间大小：
 - a. 创建表。
 - b. 创建表上的所有索引。
 - c. 在根据需要给定表名、表的估计行数和可选参数的情况下，运行 **sp_estspace**。不需要在表中插入数据。**sp_estspace** 在计算表和索引的大小时，使用系统表中的信息，而不使用表中数据的大小。
- 如果在数据库中设置了 **auto identity** 选项，**Adaptive Server** 将自动在每个新表中定义 10 位数字的 **IDENTITY** 列，而该新表是在没有指定 **primary** 键、**unique** 约束或 **IDENTITY** 列的情况下创建的。要估计该列要求的额外空间的大小：
 - a. 在 **master** 数据库中，使用 **sp_dboption** 打开数据库的 **auto identity** 选项。
 - b. 创建表。
 - c. 在表上运行 **sp_estspace** 并记录结果。
 - d. 删除表。
 - e. 关闭数据库的 **auto identity** 选项。

- f. 重新创建表。
- g. 重新在表上运行 **sp_estspace** 并记录结果。
- 要获取表或列的信息，可以使用 **sp_help tablename**。

权限

任何用户都可以执行 **sp_estspace**。

使用的表

syscolumns、*sysindexes*、*sysobjects*

参见

命令	create index、create table
系统过程	sp_help

sp_export_qpgroup

功能

将指定用户和抽象计划组的所有计划导出到用户表。

语法

```
sp_export_qpgroup usr, group, tab
```

参数

usr — 是拥有要导出的抽象计划的用户名称。

group — 是包含要导出的计划的抽象计划组名称。

tab — 是要复制计划的表的名称。该表必须是当前数据库中的表。可以指定数据库名称，但不能指定所有者名称，其格式是 *dbname.tablename*。总长度必须等于或小于 30 个字符。

示例

```
1. sp_export_qpgroup freidak, ap_stdout,  
   "tempdb..moveplans"
```

为用户 “freidak” 创建包含所有计划的、名为 *moveplans* 的表，该用户在 *ap_stdout* 组中。

注释

- **sp_export_qpgroup** 将计划从抽象计划组复制到用户表。使用 **sp_import_qpgroup** 可以在服务器和数据库之间复制抽象计划组或将用户 ID 指派给复制的计划。
- 在运行 **sp_export_qpgroup** 之前，指定的用户表名是不会存在的。该表是用与 *sysqueryplans* 相同的结构创建的。
- **sp_export_qpgroup** 使用 **select...into** 创建该表以存储复制的计划。为了使用 **sp_export_qpgroup** 或在 *tempdb* 中创建表，必须使用 **sp_dboption** 来启用 **select into/bulkcopy/pilsort**。

权限

只有系统管理员或数据库所有者才能执行 **sp_export_qpgroup**。

使用的表

sysattributes、*sysqueryplans*

参见

系统过程	sp_copy_all_qplans、sp_copy_qplan、 sp_import_qpgroup
------	--

sp_extendsegment

功能

将段的范围扩展到其它数据库设备。

语法

```
sp_extendsegment segname, dbname, devname
```

参数

segname — 是以前用 **sp_addsegment** 定义的现有段的名称。

dbname — 是要扩展的段所位于的数据库的名称。*dbname* 必须是当前数据库的名称。

devname — 是要添加到当前数据库设备范围的数据库设备的名称，而该设备范围已经被包含在 *segname* 中。

示例

```
1. sp_extendsegment indexes, pubs2, dev2
```

该命令将在数据库设备 *dev2* 上扩展数据库 *pubs2* 的段 *indexes* 的范围。

注释

- 在定义了段之后，就可以在 **create table** 和 **create index** 命令中使用它来将表或索引放置在段上了。如果在特定段上创建表或索引，则该表或索引的后续数据也位于该段上。
- 要将段与数据库设备关联起来，可以创建或变更引用该设备的数据库。可以将多个段与一个数据库设备相关联。
- 可以将一个段扩展到多个数据库设备上。
- 在扩展 *logsegment* 段时，Adaptive Server 将重新计算它的最后机会阈值。

权限

只有数据库所有者或系统管理员才能执行 **sp_extendsegment**。

使用的表

master.dbo.sysdatabases、*sysdevices*、*master.dbo.sysusages*、*sysobjects*、*syssegments*

参见

命令	alter database、create index、create table
系统过程	sp_addsegment、sp_dropsegment、sp_helpdb、 sp_helpdevice、sp_helpsegment、 sp_placeobject

sp_familylock

功能

报告有关系列（协调进程及其工作进程）持有的所有锁的信息，而该系列是以并行方式执行语句的。

语法

```
sp_familylock [fpid1 [, fpid2]]
```

参数

fpid1 — 是 *master.dbo.sysprocesses* 表中的工作进程系列的系列标识符。运行 **sp_who** 或 **sp_lock** 可以获得父进程的 *spid*。

pfid2 — 是其它锁的 Adaptive Server 进程 ID 号。

示例

1. sp_familylock 5

fid	spid	locktype	table_id	page	dbname	class	context
5	5	Sh_intent	176003658	0	userdb	Non cursor lock	Sync-pt
duration request							
5	5	Sh_intent-blk	208003772	0	userdb	Non cursor lock	Sync-pt
duration request							
5	6	Sh_page	208003772	3972	userdb	Non cursor lock	Sync-pt
duration request							
5	7	Sh_page	208003772	3973	userdb	Non cursor lock	Sync-pt
duration request							
5	8	Sh_page	208003772	3973	userdb	Non cursor lock	Sync-pt
duration request							

显示 *fid* 为 5 的系列所有成员持有的锁的有关信息。

注释

- **sp_familylock** 如果不带参数，将报告有关属于系列的所有进程的信息，而这些系列当前持有锁。该报告与 **sp_lock** 的输出内容相同；但 **sp_familylock** 允许基于系列 ID（而不是进程 ID）生成报告。它用于检测系列的死锁情况。
- 使用 **object_name** 系统函数可以从表的 ID 号生成表名。
- “locktype” 列指示锁是共享锁（前缀为 “Sh”）、排他锁（前缀为 “Ex”）还是更新锁，以及锁是被表（“table” 或 “intent”）持有还是被页（“page”）持有。

“locktype” 列的 “blk” 后缀表示该进程正在阻塞其它要求获取锁的进程。一旦该进程结束，其它进程将继续执行。“demand” 后缀表示进程正在尝试获取排他锁。

- “class” 列指示锁是否与游标关联。它显示下列内容之一：
 - “Non cursor lock” 表示锁没有与游标关联。
 - “Cursor Id number” 表示锁与该 Adaptive Server 进程 ID 的游标 ID 号相关联。
 - 游标名表示锁与游标 *cursor_name* 相关联，而执行 **sp_lock** 的当前用户拥有该游标。
- “fid” 列标识锁属于的系列（包括协调进程及其工作进程）。“fid” 的值如下：
 - 零值表示 *spid* 代表的任务是串行执行的。它不参与并行执行。
 - 非零值表示持有锁的任务 (*spid*) 是并行执行语句的进程系列的成员（用 “fid” 标识）。如果值与 *spid* 相等，则表示该任务是并行执行查询系列中的协调进程。
- “context” 列标识锁的环境。相同系列中的工作进程具有相同的环境值。“context” 的值如下所示：
 - “NULL” 表示持有该锁的任务在执行串行查询，或者在事务隔离级别 1 上执行并行查询。
 - “FAM_DUR” 表示持有锁的任务在完成查询之前将继续持有锁。

在以下条件下，锁的环境可以是 “FAM_DUR”：锁是被作为并行查询的一部分而持有的表锁；锁在事务隔离级别 3 上被工作进程持有；或者锁被并行查询中的工作进程持有，并且必须在事务的持续期间内被持有。

权限

任何用户都可以执行 **sp_familylock**。

使用的表

master.dbo.spt_values、*master.dbo.syslocks*、*sysobjects*、*master.dbo.sysprocesses*

参见

命令	kill、select
系统过程	sp_lock、sp_who

sp_find_qplan

功能

在给定的查询文本或计划文本的模式的情况下，查找抽象计划。

语法

```
sp_find_qplan pattern [, group ]
```

参数

pattern — 是要在查询或抽象计划的文本中查找的字符串。

group — 是抽象计划组的名称。

示例

```
1. sp_find_qplan "%from titles%"

gid      id
text
-----
2 921054317
select count(*) from titles
2 921054317
( plan
( i_scan t_pub_id_ix titles )
( )
)
( prop titles
( parallel 1 )
( prefetch 16 )
( lru )
)

5 937054374
select type, avg(price) from titles group by type
5 937054374
( plan
( store Worktab1
( i_scan type_price titles )
)
( t_scan ( work_t Worktab1 ) )
)
( prop titles
( parallel 1 )
( prefetch 16 )
( lru )
)
```


报告在查询中具有字符串 “from titles” 的所有抽象计划。

2. **sp_find_qplan** "%t_scan%"

查找包含表扫描运算符的所有计划。

3. **sp_find_qplan** "%table[0-9]%", dev_plans

使用范围模式匹配在 *dev_plans* 组的计划中查找字符串，如 “table1”、“table2” 等。

注释

- 使用 **sp_find_qplan** 可以查找包含特定字符串的抽象计划。既可以与查询文本中的字符串匹配，也可以与抽象计划文本中的字符串匹配。
- 对于每种匹配计划，**sp_find_qplan** 都将输出组 ID、计划 ID、查询文本和抽象计划文本。
- 如果将组名包含在内，**sp_find_qplan** 将在指定的组中搜索字符串。如果不提供组名，**sp_find_qplan** 将搜索所有组的所有查询和计划。
- 必须象示例中那样提供 “%” 通配符，除非要搜索的字符串位于查询或计划的开头或末尾。可以使用任何 Transact-SQL 模式匹配语法，如示例 3 中所示。
- *sysqueryplans* 中查询文本被拆分为 255 字节的列值。**sp_find_qplan** 可能会错过跨越这些边界之一的匹配项，但会找到所有少于 127 字节的匹配项，即使这些匹配项跨越两行，也没有问题。

权限

任何用户都可以执行 **sp_find_qplan**。除非是由系统管理员或数据库所有者来执行，否则该命令只报告执行它的用户所拥有的抽象计划。

使用的表

sysattributes、*sysqueryplans*

参见

系统过程	sp_help_qpgroup、sp_help_qplan
------	-------------------------------

sp_flushstats

功能

将内存中存储的统计信息刷新到 *systabstats* 系统表。

语法

```
sp_flushstats objname
```

参数

objname — 是表名。

示例

```
1. sp_flushstats titles
```

刷新 *titles* 表的统计信息。

注释

- *systabstats* 表中某些统计信息是在内存存储位置中刷新的，并定期刷新到 *systabstats*，以减少 *systabstats* 上的开销和争用。
- 如果使用 SQL 查询 *systabstats*，则执行 **sp_flushstats** 可确保将内存中的统计信息刷新到 *systabstats* 中。
- **optdiag** 命令总是在显示输出内容之前刷新内存中的统计信息。
- 只能通过数据定义语言命令来改变 *sysstatistics* 中的统计信息，而且不需要使用 **sp_flushstats**。

权限

只有系统管理员可以执行 **sp_flushstats**。

sp_forceonline_db

功能

通过恢复提供对数据库中所有以前被标记为可疑页的页的访问。

语法

```
sp_forceonline_db dbname,  
    {"sa_on" | "sa_off" | "all_users"}
```

参数

dbname — 是要使其联机的数据库的名称。

sa_on — 只允许具有 *sa_role* 的用户访问指定的页。

sa_off — 撤消以前通过 *sa_on* 调用 *sp_forceonline_page* 创建的访问权限。

all users — 允许所有用户访问指定的页。

示例

1. sp_forceonline_db pubs2, "sa_on"

允许系统管理员访问 *pubs2* 数据库中的所有可疑页。

2. sp_forceonline_db pubs2, "sa_off"

撤消系统管理员访问 *pubs2* 数据库中所有可疑页的权限。现在，没有人能访问 *pubs2* 中的可疑页。

3. sp_forceonline_db pubs2, "all_users"

允许所有用户访问 *pubs2* 数据库中的所有页。

注释

- 不必修复被强制联机的页。可以强制已损坏的页联机。Adaptive Server 不对强制联机的页执行一致性检查。
- 带 *all_users* 的 *sp_forceonline_page* 不能被撤消。如果已将页联机，使所有用户都可访问，则不能再将其脱机。
- 事务中不能使用 *sp_forceonline_db*。
- 要仅使特定的脱机页联机，可以使用 *sp_forceonline_page*。

权限

只有系统管理员才能执行 *sp_forceonline_db*。

使用的表

master.dbo.sysattributes

参见

系统过程	sp_forceonline_page、 sp_listsuspect_db、 sp_listsuspect_page、 sp_setsuspect_granularity、 sp_setsuspect_threshold
------	--

sp_forceonline_object

功能

通过恢复提供对以前被标记为可疑的索引的访问。

语法

```
sp_forceonline_object dbname, objname, indid,  
    {sa_on | sa_off | all_users} [, no_print]
```

参数

dbname — 是数据库的名称，该数据库包含要联机的索引。

objname — 是表名。

indid — 是要联机的可疑索引的索引 ID。

sa_on — 只允许具有 *sa_role* 的用户访问指定的索引。

sa_off — 撤消以前通过 *sa_on* 调用 **sp_forceonline_object** 创建的访问权限。

all_users — 允许所有用户访问指定的索引。

no_print — 指定对象联机后，不显示其它可疑对象的列表。

示例

1. **sp_forceonline_object pubs2, titles, 3 , sa_on**

允许系统管理员访问 *pubs2* 数据库中 *titles* 表上具有 *indid* 3 的索引。

2. **sp_forceonline_object pubs2, titles 3, sa_off**

撤消系统管理员对索引的访问。现在，所有人都不能访问该索引。

3. **sp_forceonline_object pubs2, titles, 3, all_users**

允许所有用户访问 *pubs2* 数据库中 *titles* 表上的索引。

注释

- 如果 DOL 锁定表上的索引有可疑页，则恢复期间整个索引都将脱机。查询优化程序不考虑脱机索引。**allpage** 锁定表上的索引在恢复期间不完全脱机，只有这些索引的个别页脱机。可使用 **sp_forceonline_page** 将这些页联机。
- 使用 **sp_listsuspect_object** 可查看脱机的数据库列表。
- 要修复可疑索引，使用带 **sa_on** 访问权的 **sp_forceonline_object**。然后，删除并重新创建索引。

► 注意

如果索引位于 *systabstats* 或 *sysstatistics*（唯一的仅数据锁定系统表）上，则给 Sybase 技术支持部门打电话以获得帮助。

- 带 **all_users** 的 **sp_forceonline_object** 不能被撤消。若已将索引联机，使所有用户都可访问，则不能再将其脱机。
- 被强制联机的索引不必修复。可以强制已损坏的索引联机。**Adaptive Server** 不对强制联机的索引执行一致性检查。
- 事务中不能使用 **sp_forceonline_object**。
- **sp_forceonline_object** 仅用于其恢复故障隔离模式是“页”的数据库。使用 **sp_setsuspect_granularity** 可显示数据库的恢复故障隔离模式。
- 要使用单个命令使数据库的脱机页和索引都联机，可使用 **sp_forceonline_db**。
- 有关恢复故障隔离的详细信息，参见 *系统管理指南*。

权限

只有系统管理员才能执行 **sp_forceonline_object**。

使用的表

master.dbo.sysattributes

参见

系统过程	sp_listsuspect_object
------	------------------------------

sp_forceonline_page

功能

通过恢复提供对先前标记为可疑的页的访问。

语法

```
sp_forceonline_page dbname, pgid,  
    {"sa_on" | "sa_off" | "all_users"}
```

参数

dbname — 是数据库的名称，该数据库包含要联机的页。

pgid — 是要联机的页的标识符。

sa_on — 只允许具有 **sa_role** 的用户访问指定的页。

sa_off — 撤消以前通过 **sa_on** 调用 **sp_forceonline_page** 创建的访问权限。

all_users — 允许所有用户访问指定的页。

示例

1. **sp_forceonline_page pubs2, 312, "sa_on"**

允许系统管理员访问 *pubs2* 数据库中的第 312 页。

2. **sp_forceonline_page pubs2, 312, "sa_off"**

撤消系统管理员访问 *pubs2* 数据库中第 312 页的权限。现在，所有人都不能访问该页。

3. **sp_forceonline_page pubs2, 312, "all_users"**

允许所有用户访问 *pubs2* 数据库中的第 312 页。

注释

- 带 **all_users** 的 **sp_forceonline_page** 不能被撤消。如果已将页联机，使所有用户都可访问，则不能再将其脱机。
- 不必修复被强制联机的页。可以强制已损坏的页联机。Adaptive Server 不对强制联机的页执行一致性检查。
- 事务中不能使用 **sp_forceonline_page**。
- sp_forceonline_page** 仅适用于恢复故障隔离模式是“页”的数据库。使用 **sp_setsuspect_granularity** 可以显示数据库的恢复故障隔离模式。

- 要通过单个命令使数据库的脱机页都联机，可以使用 **sp_forceonline_db**。

权限

只有系统管理员才能使用 **sp_forceonline_page**。

使用的表

master.dbo.sysattributes

参见

系统过程	sp_forceonline_db、 sp_listsuspect_db、 sp_listsuspect_page、 sp_setsuspect_granularity、 sp_setsuspect_threshold
------	--

sp_foreignkey

功能

在当前数据库中定义表或视图的外键。

语法

```
sp_foreignkey tablename, pktabname, col1 [, col2] ...  
            [, col8]
```

参数

tablename — 是包含要定义的外键的表或视图的名称。

pktabname — 是具有外键所适用的主键的表或视图的名称。主键必须已经定义。

col1 — 是构成外键的第一列的名称。外键必须至少有一列，最多可以有八列。

示例

1. sp_foreignkey titles, publishers, pub_id

publishers 表的主键是 *pub_id* 列。 *titles* 表也包含 *pub_id* 列，该列是 *publishers* 的外键。

2. sp_foreignkey orders, parts, part, subpart

通过 *sp_primarykey*，已经将 *parts* 表的主键定义为 *partnumber* 和 *subpartnumber* 列。 *orders* 表包含 *part* 和 *subpart* 列，它们构成 *parts* 的外键。

注释

- *sp_foreignkey* 将键添加到 *syskeys* 表。键使数据库设计中隐式的逻辑关系变为显式。
- *sp_foreignkey* 不会强加参照完整性约束；使用 *create table* 或 *alter table* 命令的 *foreign key* 子句可以强加外键关系。
- 构成外键的列的编号和顺序必须与构成主键的列的编号和顺序相同。主键和外键的数据类型（和长度）必须一致，但空值类型不必一致。
- 安装进程在系统表的适当列上运行 *sp_foreignkey*。
- 要显示有关已经定义的键的报告，请执行 *sp_helpkey*。

权限

只有表或视图的所有者才能执行 **sp_foreignkey**。

使用的表

syscolumns、 *sysindexes*、 *syskeys*、 *sysobjects*、 *sysreferences*

参见

命令	alter table、 create table、 create trigger
系统过程	sp_commonkey、 sp_dropkey、 sp_helpkey、 sp_helpjoins、 sp_primarykey

sp_freedll

功能

卸载先前装载到 **XP Server** 内存中的动态链接库 (DLL)，以支持扩展存储过程 (ESP) 的执行。

语法

sp_freedll *dll_name*

参数

dll_name — 是从 **XP Server** 内存卸载的 DLL 的文件名。

示例

1. **sp_freedll "sqlsrvdll.dll"**

卸载 *sqlsrvdll.dll* DLL。

注释

- **sp_freedll** 不能在事务内执行。
- **sp_freedll** 不能释放系统 ESP 的 DLL。
- 一种使用 **sp_freedll** 显式卸载 DLL 的替代方法是：指定总是在调用 DLL 的 ESP 请求终止后才卸载这些 DLL。为此，可以将 **esp unload dll** 配置参数设置为 1 或者用 **-u** 选项启动 **xpserver**。
- **sp_freedll** 可以用于在不关闭 **XP Server** 或 **Adaptive Server** 的情况下，更新 DLL 中的 ESP 函数。
- 如果使用 **sp_freedll** 卸载正在使用中的 DLL，**sp_freedll** 将成功，但会导致当前使用 DLL 的 ESP 失败。

权限

只有系统管理员才能执行 **sp_freedll**。

使用的表

master.dbo.syscomments、*sysobjects*

参见

系统过程	sp_addextendedproc、sp_dropextendedproc、sp_helpextendedproc
实用程序	

sp_getmessage

功能

在 *sysmessages* 和 *sysusermessages* 中检索存储的消息字符串，供 **print** 和 **raiserror** 语句使用。

语法

```
sp_getmessage message_num, result output [, language]
```

参数

message_num — 是要检索的消息的编号。

result output — 是接收返回消息文本的变量，它的后面跟着一个空格和关键字 **output**。该变量的数据类型必须是 *char*、*nchar*、*varchar* 或 *nvarchar*。

language — 是要检索的消息语言。*language* 必须是 *syslanguages* 表中的有效语言名。如果包括 *language*，将检索具有指示的 *message_num* 和 *language* 的消息。如果不包括 *language*，则检索缺省会话语言的消息，如变量 @@langid 所指示的。

示例

1. **declare @myvar varchar(200)**
exec sp_getmessage 20001, @myvar output
在 *sysusermessages* 中检索消息编号 20001。
2. **declare @myvar varchar(200)**
exec sp_getmessage 20010, @myvar output, french
在 *sysusermessages* 中检索消息编号 20010 法语版本。

注释

- 任何应用程序都可以使用 **sp_getmessage**，任何用户都可以读取存储在 *sysmessages* 和 *sysusermessages* 中的消息。

权限

任何用户都可以执行 **sp_getmessage**。

使用的表

master.dbo.syslanguages、*master.dbo.sysmessages*、*sysobjects*、*sysusermessages*

参见

命令	print、raiserror
系统过程	sp_addmessage、sp_dropmessage

sp_grantlogin

(仅适用于 Windows NT)

功能

当集成的安全方式或混合方式（具有命名管道）处于活动状态时，为 Windows NT 用户和组指派 Adaptive Server 角色或 **default** 权限。

语法

```
sp_grantlogin {login_name | group_name}
              ["role_list" | default]
```

参数

login_name — 是 Windows NT 用户的网络登录名。

group_name — 是 Windows NT 组名。

role_list — 是授予的 Adaptive Server 角色的列表。该角色列表可以包含以下的一个或多个角色名：**sa_role**、**sso_role**、**oper_role**。如果指定了多个角色，需要用空格（而不是逗号）将各个角色名分隔开。

default — 指定 *login_name* 或 *group_name* 获得用 **grant** 语句或 **sp_role** 过程指派的缺省权限。

示例

1. sp_grantlogin jeanluc, oper_role

将 Adaptive Server **oper_role** 指派给 Windows NT 用户 “jeanluc”。

2. sp_grantlogin valle

将 **default** 值指派给 Windows NT 用户 “valle”。用户 “valle” 获得任何通过 **grant** 命令或 **sp_role** 过程指派的权限。

3. sp_grantlogin Administrators, "sa_role sso_role"

将 Adaptive Server **sa_role** 和 **sso_role** 指派给 Windows NT 管理员组的所有成员。

注释

- 在用 **sp_grantlogin** 指派角色之前，必须创建 Windows NT 登录名或组。有关详细信息，参见 Windows NT 文档。
- 只有在连接为命名管道时，在 Adaptive Server 运行在集成的安全方式或混合方式的情况下，**sp_grantlogin** 才处于活动状态。如果 Adaptive Server 运行在标准方式或混合方式下，而连接不是命名管道，需要使用 **grant** 和 **sp_role** 作为替代。
- 如果不指定 *role_list* 或 **default**，过程将自动指派 **default** 值。
- **default** 值并不表示 Adaptive Server 角色。它指定用户或组应该获得通过 **grant** 命令或 **sp_role** 过程指派给用户或组的任何权限。
- 如果通过现有 *login_name* 或 *group_name* 使用 **sp_grantlogin**，将覆盖用户或组的现有角色。

权限

只有系统管理员才能执行 **sp_grantlogin**。

使用的表

sysobjects

参见

命令	grant、setuser
系统过程	sp_addlogin、sp_addremotelogin、sp_adduser、sp_displaylogin、sp_droplogin、sp_dropuser、sp_locklogin、sp_logininfo、sp_modifylogin、sp_revokelogin、sp_who

sp_ha_admin

功能

在配置有高可用性系统中 Sybase 故障替换的 Adaptive Server 上执行管理任务。**sp_ha_admin** 是用 UNIX 平台上的 *installhavss* 脚本或用 Windows NT 平台上的 *insthasv* 脚本安装的。

语法

```
sp_ha_admin [cleansessions | help]
```

参数

cleansessions — 将删除 *sysessions* 中的原有条目。由于 Adaptive Server 未能在重新启动时清除 *sysessions* 或客户端未能连接到 Adaptive Server，通常会将原有 *sysessions* 条目遗留下来。

help — 显示 **sp_ha_admin** 的语法。

示例

```
1. sp_ha_admin cleansessions
   (return status = 0)
```

删除 *sysessions* 中由于没有正确退出客户端连接所遗留的原有条目。

```
2. sp_ha_admin "help"
```

```
sp_ha_admin Usage: sp_ha_admin command [, option1
[, option2]]
sp_ha_admin commands:
sp_ha_admin 'cleansessions'
sp_ha_admin 'help'
(return status = 0)
```

显示 **sp_ha_admin** 的语法

注释

- **sp_ha_admin** 在配置有高可用性系统中 Sybase 故障替换的 Adaptive Server 上执行管理任务。**sp_ha_admin** 不是用 *installmaster* 脚本安装的；而是使用安装并配置了 Sybase 故障替换的 *installhavss* 脚本安装的（在 Windows NT 上则使用 *insthasv* 脚本）。
- 如果成功对 *sysessions* 进行了清除，**sp_ha_admin** 将返回 0；如果遇到错误，则返回 1。

- 如果 **sp_ha_admin** 没有删除 **sysessions** 中的任何条目（例如，如果它不能在 **sysessions** 上获得锁），它将在错误日志中输入一条消息。
- 要查看 **sysessions** 中的所有当前条目，可以输入：

```
select * from sysessions
```

权限

只有具有 **ha_role** 的系统管理员才能执行 **sp_ha_admin**。

使用的表

master.dbo.sysessions

sp_help

功能

报告有关数据库对象（列在 *sysobjects* 中的任何对象）的信息和有关系统或用户定义的数据类型的信息。

语法

```
sp_help [objname]
```

参数

objname — 是 *sysobjects* 中的任何对象的名称或 *systypes* 中任何用户定义的数据类型或系统数据类型。不能指定数据库的名称。
objname 可以包含表、视图、存储过程、日志、规则、缺省值、触发器、参照约束和检查约束。如果对象所有者不是运行该命令的用户并且不是数据库所有者，则使用所有者的名称。

示例

1. sp_help

显示 *sysobjects* 中对象的列表并显示每个对象的名称、所有者和对象类型。还显示 *systypes* 中每个用户定义的数据类型的列表，指示数据类型的名称、存储类型、长度、空值类型、缺省名称和规则名称。空值类型为 0（不允许有空值）或 1（允许有空值）。

2. sp_help publishers

Name			Owner			Type		
publishers			dbo			user table		
Data_located_on_segment			When_created					
default			Apr 12 1999 3:31PM					
Column_name	Type	Length	Prec	Scale	Nulls	Default_name	Rule_name	Identity
pub_id	char	4	NULL	NULL	0	NULL	pub_idrule	0
pub_name	varchar	40	NULL	NULL	1	NULL	NULL	0
city	varchar	20	NULL	NULL	1	NULL	NULL	0
state	char	2	NULL	NULL	1	NULL	NULL	0
attribute_class		attribute		int_value		char_value		comments
buffer manager		cache binding		1		publishers_cache		NULL

```
index_name          index_description          index_keys
      index_max_rows_per_page
-----
pubind              clustered, unique located on default  pub_id
0
name      attribute_class attribute  int_value char_value
      comments
-----
pubind     buffer manager  cache name          NULL cache for index pubind
NULL
keytype    object                      related_object
      object_keys
      related_keys
-----
primary    publishers                  - none --
      pub_id, *, *, *, *, *, *, *
      *, *, *, *, *, *, *, *
foreign    titles                      publishers
      pub_id, *, *, *, *, *, *, *
      pub_id, *, *, *, *, *, *, *
Object is not partitioned.
Lock scheme Allpages
The attribute 'exp_row_size' is not applicable to tables with allpages
lock scheme.
The attribute 'concurrency_opt_threshold' is not applicable to tables
with allpages lock scheme.

exp_row_size reservepagegap fillfactor max_rows_per_page identity_gap
-----
0              0              0              0              0
concurrency_opt_threshold
-----
0
```

显示有关 *publishers* 表的信息。**sp_help** 也可以给出特性的类、名称、整数值、字符值和注释，从而列出指派给指定表及其索引的所有特性。以上示例显示了 *publishers* 表的缓存绑定特性。

3. sp_help partitioned_table

Name				Owner	Type	
-----				-----	-----	
partitioned_table				dbo	user table	
Data_located_on_segment				When_created		
-----				-----		
data1				Mar 24 1995 10:48AM		
Column_name	Type	Length	Prec	Scale	Nulls	Default_name
-----	-----	-----	-----	-----	-----	-----
coll	char	5	NULL	NULL	0	NULL
Rule_name		Identity				
-----		-----				
NULL		0				
Object does not have any indexes.						
No defined keys for this object.						
partitionid		firstpage	controlpage			
-----		-----	-----			
1		145	146			
2		312	313			
Lock scheme Datarows						
exp_row_size		reservepagegap	fillfactor	max_rows_per_page		
-----		-----	-----	-----		
1		0	0			
concurrency_opt_threshold						

15

显示有关已分区的表的信息。

4. sp_help "mary.marytrig"

Name	Owner	Type
-----	-----	-----
marytrig	mary	trigger
Data_located_on_segment		When_created
-----		-----
not applicable		Mar 20 1992 2:03PM

显示有关用户 “mary” 拥有的触发器 *marytrig* 的信息。由于句号是一种特殊的字符，所以需要引号。

5. sp_help money

Type_name	Storage_type	Length	Prec	Scale
money	money	8	NULL	NULL
Nulls	Default_name	Rule_name	Identity	
1	NULL	NULL	0	

显示有关系统数据类型 *money* 的信息。

6. sp_help identype

Type_name	Storage_type	Length	Nulls	Default_name
identype	numeric	4	0	NULL
Rule_name	Identity			
NULL	1			

显示有关用户定义的数据类型 *identype* 的信息。报告指示了从中创建数据类型的基本类型，不管它是否允许空值、允许任何绑定到数据类型的规则和缺省值的名称，也不管它是否具有 **IDENTITY** 属性。

7. sp_help titles

Name	Owner	Type					
titles	dbo	user table					
Data_located_on_segment		When_created					
default		Dec 6 1997 12:07PM					
Column_name	Type	Length	Prec	Scale	Nulls	Default_name	Rule_name
Identity							
title_id	tid	6	NULL	NULL	0	NULL	NULL
0							
title	varchar	80	NULL	NULL	0	NULL	NULL
0							
type	char	12	NULL	NULL	0	typedflt	NULL
0							
pub_id	char	4	NULL	NULL	1	NULL	NULL
0							
price	money	8	NULL	NULL	1	NULL	NULL
0							
advance	money	8	NULL	NULL	1	NULL	NULL
0							
total_sales	int	4	NULL	NULL	1	NULL	NULL

```
0
notes          varchar      200 NULL  NULL    1 NULL    NULL
0
pubdate        datetime     8 NULL  NULL    0 datedflt  NULL
0
contract       bit          1 NULL  NULL    0 NULL    NULL
0

attribute_class      attribute          int_value
      char_value
      comments
-----
lock strategy                row lock promotion
NULL
      PCT = 95, LWM = 300, HWM = 300
      NULL

index_name          index_description
      index_keys
      index_max_rows_per_page index_fillfactor index_reservepagegap
-----
-----
titleidind          clustered, unique located on default
      title_id
                                0          0          0
titleind            nonclustered located on default
      title
                                0          0          0
type_price          nonclustered located on default
      type, price DESC
                                0          0          0
No defined keys for this object.
Object is not partitioned.
Lock scheme Datarows

exp_row_size reservedpagegap fillfactor max_rows_per_page identity_gap
-----
      224          16          0          0          0
concurrency_opt_threshold
-----
      0
```

有关 *titles* 表的报告，包括有关锁定方案、期望的行大小、保留页间距和行锁升级设置。

注释

- **sp_help** 只在当前数据库中查找对象。
- **sp_help** 遵循查找对象的这些 Adaptive Server 规则：
 - 如果您未指定所有者名，且拥有具有指定名称的对象，那么 **sp_help** 将报告该对象的信息。
 - 如果您未指定所有者名，且不拥有该名称的对象，但是数据库所有者拥有这样的对象，那么 **sp_help** 将报告数据库所有者对象的信息。
 - 如果您和数据库所有者都不拥有具有指定名称的对象，即使在数据库中存在一个具有此对象名、但所有者是其他人，**sp_help** 仍然会报告一个错误条件。限定数据库用户（除了您自己和具有所有者名的数据库所有者）拥有的对象，如示例 4 所示。
 - 如果您和数据库所有者都拥有具有指定名称的对象，并且您想访问数据库所有者的对象，就要指定名称，其形式为 *dbo.objectname*。
- 如果在 *tempdb* 中发出 **sp_help**，则它将用于临时表。
- 具有 **IDENTITY** 属性的列的 “Identity” 值为 1；其它列的 “Identity” 值为 0。在示例 2 中，没有 **IDENTITY** 列。
- **sp_help** 将列出表上的所有索引，包括通过在 **create table** 或 **alter table** 语句中定义唯一或主键约束而创建的索引。它还列出与这些索引相关的所有特性。但 **sp_help** 不对为表定义的完整性约束作任何说明。使用 **sp_helpconstraint** 可以获得有关任何完整性约束的信息。
- **sp_help** 显示以下新设置：
 - 锁定方案，它可以用 **create table** 设置，用 **alter table** 更改
 - 期望的行大小，它可以用 **create table** 设置，用 **sp_chgattribute** 更改
 - 保留页间距，它可以用 **create table** 设置，用 **sp_chgattribute** 更改
 - 行锁升级设置，它可以用 **sp_setrowlockpromote** 设置或更改，用 **sp_droprowlockpromote** 删除
- **sp_help** 包括来自 **sp_helpindex** 的报告，它显示用于创建索引和空间管理属性的键顺序。
- 当启用组件集成服务时，**sp_help** 将显示有关远程对象存储位置的信息。

权限

任何用户都可以执行 **sp_help**。

使用的表

master.dbo.spt_values、*master.sysattributes*、*sysattributes*、
syscolumns、*sysindexes*、*sysmessages*、*sysobjects*、*syspartitions*、
systypes

参见

系统过程	<i>sp_chgattribute</i> 、 <i>sp_droprowlockpromote</i> 、 <i>sp_helppartition</i> 、 <i>sp_helpconstraint</i> 、 <i>sp_helpdb</i> 、 <i>sp_helpindex</i> 、 <i>sp_helpkey</i> 、 <i>sp_helpprotect</i> 、 <i>sp_helpsegment</i> 、 <i>sp_setrowlockpromote</i> 、 <i>sp_helptext</i> 、 <i>sp_helpuser</i>
------	---

sp_helppartition

功能

列出已分区的表中每个分区的分区号、首页、控制页、数据页数和摘要大小等信息。

语法

```
sp_helppartition [table_name]
```

参数

table_name — 是当前数据库中已分区的表的名称。如果没有提供表名，将输出数据库中所有用户表的所有者、表名和分区数。

示例

1. sp_helppartition sales

partitionid	firstpage	controlpage	ptn_data_pages
1	313	314	4227
2	12802	12801	4285
3	25602	25601	4404
4	38402	38401	4523
5	51202	51201	4347
6	64002	64001	4285

(6 rows affected)

Partitions	Average Pages	Maximum Pages	Minimum Pages	Ratio (Max/Avg)
6	4345	4523	4227	1.040967

返回 *sales* 中的分区信息。

注释

- sp_helppartition** 列出已分区的表中每个分区的分区号、首页、控制页和数据页数。每个分区的页数显示数据在分区之间分布的均匀程度。

摘要信息显示分区数、每个分区的平均页数、最大和最小页数以及平均页数和最大页数的比率。在查询优化过程中，将使用该比率。如果比率等于或大于 **2**（表示最大大小是平均大小的二倍或二倍以上），优化程序将选择串行查询计划，而不选择并行计划。

- 对表进行分区将创建附加的页链。使用 **alter table** 命令的 **partition** 子句可以对表进行分区。每个页链都有最后一页，可用于并发的插入操作。这将减少对页的争用，从而更有效地执行插入操作。当一个表分布于多个物理设备时，分区也会在 **Adaptive Server** 将数据从缓存刷新到磁盘时减少 I/O 争用，从而更有效地执行插入操作。
- 对表进行分区不会影响表在执行更新或删除命令时的性能。
- 使用 **alter table** 命令的 **unpartition** 子句可以将所有现有页链并置。
- 不管表是否分区，都不会移动现有数据。
- 要更改表的分区数，需要首先使用 **alter table** 的 **unpartition** 子句将其页链并置。然后，使用 **alter table** 的 **partition** 子句对表重新分区。
- **sp_helppartition** 将只在当前数据库中查找表。
- 使用 **sp_helpsegment** 可以显示存储已分区表的段上的已用页数和可用页数。

结果的精确性

- “**data_pages**”列中报告的值可能会大于实际的值。要确定计数是否准确，可以运行 **sp_statistics** 和 **sp_helppartition** 来比较数据页计数。**sp_statistics** 提供的计数总是很准确。

如果 **sp_statistics** 报告的页计数与 **sp_helppartition** 报告的分区页数和的差别超过 5%，则需要运行以下命令更新分区统计信息：

- **dbcc checkalloc**
- **dbcc checkdb**
- **dbcc checktable**
- **update all statistics**
- **update partition statistics**

然后，重新运行 **sp_helppartition** 就可以获得准确的报告了。

权限

任何用户都可以执行 **sp_helppartition**。

使用的表

syspartitions

参见

命令	alter table、insert
系统过程	sp_help、sp_helpsegment

sp_helpcache

功能

显示绑定到数据缓存的对象信息或指定的缓存大小所需的开销量信息。

语法

```
sp_helpcache {cache_name | "cache_size[P|K|M|G]"}
```

参数

cache_name — 是现有数据缓存的名称。

cache_size — 指定缓存的大小，在指定时，**P** 表示页数，**K** 表示千字节数，**M** 表示兆字节数，**G** 表示千兆字节数。缺省值是 **K**。

示例

1. sp_helpcache pub_cache

显示绑定到 *pub_cache* 的项的信息。

2. sp_helpcache "80M"

显示创建 80MB 的数据缓存所需的开销量。

3. sp_helpcache

显示有关所有缓存和绑定到这些缓存的所有项的信息。

注释

- 要查看服务器上所有数据缓存的大小、状态和 I/O 大小，可以使用 **sp_cacheconfig**。
- 在用 **sp_cacheconfig** 配置数据缓存时，您指定的所有内存对数据缓存来说都是可用的。管理缓存的开销占用的是缺省数据缓存。**sp_helpcache** 将显示指定大小的缓存要求的内存大小。
- 要将对象绑定到缓存，可以使用 **sp_bindcache**。要解除特定对象到缓存的绑定，可以使用 **sp_unbindcache**。要解除对绑定到特定缓存的所有对象的绑定，可以使用 **sp_unbindcache_all**。
- sp_cacheconfig** 过程配置数据缓存。**sp_poolconfig** 过程用数据缓存配置内存池。
- sp_helpcache** 可以精确计算多达 74GB 的开销。

权限

任何用户都可以执行 **sp_helpcache**。

使用的表

master..sysattributes、*master..sysdatabases*、*sysattributes*、*sysindexes*、*sysobjects*

参见

系统过程	sp_bindcache 、 sp_cacheconfig 、 sp_poolconfig 、 sp_unbindcache 、 sp_unbindcache_all
------	--

sp_helpconfig

功能

报告有关配置参数的帮助信息。

语法

```
sp_helpconfig "configname", ["size"]
```

参数

configname — 是要查询的配置参数或不唯一的参数片段。

size — 是内存的大小，用 **B**（字节数）、**K**（千字节数）、**M**（兆字节数）、**G**（千兆字节数）或 **P**（页数）指定。如果在没有指定大小类型的情况下使用，*size* 将指定正在用该参数配置的实体数，如锁、打开的索引等。如果 *configname* 不是唯一的参数名，将忽略 *size*。

示例

1. sp_helpconfig "allow"

Configuration option is not unique.

option_name	config_value	run_value
allow backward scans	1	1
allow nested triggers	1	1
allow procedure grouping	1	1
allow remote access	1	1
allow resource limits	0	0
allow sendmsg	0	0
allow sql server async i/o	1	1
allow updates to system tables	0	0

2. sp_helpconfig "open objects", "421"

number of open objects sets the maximum number of database objects that are open at one time on SQL Server. The default run value is 500.

Minimum Value	Maximum Value	Default Value	Current Value	Memory Used
100	2147483647	500	500	243

Configuration parameter, 'number of open objects', will consume 207K of memory if configured at 421.

返回有关为 421 个对象描述符创建元数据缓存所需的内存大小的报告。

3. **sp_helpconfig "open databases", "1M"**

number of open databases sets the maximum number of databases that can be open at one time on SQL Server. The default run value is 12.

Minimum Value	Maximum Value	Default Value	Current Value	Memory Used
-----	-----	-----	-----	-----
5	2147483647	12	12	433

Configuration parameter, 'number of open databases', can be configured to 28 to fit in 1M of memory.

返回有关将有多少数据库描述符填充 1MB 的数据库缓存的报告。

4. **sp_helpconfig "number of locks", "512K"**

number of locks sets the number of available locks. The default run value is 5000.

Minimum Value	Maximum Value	Default Value	Current Value	Memory Used
-----	-----	-----	-----	-----
1000	2147483647	5000	5000	528

Configuration parameter 'number of locks', can be configured to 4848 to fit in 512K of memory.

返回有关将有多少锁使用 512K 内存的报告。

5. **sp_helpconfig "allow updates to system tables"**

allow updates to system tables allows system tables to be updated directly. The default is 0 (off).

Minimum Value	Maximum Value	Default Value	Current Value	Memory Used
-----	-----	-----	-----	-----
0	1	0	0	0

返回有关 allow updates to system tables 配置参数状态的报告。

注释

- **sp_helpconfig** 报告有关配置参数（例如，将参数设置为某值时所需的内存）的帮助信息。**sp_helpconfig** 还显示当前设置、该设置使用的内存量、缺省值以及最低和最高设置。
- 如果使用 *configname* 的不唯一的参数片段，**sp_helpconfig** 将返回具有配置值和当前值的匹配参数列表。参见示例 1。

规划元数据缓存配置

- 要规划服务器的元数据缓存配置，可以使用 **sp_helpconfig**。

例如，假定计划将包含 **2000** 个用户索引的数据库移到其它服务器上。要确定该服务器这样配置所需要的内存，以使它容纳数据库的用户索引，可以输入以下命令：

```
sp_helpconfig "open indexes", "2000"
```

number of open indexes sets the maximum number of indexes that can be open at one time on SQL Server. The default run value is 500.

Minimum Value	Maximum Value	Default Value	Current Value	Memory Used
-----	-----	-----	-----	-----
100	2147483647	500	500	208

Configuration parameter, 'number of open indexes', will consume 829k of memory if configured at 2000.

或者，假定有 **1MB** 内存可以用于索引缓存，而您需要知道它可以支持多少索引描述符。可以运行以下命令：

```
sp_helpconfig "open indexes", "1M"
```

number of open indexes sets the maximum number of indexes that can be open at one time on SQL Server. The default run value is 500.

Minimum Value	Maximum Value	Default Value	Current Value	Memory Used
-----	-----	-----	-----	-----
100	2147483647	500	500	208

Configuration parameter 'number of open indexes', can be configured to 2461 to fit in 1M of memory.

根据这样的输出，如果有 **1MB** 内存，则可以创建最多包含 **2461** 个索引描述符的索引描述符缓存。要创建该缓存，可以按如下方式设置 **number of open indexes** 配置参数：

```
sp_configure "number of open indexes", 2461
```

通过 sybdiagdb 使用 sp_helpconfig

Sybase 技术支持部门可能会在系统中创建 **sybdiagdb** 数据库，以供调试使用。该数据库中保存有诊断配置数据，它仅供 Sybase 技术支持部门使用。

下面的 **configname** 选项已添加到 **sp_helpconfig** 中，以便和 **sybdiagdb** 数据库一起使用：

- *number of ccbs* 是可配置的动作点控制块的数目，可用于协助调试。
- *caps per ccb* 是在一个可配置的动作点中一次可以配置的可配置动作点的最大数目。
- *average cap size* 是存储典型可配置动作点的相关信息所需内存的估计字节数。

例如：

sp_helpconfig "number of ccbs"					
Minimum Value	Maximum Value	Default Value	Current Value	Memory Used	
0	100	0	0	0	

sp_helpconfig "caps per ccb"					
Minimum Value	Maximum Value	Default Value	Current Value	Memory Used	
5	500	50	50	0	

sp_helpconfig "average cap size"					
Minimum Value	Maximum Value	Default Value	Current Value	Memory Used	
100	10000	200	200	0	

权限

只有 Sybase 技术支持部门才能使用 “通过 sybdiagdb 使用 sp_helpconfig” 中指定的选项。任何用户都可以执行具有其它 configname 选项的 sp_helpconfig。

使用的表

sysindexes、sysobjects、sysdatabases

诊断参数

以下三个配置参数是 sp_helpconfig 的一部分，它们仅供 Sybase 技术支持部门在诊断时与 sybdiagdb 数据库一起使用：

- number of ccbs* 是可配置的动作点控制块的数目，可用于协助调试。
- caps per ccb* 是在一个可配置的动作点中一次可以配置的可配置动作点的最大数目。
- average cap size* 是存储典型可配置动作点的相关信息所需内存的估计字节数。

语法

```
sp_helpconfig "number of ccbs"
sp_helpconfig "caps per ccb"
sp_helpconfig "average cap size"
```

示例

```
> sp_helpconfig "number of ccbs"
2> go
Minimum Value Maximum Value Default Value Current Value Memory Used
-----
0 100 0 0 0
(return status = 0)
```

```
1> sp_helpconfig "caps per ccb"
2> go
Minimum Value Maximum Value Default Value Current Value Memory Used
-----
5 500 50 50 0
(return status = 0)
```

```
1> sp_helpconfig "average cap size"
2> go
Minimum Value Maximum Value Default Value Current Value Memory Used
-----
100 10000 200 200 0
(return status = 0)
```

```
sp_helpconfig "average cap size"
```

参见

系统过程	sp_configure、sp_countmetadata、 sp_monitorconfig
------	--

sp_helpconstraint

功能

报告有关指定表中使用的完整性约束的信息。

语法

sp_helpconstraint [*objname*] [, *detail*]

参数

objname — 是表名，该表具有一个或多个由 **create table** 或 **alter table** 语句定义的完整性约束。

detail — 返回有关约束的用户或错误消息的信息。

示例

1. sp_helpconstraint store_employees

```
name                                defn
-----
store_empl_stor_i_272004000        store_employees FOREIGN KEY
                                   (stor_id) REFERENCES stores(stor_id)
store_empl_mgr_id_288004057        store_employees FOREIGN KEY
                                   (mgr_id) SELF REFERENCES
                                   store_employees(emp_id)
store_empl_2560039432              UNIQUE INDEX( emp_id) :
                                   NONCLUSTERED, FOREIGN REFERENCE

(3 rows affected)

Total Number of Referential Constraints: 2

Details:
-- Number of references made by this table: 2
-- Number of references to this table: 1
-- Number of self references to this table: 1

Formula for Calculation:
Total Number of Referential Constraints
= Number of references made by this table
+ Number of references made to this table
- Number of self references within this table
```

显示 *pubs3* 数据库中 *store_employees* 表的约束信息。
store_employees 表具有 *stores* 表的外键 (*stor_id*) 和对自身的参照 (*mgr_id* 引用 *emp_id*)。

2. sp_helpconstraint titles, detail

```

name                                type
      defn
      msg
-----
-----
-----
datedflt
      create default datedflt as getdate()

typedflt
      create default typedflt as "UNDECIDED"

titles_pub_id_96003373              referential constraint
      titles FOREIGN KEY (pub_id) REFERENCES publishers(pub_id)
      standard system error message number : 547

roysched_title__144003544           referential constraint
      roysched FOREIGN KEY (title_id) REFERENCES titles(title_id)
      standard system error message number : 547

salesdetai_title__368004342         referential constraint
      salesdetail FOREIGN KEY (title_id) REFERENCES titles(title_id)
      standard system error message number : 547

titleautho_title__432004570         referential constraint
      titleauthor FOREIGN KEY (title_id) REFERENCES titles(title_id)
      standard system error message number : 547

titles_800033162                    unique constraint
      UNIQUE INDEX ( title_id) : NONCLUSTERED, FOREIGN REFERENCE
      standard system error message number : 2601

(7 rows affected)

Total Number of Referential Constraints: 4

Details:
-- Number of references made by this table: 1
-- Number of references to this table: 3
-- Number of self references to this table: 0

Formula for Calculation:
Total Number of Referential Constraints
= Number of references made by this table
+ Number of references made to this table
- Number of self references within this table.
```

显示有关 *pubs3..salesdetail* 约束的详细信息，包括约束类型和任何约束错误消息。

3. sp_helpconstraint

id	name	Num_referential_constraints
80003316	titles	4
16003088	authors	3
176003658	stores	3
256003943	salesdetail	3
208003772	sales	2
336004228	titleauthor	2
896006223	store_employees	2
48003202	publishers	1
128003487	roysched	1
400004456	discounts	1
448004627	au_pix	1
496004798	blurbs	1

(11 rows affected)

显示 *pubs3* 数据库中所有表的列表。

注释

- **sp_helpconstraint** 将输出完整性约束的名称和定义以及表使用的引用数。**detail** 选项返回有关约束的用户或错误消息的信息。
- 运行不带参数的 **sp_helpconstraint**，将列出当前数据库中所有包含引用的表，并显示每个表中的引用总数。**sp_helpconstraint** 将根据每个表中的引用数，按降序顺序列出表。
- **sp_helpconstraint** 只报告有关表的完整性约束信息（用 **create table** 或 **alter table** 语句定义）。它不报告有关用 **create index** 语句创建的规则、触发器或索引的信息。使用 **sp_help** 可以查看有关表的规则、触发器和索引的信息。
- 对于没有用户定义的消息的约束，**Adaptive Server** 将报告与该约束相关的系统错误消息。查询 **sysmessages** 可以获得错误消息的实际文本。
- 可以只对当前数据库中的表使用 **sp_helpconstraint**。
- 如果查询超过配置的辅助扫描描述符数，**Adaptive Server** 将返回错误消息。可以使用 **sp_helpconstraint** 确定必要的扫描描述符数。有关详细信息，参见 *系统管理指南* 中 **number of aux scan descriptors** 配置参数的说明。

- 系统安全人员可以防止约束定义的源文本显示给执行 **sp_helpconstraint** 的多数用户。要将 *syscomments* 表的 *text* 列上的 **select** 权限限制为只授予对象所有者或系统管理员，可以使用 **sp_configure** 将 **select on syscomments.text column** 参数设置为 **0**。这种限制是在已评估的配置中运行 **Adaptive Server** 所必需的。有关已评估的配置的详细信息，参见 *系统管理指南*。

权限

任何用户都可以执行 **sp_helpconstraint**。

使用的表

syscolumns、*syscomments*、*sysconstraints*、*sysindexes*、*sysobjects*、*sysreferences*、*sysusermessages*

参见

命令	alter table、create table
系统过程	sp_help、sp_helpdb、sp_monitorconfig

sp_helpdb

功能

报告有关特定数据库或有关所有数据库的信息。

语法

```
sp_helpdb [dbname]
```

参数

dbname — 是要报告其信息的数据库的名称。如果没有该可选参数，**sp_helpdb** 将报告所有数据库的信息。

示例

1. sp_helpdb

name	db_size	owner	dbid	created	status
master	5.0 MB	sa	1	Jan 01, 1900	no options set
model	2.0 MB	sa	3	Jan 01, 1900	no options set
pubs2	2.0 MB	sa	6	Sep 20, 1995	no options set
sybsystemprocs	16.0 MB	sa	4	Sep 20, 1995	trunc log on chkp
tempdb	2.0 MB	sa	2	Sep 20, 1995	select into/bulkcopy

显示有关 Adaptive Server 中所有数据库的信息。

2. sp_helpdb pubs2

(不从 *pubs2* 发出。)

name	db_size	owner	dbid	created	status
pubs2	2.0 MB	sa	4	Mar 05, 1993	abort tran when log full
device_fragments		size	usage	free kbytes	
master		2.0 MB	data and log		576
name	attribute_class	attribute	int_value	char_value	comments
pubs2	buffer manager	cache binding	1	pubs2_cache	NULL

显示有关 *pubs2* 数据库的信息。

3. sp_helpdb pubs2

(从 *pubs2* 发出。)

name	db_size	owner	dbid	created	status
pubs2	2.0 MB	sa	4	Mar 05, 1993	abort tran when log full
device_fragments		size	usage	free kbytes	
master		2.0 MB	data and log		576
device			segment		
master			default		
master			logsegment		
master			system		
name	attribute_class	attribute	int_value	char_value	comments
pubs2	buffer manager	cache binding	1	pubs2_cache	NULL

显示有关 *pubs2* 数据库的信息，包括片段的信息。

4. sp_helpdb pubtune

name	attribute_class
attribute	int_value
char_value	
comments	

pubtune	lock strategy
row lock promotion	NULL
PCT = 95, LWM = 300, HWM = 300	

显示 *pubtune* 数据库的行锁升级特性集。

注释

- 当给定 *dbname* 时，**sp_helpdb** 将报告有关指定数据库的信息；当没有提供参数时，它将报告列在 *master.dbo.sysdatabases* 中的所有数据库的信息。
- 在 *dbname* 中执行 **sp_helpdb dbname** 时，将在报告中包含可用空间和片段的信息。
- 只要定义了特性，**sp_helpdb** 将显示有关数据库特性的信息，它将给出特性的类、名称、整数值、字符值和注释。示例 3 显示了 *pubs2* 数据库的缓存绑定特性。
- **sp_helpdb** 报告数据库是否处于脱机状态。
- 如果为数据库定义了行锁升级阈值，**sp_helpdb** 将报告行锁升级阈值的信息。

- 在 **sp_helpdb** 的输出中，用 **for load** 选项创建的数据库具有 “don't recover” 状态。
- 当启用组件集成服务时，**sp_helpdb** 将列出指定数据库或所有数据库的缺省存储位置。如果没有缺省存储位置，显示内容将指示 “NULL”。

权限

任何用户都可以执行 **sp_helpdb**。

使用的表

master.dbo.spt_values、*master.dbo.sysattributes*、*sysdatabases*、*sysdevices*、*syslogins*、*sysmessages*、*syssegments*、*sysusages*

参见

命令	alter database、create database
系统过程	sp_configure、sp_dboption、sp_renamedb

sp_helpdevice

功能

报告有关特定设备或有关所有 Adaptive Server 数据库设备和转储设备的信息。

语法

```
sp_helpdevice [devname]
```

参数

devname — 是要报告其信息的设备的名称。如果省略该参数，**sp_helpdevice** 将报告所有设备的信息。

示例

1. sp_helpdevice

device_name	physical_name	description		
-----	-----	-----		
diskdump	null	disk, dump device		
master	d_master	special, default disk, dsync on, physical disk, 10 MB		
status	cntrltype	device_number	low	high
-----	-----	-----	----	-----
16	2	0	0	20000
3	0	0	0	5120

显示有关 Adaptive Server 上所有设备的信息。

2. sp_helpdevice diskdump

报告有关名为 *diskdump* 的转储设备的信息。

注释

- 当给定 *devname* 时，**sp_helpdevice** 将显示指定设备的信息；当没有给定参数时，它将显示 *master.dbo.sysdevices* 中所有设备的信息。
- *sysdevices* 表包含转储设备和数据库设备。
可以将数据库设备指定为缺省设备，这意味着它们可以用于数据库存储。当用户发出 **create database** 或 **alter database** 并且没有指定数据库设备名或给出关键字 **default** 时，将发生这种情况。要使数据库设备成为缺省数据库设备，可以执行系统过程 **sp_diskdefault**。
- 用 **disk init** 向系统中添加数据库设备。用 **sp_addumpdevice** 添加转储设备。

- “status” 列中的编号对应于 “description” 列中的状态说明。
“cntrltype” 列指定设备的控制器编号。磁盘或文件转储设备的 “cntrltype” 为 2，而磁带转储设备的 “cntrltype” 为 3 到 8。对于数据库设备，“cntrltype” 通常为 0（除非安装了具有特殊类型的磁盘控制器）。
转储设备的 “device_number” 列为 0，对于 master 数据库设备也为 0，而对于其它数据库设备则为 1 到 255 之间的数。如果设备编号大于 126，sp_helpdevice 可能会报告错误的负数编号。
“low” 和 “high ” 列表示虚拟页号，在 Adaptive Server 的所有设备中，每个虚拟页号都是唯一的。

权限

任何用户都可以执行 sp_helpdevice。

使用的表

master.dbo.spt_values、sysdevices、sysmessages

参见

命令	disk init、dump database、dump transaction、load database、load transaction
系统过程	sp_addumpdevice、sp_configure、sp_deviceattr、sp_diskdefault、sp_dropdevice、sp_helpdb、sp_logdevice、sp_who

sp_helpextendedproc

功能

显示当前数据库中的扩展存储过程 (ESP) 以及与其相关联的 DLL 文件。

语法

```
sp_helpextendedproc [esp_name]
```

参数

esp_name — 是扩展存储过程的名称。它必须是当前数据库中的过程。

示例

```
1. use sybssystemprocs  
go  
sp_helpextendedproc xp_cmdshell
```

```
ESP Name      DLL Name  
-----  
xp_cmdshell  sybsyesp
```

列出 **xp_cmdshell** ESP 以及存储其函数的 DLL 文件的名称。

```
2. sp_helpextendedproc
```

```
ESP Name      DLL Name  
-----  
xp_freedl     sybsyesp  
xp_cmdshell   sybsyesp
```

列出当前数据库中的所有 ESP 以及存储其函数的 DLL 文件的名称。

注释

- 如果省略 *esp_name*, **sp_helpextendedproc** 将列出数据库中的所有扩展存储过程。
- *esp_name* 区分大小写。它必须与创建该 ESP 所使用的 *esp_name* 匹配。

权限

只有系统管理员才能执行 **sp_helpextendedproc** 来查看数据库中的所有 ESP。所有用户都可以执行 **sp_helpextendedproc** 来查看自己拥有的或数据库所有者拥有的 ESP。

使用的表

master.dbo.syscomments、*sysobjects*

参见

命令	create procedure、drop procedure
系统过程	sp_addextendedproc、sp_dropextendedproc

sp_helpexternlogin

(仅适用于组件集成服务)

功能

报告有关外部登录名的信息。

语法

```
sp_helpexternlogin [remote_server [, login_name]]
```

参数

remote_server — 是远程服务器的名称，该服务器已经通过 **sp_addserver** 添加到本地服务器。

login_name — 是本地服务器上的登录账号。

示例

1. sp_helpexternlogin

显示所有远程服务器、本地登录名和外部登录。

2. sp_helpexternlogin SSB

显示名为 **SSB** 的服务器的本地登录名和外部登录。

3. sp_helpexternlogin NULL, milo

显示名为 “milo” 的用户的远程服务器、本地登录名和外部登录。

4. sp_helpexternlogin SSB, trixi

显示远程服务器 **SSB** 的外部登录，而远程服务器 **SSB** 的本地用户名为 “trixi”。

注释

- **sp_helpexternlogin** 显示所有远程服务器、用户的本地登录名和用户的外部登录名。
- 用 **sp_addserver** 添加远程服务器。用 **sp_addlogin** 添加本地登录。

权限

任何用户都可以执行 **sp_helpexternlogin**。

使用的表

master.dbo.syslogins、*master.dbo.sysattributes*、*master.dbo.sysservers*

参见

系统过程	sp_addexternlogin、sp_addlogin、 sp_addserver、sp_helpserver
------	--

sp_helpgroup

功能

报告当前数据库中有关特定组或有关所有组的信息。

语法

```
sp_helpgroup [grpname]
```

参数

grpname — 是数据库中用 **sp_addgroup** 创建的组的名称。

示例

1. **sp_helpgroup**

Group_name	Group_id
-----	-----
hackers	16384
public	0

显示当前数据库中所有组的信息。

2. **sp_helpgroup hackers**

Group_name	Group_id	Users_in_group	Userid
-----	-----	-----	-----
hackers	16384	ann	4
hackers	16384	judy	3

显示组 “hackers” 的信息。

注释

- 要获得有关缺省组 “public” 的报告，可以用单引号或双引号将名称 “public” 引起来（“public” 是保留字）。
- 如果指定的组中没有成员， **sp_helpgroup** 将显示标题，但不列出用户，如下所示：

Group_name	Group_id	Users_in_group	Userid
-----	-----	-----	-----

权限

任何用户都可以执行 **sp_helpgroup**。

使用的表

syssrvroles、 *sysusers*

参见

命令	grant、revoke
系统过程	sp_addgroup、sp_changegroup、 sp_dropgroup、sp_helprotect、sp_helpuser

sp_helpindex

功能

报告表上创建的索引的信息。

语法

```
sp_helpindex objname
```

参数

objname — 是当前数据库的表名。

示例

1. sp_helpindex sysobjects

index_name	index_description		
index_keys			
index_max_rows_per_page	index_fillfactor	index_reservepagegap	

sysobjects	clustered, unique located on system		
id			
	0	0	0
ncsysobjects	nonclustered, unique located on system		
name,uid			
	0	0	0

显示 *sysobjects* 表上索引的类型。

2. sp_helpindex titles

index_name	index_description		
index_keys			
index_max_rows_per_page	index_fillfactor	index_reservepagegap	

title_id_ix	nonclustered, unique located on default		
title_id			
	0	0	0
publ_ix	nonclustered located on default		
pub_id, pubdate DESC			
	0	0	8
title_ix	clustered, allow duplicate rows located on default		
title			
	0	90	0

publ_ix 上的索引是按 *pub_id* 的升序和 *pubdate* 的降序创建的。

注释

- **sp_helpindex** 将列出表上的所有索引，包括通过定义唯一或主键约束创建的索引，而唯一或主键约束是用 **create table** 或 **alter table** 语句定义的。
- **sp_helpindex** 显示所有指派给表上索引的特性（如缓存绑定）。
- **sp_helpindex** 显示：
 - 索引的 **max_rows_per_page** 设置。
 - 关于 DOL 锁定表上集群索引的信息
DOL 锁定表中集群索引的索引 ID (*indid*) 不等于 1。
 - 键的列顺序，指示它们是升序还是降序。
 - 空间管理属性值
 - 键列名后跟着的是其顺序。仅显示降序顺序。例如，如果在列 a ASC、b DESC 和 c ASC 上存在索引，“Index_keys”将显示“a, b DESC, c”。

权限

任何用户都可以执行 **sp_helpindex**。

使用的表

master.dbo.spt_values、*sysattributes*、*sysindexes*、*sysobjects*、*syssegments*

参见

命令	create index、drop index、update statistics
系统过程	sp_help、sp_helpkey

sp_helpjava

功能

显示有关数据库上安装的 **Java** 类和相关 **JAR** 的信息。

关于数据库中 **Java** 的详细信息，参见 *Adaptive Server Enterprise 中的 Java*。

语法

```
sp_helpjava ["class" [, java_class_name [, detail]] |
            "jar" [, jar_name] ]
```

参数

"class" | "jar" — 指定是否显示有关类或 **JAR** 的信息。“**class**”和“**jar**”都是关键字，所以要求带引号。

java_class_name — 要查看其信息的类的名称。该类必须是安装在数据库上的系统类或用户定义类。

detail — 指定要查看有关该类的详细信息。

jar_name — 要查看其信息的 **JAR** 的名称。该 **JAR** 必须用 **installjava** 安装在数据库中。

示例

1. sp_helpjava

显示安装在数据库上的所有类和相关 **JAR** 文件的名称。

2. sp_helpjava "class"

显示所有类的名称。

3. sp_helpjava "class" Address, detail

显示有关 **Address** 类的详细信息。例如：

```
Class
-----
Address

(1 row affected)
Class Modifiers
-----
public synchronized

Implemented Interfaces
-----
java.io.Serializable
```

Extended Superclass

java.lang.Object

Constructors

public Address()
public Address(java.lang.String, java.lang.String)

Methods

public final native java.lang.Class
java.lang.Object.getClass()
public native int java.lang.Object.hashCode()
public boolean
java.lang.Object.equals(java.lang.Object)
public java.lang.String
java.lang.Object.toString()
public final native void java.lang.Object.notify()
public final native void
java.lang.Object.notifyAll()
public final native void
java.lang.Object.wait(long) throws
java.lang.InterruptedException
public final void java.lang.Object.wait(long, int)
throws java.lang.InterruptedException
public final void java.lang.Object.wait() throws
java.lang.InterruptedException
public java.lang.String Address.display()
public void Address.removeLeadingBlanks()

Fields

public java.lang.String Address.street
public java.lang.String Address.zip**权限**任何用户都可以执行 **sp_helpjava**。**使用的表***sysjars*、*sysxtypes*

参见

命令	remove java
实用程序	extractjava、installjava

sp_helpjoins

功能

列出两个表或视图中适于连接的列。

语法

```
sp_helpjoins lefttab, righttab
```

参数

- lefttab* — 是第一个表或视图。
- righttab* — 是第二个表或视图。参数的顺序无关紧要。

示例

1. sp_helpjoins sales, salesdetail

a1	a2	b1	b2	c1	c2
d1	d2	e1	e2	f1	f2
g1	g2	h1	h2		

stor_id	stor_id	ord_num	ord_num	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL		

列出 *sales* 和 *salesdetail* 表中适于连接的列。

2. sp_helpjoins sysobjects, syscolumns

a1	a2	b1	b2	c1	c2	d1	d2	e1	e2
	f1	f2	g1	g2	h1	h2			

id	id	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
		NULL	NULL	NULL	NULL	NULL	NULL		

列出 *sysobjects* 和 *syscolumns* 系统表中适于连接的列。

注释

- **sp_helpjoins** 显示的列分别来自这两个表。**sp_helpjoins** 将检查当前数据库中的 **syskeys** 表，查看是否用 **sp_foreignkey** 在这两个表上定义了外键，然后查看是否用 **sp_commonkey** 在这两个表上定义了公用键。如果 **sp_helpjoins** 未找到任何外键或公用键，它将检查具有相同用户定义数据类型的键。如果仍找不到，它将检查具有相同名称和数据类型的列。
- **sp_helpjoins** 不创建任何连接。

权限

任何用户都可以执行 **sp_helpjoins**。

使用的表

syscolumns、*syskeys*、*sysobjects*

参见

系统过程	sp_commonkey 、 sp_foreignkey 、 sp_help 、 sp_helpkey 、 sp_primarykey
------	--

sp_helpkey

功能

报告有关特定表或视图中主键、外键或公用键的信息，或报告有关当前数据库中所有键的信息。

语法

```
sp_helpkey [tablename]
```

参数

tablename — 是当前数据库中某个表或视图的名称。如果不指定名称，此过程将报告当前数据库中定义的所有键。

示例

1. sp_helpkey

keytype	object	related_object	object_keys	related_keys
primary	authors	-- none --	au_id,*,*,*,*,*,*	*,*,*,*,*,*,*
foreign	titleauthor	authors	au_id,*,*,*,*,*,*	au_id,*,*,*,*,*,*,*

显示有关当前数据库中定义的键的信息。“object_keys”和“related_keys”列显示组成该键的列的名称。

注释

- **sp_helpkey** 列出引用 *tablename* 表的所有主键、外键和公用键定义的相关信息（如果省略 *tablename*，则列出有关数据库中所有键的信息）。使用 **sp_primarykey**、**sp_foreignkey** 和 **sp_commonkey** 系统过程可以定义这些键。
- **sp_helpkey** 不提供 **create table** 语句定义的 **unique** 或 **primary key** 完整性约束的有关信息。使用 **sp_helpconstraint** 可以确定为表定义的约束。
- 创建键使数据库中隐式的逻辑关系变成显式的，以便应用程序可以使用这些信息。

- 如果指定了对象名， **sp_helpkey** 将遵循查找对象的 **Adaptive Server** 规则：
 - 如果未指定所有者名称，而您拥有一个具有指定名称的对象，那么 **sp_helpkey** 将报告有关该对象的信息。
 - 如果未指定所有者名称，而您没有具有该名称的对象，但数据库所有者拥有这样的对象，那么 **sp_helpkey** 将报告有关数据库所有者的对象的信息。
 - 如果您和数据库所有者都没有具有指定名称的对象，即使在数据库中存在一个具有此对象名、但所有者不同的对象，**sp_helpkey** 仍会报告错误。
 - 如果您和数据库所有者都拥有具有指定名称的对象，而您想访问数据库所有者的对象，就需要按以下格式指定名称：
dbo.objectname。
- 对于除您自己和数据库所有者以外的其他数据库用户拥有的对象，请使用该所有者的名称限定这些对象，如“*mary.myproc*”。

权限

任何用户都可以执行 **sp_helpkey**。

使用的表

master.dbo.spt_values、 *syskeys*、 *sysobjects*

参见

命令	create trigger
系统过程	sp_commonkey、 sp_foreignkey、 sp_help、 sp_primarykey

sp_helplanguage

功能

报告有关特定替代语言或有关所有语言的信息。

语法

sp_helplanguage [*language*]

参数

language — 是要获取其相关信息的替代语言的名称。

示例

1. sp_helplanguage french

langid	dateformat	datefirst	upgrade	name
alias				
months				
shortmonths				
days				

1	dmy	1	0	french
	french			
	janvier,fevrier,mars,avril,mai,juin,juillet,aout,septembre,			
	octobre,novembre,decembre			
	jan,fev,mar,avr,mai,juin,juil,aou,sep,oct,nov,dec			
	lundi,mardi,mercredi,jeudi,vendredi,samedi,dimanche			

显示有关替代语言 “french” 的信息。

2. sp_helplanguage

显示有关所有已安装的替代语言的信息。

注释

- 如果指定了语言， **sp_helplanguage** 将报告有关该语言的信息；或者，如果未指定语言，该系统过程将报告有关 *master.dbo.syslanguages* 中所有语言的信息。

权限

任何用户都可以执行 **sp_helplanguage**。

使用的表

master.dbo.syslanguages

参见

系统过程	sp_addlanguage、sp_droplanguage、 sp_setlangalias
------	--

sp_helplog

功能

报告含有事务日志首页的设备的名称。

语法

sp_helplog

参数

无。

示例

1. **sp_helplog**
In database 'master', the log starts on device 'master'.

注释

- sp_helplog** 显示当前数据库中含有事务日志首页的设备的名称。

权限

任何用户都可以执行 **sp_helplog**。

使用的表

master.dbo.sysdevices、*master.dbo.sysusages*、*sysindexes*、*sysobjects*

参见

命令	alter database、create database
系统过程	sp_helpdevice、sp_logdevice

sp_helpobjectdef

(仅适用于组件集成服务)

功能

报告远程对象定义的所有者、对象和类型信息。

语法

```
sp_helpobjectdef [object_name]
```

参数

object_name — 是在 *sysattributes* 表中定义的对象名。 *object_name* 可以是以下格式之一：

- *dbname.owner.object*
- *dbname..object*
- *owner.object*
- *object*

dbname 和 *owner* 是可选的。 *object* 是必需的。如果未指定 *owner*, *owner* 将缺省为当前用户名。如果指定 *dbname*, 它必须是当前数据库, 而且必须指定 *owner* 或用占位符标记 *dbname..object*。请用引号将由多个部分组成的 *object_name* 引起。

示例

1. sp_helpobjectdef

显示当前数据库中的所有远程对象定义。

2. sp_helpobjectdef "dbo.tb1"

显示 *tb1* 表 (属于数据库所有者) 的远程对象定义。

注释

- 如果未指定 *object_name*, *sp_helpobjectdef* 将显示所有远程对象定义。
- *object_name* 参数中不允许使用服务器名。

权限

任何用户都可以执行 *sp_helpobjectdef*。

使用的表

sysattributes、*sysobjects*、*sysservers*、*spt_values*

参见

命令	create table、create existing table、drop table
系统过程	sp_addlogin、sp_addserver、sp_addobjectdef、 sp_defaultloc、sp_dropobjectdef、sp_help、 sp_helpserver

sp_help_qpgroup

功能

报告有关抽象计划组的信息。

语法

```
sp_help_qpgroup [ group [, mode ]]
```

参数

- group* — 是抽象计划组的名称。
- mode* — 是要打印的报告类型，为下列类型之一：

模式	返回的信息
full	组中的行数和计划数、使用两行或多行的计划数、最长的计划的行数和计划 ID、以及散列键数和散列键冲突信息。这是缺省报告模式。
stats	“full” 报告中除散列键信息之外的所有信息。
hash	组中的行数和抽象计划数、散列键数以及散列键冲突信息。
list	组中的行数和抽象计划数，以及每个查询/计划对的以下信息：散列键、计划 ID、查询的前几个字符和计划的前几个字符。
queries	组中的行数和抽象计划数，以及每个查询的以下信息：散列键、计划 ID、查询的前几个字符。
plans	组中的行数和抽象计划数，以及每个计划的以下信息：散列键、计划 ID、计划的前几个字符。
counts	组中的行数和抽象计划数，以及每个计划的以下信息：行数、字符数、散列键、计划 ID、查询的前几个字符。

示例

1. sp_help_qpgroup

Group	GID	Plans
-----	-----	-----
ap_stdin	1	0
ap_stdout	2	0
dev_test	3	209

报告有关数据库中所有抽象计划组的摘要信息。

2. **sp_help_qpgroup test_plans**

```
Query plans group 'test_plans', GID 8

Total Rows    Total QueryPlans
-----
                6                3
sysqueryplans rows consumption, number of query
plans per row count

Rows          Plans
-----
                2                3

Hashkeys
-----
                3
There is no hash key collision in this group.
报告有关 test_plans 组的信息。
```

注释

- 如果指定了抽象计划组的名称，但未指定 *mode* 参数，**sp_help_qpgroup** 的缺省模式是 **full**。
- 散列键冲突表示特定用户的多个计划具有相同的散列键值。发生散列键冲突时，必须对每个具有匹配散列键的查询的查询文本和用户的查询文本进行比较，才能标识出匹配的查询，所以性能将略有下降。

权限

任何用户都可以执行 **sp_help_qpgroup**。

使用的表

sysattributes、*sysqueryplans*

参见

系统过程	sp_help_qplan
------	----------------------

sp_help_qplan

功能

报告有关抽象计划的信息。

语法

```
sp_help_qplan id [, mode ]
```

参数

- id* — 是抽象计划的 ID。
- mode* — 是要打印的报告类型，为下列类型之一：

模式	返回的信息
full	计划 ID、组 ID 和散列键，以及完整的查询和计划文本。
brief	与 full 相同，但是只打印查询和计划的大约 80 个字符，而不是完整的查询和计划。这是缺省模式。
list	散列键、ID 以及查询和计划的前 20 个字符。

示例

```
1. sp_help_qplan 800005881

gid          hashkey      id
-----
          5  2054169974   937054374

query

-----
select type, avg(price) from titles group by type

plan

-----
( plan
  ( store Worktab1
    ( i_scan type_price titles )
  )
  ( t_scan ( ...
```

打印简要的抽象计划报告。

2. **sp_help_qplan 784005824, full**

打印完整的抽象计划报告。

注释

- 如果不指定 *mode* 参数的值，缺省值为 **brief**。

权限

任何用户都可以执行 **sp_help_qplan**，查看他或她拥有的查询的抽象计划。只有系统管理员和数据库所有者可以查看其他用户拥有的抽象计划。

使用的表

sysqueryplans

参见

系统过程	sp_find_qplan、sp_help_qpgroup
------	-------------------------------

sp_helpremotelogin

功能

报告有关特定远程服务器登录的信息，或有关所有远程服务器登录的信息。

语法

```
sp_helpremotelogin [remoteserver [, remotename]]
```

参数

remoteserver — 是要报告其远程登录信息的服务器的名称。

remotename — 是远程服务器上特定远程用户的名称。

示例

```
1. sp_helpremotelogin GATEWAY
```

显示远程服务器 GATEWAY 上所有远程用户的信息。

```
2. sp_helpremotelogin
```

显示本地服务器知道的远程服务器上所有远程用户的信息。

注释

- 如果指定了 *remoteserver* 参数，**sp_helpremotelogin** 将报告指定服务器的远程登录信息；如果未指定任何参数，该系统过程将报告所有服务器的远程登录信息。

权限

任何用户都可以执行 **sp_helpremotelogin**。

使用的表

master.dbo.spt_values、*master.dbo.sysmessages*、
master.dbo.sysremotelogins、*master.dbo.sysservers*、*sysobjects*

参见

系统过程	sp_addremotelogin、sp_dropremotelogin、 sp_helpserver
------	--

sp_help_resource_limit

功能

报告有关资源限制的信息。

语法

```
sp_help_resource_limit [name [, appname [, limittime  
                        [, limitday [, scope [, action]]]]]]
```

参数

name — 是受这些限制约束的 Adaptive Server 登录。要查看约束特定登录的限制的信息，需指定该登录的 **name**。要查看有关限制的信息而不考虑登录，需指定 **null**。

► 注意

如果您不是系统管理员，但要查看有关适用于您的资源限制的信息，需指定您自己的登录或指定登录为 **NULL**。

appname — 是受这些限制约束的应用程序的名称。要查看约束特定应用程序的限制的信息，需指定客户程序在登录包中传递给 Adaptive Server 的应用程序名。要查看有关限制的信息而不考虑应用程序，需指定 **null**。

limittime — 是限制有效的时间。要查看在指定时间有效的限制的信息，需指定 “00:00” 到 “23:59” 之间的时间，格式如下：

"HH:MM"

要查看有关限制的信息而不考虑时间，需指定 **null**。

limitday — 是限制有效的日期（星期几）。要查看在一星期中某一天有效的资源限制的信息，需使用缺省服务器语言（存储在 **master** 数据库的 **syslanguages** 系统表中）指定完整的日期（星期几）。要查看有关限制的信息而不考虑限制的有效日期，需指定 **null**。

scope — 是限制的范围。请指定以下范围代码之一：

范围代码	显示所有约束以下对象的限制的相关帮助信息
1	查询
2	查询批处理 （通过客户端发送到服务器的一个或多个 SQL 语句）
4	事务
6	查询批处理和事务
NULL	指定的 <i>name</i> 、 <i>appname</i> 、 <i>limittime</i> 、 <i>limitday</i> 和 <i>action</i> ，不考虑其 <i>scope</i>

action — 是超过限制时采取的操作。请指定以下操作之一：

操作代码	显示所有执行以下操作的限制的相关帮助信息
1	发出警告
2	中止查询批处理
3	中止事务
4	注销会话
NULL	约束指定的 <i>name</i> 、 <i>appname</i> 、 <i>limittime</i> 、 <i>limitday</i> 和 <i>scope</i> ，而不考虑它们执行的 <i>action</i>

示例

1. `sp_help_resource_limit`
列出所有存储在 `sysresourcelimits` 系统表中的资源限制。
2. `sp_help_resource_limit joe_user`
列出所有对用户 “joe_user” 的限制。
3. `sp_help_resource_limit NULL, my_app`
列出所有对应用程序 `my_app` 的限制。
4. `sp_help_resource_limit NULL, NULL, "09:00"`
列出所有在上午 9:00 有效的限制。
5. `sp_help_resource_limit @limittype = "09:00"`
另外一种列出所有在上午 9:00 有效的限制的方法。

6. `sp_help_resource_limit NULL, NULL, NULL, Monday`

列出所有在星期一有效的限制。

7. `sp_help_resource_limit joe_user, NULL, "09:00", Monday`

列出在星期一上午 9:00 对 “joe_user” 有效的限制。

注释

- `sp_help_resource_limit` 报告所有有关资源限制的信息，包括对指定登录或应用程序的限制，在指定的时间或一星期中某一天生效的限制，或具有指定范围或操作的限制。
- 有关资源限制的详细信息，参见 *系统管理指南*。

权限

任何用户都可以执行 `sp_help_resource_limit` 列出自己的资源限制。只有系统管理员才可以执行 `sp_help_resource_limit` 列出对其他用户的限制。

使用的表

master..sysresource_limits、*master..systimeranges*、*master..spt_limit_types*

参见

系统过程	<code>sp_add_resource_limit</code> 、 <code>sp_drop_resource_limit</code> 、 <code>sp_modify_resource_limit</code>
------	--

sp_helpprotect

功能

报告有关数据库对象、用户、组或角色具有的权限的信息。

语法

```
sp_helpprotect [name [, username [, "grant"  
                [,"none"|"granted"|"enabled"|"role_name"]]]]
```

参数

name — 是表、视图或存储过程的名称，或者是当前数据库中用户、用户定义角色或组的名称。如果没有提供名称，则 **sp_helpprotect** 报告数据库中的所有权限。

username — 是当前数据库中的用户名。

grant — 显示使用 **with grant option** 选项授予 **name** 的特权。

none — 确定所授权限时，忽略授予用户的角色。

granted — 确定所授权限时，包括授予用户的所有角色的信息。

enabled — 确定所授权限时，包括用户激活的所有角色的信息。

role_name — 只显示指定角色的权限信息，而不考虑是否已将此角色授予该用户。

示例

```
1. grant select on titles to judy  
   grant update on titles to judy  
   revoke update on titles(price) from judy  
   grant select on publishers to judy  
   with grant option
```


执行了这一系列 **grant** 和 **revoke** 语句后，执行 **sp_helpprotect titles** 将显示如下结果：

grantor	grantee	type	action	object	column	grantable
-----	-----	-----	-----	-----	-----	-----
dbo	judy	Grant	Select	titles	All	FALSE
dbo	judy	Grant	Update	titles	advance	FALSE
dbo	judy	Grant	Update	titles	notes	FALSE
dbo	judy	Grant	Update	titles	pub_id	FALSE
dbo	judy	Grant	Update	titles	pubdate	FALSE
dbo	judy	Grant	Update	titles	title	FALSE
dbo	judy	Grant	Update	titles	title_id	FALSE
dbo	judy	Grant	Update	titles	total_sales	FALSE
dbo	judy	Grant	Update	titles	type	FALSE
dbo	judy	Grant	Select	publishers	all	TRUE

```
2. grant select, update on titles(price, advance)
   to mary
   with grant option
go
sp_helpprotect titles
```

执行了此 **grant** 语句后，**sp_helpprotect** 将显示如下结果：

grantor	grantee	type	action	object	column	grantable
-----	-----	-----	-----	-----	-----	-----
dbo	mary	Grant	Select	titles	advance	TRUE
dbo	mary	Grant	Select	titles	price	TRUE
dbo	mary	Grant	Update	titles	advance	TRUE
dbo	mary	Grant	Update	titles	price	TRUE

```
3. sp_helpprotect judy
```

显示 “judy” 在数据库中具有的所有权限。

```
4. sp_helpprotect sysusers, csmith, null, doctor,
   "grant"
```

显示 “csmith” 在 **sysusers** 表中具有的所有权限，以及
“csmith” 是否可将权限授予其它用户（使用 **with grant option** 选项设置）。

grantor	grantee	type	action	object	column	grantable
-----	-----	-----	-----	-----	-----	-----
dbo	doctor	Grant	Delete	sysusers	All	FALSE
dbo	doctor	Grant	Insert	sysusers	All	FALSE
dbo	doctor	Grant	References	sysusers	All	FALSE
dbo	doctor	Grant	Select	sysattributes	All	FALSE

```
(1 row affected)
(return status = 0)
```

5. sp_helpprotect doctor_role

显示 **doctor** 角色在数据库中所具有的权限的信息。

grantor	grantee	type	action	object	column	grantable
dbo	doctor	Grant	Delete	sysusers	All	FALSE
dbo	doctor	Grant	Insert	sysusers	All	FALSE
dbo	doctor	Grant	References	sysusers	All	FALSE
dbo	doctor	Grant	Select	sysattributes	All	FALSE

(1 row affected)

(return status = 0)

6. sp_helpprotect sysusers, csmith, null, doctor_role, "granted"

显示授予 “csmith” 的所有角色的信息。

grantor	grantee	type	action	object	column	grantable
dbo	csmith	Grant	Update	sysusers	All	FALSE
dbo	doctor	Grant	Delete	sysusers	All	FALSE
dbo	doctor	Grant	Insert	sysusers	All	FALSE
dbo	doctor	Grant	References	sysusers	All	FALSE

(1 row affected)

(return status = 0)

7. sp_helpprotect sysattributes, rpillai, null, intern, "enabled"

显示授予 “rpillai” 的所有活动角色的信息。

grantor	grantee	type	action	object	column	grantable
dbo	public	Grant	Select	sysattributes	All	FALSE

(1 row affected)

(return status = 0)

注释

- **sp_helpprotect** 报告对数据库对象的权限。如果指定了 **username** 参数，则只报告该用户在数据库对象上的权限。如果 **name** 不是对象，**sp_helpprotect** 将检查它是用户、组，还是角色。如果是对象，**sp_helpprotect** 将列出用户、组或角色的权限。
- **sp_helpprotect** 只在当前数据库中查找对象和用户。
- 如果不指定可选值，例如 **granted**、**enabled**、**none** 或 **role_name**，**Adaptive Server** 将返回由当前指定的用户激活的所有角色的信息。
- 如果指定的用户不是当前用户，**Adaptive Server** 将返回授予指定用户的所有角色的信息。
- 如果指定的用户是某个组的成员，还将显示授予该组的权限。
- 授予权限时，系统管理员被视作对象所有者。如果授予权限的人是系统管理员，但对象属于其他用户，则 **sp_helpprotect** 的输出中显示的授权者是对象所有者的名称。

权限

任何用户都可以执行 **sp_helpprotect** 来查看自己的权限。只有系统安全人员才可以执行 **sp_helpprotect** 来查看授予其他用户的权限。

使用的表

master.dbo.spt_values、*syscolumns*、*sysobjects*、*sysprotects*、*sysusers*

参见

命令	grant、revoke、create、drop、set
系统过程	sp_help、sp_activeroles、sp_configure、sp_displaylogin、sp_displayroles、sp_modifylogin

sp_helpsegment

功能

报告有关当前数据库中特定段或所有段的信息。

语法

```
sp_helpsegment [segname]
```

参数

segname — 是要获取其相关信息的段的名称。如果省略该参数，将显示有关当前数据库中所有段的信息。

示例

1. sp_helpsegment

segment name	status
0 system	0
1 default	1
2 logsegment	0

报告当前数据库中所有段的信息。

2. sp_helpsegment order_seg

segment name	status
3 order_seg	0

device	size	free_pages
tpcd_data1	25.0MB	8176
tpcd_data2	25.0MB	8512
tpcd_data3	25.0MB	8392
tpcd_data4	25.0MB	8272
tpcd_data5	25.0MB	8448
tpcd_data6	25.0MB	8512

table_name	index_name	indid
orders	orders	0

total_size	total_pages	free_pages	used_pages
150.0MB	76800	50312	26488

报告有关名为 *order_seg* 的段的信息，包括哪些数据库表和索引使用该段，以及该段上的总页数、可用页数和已用页数。

3. **sp_helpsegment "default"**

报告有关 *default* 段的信息。必须用引号将关键字 **default** 引起。

4. **sp_helpsegment logsegment**

segment name		status	
2 logsegment		0	
device		size	free_pages
tpcd_log1		20.0MB	10200
table_name	index_name	indid	
syslogs	syslogs	0	
total_size	total_pages	free_pages	used_pages
20.0MB	10240	10200	40

报告存储事务日志的段的信息。

注释

- 如果指定了 *segname*，**sp_helpsegment** 将显示有关指定段的信息；如果未指定任何参数，该系统过程将显示有关当前数据库上所有段的信息。
- 首次创建数据库时，Adaptive Server 自动创建 *system*、*default* 和 *logsegment* 段。使用 **sp_addsegment** 可在当前数据库中添加段。
- *system*、*default* 和 *logsegment* 段分别编号为 0、1 和 2。
- “status” 列显示哪个段是缺省空间池。使用 **sp_placeobject** 命令，或 **create table** 或 **create index** 命令的 **on segment_name** 子句，可将对象放置在特定段上。
- 如果表没有集群索引，“indid” 列为 0；如果表具有集群索引，该列则为 1。

权限

任何用户都可以执行 **sp_helpsegment**。

使用的表

master.dbo.sysdevices、*master.dbo.sysusages*、*sysindexes*、*sysobjects*、*syssegments*

参见

命令	create index、create table
系统过程	sp_addsegment、sp_dropsegment、 sp_extendsegment、sp_helpdb、 sp_helpdevice、sp_placeobject

sp_helpserver

功能

报告有关特定远程服务器的信息，或有关所有远程服务器的信息。

语法

sp_helpserver [*server*]

参数

server — 是要获取其相关信息的远程服务器的名称。

示例

1. sp_helpserver GATEWAY

显示有关远程服务器 GATEWAY 的信息。

2. sp_helpserver SYB_BACKUP

name	network_name	status	id
SYB_BACKUP	SYB_BACKUP	timeouts, no net password encryption	1

显示有关本地 Backup Server 的信息。

3. sp_helpserver

显示本地服务器知道的所有远程服务器的信息。

注释

- **sp_helpserver** 报告有关 *master.dbo.sysservers* 中所有服务器的信息；或者，如果指定了 *server*，则报告有关特定远程服务器的信息。
- 如果安装了组件集成服务，**sp_helpserver** 将列出每个服务器的服务器类。

权限

任何用户都可以执行 **sp_helpserver**。

使用的表

master.dbo.spt_values、*master.dbo.sysservers*、*sysobjects*

参见

系统过程	sp_addserver、sp_dropserver、 sp_helpremotelogin、sp_serveroption
------	---

sp_helpsort

功能

显示 Adaptive Server 的缺省排序顺序和字符集。

语法

sp_helpsort

参数

无。

示例

1. **sp_helpsort**

对于一类（单字节）字符集，**sp_helpsort** 将显示服务器缺省排序顺序的名称、其字符集以及主排序值表。在 7 位终端上，显示如下：

```
Sort Order Description
-----
Character Set = 1, iso_1
      ISO 8859-1 (Latin-1) - Western European 8-bit character set.
Sort Order = 50, bin_iso_1
      Binary sort order for the ISO 8859/1 character set (iso_1).
Characters, in Order

-----
! " # $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ?
@ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [ \ ] ^ _
` a b c d e f g h i j k l m n o p q r s t u v w x y z { | } ~
! " # $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ?
@ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [ \ ] ^ _
` a b c d e f g h i j k l m n o p q r s t u v w x y z { | } ~
```

在 8 位终端上，显示如下：

Sort Order Description

```
-----
Character Set = 1, iso_1
    ISO 8859-1 (Latin-1) - Western European 8-bit character set.
Sort Order = 50, bin_iso_1
    Binary sort order for the ISO 8859/1 character set (iso_1).
Characters, in Order
```

```
-----
! " # $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ?
@ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [ \ ] ^ _
` a b c d e f g h i j k l m n o p q r s t u v w x y z { | } ~
ı ç £ ¤ ¥ ¦ § ¨ © ª « ¬ ® ¯ ° ± ² ³ ´ µ ¶ · ¸ ¹ º » ¼ ½ ¾
/4 ¿ À Á Â Ã Ä Å Æ Ç È É Ê Ë Ì Í Î Ï Ð Ñ Ò Ó Ô Õ Ö × Ø Ù Ú Û Ü Ý Þ à
á â ã ä å æ ç è é ê ë ì í î ï ð ñ ò ó ô õ ö ÷ ø ù ú û ü ý þ ÿ
```

对于二类（多字节）字符集，不列出字符，但显示对该字符集的说明。例如：

Sort Order Description

```
-----
Character Set = 140, euc_jis
    Japanese. Extended Unix Code mapping for JIS-X0201
    (hankaku katakana) and JIS-X0208 (double byte) roman,
    kana, and kanji.
Class 2 character set
Sort Order = 50, bin_eucjis
    Binary sort order for Japanese using the EUC JIS
    character set as a basis.
```

注释

- 缺省为二进制排序顺序。

权限

任何用户都可以执行 **sp_helpsort**。

使用的表

master.dbo.syscharsets、*master.dbo.syscurconfigs*、*sysobjects*

sp_helptext

功能

显示已编译对象的源文本。

语法

```
sp_helptext objname
```

参数

objname — 是要显示其源文本的已编译对象的名称。该已编译对象必须在当前数据库中。

示例

1. sp_helptext pub_idrule

```
# Lines of Text
-----
1
text
-----
create rule pub_idrule
as @pub_id in ("1389", "0736", "0877",
              "1622", "1756")
   or @pub_id like "99[0-9][0-9]"
```

显示 *pub_idrule* 的源文本。由于该规则在 *pubs2* 数据库中，因此应在 *pubs2* 中执行该命令。

2. sp_helptext sp_helptext

显示 *sp_helptext* 的源文本。由于系统过程存储在 *sybsystemprocs* 中，因此应在 *sybsystemprocs* 中执行该命令。

注释

- **sp_helptext** 显示已编译对象占用 *syscomments* 中的行数（每行的长度为 255 个字符），后面是已编译对象的源文本。
- **sp_helptext** 在当前数据库的 *syscomments* 表中查找源文本。
- 使用 **sp_hidetext** 可以对源文本进行加密。

- 系统安全人员可以防止大多数用户通过执行 **sp_helptext** 来查看已编译对象的源文本。要限制只有对象所有者或系统管理员才具有对 *syscomments* 表中 *text* 列的 **select** 权限，可以使用 **sp_configure** 将 **select on syscomments.text column** 参数设置为 **0**。要这样限制权限，需要在已评估的配置中运行 **Adaptive Server**。有关已评估的配置的详细信息，参见 *系统管理指南*。

权限

任何用户都可以执行 **sp_helptext**。

使用的表

syscomments、*sysobjects*

参见

命令	create default、create procedure、create rule、create trigger、create view
系统过程	sp_checksource、sp_help、sp_hidetext

sp_helpthreshold

功能

报告与当前数据库中所有阈值或特定段中所有阈值相关联的段、可用空间值、状态和存储过程。

语法

sp_helpthreshold [*segname*]

参数

segname — 是当前数据库中某个段的名称。

示例

- 1. **sp_helpthreshold logsegment**
显示日志段上的所有阈值。
- 2. **sp_helpthreshold**
显示当前数据库中所有段上的所有阈值。
- 3. **sp_helpthreshold "default"**
显示缺省段上的所有阈值。注意用引号引起保留词 “default”。

注释

- **sp_helpthreshold** 显示当前数据库中所有段的阈值信息。如果指定了段名，**sp_helpthreshold** 将列出该段的所有阈值。
- 对于最后机会阈值，*status* 列是 1；对于所有其它阈值，该列是 0。将事务日志存储在不同段上的数据库没有最后机会阈值。

权限

任何用户都可以执行 **sp_helpthreshold**。

使用的表

sysobjects、*syssegments*、*systhresholds*

参见

系统过程	sp_addthreshold 、 sp_droptreshold 、 sp_helpsegment 、 sp_modifythreshold 、 sp_thresholdaction
------	---

sp_helpuser

功能

报告有关当前数据库中特定用户、组或别名的信息，或有关当前数据库中所有用户的信息。

语法

sp_helpuser [*name_in_db*]

参数

name_in_db — 是当前数据库中的用户名。

示例

1. sp_helpuser

Users_name	ID_in_db	Group_name	Login_name
-----	-----	-----	-----
ann	4	hackers	ann
dbo	1	public	sa
guest	2	public	NULL
judy	3	hackers	judy

显示有关当前数据库中所有用户的信息。

2. sp_helpuser dbo

Users_name	ID_in_db	Group_name	Login_name
-----	-----	-----	-----
dbo	1	public	sa

Users aliased to user.
Login_name

andy
christa
howard
linda

显示有关数据库所有者（用户名为 “dbo”）的信息。

注释

- **sp_helpuser** 报告当前数据库中所有用户的信息。如果指定了 *name_in_db*, **sp_helpuser** 只报告指定用户的信息。
- 如果当前数据库的 **sysusers** 表中未列出指定的用户, **sp_helpuser** 将查看该用户是否是另一个用户的别名, 或者是组名。

权限

任何用户都可以执行 **sp_helpuser**。

使用的表

master.dbo.syslogins、 *sysalternates*、 *sysobjects*、 *sysusers*

参见

命令	grant、 revoke、 use
系统过程	sp_adduser、 sp_dropuser、 sp_help、 sp_helpgroup

sp_hidetext

功能

隐藏指定已编译对象的源文本。

语法

```
sp_hidetext [objname [, tabname [, username]]]
```

参数

objname — 指定要隐藏其源文本的已编译对象。

tabname — 指定要隐藏其源文本的表或视图的名称。

username — 指定用户名，该用户拥有要隐藏源文本的已编译对象。

示例

1. sp_hidetext

隐藏当前数据库中所有已编译对象的源文本。

2. sp_hidetext @objname = "sp_sort_table", @username = "Mary"

隐藏 Mary 拥有的用户定义存储过程 **sp_sort_table** 的源文本。

3. sp_hidetext "pr_phone_list"

隐藏存储过程 **pr_phone_list** 的源文本。

4. sp_hidetext @tabname = "my_tab"

隐藏在表 *my_tab* 上定义的所有检查约束、缺省值和触发器的源文本。

5. sp_hidetext "my_vu", "my_tab"

隐藏在表 *my_tab* 上定义的所有检查约束、缺省值和触发器的源文本，以及视图 *my_vu* 的源文本。

6. sp_hidetext @username = "Tom"

隐藏 Tom 拥有的所有已编译对象的源文本。

注释

- **sp_hidetext** 隐藏指定已编译对象的源文本。

◆ 警告!

执行 **sp_hidetext** 之前，确保已备份源文本。执行 **sp_hidetext** 的结果不能被撤消。

- 如果不指定任何参数，**sp_hidetext** 将隐藏当前数据库中所有已编译对象的源文本。
- 有关隐藏源文本的详细信息，参见 *Transact-SQL User's Guide*。

权限

任何用户都可以使用 **sp_hidetext** 隐藏自己的已编译对象的源文本。只有数据库所有者或系统管理员才可以隐藏其他用户拥有的已编译对象的源文本，或使用不带参数的 **sp_hidetext**。

使用的表

syscolumns、*syscomments*、*sysconstraints*、*sysobjects*、*sysprocedures*

参见

系统过程	sp_checksourc e
------	------------------------

sp_import_qpgroup

功能

将抽象计划从用户表导入抽象计划组。

语法

```
sp_import_qpgroup tab, usr, group
```

参数

tab — 是要复制的计划所在的表的名称。可以指定数据库名称，但不能指定所有者名称，其格式是 *dbname..tablename*。总长度必须等于或小于 30 个字符。

usr — 是在导入抽象计划后，要将其 ID 指派到这些抽象计划的用户的名称。

group — 是包含要导入的计划的抽象计划组的名称。

示例

```
1. sp_import_qpgroup moveplans, dbo, new_plans
```

将计划从表 *moveplans* 复制到 *new_plans* 组，并将数据库所有者的用户 ID 指派给这些计划。

注释

- **sp_import_qpgroup** 将计划从用户表复制到 *sysqueryplans* 中的抽象计划组。与 **sp_export_qpgroup** 结合使用，可在服务器和数据库之间复制抽象计划组，或复制属于一个用户的计划并为其指派另一个用户 ID。
- 如果执行 **sp_import_qpgroup** 时没有该抽象计划组，此过程将创建该组。
- 如果在执行 **sp_import_qpgroup** 时有抽象计划组，该组不能包含指定用户的计划。**sp_import_qpgroup** 不会检查查询文本来确定该组中是否已经存在查询。如果需要将某个用户的计划导入一个抽象计划组，而该组中已经存在该用户的计划，请执行以下操作：
 - 使用 **sp_import_qpgroup** 将计划导入到一个新计划组中。
 - 使用 **sp_copy_all_qplans** 将计划从新创建的组复制到目标组中。**sp_copy_all_qplans** 将检查查询以确保不会创建重复的查询。
 - 如果您不再需要导入时所创建的组，可以使用 **sp_drop_all_qplans** 删除该组中的计划，然后用 **sp_drop_qpgroup** 删除该组。

- 要创建一个空表来批量复制抽象计划，请使用以下语句：

```
select * into load_table
from sysqueryplans
where 1 = 2
```

权限

只有系统管理员或数据库所有者才能执行 **sp_import_qpgroup**。

使用的表

sysattributes、*sysqueryplans*

参见

命令	create plan
系统过程	sp_copy_all_qplans、sp_copy_qplan、 sp_export_qpgroup、sp_help_qpgroup

sp_indsuspect

功能

检查用户表中是否有索引在排序顺序更改后的恢复过程中被标记为可疑索引。

语法

sp_indsuspect [*tab_name*]

参数

tab_name — 是要检查的用户表的名称。

示例

1. **sp_indsuspect newaccts**

检查表 *newaccts* 中是否存在被标记为可疑的索引。

注释

- 不带参数的 **sp_indsuspect** 将创建一个列表，列出当前数据库中所有包含因排序顺序更改而需要重建的索引的表。如果指定 *tab_name* 参数， **sp_indsuspect** 将检查指定的表中是否有索引在排序顺序更改后的恢复过程中被标记为可疑索引。
- 使用 **sp_indsuspect** 将列出所有可疑索引。表的所有者或系统管理员可以使用 **dbcc reindex** 检查所列出的索引的完整性，并根据需要重建这些索引。

权限

任何用户都可以执行 **sp_indsuspect**。

使用的表

sysindexes、*sysobjects*、*sysusers*

参见

命令	dbcc
----	------

sp_listsuspect_db

功能

列出所有当前具有脱机页（因在恢复过程中检测到损坏）的数据库。

语法

```
sp_listsuspect_db
```

参数

无。

示例

1. sp_listsuspect_db

列出具有可疑页的数据库。

注释

- **sp_listsuspect_db** 列出数据库名、可疑页的数量和含有可疑页的对象数。
- 使用 **sp_listsuspect_page** 可标识可疑页。

权限

任何用户都可以执行 **sp_listsuspect_db**。

使用的表

```
master.dbo.sysattributes
```

参见

系统过程	sp_forceonline_db、sp_forceonline_page、 sp_listsuspect_page、 sp_setsuspect_granularity、 sp_setsuspect_threshold
------	---

sp_listsuspect_object

功能

列出数据库中所有因在恢复过程中检测到损坏而当前处于脱机状态的索引。

语法

sp_listsuspect_object [*dbname*]

参数

dbname — 是数据库的名称。

示例

- 1. **sp_listsuspect_object**
列出当前数据库中的可疑索引。
- 2. **sp_listsuspect_object pubs2**
列出 *pubs2* 数据库中的可疑索引。

注释

- 如果 DOL 锁定表上的索引有可疑页，则恢复期间整个索引都将脱机。查询优化程序不考虑脱机索引。
- 使用系统过程 **sp_forceonline_object** 使脱机索引联机，以便于修复。
- **allpage** 锁定表上的索引在恢复期间不完全脱机，只有这些索引的个别页脱机。可使用 **sp_forceonline_page** 将这些页联机。
- **sp_listsuspect_object** 列出指定数据库或当前用户数据库（如果忽略 *dbname*）中的数据库名、对象 ID、对象名、索引 ID 和每个可疑索引的访问状态。
- **access** 列中 **SA_ONLY** 值意味着索引已被强制联机，且只有系统管理员可使用。**BLOCK_ALL** 的值意味着索引已对每个人脱机。
- 有关恢复故障隔离的详细信息，参见 *系统管理指南*。

权限

任何用户都可以执行 **sp_listsuspect_object**。

参见

系统过程	sp_forceonline_object
------	------------------------------

sp_listsuspect_page

功能

列出数据库中所有因在恢复过程中检测到损坏而当前处于脱机状态的页。

语法

```
sp_listsuspect_page [dbname]
```

参数

dbname — 是数据库的名称。

示例

1. **sp_listsuspect_page**

列出当前数据库中的可疑页。

2. **sp_listsuspect_page pubs2**

列出 *pubs2* 数据库中的可疑页。

注释

- 对于指定的数据库或当前用户数据库（如果省略了 *dbname*）中的每个可疑页， **sp_listsuspect_page** 将列出数据库名、页 ID、对象、索引 ID 和访问状态。
- 如果 “access” 列的值为 **SA_ONLY**，表示该页已被强制联机，且只有系统管理员可使用。如果值为 **BLOCK_ALL**，表示该页对每个人均已脱机。

权限

任何用户都可以执行 **sp_listsuspect_page**。

使用的表

master.dbo.sysattributes

参见

系统过程	sp_forceonline_db、 sp_forceonline_page、 sp_listsuspect_db、 sp_setsuspect_granularity、 sp_setsuspect_threshold
------	---

sp_lock

功能

报告关于当前持有锁的进程的信息。

语法

sp_lock [*spid1* [, *spid2*]]

参数

spid1 — 是 *master.dbo.sysprocesses* 表中的 Adaptive Server 进程 ID 号。运行 **sp_who** 可获得锁定进程的 *spid*。

spid2 — 是要进行锁检查的另一个 Adaptive Server 进程 ID 号。

示例

1. sp_lock

The class column will display the cursor name for locks associated with a cursor for the current user and the cursor id for other users.

fid	spid	locktype	table_id	page	dbname	class	context
0	7	Sh_intent	480004741	0	master	Non Cursor Lock	NULL
0	18	Ex_intent	16003088	0	pubtune	Non Cursor Lock	NULL
0	18	Ex_page	16003088	587	pubtune	Non Cursor Lock	NULL
0	18	Ex_page	16003088	590	pubtune	Non Cursor Lock	NULL
0	18	Ex_page	16003088	1114	pubtune	Non Cursor Lock	NULL
0	18	Ex_page	16003088	1140	pubtune	Non Cursor Lock	NULL
0	18	Ex_page	16003088	1283	pubtune	Non Cursor Lock	NULL
0	18	Ex_page	16003088	1362	pubtune	Non Cursor Lock	NULL
0	18	Ex_page	16003088	1398	pubtune	Non Cursor Lock	NULL
0	18	Ex_page-blk	16003088	634	pubtune	Non Cursor Lock	NULL
0	18	Update_page	16003088	1114	pubtune	Non Cursor Lock	NULL
0	18	Update_page-blk	16003088	634	pubtune	Non Cursor Lock	NULL
0	23	Sh_intent	16003088	0	pubtune	Non Cursor Lock	NULL
0	23	Sh_intent	176003658	0	pubtune	Non Cursor Lock	NULL
0	23	Ex_intent	208003772	0	pubtune	Non Cursor Lock	NULL
1	1	Sh_intent	176003658	0	tpcd	Non Cursor Lock	Sync-pt
duration request							
1	1	Sh_intent-blk	208003772	0	tpcd	Non Cursor Lock	Sync-pt

duration request						
1	8	Sh_page	176003658	41571	tpcd	Non Cursor Lock NULL
1	9	Sh_page	176003658	41571	tpcd	Non Cursor Lock NULL
1	10	Sh_page	176003658	41571	tpcd	Non Cursor Lock NULL
11	11	Sh_intent	176003658	0	tpcd	Non Cursor Lock Sync-pt
duration request						
11	12	Sh_page	176003658	41571	tpcd	Non Cursor Lock NULL
11	13	Sh_page	176003658	41571	tpcd	Non Cursor Lock NULL
11	14	Sh_page	176003658	41571	tpcd	Non Cursor Lock NULL

此示例显示 *spid* 为 7、18 和 23 的串行进程以及两个进程系列的锁定状态。*fid* 为 1 的系列具有 *spid* 为 1 的协调进程和 *spid* 为 8、9 和 10 的工作进程。*fid* 为 11 的系列具有 *spid* 为 11 的协调进程和 *spid* 为 12、13 和 14 的工作进程。

2. sp_lock 7

The class column will display the cursor name for locks associated with a cursor for the current user and the cursor id for other users.

fid	spid	locktype	table_id	page	dbname	class	context
0	7	Sh_intent	480004741	0	master	Non Cursor Lock	NULL

显示当前由 *spid* 7 所持有的锁的有关信息。

注释

- 如果不指定参数，**sp_lock** 将报告当前持有锁的所有进程的信息。
- 用户只能通过在 **select** 语句中使用 **holdlock** 关键字对锁定进行控制。
- 使用 **object_name** 系统函数可以从表的 ID 号中派生其名称。
- **sp_lock** 的输出将按照 *fid* 及 *spid* 进行排序。
- *loid* 列标识阻塞事务的唯一锁所有者 ID。如果 *loid* 值为偶数，表示本地事务拥有该锁。如果值为奇数，表示外部事务拥有该锁。
- *locktype* 列显示该锁为共享锁（前缀为 “Sh”）、排它锁（前缀为 “Ex”）还是更新锁，以及锁定的是表（“table” 或 “intent”）还是页（“page”）。

“locktype” 列中的 “blk” 后缀表示该进程阻塞了另一个需要获取锁的进程。一旦该进程结束，其它进程将继续执行。
“locktype” 列中的 “demand” 后缀表示该进程正在试图获取一个排它锁。有关锁类型的详细信息，参见 *Performance and Tuning Guide*。

- **class** 列表示锁是否与游标相关联。它显示下列内容之一：
 - “Non Cursor Lock” 表示锁不与游标相关联。
 - “Cursor Id number” 表示锁与该 Adaptive Server 进程 ID 的游标 ID 号相关联。
 - 游标名表示锁与游标 **cursor_name** 相关联，而该游标属于执行 **sp_lock** 的当前用户。
- **fid** 列标识锁所属的系列（包括协调进程及其工作进程）。**fid** 的值为：
 - 如果值为零，表示 **spid** 代表的任务是串行执行的。它不参与并行执行。
 - 如果值不为零，表示持有锁的任务 (**spid**) 属于并行执行语句的进程系列（由 **fid** 标识）。如果值与 **spid** 相等，则表示该任务是并行执行查询的系列中的协调进程。
- **context** 列标识锁的环境。相同系列中的工作进程具有相同的环境值。“context” 的合法值如下所示：
 - “NULL” 表示持有该锁的任务是一个串行执行的查询，或在事务隔离级别 1 中并行执行的查询。
 - “Sync-pt duration request” 表示在该查询完成之前，持有该锁的任务将一直持有该锁。
 在以下条件下，锁的环境可以是 “Sync-pt duration request”：锁是被作为并行查询的一部分而持有的表锁；锁在事务隔离级别 3 上被工作进程持有；或者锁被并行查询中的工作进程持有，并且必须在事务的持续期间内被持有。
 - “Ind pg” 表示索引页上的锁（仅适用于 **allpage** 锁定表）
 - “Inf key” 表示无限键锁定（用于 DOL 锁定表上事务隔离级别为 3 的某些域查询）
 - “Range” 表示域锁定（用于 DOL 锁定表上事务隔离级别为 3 的某些域查询）

只要适用，这些新值能以组合的方式显示，包括与 “Fam dur”（代替 “Sync pt duration”）的组合以及相互之间的组合。
- **row** 列显示行级锁的行号。
- **sp_lock** 输出还显示下列锁类型：
 - “Sh_row” 表示共享行锁
 - “Update_row” 表示更新行锁
 - “Ex_row” 表示排它行锁

权限

任何用户都可以执行 **sp_lock**。

使用的表

master.dbo.spt_values、*master.dbo.syslocks*、*master.dbo.sysobjects*

参见

命令	kill、select
系统过程	sp_familylock、sp_who

sp_locklogin

功能

锁定一个 **Adaptive Server** 帐号从而使该用户不能登录，或者显示所有锁定帐号的列表。

语法

```
sp_locklogin [loginame, "{lock | unlock}"]
```

参数

loginame — 是要锁定或解锁的帐号的名称。

lock | unlock — 指定锁定该帐号，还是解锁该帐号。

示例

1. **sp_locklogin charles, "lock"**

锁定用户 “charles” 的登录帐号。

2. **sp_locklogin**

显示所有锁定帐号的列表。

注释

- 锁定 **Adaptive Server** 登录帐号后，该用户将无法登录。使用 **sp_locklogin** 而不使用 **sp_droplogin** 有以下几个原因：
 - 如果登录是数据库中的用户，则不能删除该登录，而且，如果用户拥有该数据库中的对象或者曾授予其他用户对这些对象的权限，就不能从数据库中删除该用户。
 - 创建了下一个登录帐号后，**Adaptive Server** 就可以重新使用已删除登录帐号的服务器用户 **ID (suid)**。这只会出现在被删除的登录持有 **syslogins** 中的最高 **suid** 的情况下才会出现；然而，如果未对 **sp_droplogin** 的执行进行审计，则可能会降低可靠性。另外，拥有重新使用的 **suid** 的用户实际上也可以访问授权给旧 **suid** 的数据库对象。
 - 不能删除剩下的最后一个系统安全员或系统管理员的登录帐号。
- 不带参数的 **sp_locklogin** 将返回所有锁定帐号的列表。
- 可以锁定当前登录的某个帐号。该用户会收到一条警告，显示其帐号已被锁定，但是在其注销之前仍可以使用该帐号。

- 锁定的帐号可以被指定为数据库所有者，并且可以拥有任一数据库中的对象。
- 锁定某个已经被锁定的帐号，或者解锁某个未锁定的帐号不会起任何作用。
- 锁定某个系统安全员的登录帐号时，**sp_locklogin** 会校验至少有另一个未锁定的系统安全员存在。同样，**sp_locklogin** 会校验总是至少有一个未锁定的系统管理员帐号。如果试图锁定剩下的最后一个未锁定的系统管理员或系统安全员帐号，**sp_locklogin** 将返回一条错误消息并失败。

权限

只有系统管理员或系统安全员才可以执行 **sp_locklogin**。

使用的表

master.dbo.sysloginroles、*master.dbo.syslogins*、
master.dbo.sysprocesses、*sysobjects*

参见

系统过程	sp_addlogin 、 sp_modifylogin 、 sp_password
------	---

sp_logdevice

功能

对于日志和数据保存在同一设备上的数据库，将其事务日志移动到一个单独的数据库设备上。

语法

```
sp_logdevice dbname, devname
```

参数

dbname — 是要将其 *syslogs* 表（含有事务日志）存放到某个逻辑设备上的数据库的名称。

devname — 是要存放 *syslogs* 表的设备的逻辑名。这个设备必须是与该数据库相关联的数据库设备（在 **create database** 或 **alter database** 中指定的设备）。要获得该数据库的设备的信息，请运行 **sp_helpdb**。

示例

```
1. create database products on default = 10, logs = 2  
go  
sp_logdevice products, logs  
go
```

创建数据库 *products*，并将表 *products.syslogs* 存放到数据库设备 *logs* 上。

```
2. alter database test log on logdev  
go  
sp_logdevice test, logdev  
go
```

对于日志和数据保存在同一设备上的数据库 *test*，将其日志存放到日志设备 *logdev* 上。

注释

- **sp_logdevice** 过程只影响将来对 *syslogs* 的空间分配。这将会创建一个易损性窗口，在此期间日志的前几页将与数据保存在同一设备上。因此，要将事务日志存放到单独的设备上，首选方法是在 **create database** 中使用 **log on** 选项，这样会立即将整个事务日志存放到单独的设备上。

- 将事务日志存放到单独的数据库设备上，不但可以便于进行恢复，而且可以提高性能。
如果数据库非常小，而且不是关键数据库，可以将其日志和其余部分存放在一起。这样的数据库使用 **dump database** 来备份数据库和日志，并使用 **dump transaction with truncate_only** 来截断日志。
- 在执行下一个 **dump transaction** 之前，执行 **dbcc checkalloc** 和 **sp_helplog** 仍将显示 *syslogs* 中分配到数据库设备上的某些页。之后，在执行 **sp_logdevice** 时，事务日志将全部被传送到所指定的设备。
- 事务日志所需的设备大小根据更新活动的数量和事务日志转储的频率而有所不同。一般来说，应将分配给数据库自身的空间的 10% 到 25% 分配给日志设备。
- 只应对日志和数据存储在单一设备上的数据库使用 **sp_logdevice**。不能对日志和数据存储在不同设备上的数据库使用 **sp_logdevice**。
- 要增加分配给事务日志的存储空间，请使用 **alter database**。如果在 **create database** 中使用 **log on** 选项来将事务日志存放到单独的设备上，请使用：
sp_extendsegment segname, devname
来增加事务段的大小。如果未使用 **log on**，请执行 **sp_logdevice**。
存放 *syslogs* 的设备或段只用于存放 *syslogs* 表。要增加分配给数据库其余部分的存储空间，请在执行 **alter database** 命令时指定除日志设备外的其它设备。
- 请使用 **disk init** 命令为数据库或事务日志初始化新的数据库设备。
- 详细信息，参见 *系统管理指南*。

权限

只有数据库所有者或系统管理员才可以执行 **sp_logdevice**。

使用的表

master.dbo.sysdatabases、*master.dbo.sysdevices*、*master.dbo.sysusages*、*sysobjects*

参见

命令	alter database 、 create database 、 dbcc 、 disk init 、 dump database 、 dump transaction 、 select
系统过程	sp_extendsegment 、 sp_helpdevice

sp_loginconfig

(仅适于 Windows NT)

功能

显示一个或所有集成安全参数的值。

语法

```
sp_loginconfig ["parameter_name"]
```

参数

parameter_name — 是要检查的集成安全参数的名称。这些值为: **login mode**、**default account**、**default domain**、**set host**、**key _**、**key \$**、**key @**、和 **key #**。

示例

1. sp_loginconfig

name	config_item

login mode	standard
default account	NULL
default domain	NULL
set host	false
key _	domain separator
key \$	space
key @	space
key #	-

显示所有集成安全参数的值。

2. sp_loginconfig "login mode"

name	config_item

login mode	standard

显示 **login mode** 安全参数的值。

注释

- 集成安全参数的值存储在 Windows NT 注册表中。有关更改这些参数的说明，参见 *Configuring Adaptive Server for Windows NT* 中有关登录安全章节。
- **sp_loginconfig** 显示在启动了 Adaptive Server 之后有效的 *config_item* 值。如果在启动 Adaptive Server 之后更改了注册表的值，那么这些值将不会在 **sp_loginconfig** 的输出结果中反映出来。

权限

只有系统管理员才可以执行 **sp_loginconfig**。

使用的表

sysobjects

参见

系统过程	sp_revokelogin
------	-----------------------

sp_logininfo

(仅适于 Windows NT)

功能

显示使用 **sp_grantlogin** 授予 Windows NT 用户和组的所有角色。

语法

sp_logininfo ["*login_name*" | "*group_name*"]

参数

login_name — 是 Windows NT 用户的网络登录名。

group_name — 是 Windows NT 组名。

示例

1. sp_logininfo regularjoe

account name	mapped login name	type	privilege
HAZE\regularjoe	HAZE_regularjoe	user	'oper_role'

显示授予 Windows NT 用户 “regularjoe” 的权限。

2. sp_logininfo

account name	mapped login name	type	privilege
BUILTIN\Administrators	BUILTIN\Administrators	group	'sa_role sso_role oper_role sybase_ts_role navigator_role replication_role'
HAZE\regularjoe	HAZE_regularjoe	user	'oper_role'
PCSRE\randy	PCSRE_alexander	user	'default'

显示使用 **sp_grantlogin** 授予 Windows NT 用户和组的所有权限。

注释

- **sp_logininfo** 显示使用 **sp_grantlogin** 授予 Windows NT 用户和组的所有角色。
- 指定 Windows NT 用户名和组名时，可以省略域名和域分隔符 (\)。

权限

只有系统管理员才可以执行 **sp_logininfo**。

使用的表

sysobjects

参见

命令	grant、setuser
系统过程	sp_displaylogin、sp_grantlogin、 sp_revokelogin、sp_role、sp_who

sp_logiosize

功能

在为当前数据库的事务日志执行 I/O 操作时，将 Adaptive Server 使用的日志 I/O 大小更改为其它的内存池。

语法

```
sp_logiosize ["default" | "size" | "all"]
```

参数

default — 将当前数据库的日志 I/O 大小设置为 Adaptive Server 的缺省值 (4K)，如果缓存中有大小为 4K 的内存池可用。否则，Adaptive Server 会把日志 I/O 大小设置为 2K。由于 **default** 是关键字，所以在指定该参数时需要使用引号。

size — 是要为当前数据库的日志 I/O 设置的大小。其值可以为 2、4、8 和 16。必须用引号将该值引起来。

all — 显示为所有数据库配置的日志 I/O 大小（数据库按照缓存名分组）。

示例

1. sp_logiosize

```
The transaction log for database 'master' will use  
I/O size of 2 Kbytes.
```

显示为当前数据库配置的日志 I/O 大小。

2. sp_logiosize "8"

将当前数据库的日志 I/O 大小更改为使用大小为 8K 的内存池。如果数据库事务日志绑定的缓存中没有大小为 8K 的内存池，Adaptive Server 会返回一条错误消息，指出不存在这样的内存池，而当前的日志 I/O 大小不会被更改。

3. sp_logiosize "default"

将当前数据库的日志 I/O 大小更改为 Adaptive Server 的缺省值 (4K)。如果该事务日志使用的缓存没有大小为 4K 的内存池，Adaptive Server 将使用大小为 2K 的内存池。

4. sp_logiosize "all"

Cache name:default data cache	
Data base	Log I/O Size
-----	-----
master	2 Kb
tempdb	2 Kb
model	2 Kb
sybsystemprocs	2 Kb
pubs3	2 Kb
pubtune	2 Kb
dbccdb	2 Kb
sybsyntax	2 Kb

显示为所有数据库配置的日志 I/O 大小。

注释

- **sp_logiosize** 显示或更改当前数据库的日志 I/O 大小。任何用户都可以执行 **sp_logiosize** 来显示所配置的日志 I/O 大小。只有系统管理员才可以更改日志 I/O 大小。
- 如果使用不带参数的 **sp_logiosize**， Adaptive Server 将显示当前数据库的日志 I/O 大小。
- 对 I/O 大小的更改将会立即生效。 Adaptive Server 在 *sysattributes* 表中记录数据库的新 I/O 大小。
- 为 **sp_logiosize** 指定的任何值都必须与一个为数据库事务日志所用缓存配置的现有内存池相对应。使用 **sp_poolconfig** 系统过程可以指定这些池。

如果缓存中有一个大小为 4K 的内存池可用， Adaptive Server 会将数据库的缺省日志 I/O 大小定义为 4K。否则， Adaptive Server 会将日志 I/O 大小设置为 2K（在任一缓存中总是会存在一个 2K 大小的内存池存在）。对多数工作负载而言， 4K 大小的日志 I/O 性能要比 2K 的好得多，因此事务日志使用的每个缓存都应该有大小为 4K 的内存池。有关配置缓存和内存池的详细信息，参见 *系统管理指南* 和 *Performance and Tuning Guide*。

- 如果一个或多个数据库的事务日志绑定到类型为 **logonly** 的缓存，该缓存中所有 I/O 大小超出为这些数据库定义的日志 I/O 大小的内存池将不会被使用。

例如，假定只有两个数据库将其日志绑定到一个包含 2K、 4K 和 8K 内存池的 “log only” 缓存。在缺省情况下， **sp_logiosize** 将这些参数的日志 I/O 大小设置为 4K，而不使用 8K 的内存池。因此，为了避免浪费缓存空间，在配置日志 I/O 大小时一定要谨慎。

- 在恢复过程中，不管为数据库配置的日志 I/O 大小如何，只有缺省缓存中大小为 **2K** 的内存池处于活动状态。事务日志会被读入缺省缓存的 **2K** 内存池中，而所有必须被回退或前滚的事务都会将数据页读入缺省数据缓存中。

权限

只有系统管理员才可以执行 **sp_logiosize** 来更改当前数据库的日志 I/O 大小。任何用户都可以执行 **sp_logiosize** 来显示日志 I/O 大小的值。

使用的表

sysattributes

参见

系统过程	sp_cacheconfig、sp_poolconfig
------	------------------------------

sp_modifylogin

功能

修改 Adaptive Server 登录帐户的缺省数据库、缺省语言、缺省激活角色或全名。

语法

```
sp_modifylogin account, column, value
```

参数

account — 是要修改的登录帐户。

column — 指定要更改的选项的名称。这些选项包括：

选项	定义
defdb	用户在登录时连接到的“主”数据库。
deflanguage	用户缺省语言的正式名称。
fullname	用户的全名。
"add default role"	登录时缺省激活的角色。
"drop default role"	要从登录时缺省激活的角色列表中删除的角色。

value — 是指定选项的新值。

示例

1. `sp_modifylogin sarah, defdb, "pubs2"`
将“sarah”的缺省数据库更改为 *pubs2*。
2. `sp_modifylogin claire, deflanguage, "french"`
将“claire”的缺省语言设置为法语。
3. `sp_modifylogin clemens, fullname, "Samuel Clemens"`
将用户“clemens”的全名更改为“Samuel Clemens”。
4. `sp_modifylogin csmith, "add default role", specialist_role`
将 *specialist* 角色添加到用户 *csmith* 登录时缺省激活的角色列表。
5. `sp_modifylogin hpillai, "drop default role", intern_role`
将 *intern* 角色从用户“hpillai”登录时缺省激活的角色列表中删除。

注释

- 当首次向 **Adaptive Server** 添加用户的登录时，应使用 **sp_modifylogin** 或 **sp_addlogin** 设置缺省数据库、语言或全名。
 - 如果不指定缺省数据库，用户的缺省数据库为 *master*。
 - 如果不指定语言，用户的缺省语言将设置为服务器的缺省语言。
 - 如果不指定全名，*syslogins* 中的该列将留为空白。

更改用户的缺省数据库

- 当通过执行 **sp_modifylogin** 更改用户的缺省数据库后，用户在下次登录时将连接到新的 *defdb*。然而，如果数据库所有者未通过 **sp_adduser** 或 **sp_addalias** 向该用户授予访问权限，该用户将无法访问数据库（除非数据库的 *sysusers* 表中存在 “*guest*” 用户）。如果用户不具有通过上述任何方式授予的数据库访问权限，则将连接到 *master* 并显示错误消息。
- 如果删除了用户的缺省数据库，或者将用户从数据库中删除，用户就会在下次登录时连接到 *master*，并显示错误消息。
- 如果将用户的缺省语言从服务器中删除，全服务器范围的缺省语言就将用作初始的语言设置，并显示消息。

更改用户的激活角色

- 使用 **sp_modifylogin** 可设置登录时缺省激活的角色，或从登录时缺省激活的角色中删除角色。

权限

只有系统管理员才能执行 **sp_modifylogin** 来更改其它用户的缺省数据库、缺省语言或全名。只有系统安全员才能执行 **sp_modifylogin**，在登录时缺省激活其它用户的角色。任何用户都可以执行 **sp_modifylogin** 来更改自己的登录帐户。

使用的表

master..syslogins、*sysobjects*、*sysloginroles*

参见

系统过程	sp_activeroles 、 sp_addlogin 、 sp_displaylogin 、 sp_displayroles 、 sp_helprotect
------	---

sp_modify_resource_limit

功能

通过指定新的限制值（和/或指定超过限制时所采取的措施）来更改资源限制。

语法

```
sp_modify_resource_limit {name, appname }  
    , rangename , limittype , limitvalue , enforced  
    , action , scope
```

参数

- name* — 是应用限制的 Adaptive Server 登录。必须指定 *name* 或 *appname*，或指定两者。要修改应用于特定应用程序所有用户的限制，可将 *name* 指定为 *null*。
- appname* — 是应用限制的应用程序的名称。必须指定 *name* 或 *appname*，或指定两者。如果要限制应用于 *name* 使用的所有应用程序，可将 *appname* 指定为 *null*。如果该限制约束特定的应用程序，则可指定客户端程序在登录包中传递给 Adaptive Server 的应用程序名。
- rangename* — 是实施限制的时间范围。您不能修改该值，但必须指定非空值来唯一标识资源限制。
- limittype* — 是应用限制的资源类型。您不能修改该值，但必须指定非空值来唯一标识资源限制。该值必须是下列值之一：

限制类型	说明
row_count	限制查询可以返回的行数
elapsed_time	限制查询批处理或事务可以运行的挂钟时间（以秒表示）
io_cost	限制处理查询的实际成本或优化程序的估计成本

limit_value — 是在 **Adaptive Server** 施加限制之前，登录或应用程序可以使用的服务器资源的最大值。它必须是小于或等于 2^{31} 的正整数，也可以为 **null**，以保留现有值。下表列出了为每种限制类型指定的值：

限制类型	限制值
row_count	在施加限制之前，查询可以返回的最大行数
elapsed_time	在施加限制之前，查询批处理或事务可以运行的最长的挂钟时间（以秒表示）
io_cost	从优化程序的成本计算公式得到的无单位度量值

enforced — 确定是在查询执行之前还是在查询执行期间施加限制。不能修改该值。将 **null** 用作占位符。

action — 是超过限制时所采取的措施。以下代码适用于所有限制类型：

操作代码	说明
1	发出警告
2	中止查询批处理
3	中止事务
4	注销会话
null	保留现有值

scope — 是限制的范围。不能修改该值。可以将 **null** 用作占位符。

示例

1. `sp_modify_resource_limit robin, NULL, weekends, row_count, 3000, NULL, 1, NULL`

修改在 **weekends** 时间范围内应用于 “robin” 使用的所有应用程序的资源限制。如果预计查询返回的行数多于 3000 行，该限制将发出警告。
2. `sp_modify_resource_limit NULL, acctg, "at all times", elapsed_time, 45, 2, 2, 6`

修改在整周和全天应用于 **acctg** 应用程序的资源限制。如果预计的查询处理时间超过 45 秒，该限制将中止查询批处理。

注释

- 不能更改已应用限制的登录或应用程序，或指定新的时间范围、限制类型、施加时间或范围。
- 修改资源限制后，该登录和/或应用程序中每个会话的限制将在该会话的下一查询批处理开始时重新绑定。
- 有关详细信息，参见 *系统管理指南*。

权限

只有系统管理员才能执行 **sp_modify_resource_limit**。

使用的表

master..sysresourcelimits、 *master..systimeranges*、
master..spt_limit_types

参见

系统过程	sp_add_resource_limit 、 sp_drop_resource_limit 、 sp_help_resource_limit
------	---

sp_modify_time_range

功能

更改与指定的时间范围相关联的起始日（星期）、开始时间、结束日（星期）和/或结束时间。

语法

```
sp_modify_time_range name, startday, endday,  
                    starttime, endtime
```

参数

name — 是时间范围的名称。它必须是存储在 *master* 数据库 *systimeranges* 系统表中的时间范围的名称。

startday — 是时间范围在一周中开始的那一天。它必须是缺省服务器语言的完整星期名称（存储在 *master* 数据库的 *syslanguages* 系统表中），或者为 **null**，以保持现有的 **startday**。

endday — 是时间范围在一周结束的那一天。它必须是缺省服务器语言的完整星期名称（存储在 *master* 数据库的 *syslanguages* 系统表中），或者为 **null**，以保持现有结束日（星期）不变。一周中的 **endday** 可以早于或晚于 **startday**，也可以与 **startday** 相同。

starttime — 是时间范围在一天中开始的时间。按 24 小时制，应在 00:00 到 23:59 的范围内指定 **starttime** 的值。其形式如下：

"HH:MM"

或为 **null**，以保留现有的 **starttime**。

endtime — 是时间范围在一天中结束的时间。按 24 小时制，应在 00:00（午夜）到 23:59 的范围内指定 **endtime** 的值。其形式如下：

"HH:MM"

或为 **null**，以保持现有的 **endtime** 不变。除非 **endtime** 是 00:00，否则一天中的 **endtime** 必须晚于 **starttime**。

► 注意

对于包括全天的时间范围，将开始时间指定为“00:00”，将结束时间指定为“23:59”。

示例

```
1. sp_modify_time_range business_hours, NULL,  
   Saturday, NULL, NULL
```

将 *business_hours* 时间范围的结束日从星期五更改为星期六。保留现有的起始日（星期）、开始时间和结束时间。

```
2. sp_modify_time_range before_hours, Monday,  
   Saturday, NULL, "08:00"
```

为 *before_hours* 时间范围指定新的结束日（星期）和结束时间。

注释

- 不能修改“所有时间”时间范围。
- 修改后的时间范围可以和一个或多个其它的时间范围相重叠。
- 当通过系统存储过程修改时间范围后，不会影响正进行的会话的活动时间范围。
- 如果所更改的资源限制具有作为其范围的事务，则不会影响正进行的任何事务。
- 有关详细信息，参见 *系统管理指南*。

权限

只有系统管理员才能执行 `sp_modify_time_range`。

使用的表

master..systimeranges、*master..syslanguages*

参见

系统过程	<code>sp_add_resource_limit</code> 、 <code>sp_add_time_range</code> 、 <code>sp_drop_time_range</code>
------	---

sp_modifythreshold

功能

要修改阈值，可以将该阈值与其它阈值过程、可用空间级别或段名相关联。**不能**使用 **sp_modifythreshold** 来更改最后机会阈值的可用空间量或段名。

语法

```
sp_modifythreshold dbname, segname, free_space  
    [, new_proc_name] [, new_free_space]  
    [, new_segname]
```

参数

dbname — 是要更改阈值的数据库。它必须是当前数据库的名称。

segname — 是要监控其可用空间的段。指定为 “default” 段时应使用引号。

free_space — 是超过阈值的可用页数。如果段上的可用空间低于此级别，Adaptive Server 将执行相关联的存储过程。

new_proc_name — 是超过阈值时所执行的新存储过程。此过程可在当前 Adaptive Server 或 Open Server 的任意数据库中找到。阈值不能执行远程 Adaptive Server 上的过程。

new_free_space — 是与该阈值相关联的新的可用页数。如果段上的可用空间低于此级别，Adaptive Server 将执行相关联的存储过程。

new_segname — 是要监控其可用空间的新段。指定为 “default” 段时应使用引号。

示例

1. **sp_modifythreshold mydb, "default", 200, NULL, 175**

修改 **mydb** 数据库的 “default” 段的阈值，以便在段的可用空间少于 175 页（而非 200 页）时执行该阈值。NULL 是占位符，表示不更改过程名。

2. **sp_modifythreshold mydb, data_seg, 250, new_proc**

修改 **mydb** 的 **data_seg** 段的阈值，以使其执行 **new_proc** 过程。

注释

- 有关详细信息，参见 *系统管理指南*。

超过阈值

- 如果超过阈值，**Adaptive Server** 将执行相关联的存储过程。**Adaptive Server** 为阈值过程使用以下搜索路径：
 - 如果过程名未指定数据库，**Adaptive Server** 将在超过阈值的数据库中进行搜索。
 - 如果在此数据库中未找到该过程，而该过程名以 “sp_” 开头，则 **Adaptive Server** 在 *sybsystemprocs* 数据库中搜索。

如果在这两个数据库中都没有找到该过程，**Adaptive Server** 将向错误日志发送错误消息。

- **Adaptive Server** 使用一个**停滞值**（即全局变量 *@@thresh_hysteresis*）来确定阈值对可用空间变化的敏感程度。一旦阈值执行其过程，该阈值将失效。该阈值会一直保持不活动状态，直到段中可用空间量增加到该阈值之上的 *@@thresh_hysteresis* 页。这样可防止因响应可用空间的微小波动而使阈值重复执行其过程。

最后机会阈值

- 缺省情况下，**Adaptive Server** 监控日志所在段的可用空间，如果可用空间量小于成功转储事务日志所需的空間，就将执行 **sp_thresholdaction**。可用空间量也称作**最后机会阈值**，它由 **Adaptive Server** 来计算，用户不能更改。
- 在记录事务前，如果超过最后机会阈值，**Adaptive Server** 将挂起该事务，直到释放日志空间为止。使用 **sp_dboption** 可更改特定数据库的这一行为。将 **abort tran on log full** 选项设置为 **true** 后，会使 **Adaptive Server** 回退所有在超过最后机会阈值时尚未记录的事务。
- 不能使用 **sp_modifythreshold** 来更改与最后机会阈值相关联的可用空间值或段名。

其它阈值

- 每个数据库最多可以有 256 个阈值，其中包括最后机会阈值。
- 每个阈值必须至少是下一个最接近的阈值中 *@@thresh_hysteresis* 页数的 2 倍。
- 使用 **sp_helpthreshold** 可获得有关现有阈值的信息。
- 使用 **sp_droptreshold** 可从段中删除阈值。

创建阈值过程

- 任何具有 **create procedure** 权限的用户都可以在数据库中创建阈值过程。通常，系统管理员会在 **master** 数据库中创建 **sp_thresholdaction**，数据库所有者会在用户数据库中创建阈值过程。
- **sp_modifythreshold** 不校验指定过程是否存在。可以将阈值与尚不存在的过程相关联。
- **sp_modifythreshold** 将检查并确保修改阈值过程的用户已被直接授予“**sa_role**”。所有在修改阈值过程时处于活动状态的系统角色都将输入到 **systhresholds** 中，作为编写该过程的用户的有效角色。然而，引发阈值时，仅激活直接授予的系统角色。间接授予的系统角色和用户定义的角色则不被激活。
- Adaptive Server 向阈值过程传递四个参数：
 - **@dbname, varchar(30)**，用于标识数据库
 - **@segment_name, varchar(30)**，用于标识段
 - **@space_left, int**，用于表示与该阈值相关联的可用页数
 - **@status, int**，其中包含一个值，该值对最后机会阈值为 **1**，对其它阈值为 **0**。

这些参数按位置（而非名称）传递；阈值过程可以使用它们的其它名称，但该过程必须按显示的顺序声明这些参数，而且数据类型必须正确。

- 无需为每个阈值创建不同的过程。要进行最低程度的维护，可在 **sybsystemprocs** 数据库中创建一个可供所有阈值执行的阈值过程。
- 将 **print** 和 **raiserror** 语句包括在阈值过程中，以便向错误日志输出信息。

执行阈值过程

- 超过阈值时开始的任务将作为后台任务来执行。这些任务不具有相关联的终端或用户会话。如果在运行这些任务时执行 **sp_who, status** 列将显示“**background**”。
- 当用户执行 **sp_modifythreshold** 时，Adaptive Server 将以修改阈值的用户的权限来执行该阈值过程，但不使用任何已撤消的权限。
- 每个阈值过程都将使用一个用户连接，连接时间为执行过程所需的时间。

禁用可用空间统计

- 使用 **sp_dboption** 的 **no free space acctg** 选项可禁用非日志段上的可用空间统计。
- 您不能对日志段禁用可用空间统计。

◆ 警告!

如果禁用可用空间统计，系统过程将无法提供有关空间分配的准确信息。

创建 10.0 之前版本数据库的最后机会阈值

- 如果将 10.0 之前版本的数据库升级到当前版本，将不能自动获得最后机会阈值。使用 **lct_admin** 系统函数可在现有数据库中创建最后机会阈值。
- 只有将其日志存储在不同段上的数据库才能具有最后机会阈值。使用 **sp_logdevice** 可将事务日志移动到不同设备上。

权限

只有数据库所有者或系统管理员才能执行 **sp_modifythreshold**。

使用的表

master..sysusages、sysobjects、syssegments、systhresholds

参见

命令	create procedure、dump transaction
系统过程	sp_addthreshold、sp_dboption、sp_droptreshold、sp_helpthreshold、sp_thresholdaction

sp_monitor

功能

显示有关 Adaptive Server 的统计信息。

语法

sp_monitor

参数

无。

示例

1. sp_monitor

last_run	current_run	seconds
-----	-----	-----
Jan 29 1987 10:11AM	Jan 29 1987 10:17AM	314
cpu_busy	io_busy	idle
-----	-----	-----
4250(215)-68%	67(1)-0%	109(100)-31%
packets_received	packets_sent	packet_errors
-----	-----	-----
781(15)	10110(9596)	0(0)
total_read	total_write	total_errors
-----	-----	-----
394(67)	5392(53)	0(0)
		connections

		15(1)

报告有关 Adaptive Server 繁忙程度的信息。

注释

- Adaptive Server 使用一系列全局变量来跟踪它所完成的工作量。**sp_monitor** 显示这些全局变量的当前值，以及自上次执行该过程后这些变量的更改情况。
- 对于每列而言，统计信息将以 *number(number)-number%* 或 *number(number)* 形式来显示。
 - 第一个数字表示 Adaptive Server 重新启动后的秒数（对于“cpu_busy”、“io_busy”和“idle”）或总数（对于其它列）。

- 小括号中的数字表示上次运行 **sp_monitor** 后的秒数或总数。百分比符号表示上次运行 **sp_monitor** 后时间的百分比。

例如，如果报告中将 “**cpu_busy**” 显示为 “**4250(215)-68%**”，则表示自 **Adaptive Server** 上次启动以来，**CPU** 占用时间已达 **4250** 秒，自上次运行 **sp_monitor** 以来，**CPU** 占用时间为 **215** 秒，自上次运行 **sp_monitor** 以来，**CPU** 占用时间为总时间的 **68%**。

对于 “**total_read**” 列，值 **394(67)** 表示自上次启动 **Adaptive Server** 以来，已发生 **394** 次磁盘读取，其中的 **67** 次发生于上次运行 **sp_monitor** 以后。

- 表 7-19 说明了 **sp_monitor** 报告中的列、等同的全局变量（如果有）及其含义。除 “**last_run**”、“**current_run**” 和 “**seconds**” 之外，这些列标题同时也是全局变量的名称，但是所有全局变量之前都需带有 **@@**。另一个不同之处是全局变量所报告数字的单位，它们以计算机的时钟周期为单位，而非以毫秒表示 **CPU** 时间。

表 7-19: **sp_monitor** 报告中的列

列标题	等同的变量	含义
last_run		sp_monitor 过程上次运行的时钟时间。
current_run		当前的时钟时间
seconds		自 sp_monitor 上次运行以来的秒钟数。
cpu_busy	@@cpu_busy	Adaptive Server 的 CPU 执行 Adaptive Server 任务的 CPU 时间（以秒为单位）。
io_busy	@@io_busy	Adaptive Server 执行输入和输出操作所用的 CPU 时间（以秒为单位）。
idle	@@idle	Adaptive Server 一直处于空闲状态的 CPU 时间（以秒为单位）。
packets_received	@@pack_received	由 Adaptive Server 所读取的输入包的数目。
packets_sent	@@pack_sent	由 Adaptive Server 所写入的输出包的数目。
packet_errors	@@packet_errors	读取和写入包时 Adaptive Server 所检测到的错误数。
total_read	@@total_read	Adaptive Server 读取磁盘的次数。
total_write	@@total_write	Adaptive Server 写入磁盘的次数。
total_errors	@@total_errors	读取和写入时 Adaptive Server 所检测到的错误数。
connections	@@connections	登录或尝试登录 Adaptive Server 的次数。

- 在 Adaptive Server 启动后首次运行 **sp_monitor** 时，小括号中的数字无意义。
- Adaptive Server 的管家任务利用服务器的空闲周期将更改的页从缓存写入磁盘。此过程影响 **sp_monitor** 所报告的 “cpu_busy”、“io_busy” 和 “idle” 列的值。要禁用管家任务并消除这些影响，可将 **housekeeper free write percent** 配置参数设置为 0:

```
sp_configure "housekeeper free write percent", 0
```

权限

只有系统管理员才能执行 **sp_monitor**。

使用的表

master.dbo.sysengines、 *master.dbo.spt_monitor*、 *sysobjects*

参见

系统过程	sp_who
------	--------

sp_monitorconfig

功能

显示有关用于索引、对象和数据库的元数据描述符的缓存使用情况统计信息。**sp_monitorconfig** 还报告用于参照完整性查询的辅助扫描描述符的统计信息，以及事务描述符和 DTX 参与者的使用情况统计信息。

语法

```
sp_monitorconfig "configname"
```

参数

configname — 是正查询其监控信息的配置参数的完整或部分名称。
有效配置参数包括 **open indexes**、**open objects**、**open databases**、**aux scan descriptors**、**txn to pss ratio**、**number of dtx participants** 和 **all**。指定 **all** 时，将显示服务器中所有索引、对象、数据库和辅助扫描描述符的描述符帮助信息。

示例

1. sp_monitorconfig "open"

```
Configuration option is not unique.
option_name                config_value run_value
-----
current read change w/ open cursors      1          1
number of open databases                  12          12
number of open indexes                    500          500
number of open objects                    500          500
open index hash spinlock ratio            100          100
open index spinlock ratio                 100          100
open object spinlock ratio                100          100
```

2. sp_monitorconfig "open objects"

```
Usage information at date and time: Jan 14 1997 8:54AM.
Name                # Free    # Active    % Active    # Max Ever Used    Re-used
-----
number of open      217        283         56.60        300                No
objects
```

在上例中，存在 **283** 个活动的对象元数据描述符，其中 **217** 个可用。自 **Adaptive Server** 上次启动以来，在高峰期使用的描述符最多为 **300** 个。因此，可以将大小重新设置为 **330**，以便包括所用元数据描述符数的最大值（**300** 个），并另外留出 **10%** 的空间：

```
sp_configure "number of open objects", 330
```

3. sp_monitorconfig "open indexes"

Usage information at date and time: Jan 14 1997 8:55:00AM.

Name	# Free	# Active	% Active	# Max Ever Used	Re-used
number of open indexes	556	44	7.33	44	No

在此示例中，索引元数据描述符的最多个数为 **44** 个。可以将该大小重新设置为可接受的最小值 **100**：

```
sp_configure "number of open indexes", 100
```

4. sp_monitorconfig "aux scan descriptors"

Usage information at date and time: Jan 14 1997 8:56:00AM.

Name	# Free	# Active	% Active	# Max Ever Used	Re-used
number of aux scan descriptors	170	30	15.00	32	NA

在上例中，尽管 **Adaptive Server** 配置为使用 **200** 个描述符，但活动的扫描描述符是 **30** 个。使用 **number of aux scan descriptors** 配置参数可将该值重新设置为至少 **32**。安全的设置为 **36**，可包括 **32** 个扫描描述符，并另外留出 **10%** 的空间。

5. sp_monitorconfig "number of open databases"

Usage information at date and time: Jan 14 1997 8:57:00AM.

Name	# Free	# Active	% Active	# Max Ever Used	Re-used
number of open databases	0	5	100.00	5	Yes

在上例中，**Adaptive Server** 配置为用于 **5** 个打开的数据库，这些数据库已在当前会话中使用。然而，正如“**Re-used**”列所示，还需要打开另外一个数据库。如果 **5** 个数据库都在使用，就会导致错误，除非可以重新使用未使用的数据库的描述符。要防止出现错误，可将 **number of open databases** 重新设置为较大的值。

6. sp_monitorconfig "txn to pss ratio"

Usage information at date and time: Jun 18 1999 8:54:00AM.

Name	# Free	# Active	% Active	# Max Ever Used	Re-used
txn to pss ratio	784	80	10.20	523	NA

在上例中，当前只有 10.2% 的事务描述符正在使用。然而，自 Adaptive Server 上次启动以来，在高峰期使用的事务描述符最多个数是 523。

注释

- **sp_monitorconfig** 显示有关用于索引、对象和数据库的元数据描述符的缓存使用情况统计信息，如服务器当前使用的元数据描述符的个数。
- **sp_monitorconfig** 还报告正使用的辅助扫描描述符的个数。扫描描述符可管理对运行查询的表进行的单扫描。
- **sp_monitorconfig** 输出中的列提供以下信息：
 - “#Free” 指定当前未使用的可用元数据或辅助扫描描述符的个数。
 - “#Active” 指定在缓存中安装（即处于活动状态）的元数据或辅助扫描描述符的个数。
 - “%Active” 指定缓存的或活动的元数据或辅助扫描描述符的百分比。
 - “#Max Ever Used” 指定自服务器启动以来使用的元数据或辅助扫描描述符的最多个数。
 - “Re-used” 指定是否重新使用元数据描述符，以满足服务器中索引、对象或数据库增加的需要。返回的值包括 “Yes”、“No” 或 “N/A”（用于不支持重用机制的配置参数，如 **aux scan descriptors** 的个数）。
- 根据 “#Max Ever Used” 列中的值来确定描述符的适当个数；务必要添加大约 10% 的空间以用于最终设置。例如，如果所用索引元数据描述符的最多个数是 142，最好将 **number of open indexes** 配置参数设置为 157。
- 如果 “Re-used” 列的状态是 “Yes”，可将配置参数重新设置为较大的值。如果需要重新使用描述符，可能出现性能问题，特别是对于打开的数据库。打开的数据库包含大量的元数据信息，这意味着要填充打开的数据库，Adaptive Server 需要多次访问磁盘上的元数据；服务器也可能存在旋转锁争用问题。要检查旋转锁争用，可使用系统过程 **sp_sysmon**。有关详细信息，参见

Performance and Tuning Guide。要查找索引、对象或数据库的当前个数，可使用 **sp_countmetadata**。

- 要获得精确的读数，可在常规的 **Adaptive Server** 高峰期间运行 **sp_monitorconfig**。高峰期间可多次运行 **sp_monitorconfig**，以确保实际查找到所用描述符的最多个数。

权限

只有系统管理员才能执行 **sp_monitorconfig**。

使用的表

master..sysindexes、*sysobjects*、*sysdatabases*

参见

系统过程	sp_configure 、 sp_countmetadata 、 sp_helpconfig 、 sp_helpconstraint
------	---

sp_object_stats

功能

输出表和索引的锁争用、锁等待时间和死锁统计信息。

语法

```
sp_object_stats interval [, top_n  
[, dbname, objname [, rpt_option ]]]
```

参数

interval — 指定采样的时间间隔。必须使用 HH:MM:SS 格式，例如“00:20:00”。

top_n — 要按争用顺序报告的对象数目。缺省值为 10。

dbname — 是要报告的数据库的名称。如果未给定数据库名称，将报告所有数据库中的对象争用。

objname — 是要报告的表的名称。如果指定了表名，同时还必须指定数据库名。

rpt_option — 必须为 **rpt_locks** 或 **rpt_objlist**。

示例

1. **sp_object_stats "00:20:00"**

报告全服务器范围的前 10 个对象的锁统计信息。

2. **sp_object_stats "00:20:00", 5, pubtune**

只报告 *pubtune* 数据库中的表，并列出遇到最高争用的前 5 个表。

3. **sp_object_stats "00:15:00", @rpt_option =
"rpt_objlist"**

即使没有发生争用和死锁，只显示具有最高级别锁活动的表的名称。

注释

- **sp_object_stats** 报告在指定的采样期间在表上所获得的共享锁、更新锁和排它锁的数目。以下报告显示 *titles* 表:

Object Name: pubtune..titles (dbid=7, objid=208003772,lockscheme=Datapages)

Page Locks	SH_PAGE	UP_PAGE	EX_PAGE\$
-----	-----	-----	-----
Grants:	94488	4052	4828
Waits:	532	500	776
Deadlocks:	4	0	24
Wait-time:	20603764 ms	14265708 ms	2831556 ms
Contention:	0.56%	10.98%	13.79%

*** Consider altering pubtune..titles to Datarows locking.

- 表 7-20 显示了这些值的含义。

表 7-20: sp_object_stats 的输出

输出行	值
Grants	立即授予锁的次数。
Waits	需要锁的任务要等待的次数。
Deadlocks	发生死锁的数目。
Wait-times	所有任务在等待锁时所用的总毫秒数。
Contention	任务必须等待或遇到死锁的次数所占的百分比。

- **sp_object_stats** 建议在表中的总争用大于 15% 时更改锁方案，具体建议如下:
 - 如果表使用 **allpage** 锁，建议更改为数据页锁
 - 如果表使用数据页锁，建议更改为数据行锁。
- **rpt_option** 指定报告类型:
 - **rpt_locks** 为具有最高争用的表报告授予、等待、死锁和等待次数。**rpt_locks** 为缺省类型。
 - **rpt_objlist** 只报告具有最高级别锁活动的对象的名称。
- **sp_object_stats** 创建名为 *tempdb..syslkstats* 的表。存储过程完成后不会删除该表，这样系统管理员就可查询该表。
- 一次只应有一个用户执行 **sp_object_stats**。如果多个用户试图同时运行 **sp_object_stats**，则第二个命令将会被阻塞，或结果可能无效。

- 每次执行 **sp_object_stats** 时，都将删除 *tempdb..syslkstats* 表并重新创建它。
- *tempdb..syslkstats* 的结构在表 7-21 中进行介绍。

表 7-21: tempdb..syslkstats 表中的列

列名	数据类型	说明
<i>dbid</i>	<i>smallint</i>	数据库 ID
<i>objid</i>	<i>int</i>	对象 ID
<i>lockscheme</i>	<i>smallint</i>	整数值 1-3: Allpage = 1, 数据页 = 2, 数据行 = 3
<i>page_type</i>	<i>smallint</i>	数据页 = 0, 索引页 = 1
<i>stat_name</i>	<i>char(30)</i>	该行提供的统计信息
<i>stat_value</i>	<i>float</i>	授予、等待或死锁的次数或总的等待时间

stat_name 列中的值包括三部分：

- 第一部分为 “ex”（表示排它锁）、“sh”（表示共享锁）或 “up”（表示更新锁）。
 - 第二部分为 “pg”（表示页锁）或 “row”（表示行锁）。
 - 第三部分为 “grants”（表示立即授予的锁）、“waits”（表示需等待其它锁释放的锁）、“deadlocks”（表示死锁）和 “waittime”（表示等待获得锁的时间）。
- 如果指定表名， **sp_object_stats** 将显示具有该名的所有表。如果多个用户拥有具有指定名的表，这些表的输出将显示对象 ID，而不是所有者名。

权限

只有系统管理员才能执行 **sp_object_stats**。

使用的表

创建表 *tempdb..syslkstats*。该表在执行结束不会被删除，它可以通过 Transact-SQL 来查询。

参见

命令	alter table
----	-------------

sp_passthru

(仅适用于组件集成服务)

功能

使用户可以将 SQL 命令缓冲区传递到远程服务器。

语法

```
sp_passthru server, command, errcode, errmsg, rowcount  
[, arg1, arg2, ... argn]
```

参数

server — 是要将 SQL 命令缓冲区传递到的远程服务器的名称。该服务器的类必须是受支持的、非本地的服务器类。

command — 是 SQL 命令缓冲区。它最多可以包含 255 个字符。

errcode — 是远程服务器返回的错误代码（如果有的话）。如果远程服务器上沒有发生错误，返回的值为 0。

errmsg — 是远程服务器返回的错误消息。它最多可以包含 255 个字符。只有在 **errcode** 为非零值时，才设置该参数；否则，返回 NULL。

rowcount — 是命令缓冲区中的最后一条命令影响的行数。如果命令是 **insert**、**delete** 或 **update**，该值表示受影响的行数，即使没有返回任何值，也是如此。如果最后一条命令是查询，该值表示从外部服务器返回的行数。

arg1 ... argn — 接收命令缓冲区中最后一条命令返回的最后一行的结果。最多可以指定 250 个 **arg** 参数。必须将所有参数声明为输出参数。

示例

```
1. sp_passthru ORACLE, "select date from dual",  
   @errcode output, @errmsg output, @rowcount output,  
   @oradate output
```

在输出参数 **@oradate** 中返回来自 Oracle 服务器的日期。如果 Oracle 发生错误，错误代码将放置在 **@errcode** 中，而相应的消息放置在 **@errmsg** 中。将 **@rowcount** 参数设置为 1。

注释

- **sp_passthru** 使用户可以将 SQL 命令缓冲区传递到远程服务器。假定传递的 SQL 语句的语法是接收缓冲区的服务器类的原始语法。不执行任何转换和解释。远程服务器的结果被放置在输出参数中（可选）。

只有在安装并配置了组件集成服务的情况下，才能使用 **sp_passthru**。
- 可以在命令缓冲区中包含多条命令。对于某些服务器类来说，必须用分号将各个命令分隔开。有关在直通模式下如何处理查询缓冲区的详细论述，参见 *Component Integration Services 用户指南*。

返回参数

- 将输出参数 *arg1 ... argn* 设置为相应列的值，这些列位于命令缓冲区中最后一条命令所返回的最后一行中。参数的位置决定了该参数将包含的列值。*arg1* 接收列 1 中的值，*arg2* 接收列 2 中的值，依此类推。
- 如果可选参数个数比返回的列数少，将忽略超出的列。如果可选参数个数比列数多，则将多余的参数设置为 NULL。
- 将尝试把每一列都转换为输出参数的数据类型。如果数据类型相似到允许进行隐式转换的程度，这种尝试将成功。有关隐式转换的信息，参见第 2 章“Transact-SQL 函数”中的“数据类型转换函数”。有关在直通模式下哪种数据类型表示每个服务器类中数据类型的信息，参见*组件集成服务用户指南*。

权限

任何用户都可以执行 **sp_passthru**。

使用的表

sys.servers、*sys.remotelogs*

参见

系统过程	sp_autoconnect 、 sp_remotesql
------	---

sp_password

功能

添加或更改 Adaptive Server 登录账户的口令。

语法

```
sp_password caller_passwd, new_passwd [, loginame]
```

参数

caller_passwd — 是您的口令。在您更改自己的口令时，这就是您的原有口令。当系统安全员使用 **sp_password** 更改其他用户的口令时，**caller_passwd** 则是系统安全员的口令。

new_passwd — 是用户或 **loginame** 的新口令。口令的长度至少必须为 6 个字节。口令中如果包含字符 **A-Z**、**a-z** 或 **0-9** 以外的字符，需要用引号将口令引起来。对于以 **0-9** 的数字开头的口令，也要用引号引起来。

loginame — 系统安全员正在更改其账户口令的用户的登录名。

示例

1. **sp_password "3blindmice", "2mediumhot"**

将口令从 “3blindmice” 更改为 “2mediumhot”（由于口令以数字开头，所以用引号将它们引起来。）。

2. **sp_password "2tomato", sesame1, victoria**

系统安全员（口令为 “2tomato”）已经将 Victoria 的口令更改为 “sesame1”。

3. **sp_password null, "16tons"**

将您的口令从 NULL 更改为 “16tons”。注意，NULL 不需要用引号引起来。（NULL 不是允许使用的新口令。）

4. **PRODUCTION...sp_password figaro, lilacs**

将您在 PRODUCTION 服务器上的口令从 “figaro” 更改为 “lilacs”。

注释

- 任何用户都可以用 **sp_password** 更改自己的口令。
- 新口令的长度至少必须包含 6 个字符。新口令不能是 NULL。
- *caller_passwd* 的加密文本必须与调用方的现有加密口令匹配。如果二者不匹配，**sp_password** 将返回错误消息并失败。
master.dbo.syslogins 以加密的形式列出口令。
- 如果客户端程序要求用户在远程服务器上的口令与在本地服务器上的口令相同，则用户在更改其本地口令之前，必须首先更改他们在所有远程服务器上的口令。**sp_password** 将作为各远程服务器上的远程过程调用来执行。参见示例 4。
- 可以对 **systemwide password expiration** 配置参数进行设置，从而指定口令的有效期，强制所有 Adaptive Server 登录账户定期更改口令。详细信息，参见 *系统管理指南*。

权限

只有系统安全员才能执行 **sp_password** 来更改其他用户的口令。任何用户都可以执行 **sp_password** 来更改自己的口令。

使用的表

master.dbo.syslogins、 *sysobjects*

参见

系统过程	sp_addlogin、 sp_adduser
------	-------------------------

sp_placeobject

功能

将表或索引的未来空间分配放置在特定段上。

语法

```
sp_placeobject segname, objname
```

参数

segname — 是要放置表或索引的段的名称。

objname — 是表或索引的名称，将把该表或索引的后续空间分配放置在段 *segname* 上。以 “*tablename.indexname*” 的形式指定索引名。

示例

1. **sp_placeobject segment3, authors**

该命令可以将表 *authors* 的所有后续空间分配放置在名为 “segment3” 的段上。

2. **sp_placeobject indexes, 'employee.employee_nc'**

该命令可以将 *employee* 表的 *employee_nc* 索引的所有后续空间分配放置在名为 *indexes* 的段上。

注释

- 不能更改系统表的未来空间分配位置。
- 将表或索引放置在特定段上不会影响任何现有表或索引数据的位置。它只影响未来的空间分配。通过更改表或索引使用的段，可以使数据分布到多个段上。
- 如果将集群索引与 **sp_placeobject** 一起使用，表将随该索引一起移动。
- 用 **create table** 或 **create index** 创建表或索引时，可以指定段。如果未指定段，数据将放置在 *default* 段上。
- 当 **sp_placeobject** 将表或索引拆分并使其分布到多个磁盘片段上时，在执行 **sp_placeobject** 之前，诊断命令 **dbcc** 将显示有关驻留在某些片段（这些片段已用于存储）上的数据的信息。忽略这些消息。
- 不能在已分区的表上使用 **sp_placeobject**。

权限

只有表所有者、数据库所有者或系统管理员才能执行 **sp_placeobject**。

使用的表

sysindexes、*sysobjects*、*syspartitions*、*syssegments*

参见

命令	alter table、dbcc
系统过程	sp_addsegment、sp_dropsegment、 sp_extendsegment、sp_help、sp_helpindex、 sp_helpsegment

sp_plan_dbccdb

功能

为新的 *dbccdb* 和 *dbccalt* 数据库推荐合适的大小，为 *dbccdb* 和 *dbccalt* 列出合适的设备，并为目标数据库建议缓存大小和合适的工作进程数。

语法

```
sp_plan_dbccdb [dbname]
```

参数

dbname — 指定目标数据库的名称。如果未指定 *dbname*，*sp_plan_dbccdb* 将为 *master.sysdatabases* 中的所有数据库提出建议。

示例

```
1. sp_plan_dbccdb master
```

Recommended size for dbccdb is 4MB.

dbccdb database already exists with size 8MB.

Recommended values for workspace size, cache size and process count are:

dbname	scan ws	text ws	cache	process count
master	64K	64K	640K	1

为创建 *dbccdb* 数据库返回配置建议，该数据库将适合于检查 *master* 数据库。由于在运行该命令时，*dbccdb* 数据库已经存在，所以提供现有数据库的大小以供比较。

2. sp_plan_dbccdb

Recommended minimum size for dbccdb is 4MB.

Recommended values for workspace size, cache size and process count are:

dbname	scan ws	text ws	cache	process count
master	64K	64K	640K	1
tempdb	64K	64K	640K	1
model	64K	64K	640K	1
sybsystemprocs	272K	80K	640K	1
dbccdb	128K	64K	640K	1

为创建 *dbccdb* 数据库返回配置建议，该数据库将适合于检查服务器中的所有数据库。在运行该命令时，*dbccdb* 数据库并不存在。

3. sp_plan_dbccdb pubs2

Recommended size for dbccdb is 4MB.

Recommended devices for dbccdb are:

Logical Device Name	Device Size
Physical Device Name	
sprocdev	28672
/remote/sybase/devices/srv_sprocs_dat	
tun_dat	8192
/remote/sybase/devices/srv_tun_dat	
tun_log	4096
/remote/sybase/devices/srv_tun_log	

Recommended values for workspace size, cache size and process count are:

dbname	scan ws	text ws	cache	process count
pubs2	64K	64K	640K	1

为创建 *dbccdb* 数据库提供配置建议，该数据库将适合于检查 *pubs2*。

注释

- **sp_plan_dbccdb** 为创建新的 *dbccdb* 和 *dbccalt* 数据库推荐合适的大小，为新数据库列出合适的设备，并为目标数据库建议缓存大小和合适的工作进程数。
- 如果指定 *dbccdb*，**sp_plan_dbccdb** 将为替代数据库 *dbccalt* 推荐值。如果指定 *dbccalt*，**sp_plan_dbccdb** 将为 *dbccdb* 推荐值。
- **sp_plan_dbccdb** 不报告现有 *dbccdb* 和 *dbccalt* 数据库的值。要收集现有 *dbccdb* 或 *dbccalt* 数据库的配置参数，可以使用 **sp_dbcc_evaluatedb**。
- 有关 **dbcc** 存储过程在维护 *dbccdb* 和从 *dbccdb* 生成报告方面的信息，参见第 10 章 “**dbcc** 存储过程”。

权限

只有系统管理员或数据库所有者才能执行 **sp_plan_dbccdb**。在未指定数据库名的情况下，只有系统管理员才能执行 **sp_plan_dbccdb**。

使用的表

master..sysdatabases、*master..sysdevices*、*master..sysusages*

参见

命令	dbcc
系统过程	sp_dbcc_evaluatedb

sp_poolconfig

功能

创建、删除内存池或调整内存池大小，并提供有关数据缓存中内存池的信息。

语法

要在现有缓存中创建内存池或更改内存池大小：

```
sp_poolconfig cache_name [, "mem_size[P|K|M|G]",
    "config_poolK" [, "affected_poolK"]]
```

要更改内存池的清洗大小：

```
sp_poolconfig cache_name, "io_size",
    "wash=size[P|K|M|G]"
```

要更改内存池的异步预取百分比：

```
sp_poolconfig cache_name, "io_size",
    "local async prefetch limit=percent"
```

参数

cache_name — 是现有数据缓存的名称。

mem_size — 是要创建的内存池的大小；如果具有指定 I/O 大小的内存池已经存在，则表示为现有内存池新指定的大小。内存池的最小大小为 512K。在指定内存池的大小单位时，用 **P** 表示页，用 **K** 表示千字节，用 **M** 表示兆字节，或用 **G** 表示千兆字节。缺省值为千字节。

config_pool — 是在要分配或删除内存的内存池中执行的 I/O 大小。
有效 I/O 大小为 2K、4K、8K 和 16K。

affected_pool — 在要释放内存的内存池中执行的 I/O 大小。如果未指定 **affected_pool**，内存将取自 2K 内存池。

io_size — 是在要重新配置清洗大小的内存池中执行的 I/O 大小。缓存名和 I/O 大小的组合将唯一地标识内存池。

wash=size — 更改内存池的清洗大小（缓存中 Adaptive Server 将脏页写入磁盘的点）。

local async prefetch limit=percent — 设置内存池中缓冲区的百分比，它用于保留已通过异步预取读入缓存、但还没有使用的缓冲区。

示例

1. `sp_poolconfig pub_cache, "10M", "16K"`

在具有 10MB 空间的数据缓存 `pub_cache` 中创建 16K 内存池。
所有空间都取自缺省的 2K 内存池。

2. `sp_poolconfig pub_cache, "3M", "8K", "16K"`

将 3MB 空间从 `pub_cache` 的 16K 内存池移到 8K 内存池。

3. `sp_poolconfig "pub_cache"`

报告 `pub_cache` 的当前配置。

4. `sp_poolconfig pub_cache, "0K", "16K"`

将 16K 内存池从 `pub_cache` 中删除，并将所有分配给它的内存放置在 2K 内存池中。

5. `sp_poolconfig pub_cache, "2K", "wash=508K"`

将 `pubs_cache` 中 2K 内存池的清洗大小更改为 508K。

6. `sp_poolconfig pub_cache, "2K", "local async prefetch limit=15"`

将 2K 内存池的异步预取限制更改为 15%。

注释

- 当用 `sp_cacheconfig` 创建数据缓存时，将把所有空间分配给 2K 内存池。`sp_poolconfig` 将把数据缓存拆分为具有较大 I/O 大小的附加池。
- 如果缓存中没有较大的 I/O 内存池，**Adaptive Server** 将对所有绑定到缓存的对象以 2K 为单位（数据页的大小）执行 I/O。通常可以通过配置执行较大 I/O 的内存池来改进性能。16K 内存池在单个 I/O 操作中可以读和写八个数据页。
- 缓存名和 I/O 大小的组合必须是唯一的。换句话说，在特定的数据缓存中，只能有一个内存池具有给定的 I/O 大小。
- 在一个缓存上，每次只能有一个 `sp_poolconfig` 命令处于活动状态。如果在第一条 `sp_poolconfig` 命令完成之前，又发出了同样的命令，则第二条命令将处于休眠状态，直到第一条命令完成为止。
- 图 7-3 显示了具有以下各项的数据缓存：
 - 缺省数据缓存，包含一个 2K 内存池和一个 16K 内存池
 - 用户缓存，包含一个 2K 内存池和一个 16K 内存池
 - 日志缓存，包含一个 2K 内存池和一个 4K 内存池



图 7-3：带有缺省缓存和用户定义缓存的数据缓存

- 可以在缺省数据缓存中创建 I/O 大小多达 16K 的内存池。
- 内存池的最小大小为 512K。不能通过将存储空间迁移到其它内存池的方式，将任何缓存中的任何内存池的大小减小到 512K 以下。
- 在以下两种情况下，可以创建小于 512K 的内存池：
 - 如果尝试通过将其大小设置为零来删除内存池，而某些页正在使用中，则 **sp_poolconfig** 将尽量减小池的大小，并输出警告消息。池的状态被设置为 “Unavailable/deleted”。
 - 如果尝试通过移动缓冲区来创建新的内存池，但却没有足够的缓冲区可以移到新池中，则 **sp_poolconfig** 将移动尽可能多的缓冲区，并将缓存的状态设置为 “Unavailable/too small”。

在这两种情况下，都可以在以后的时间重新执行命令。重新启动服务器时，还将删除内存池或将其更改为所需的大小。

- 可以在 **Adaptive Server** 处于活动状态时创建内存池；不必重新启动，这些内存池就可以生效。但是，**Adaptive Server** 只能移动“可用”缓冲区（即处于空闲状态的缓冲区或不包含没有被写入磁盘的更改的缓冲区）。在配置内存池或更改其大小时，**Adaptive Server** 将把尽量多的内存移到池中，并输出信息性消息显示池的要求大小和实际大小。在重新启动 **Adaptive Server** 后，将按配置的大小创建所有内存池。

- 以下命令只执行 2K 的 I/O 操作：**create database**、**alter database**、某些 **dbcc** 命令、**disk init** 和 **drop table**。**dbcc checktable** 可以执行更大的 I/O，而 **dbcc checkdb** 可以在表上执行更大的 I/O 以及在索引上执行 2K I/O。恢复进程也只使用 2K 的内存池：将把所有页读入缺省缓存的 2K 内存池并在那里进行更改。请确保缺省的 2K 内存池对于这些活动来说足够大。
- 当为事务日志配置了 4K 的 I/O 时，多数 Adaptive Server 的性能最佳。如果为缺省缓存或绑定了事务日志的缓存配置了 4K 的内存池，Adaptive Server 将使用缺省的 4K I/O 大小。否则，它将使用 2K 的内存池。
- 可以使用 **sp_logiosize** 系统过程来增加缺省的日志 I/O 大小。但是，您指定的 I/O 大小必须使缓存中相同大小的内存池绑定到事务日志。如果不是这样，Adaptive Server 将使用 4K 或 2K 内存池。

清洗百分比

- 清洗大小的缺省值是按以下方法计算的：
 - 如果内存池小于 300MB，则将缺省清洗大小设置为该内存池中缓冲区数的 20%
 - 如果内存池大于 300MB，则将缺省清洗大小设置为 300MB 中缓冲区数的 20%
- 清洗大小的最低设置为 10 个缓冲区，最高设置为内存池大小的 80%。
- 每个内存池在内存池缓存链的最近最少使用 (LRU) 端都包含一个清洗区。一旦将脏页（在缓存中被更改过的页）移到清洗区，Adaptive Server 将对这些页启动异步写入操作。清洗区必须足够大，以便在页到达内存池的 LRU 端之前，可以将页写入磁盘。当 Adaptive Server 需要等待清除缓冲区时，性能将会下降。

缺省百分比，即在清洗区中放置 20% 的缓冲区，对于多数应用程序来说已经足够了。如果要使用超大内存池，而您的应用程序具有非常高的数据修改率，可能需要将该大小增加内存池的 1% 或 2%。请与 Sybase 技术支持部门联系，以获取有关选择有效清洗大小的详细信息。

局部异步预取百分比

- 内存池异步预取百分比的缺省值是用配置参数 **global async prefetch limit** 设置的。内存池限制总是替换全局限制。
- 要在内存池中禁用预取（如果全局限制是非零值），可以将内存池的限制设置为 **0**。
- 有关更改异步预取限制如何影响性能的信息，参见 *Performance and Tuning Guide*。

权限

只有系统管理员才能执行 **sp_poolconfig**，以重新配置数据缓存中的内存池。任何用户都可以使用 **sp_poolconfig** 来获取有关内存池的信息。

使用的表

master..sysconfigures

参见

系统过程	sp_cacheconfig、sp_helpcache、 sp_unbindcache、sp_unbindcache_all
------	---

sp_primarykey

功能

定义表或视图的主键。

语法

```
sp_primarykey tablename, col1 [, col2, col3, ..., col8]
```

参数

tablename — 是要定义主键的表或视图的名称。

col1 — 是构成主键的第一列的名称。主键可以由一到八列组成。

示例

1. **sp_primarykey authors, au_id**

将 *au_id* 域定义为表 *authors* 的主键。

2. **sp_primarykey employees, lastname, firstname**

将域 *lastname* 和 *firstname* 的组合定义为表 *employees* 的主键。

注释

- 执行 **sp_primarykey** 可以将键添加到 *syskeys* 表。只有表或视图的所有者才能定义其主键。**sp_primarykey** 不会强加参照完整性约束；使用 **create table** 或 **alter table** 命令的 **primary key** 子句可以强加主键关系。
- 通过用 **sp_primarykey**、**sp_commonkey** 和 **sp_foreignkey** 定义键，可以使数据库设计中的隐式逻辑关系变为显式逻辑关系。这样，应用程序就可以使用这些信息了。
- 表或视图只能有一个主键。要显示有关已经定义的键的报告，可以执行 **sp_helpkey**。
- 安装进程在系统表的适当列上运行 **sp_primarykey**。

权限

只有指定表或视图的所有者才能执行 **sp_primarykey**。

使用的表

syscolumns、*syskeys*、*sysobjects*

参见

命令	alter table、create table、create trigger
系统过程	sp_commonkey、sp_dropkey、sp_foreignkey、 sp_helpjoins、sp_helpkey

sp_processmail

(仅适于 Windows NT)

功能

通过使用 **xp_findnextmsg**、**xp_readmail**、**xp_sendmail** 和 **xp_deletemail** 系统扩展存储过程 (ESP)，来读取、处理、发送和删除 Adaptive Server 收件箱中的消息。

语法

```
sp_processmail [subject] [, originator [, dbuser  
[, dbname [, filetype [, separator]]]]]
```

参数

subject — 是消息的标题。如果指定了 **subject**，但没有指定 **originator**，**sp_processmail** 将处理收件箱中所有具有指定标题的未读消息。如果既指定了 **subject**，又指定了 **originator**，**sp_processmail** 将处理具有指定标题并且由指定发件人发送的所有未读消息。如果既没有指定 **subject**，又没有指定 **originator**，则 **sp_processmail** 将处理 Adaptive Server 收件箱中的所有未读消息。

originator — 是传入消息的发送者。如果指定了 **originator**，但没有指定 **subject**，则 **sp_processmail** 将处理收件箱中由指定发件人发送的所有未读消息。

dbuser — 指定要用于用户环境以执行消息查询的 Adaptive Server 登录名。缺省值为 “guest”。

dbname — 指定要用于数据库环境以执行消息查询的数据库的名称。缺省值为 “master”。

filetype — 指定包含查询结果的所附文件的扩展名。缺省值为 “.txt”。

separator — 指定要在查询结果中用作列分隔符的字符。它与 **isql** 的 **/s** 选项作用相同。缺省值为制表符。

示例

```
1. sp_processmail @subject="SQL REPORT",  
   @originator="janet", @dbuser="sa",  
   @dbname="salesdb", @filetype="res", @separator=";"
```

处理 Adaptive Server 收件箱中具有“SQL Report”标题、由邮件用户“janet”提交的所有未读消息，作为 *salesdb* 数据库中的用户“sa”来处理收到的查询，并在附加到邮件消息的 *.res* 文件中将查询结果返回给“janet”。返回结果中的各列用分号分隔。

```
2. sp_processmail @dbuser="sa"
```

在 *master* 数据库中作为用户“sa”处理 Adaptive Server 收件箱中的所有未读消息，并将查询结果返回到 *.txt* 文件中，该文件将附加到邮件消息上。返回结果中的各列用制表符分隔。

注释

- **sp_processmail** 通过使用 **xp_findnextmsg**、**xp_readmail**、**xp_sendmail** 和 **xp_deletemail** 系统 ESP，可以读取、处理、发送和删除 Adaptive Server 收件箱中的消息。
- **sp_processmail** 将传出邮件发送到正在处理的传入邮件消息的发件人。
- **sp_processmail** 在调用 ESP 时使用缺省参数，但 **xp_sendmail** 的 *dbuser*、*dbname*、*attachname* 和 *separator* 参数除外，这些参数可以用 **sp_processmail** 的参数替换。
- **sp_processmail** 将所有消息作为 Adaptive Server 查询来处理。它从 Adaptive Server 收件箱读取消息，并通过 Adaptive Server 消息的附件将查询结果返回给消息的发件人、所有抄送的收件人和匿名收件人。**sp_processmail** 将为所附的文件生成文件名，该文件名由三部分组成：首先是“syb”，随后是五个随机数字，再后是用 *filetype* 参数指定的扩展名，如“syb84840.txt”。
- 处理完成后，**sp_processmail** 将消息从收件箱中删除。
- *subject* 和 *originator* 参数指定应该处理的消息。如果没有提供这些参数中的任何一个，**sp_processmail** 将处理 Adaptive Server 收件箱中的所有未读消息。
- **sp_processmail** 不处理传入邮件的附件。查询必须在传入消息的正文中。

权限

只有系统管理员才能执行 **sp_processmail**。

参见

系统 ESP	xp_deletemail、xp_findnextmsg、xp_readmail、xp_sendmail、xp_startmail
实用程序	isql

sp_procqmode

功能

显示存储过程、视图或触发器的查询处理模式。

语法

```
sp_procqmode [object_name [, detail]]
```

参数

object_name — 是要检查其查询处理模式的存储过程、视图或触发器的名称。如果未指定 *object_name*, **sp_procqmode** 将报告当前数据库中的所有存储过程、视图和触发器。

detail — 返回有关对象是否包含子查询的信息, 以及有关是否存在 *syscomments* 中的对象的信息。

示例

1. sp_procqmode

Object Owner.name	Object Type	Processing Mode
-----	-----	-----
dbo.au_info	stored procedure	pre-System 11
dbo.titleview	view	System 11 or later

显示当前数据库中所有存储过程的查询处理模式。

2. sp_procqmode old_sproc, detail

Object Owner.Name	Object Type	Processing Mode	Subq	Text
-----	-----	-----	-----	-----
dbo.au_info	stored procedure	pre-System 11	no	yes

显示存储过程 *old_sproc* 的查询处理模式, 报告 *old_sproc* 是否包含子查询, 并报告 *syscomments* 是否具有有关 *old_sproc* 的信息。

3. sp_procqmode null, detail

显示数据库中所有对象的详细报告。

注释

- 处理模式用于标识对象是在 SQL Server 10.0 版中创建的, 还是在更早的版本中创建的。在 10.x (或更早版本) Server 上创建的对象是 “pre-System 11” 对象。在 11.0 版或更高版本的 Server 上创建的对象是 “System 11 or later” 对象。

- “pre-System 11”对象中的子查询与“System 11 or later”对象中的子查询分别使用不同的处理模式。升级到 11.0 版或更高版本并不会自动更改子查询的处理模式。
一般来说，“System 11 or later”处理模式比“pre-System 11”处理模式快。要将处理模式更改为“System 11 or later”的处理模式，需要删除并重新创建对象。由于在当前版本的 Adaptive Server 上是不能用“pre-System 11”创建对象的，因此在删除使用“pre-System 11”处理模式的版本之前，可能需要用其它名称创建对象并对其进行检查。
- 给定对象所显示的处理模式与该对象是否实际包含子查询无关，它只与指定对象有关，而与任何依赖对象都无关。必须对每个对象分别进行检查。
- 详细报告显示了对象是否包含子查询，并报告 *syscomments* 中是否有可用文本（供 *sp_helptext* 进行报告或供 *defncopy* 实用程序进行复制）。*sp_procqmode* 并不检查 *syscomments* 中的文本是否有效或完整。

权限

只有数据库所有者或对象所有者才能执行 *sp_procqmode*。

使用的表

syscomments、*sysobjects*、*sysprocedures*

参见

存储过程	sp_helptext
实用程序命令	defncopy

sp_procxmode

功能

显示或更改与存储过程相关联的事务模式。

语法

```
sp_procxmode [procname [, tranmode]]
```

参数

- procname* — 是要检查或更改其事务模式的存储过程的名称。
- tranmode* — 是存储过程的新事务模式。其值为 "chained"、"unchained" 和 "anymode"。

示例

1. sp_procxmode

procedure name	user name	transaction mode
-----	-----	-----
byroyalty	dbo	Unchained
discount_proc	dbo	Unchained
history_proc	dbo	Unchained
insert_sales_proc	dbo	Unchained
insert_detail_proc	dbo	Unchained
storeid_proc	dbo	Unchained
storename_proc	dbo	Unchained
title_proc	dbo	Unchained
titleid_proc	dbo	Unchained

显示当前数据库中所有存储过程的事务模式。

2. sp_procxmode byroyalty

procedure name	transaction mode
-----	-----
byroyalty	Unchained

显示存储过程 *byroyalty* 的事务模式。

3. sp_procxmode byroyalty, "chained"

将 *pubs2* 数据库中的存储过程 *byroyalty* 的事务模式从 “unchained” 更改为 “chained”。

注释

- 要更改存储过程的事务模式，您必须是该存储过程的所有者、包含该存储过程的数据库的所有者或者是系统管理员。数据库所有者或系统管理员可以通过用数据库和用户名限定其他用户的存储过程，来更改其他用户存储过程的模式。例如：

```
sp_procxmode "otherdb.otheruser.newproc", "chained"
```

- 要使用 **sp_procxmode**，可以用 **set** 命令的 **chained** 选项，关闭链式事务模式。缺省情况下，该选项是关闭的。
- 在使用不带参数的 **sp_procxmode** 时，它将报告当前数据库中每个存储过程的事务模式。
- 要检查存储过程的事务模式（不作更改），可以输入：

```
sp_procxmode procname
```

- 要更改存储过程的事务模式，可以输入：

```
sp_procxmode procname, tranmode
```
- 在创建存储过程时，**Adaptive Server** 将用当前会话的事务模式对其进行标记。这意味着：
 - 只有在使用链式事务模式的会话中才能执行 “**chained**” 存储过程。
 - 只有在使用非链接事务模式的会话中才能执行 “**unchained**” 存储过程。

要在链式或非链接会话中执行特定的存储过程，可以将其事务模式设置为 “**anymode**”。

- 如果要尝试在错误的事务模式下运行存储过程，**Adaptive Server** 将返回警告消息，但不会影响当前的事务（如果有）。

权限

只有系统管理员、数据库所有者或过程所有者才能执行 **sp_procxmode** 来更改事务模式。任何用户都可以执行 **sp_procxmode** 来显示事务模式。

使用的表

sysobjects

参见

命令	begin transaction、commit、save transaction、set
----	---

sp_recompile

功能

使每个使用指定表的存储过程和触发器在下次运行时进行重新编译。

语法

```
sp_recompile objname
```

参数

objname — 是当前数据库中的表的名称。

示例

```
1. sp_recompile titles
```

重新编译每个触发器和存储过程，这些触发器和存储过程在下次运行时使用表 *titles*。

注释

- 存储过程和触发器所用的查询只能在编译时优化一次。在添加索引或对数据库进行其它影响其统计信息的更改时，所编译的存储过程和触发器可能会失效。通过重新编译作用于表的存储过程和触发器，可以将查询优化到最有效的状态。
- **sp_recompile** 只在当前数据库中查找 *objname*，并且只在当前数据库中重新编译触发器和存储过程。**sp_recompile** 不会影响其它数据库中依赖于该表的对象。
- 不能对系统表使用 **sp_recompile**。

权限

任何用户都可以执行 **sp_recompile**。

使用的表

sysobjects

参见

命令	create index、update statistics
----	--------------------------------

sp_remap

功能

重新映射 4.8 版和 10.0 版之间的存储过程、触发器、规则、缺省值或视图，使其与 10.0 版和更高版本兼容。对升级过程未能重新映射的预存在对象使用 **sp_remap**。

语法

```
sp_remap objname
```

参数

objname — 是当前数据库中存储过程、触发器、规则、缺省值或视图的名称。

示例

1. **sp_remap myproc**

重新映射名为 *myproc* 的存储过程。

2. **sp_remap "my_db..default_date"**

重新映射名为 *default_date* 的规则。运行该过程之前，需执行 **use my_db** 语句打开 *my_db* 数据库。

注释

- 如果 **sp_remap** 未能重新映射对象，可以将该对象从数据库中删除，然后重新创建它。在对某个对象运行 **sp_remap** 之前，最好用 **defncopy** 实用程序将其定义复制到操作系统文件中。有关 **defncopy** 的详细信息，参见针对于所用平台的 *实用程序手册*。
- **sp_remap** 可以使事务日志迅速填充。在运行 **sp_remap** 之前，可以根据需要使用 **dump transaction** 命令转储事务日志。
- 只能对当前数据库中的对象使用 **sp_remap**。
- 对于已成功升级到当前版本的对象，**sp_remap** 不作任何更改。

权限

只有系统管理员或对象所有者才能执行 **sp_remap**。

使用的表

master.dbo.sysdatabases、*sysobjects*

参见

命令	create default、create procedure、create rule、create trigger、create view、drop default、drop procedure、drop rule、drop trigger、drop view、dump transaction
系统过程	sp_helptext
实用程序	defncopy

sp_remotoption

功能

显示或更改远程登录选项。

语法

```
sp_remotoption [remoteserver [, loginname
                [, remotename [, optname [, optvalue]]]]]
```

参数

remoteserver — 是将要在其上执行 RPC 的服务器的名称。

► 注意

本手册页使用术语“本地服务器”表示正执行从“远程服务器”运行的远程过程的服务器。

loginname — 即登录名，它标识 *remoteserver*、*loginname*、*remotename* 组合的本地登录。

remotename — 即远程用户名，它标识 *remoteserver*、*loginname*、*remotename* 组合的远程登录。

optname — 是要更改的选项的名称。当前只有一个选项，即 **trusted**。这意味着本地服务器将接受来自其它服务器的远程登录，而不会对特定远程登录进行用户访问验证。缺省情况下将使用口令验证。**Adaptive Server** 可以识别选项名中的任何唯一字符串。如果选项名包括嵌入的空白，请用引号将其引起来。

optvalue — 是 **true** 或 **false**。**true** 将打开选项，而 **false** 则关闭选项。

示例

1. sp_remotoption

```
Settable remote login options.
```

```
remotelogin_option
```

```
-----
```

```
trusted
```

显示远程登录选项的列表。

2. **sp_remotoption GATEWAY, churchy, pogo, trusted, true**

将来自远程服务器 GATEWAY 的远程登录定义为 **trusted**（即不检查口令）。

3. **sp_remotoption GATEWAY, churchy, pogo, trusted, false**

将来自远程服务器 GATEWAY 的远程登录 “pogo” 定义为不受托（将检查口令）。

4. **sp_remotoption GATEWAY, albert, NULL, trusted, true**

将 GATEWAY 中映射到本地服务器上登录 “albert” 的所有登录定义为 **trusted**。

注释

- 要显示远程登录选项的列表，可执行不带参数的 **sp_remotoption**。
- 如果已使用 **sp_addremotelogin** 将远程服务器中的所有用户映射为相同的本地名，则可以为这些用户指定 **trusted**。例如，要使服务器 GOODSrv 中所有映射为 “albert” 的用户受托，可指定：

```
sp_remotoption GOODSrv, albert, NULL, trusted true
```

如果没有将这些登录指定为 **trusted**，它们将不能在本地服务器上执行 RPC，除非它们在登录到远程服务器时指定了本地服务器口令。当使用 **Open Client Client-Library** 时，用户可以用例程 **ct_remote_pwd** 指定服务器到服务器连接的口令。**isql** 和 **bcp** 不允许用户指定 RPC 连接的口令。

如果用户使用 “统一登录” 登录到远程服务器，那么这些登录在本地服务器上也必须是受托的，否则当用户登录到远程服务器时，他们必须指定服务器的口令。

有关为远程过程调用设置服务器和使用 “统一登录” 的详细信息，参见 *系统管理指南*。

权限

只有系统安全员才能执行 **sp_remotoption**。

使用的表

master.dbo.spt_values、*master.dbo.sysmessages*、*master.dbo.sysremotelogins*、*master.dbo.sysservers*、*sysobjects*

参见

系统过程	sp_addremotelogin、 sp_dropremotelogin、 sp_helpremotelogin
实用程序	isql

sp_remotesql

(仅适用于组件集成服务)

功能

建立与远程服务器的连接，将查询缓冲区从客户端传递到远程服务器，然后再将结果传递回客户端。

语法

```
sp_remotesql server, query  
[, query2, ... , query254]
```

参数

server_name — 是用 **sp_addserver** 定义的远程服务器的名称。

query — 是最大长度可达 255 个字符的查询缓冲区。

query2 到 *query254* — 是最大长度可达 255 个字符的查询缓冲区。如果提供了这些参数，它们将与 *query1* 的内容一起并置到单个查询缓冲区中。

示例

```
1. sp_remotesql FREDs_SERVER, "select @@version"
```

将查询缓冲区传递到 FREDs_SERVER 中，而后者将解释 **select @@version** 并将结果返回到客户端。Adaptive Server 不解释此结果。

```
2. create procedure fred's_version  
as  
exec sp_remotesql FREDs_SERVER, "select @@version"  
go  
exec fred's_version  
go
```

说明 **sp_remotesql** 在存储过程中的用法。该示例和示例 1 将向客户端返回相同的信息。


```

3. sp_remotesql DCO_SERVER,
   "insert into remote_table
   (numbercol,intcol, floatcol,datecol )",
   "values (109.26,75, 100E5,'10-AUG-85')",
   select @@error

```

服务器将两个查询缓冲区并置到单个缓冲区中，然后将完整的 **insert** 语句传递到服务器 **DCO_SERVER** 中进行处理。**insert** 语句的语法采用的是 **DCO_SERVER** 可以理解的格式。此服务器不解释返回的信息。该示例还将检查在 **@@error** 中返回的值。

```

4. declare @servname varchar(30)
   declare @querybuf varchar(200)
   select @servname = "DCO_SERV"
   select @querybuf = "select table_name
                       from all_tables
                       where owner = 'SYS'"
   exec sp_remotesql @servname, @querybuf

```

说明将本地变量用作 **sp_remotesql** 参数方法。

注释

- **sp_remotesql** 建立与远程服务器的连接，将查询缓冲区从客户端传递到远程服务器，然后再将结果传递回客户端。本地服务器不对结果进行解释。
- 可以在其它存储过程内使用 **sp_remotesql**。
- 查询缓冲区参数必须是最大长度为 255 个字符的字符表达式。如果使用的查询缓冲区不是 **char** 或 **varchar**，您将收到数据类型转换错误。
- 如果消息的严重级别大于 10，**sp_remotesql** 将把全局变量 **@@error** 设置为从远程服务器返回的最后一条错误消息的值。
- 如果 **sp_remotesql** 是从事务内部发出的，**Adaptive Server** 将在传递查询缓冲区进行执行之前，确保事务已经在远程服务器上开始。当事务终止时，将指示远程服务器提交该事务。查询缓冲区内容执行的工作是事务所定义的工作单元的一部分。

如果事务控制语句是查询缓冲区的一部分，客户端有责任确保事务按预期的方式进行 **commit**（提交）和 **rollback**（回退）。如果在查询缓冲区中混合使用 **Transact-SQL** 和事务控制命令，可能会导致无法预料的结果。

- 本地服务器管理与远程服务器的连接。如果将 **connect to** 或 **disconnect** 命令嵌入查询缓冲区，则会产生要求远程服务器来解释的结果。这不是必需或推荐的方法。其结果通常是语法错误。

权限

任何用户都可以执行 **sp_remotesql**。

使用的表

不使用任何表。

参见

命令	connect to...disconnect
系统过程	sp_autoconnect、 sp_passthru

sp_rename

功能

更改当前数据库中用户创建对象或用户定义数据类型的名称。

语法

```
sp_rename objname, newname
```

参数

objname — 是用户创建对象（表、视图、列、存储过程、索引、触发器、缺省值、规则、检查约束、参照约束或用户定义的数据类型）的初始名称。如果要重命名的对象是表中的列，**objname** 的格式必须为 “**table.column**”。如果对象是索引，**objname** 的格式必须为 “**table.indexname**”。

newname — 是对象或数据类型的新名称。该名称必须符合标识符规则，并且在当前数据库中必须是唯一的。

示例

1. **sp_rename titles, books**

将 *titles* 表重命名为 *books*。

2. **sp_rename "books.title", bookname**

将 *books* 表中的 *title* 列重命名为 *bookname*。

3. **sp_rename "books.titleind", titleindex**

将 *books* 表中的 *titleind* 索引重命名为 *titleindex*。

4. **sp_rename tid, bookid**

将用户定义的数据类型 *tid* 重命名为 *bookid*。

注释

- **sp_rename** 更改用户创建对象或数据类型的名称。可以只更改发出 **sp_rename** 时所在数据库的对象或数据类型的名称。
- 重命名列或索引时，不要在 **newname** 中指定表名。参见示例 2 和 3。
- 可以更改视图所引用对象的名称。例如，如果视图引用 *new_sales* 表，而您将 *new_sales* 重命名为 *old_sales*，该视图就将引用 *old_sales*。
- 不能更改系统对象和系统数据类型的名称。

◆ 警告!

如果过程、触发器和视图依赖于名称已更改的对象，它们将继续有效，直到它们被删除并重新创建。同样，旧对象名一直会在查询结果中出现，直到用户更改并重新创建过程、触发器或视图。执行 `sp_rename` 时，应更改任何相关对象的定义。使用 `sp_depends` 可查找相关对象。

权限

只有数据库所有者或系统管理员才能使用 `setuser` 命令，以其他数据库用户的身份来重命名其他用户拥有的对象。所有用户都可以执行 `sp_rename` 来重命名自己的对象。

使用的表

syscolumns、*sysindexes*、*sysobjects*、*systypes*

参见

命令	<code>alter table</code> 、 <code>create default</code> 、 <code>create procedure</code> 、 <code>create rule</code> 、 <code>create table</code> 、 <code>create trigger</code> 、 <code>create view</code>
系统过程	<code>sp_addtype</code> 、 <code>sp_checkreswords</code> 、 <code>sp_depends</code> 、 <code>sp_renamedb</code>

sp_renamedb

功能

更改用户数据库的名称。

语法

```
sp_renamedb dbname, newname
```

参数

dbname — 是数据库的初始名称。

newname — 是数据库的新名称。数据库名称必须符合标识符规则，并且必须是唯一的。

示例

```
1. sp_renamedb accounting, financial
```

将 *accounting* 数据库重命名为 *financial*。

```
2. sp_dboption work, single, true
```

```
go  
use work  
go  
checkpoint  
go  
sp_renamedb work, workdb  
go  
use master  
go  
sp_dboption workdb, single, false  
go  
use workdb  
go  
checkpoint  
go
```

将名为 *work* （它是 Transact-SQL 保留字）的数据库重命名为 *workdb*。

注释

- **sp_renamedb** 更改数据库的名称。**不能**重命名系统数据库或具有外部参照完整性约束的数据库。
- 重命名数据库之前，系统管理员必须用 **sp_dboption** 将数据库置于单用户模式下，重命名后必须将其恢复到多用户模式。
- 如果该数据库中的表引用其它数据库中的表（或者被其它数据库中的表引用），**sp_renamedb** 将会失败。使用以下查询可确定哪些表和外部数据库具有当前数据库中主键表的外键约束：

```
select object_name(tableid), db_name(frgndbid)
from sysreferences
where frgndbid is not null
```

使用以下查询可确定哪些表和外部数据库具有当前数据库中表的主键约束：

```
select object_name(reftabid), db_name(pmrydbid)
from sysreferences
where pmrydbid is not null
```

使用 **alter table** 删除这些表中的跨数据库约束。然后，重新运行 **sp_renamedb**。

- 当更改数据库名时：
 - 删除所有包含该数据库名的存储过程、触发器和视图
 - 更改被删除对象的源文本，以反映新的数据库名
 - 重新创建被删除的对象
 - 更改所有引用数据库的应用程序和 SQL 源脚本，无论数据库是在 **use database_name** 命令中引用的，还是作为完全限定标识符的一部分（其形式为 **dbname.[owner].objectname**）引用的。
- 如果使用脚本在数据库上运行 **dbcc** 命令或 **dump database** 和 **dump transaction** 命令，务必要更新这些脚本。

◆ 警告！

如果过程、触发器和视图所依赖的数据库的名称已被更改，它们在被重新创建之前将一直有效。当执行 **sp_renamedb** 时，应更改任何相关对象的定义。使用 **sp_depends** 可查找相关对象。

权限

只有系统管理员才能执行 **sp_renamedb**。

使用的表

master.dbo.spt_values、 *master.dbo.sysdatabases*、 *sysobjects*

参见

命令	create database
系统过程	sp_changedbowner、 sp_dboption、 sp_depends、 sp_helpdb、 sp_rename

sp_rename_qpgroup

功能

重命名抽象计划组。

语法

```
sp_rename_qpgroup old_name, new_name
```

参数

- old_name* — 是抽象计划组的当前名称。
- new_name* — 是抽象计划组的新名称。指定的 *new_name* 不能是数据库中现有抽象计划组的名称。

示例

```
1. sp_rename_qpgroup dev_plans, prod_plans
```

将组的名称从 *dev_plans* 更改为 *prod_plans*。

注释

- 使用 **sp_rename_qpgroup** 可重命名抽象计划组。不能将现有抽象计划组的名称用作新名称。
- sp_rename_qpgroup** 不会影响已重命名的组的内容。现有抽象计划组的 ID 不会更改。
- 不能重命名缺省的抽象计划组 (*ap_stdin* 和 *ap_stdout*)。
- sp_rename_qpgroup** 不能在事务中运行。

权限

只有系统管理员或数据库所有者才能执行 **sp_rename_qpgroup**。

使用的表

sysattributes

参见

系统过程	sp_help_qpgroup
------	-----------------

sp_reportstats

功能

报告有关系统使用情况的统计信息。

语法

```
sp_reportstats [loginame]
```

参数

loginame — 要向其显示会计总和的用户的登录名。

示例

1. sp_reportstats

Name	Since	CPU	Percent CPU	I/O	Percent I/O
-----	-----	-----	-----	-----	-----
julie	jun 19 1993	10000	24.9962%	5000	24.325%
jason	jun 19 1993	10002	25.0013%	5321	25.8866%
ken	jun 19 1993	10001	24.9987%	5123	24.9234%
kathy	jun 19 1993	10003	25.0038%	5111	24.865%
		Total CPU	Total I/O		
		-----	-----		
		40006	20555		

显示所有 Adaptive Server 用户的当前会计总和的报告。

2. sp_reportstats kathy

Name	Since	CPU	Percent CPU	I/O	Percent I/O
-----	-----	-----	-----	-----	-----
kathy	Jul 24 1993	498	49.8998%	48392	9.1829%
		Total CPU	Total I/O		
		-----	-----		
		998	98392		

显示用户 “kathy” 的当前会计总和的报告。

注释

- **sp_reportstats** 输出所有登录的当前会计总和，以及各个登录的统计信息和占总统计信息的百分比。**sp_reportstats** 将接受一个参数，即要报告的帐户的登录名。如果不提供参数，**sp_reportstats** 将报告所有帐户的信息。
- **sp_reportstats** 不报告任何系统用户 ID (*suid*) 为 0 或 1 的进程的统计信息。其中包括死锁检测、检查点、管家任务、网络、审计、镜像处理程序和具有 **sa_role** 的所有用户。
- “CPU” 的报告单元是**计算机**时钟周期，而不是 Adaptive Server 时钟周期。
- “probe” 用户主要用于两阶段提交探查进程，该进程使用 “询问和答复” 机制来访问 Adaptive Server。

权限

只有系统管理员才能执行 **sp_reportstats**。

使用的表

master.dbo.syslogins、 *sysobjects*

参见

系统过程	sp_clearstats 、 sp_configure
------	--

sp_revokelogin

(仅适用于 Windows NT)

功能

当“集成安全性”模式或“混合”模式（具有命名管道）处于活动状态时，撤消 Windows NT 用户和组的 Adaptive Server 角色和缺省权限。

语法

```
sp_revokelogin {login_name | group_name}
```

参数

login_name — 是 Windows NT 用户的网络登录名。

group_name — 是 Windows NT 组名。

示例

1. **sp_revokelogin jeanluc**

撤消名为“jeanluc”的 Windows NT 用户的所有权限。

2. **sp_revokelogin Administrators**

撤消 Windows NT 管理员组的所有角色。

注释

- 当连接为命名管道时，只有在 Adaptive Server 以“集成安全性”模式或“混合”模式运行的情况下，才使用 **sp_revokelogin**。如果 Adaptive Server 以“标准”模式运行，或者使用命名管道之外的连接以“混合”模式运行，则应使用 **revoke** 命令。
- 如果用 **sp_revokelogin** 撤消用户的角色和缺省权限，该用户将再也不能通过受托连接登录到 Adaptive Server。

权限

只有系统管理员才能执行 **sp_revokelogin**。

使用的表

sysobjects

参见

命令	grant、revoke、setuser
系统过程	sp_droplogin、sp_dropuser、sp_logininfo

sp_role

功能

授予或撤消 Adaptive Server 登录帐户的角色。

语法

```
sp_role { "grant" | "revoke" }, rolename, loginame
```

参数

grant | **revoke** — 指定是向 *loginame* 授予角色还是撤消其角色。

rolename — 是要授予或撤消的角色。

loginame — 是要授予或撤消角色的登录帐户。

示例

```
1. sp_role "grant", sa_role, alexander
```

将系统管理员角色授予名为 “alexander” 的登录帐户。

注释

- **sp_role** 可以向 Adaptive Server 登录帐户授予角色或撤消其角色。
- 向用户授予角色时，角色将在该用户下次登录到 Adaptive Server 时生效。或者，用户可以通过使用 **set role** 命令立即启用角色。例如，命令：

```
set role sa_role on
```

对该用户启用系统管理员角色。

- 不能在用户登录时撤消该用户的角色。
- 当用户登录时，已向其授予的所有角色都将处于活动状态 (**on**)。要关闭角色，可使用 **set** 命令。例如，要使系统管理员角色失活，可使用命令：

```
set role "sa_role" off
```

权限

只有系统管理员才能执行 **sp_role** 将系统管理员角色授予其他用户。只有系统安全员才能执行 **sp_role** 将除 “sa” 以外的任何角色授予其他用户。

使用的表

*master.dbo.sysloginroles、 master.dbo.syslogins、
master.dbo.sysprocesses、 master.dbo.syssrvroles、 sysobjects*

参见

命令	grant、 revoke、 set
函数	proc_role
系统过程	sp_displaylogin

sp_sendmsg

功能

将消息发送到 UDP（用户数据报协议）端口。

语法

```
sp_sendmsg ip_address, port_number, message
```

参数

ip_address — 是运行 UDP 应用程序的计算机的 IP 地址。

port_number — 是 UDP 端口的端口号。

示例

```
1. sp_sendmsg "120.10.20.5", 3456, "Hello World"
```

注释

- Windows NT 或 NCR 平台不支持 **sp_sendmsg**。
- 系统安全员必须将配置参数 **allow sendmsg** 设置为 1，才能启用 UDP 消息传送功能。
- 对 **syb_sendmsg** 没有安全性检查。Sybase 强烈建议您使用 **syb_sendmsg** 在网络上发送敏感信息时一定要谨慎。如果启用此功能，则表示用户接受使用该功能所导致的任何安全性问题。
- 该 C 程序的示例监听您指定的端口并回送它收到的消息。例如，要接收上面示例中的 **syb_sendmsg** 调用，可使用：

```
updmon 3456
```

```
#include <stdlib.h>
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <unistd.h>
#include <fcntl.h>

main(argc, argv)
int argc; char *argv[];
{
    struct sockaddr_in saddr;
    int portnum, sck, dummy, msglen;
    char msg[256];
```

```

if (argc < 2) {
    printf("Usage: udpmon <udp portnum>\n");
    exit(1);
}

if ((portnum=atoi(argv[1])) < 1) {
    printf("Invalid udp portnum\n");
    exit(1);
}

if ((sck=socket(AF_INET,SOCK_DGRAM,IPPROTO_UDP)) < 0) {
    printf("Couldn't create socket\n");
    exit(1);
}

sadr.sin_family = AF_INET;
sadr.sin_addr.s_addr = inet_addr("0.0.0.0");
sadr.sin_port = portnum;

if (bind(sck,&sadr,sizeof(sadr)) < 0) {
    printf("Couldn't bind requested udp port\n");
    exit(1);
}

for (;;)
{
    if((msglen=recvfrom(sck,msg,sizeof(msg),0,NULL,&dummy)) < 0)
        printf("Couldn't recvfrom() from udp
port\n");

        printf("%.s\n", msglen, msg);
    }
}

```

权限

任何用户都可以执行 **sp_sendmsg**。

参见

函数	syb_sendmsg
----	-------------

sp_serveroption

功能

显示或更改远程服务器选项。

语法

```
sp_serveroption [server, optname, optvalue]
```

参数

- server* — 是要设置选项的远程服务器的名称。
- optname* — 是要设置或不设置的选项名。表 7-22 列出了选项名。

表 7-22: sp_serveroption 选项

选项	含义
net password encryption	指定开始与远程服务器连接时，是使用客户端口令加密握手还是使用常规（未加密口令）握手序列。缺省值是 false ，没有进行网络加密。
readonly	指定对已命名的服务器的访问为只读访问。该选项仅适用于组件集成服务。
rpc security model A	处理 RPC 的缺省模式。该模式不支持本地服务器和远程服务器之间的相互鉴定、消息完整性或消息保密性。
timeouts	由于不设置该选项 (false) 时，禁用本地服务器使用的常规超时代码，因此在缺少逻辑连接的情况持续一分钟后，站点连接处理程序并不自动删除物理连接。缺省值是 true 。

Adaptive Server 接受作为选项名一部分的任何唯一的字符串。如果选项名包括嵌入的空白，请用引号将其引起来。

示例**1. sp_serveroption**

Settable server options.

```
-----
net password encryption
readonly
rpc security model A
timeouts
timeouts
net password encryption
```

显示服务器选项的列表。

2. sp_serveroption GATEWAY, "timeouts", false

通知服务器不要使与远程服务器 GATEWAY 的非活动物理连接超时。

3. sp_serveroption GATEWAY, "net password encryption", true

指定当连接到远程服务器 GATEWAY 时，GATEWAY 送回加密密钥，为发送到该服务器的口令加密。

注释

- 要显示用户可设置的服务器选项的列表，请使用不带参数的 **sp_serveroption**。
- 一旦将 **timeouts** 设置为 **false**，站点处理程序将继续运行，直到关闭两个服务器中的一个。
- **net password encryption** 选项允许客户端在开始远程过程调用时，指定在网络上是以明文还是以加密的形式来发送口令。如果 **net password encryption** 是 **true**，则发送初始登录包时无口令，而且客户端向远程服务器表明需要加密。远程服务器送回加密密钥，客户端可以用它来加密其口令。客户端加密其口令，然后，当加密的口令到达远程服务器后，远程服务器使用该密钥来对口令进行鉴定。
- 要为特定 **isql** 会话设置网络口令加密，可使用 **isql** 的命令行选项。有关详细信息，参见针对于所用平台的**实用程序手册**。
- 连接到版本 10.0 之前的 SQL Server 时，不能使用 **net password encryption** 选项。
- 有关服务器选项的详细信息，参见**系统管理指南**。

权限

只有系统管理员才能执行 **sp_serveroption** 来设置 **timeouts** 选项。任何用户都可以执行不带参数的 **sp_serveroption** 来显示选项列表。

使用的表

master.dbo.sysservers、*sysobjects*、*syssecmechs*

参见

系统过程	sp_helpserver、sp_password
实用程序	isql

sp_setlangalias

功能

指派或更改替代语言的别名。

语法

```
sp_setlangalias language, alias
```

参数

language — 是替代语言的正式语言名称。

alias — 是替代语言的新的本地别名。

示例

```
1. sp_setlangalias french, français
```

该命令为正式语言名称 “french” 指派别名 “français”。

注释

- *alias* 为正式名称替换 *syslanguages.alias* 的当前值。
- **set language** 命令可使用新的 *alias* 来代替正式语言名称。

权限

只有系统管理员才能执行 **sp_setlangalias**。

使用的表

master.dbo.syslanguages、*sysobjects*

参见

命令	set
系统过程	sp_addlanguage、sp_droplanguage、 sp_helplanguage

sp_setpglockpromote

功能

设置或更改数据库、表或 Adaptive Server 的锁升级阈值。

语法

```
sp_setpglockpromote {"database" | "table"}, objname,  
                    new_lwm, new_hwm, new_pct  
  
sp_setpglockpromote server, NULL, new_lwm, new_hwm,  
                    new_pct
```

参数

server — 为锁升级阈值设置全服务器范围的值。

"database" | "table" — 指定是否设置数据库或表的锁升级阈值。由于“database”和“table”是 Transact-SQL 关键字，所以需要引用号引起来。

objname — 是要为其设置锁升级阈值或 **null**（如果要设置全服务器范围的值）的表名或数据库名。

new_lwm — 指定要为低水平 (LWM) 阈值设置的值。LWM 必须小于或等于高水平 (HWM)。LWM 的最小值为 2。该参数可以为 **null**。

new_hwm — 指定要为锁升级 HWM 阈值设置的值。HWM 必须大于或等于 LWM。HWM 的最大值为 2,147,483,647。该参数可以为 **null**。

new_pct — 指定要为锁升级百分比 (PCT) 阈值设置的值。PCT 必须介于 1 与 100 之间。该参数可以为 **null**。

示例

1. **sp_setpglockpromote "server", NULL, 200, 300, 50**
设置全服务器范围的锁升级 LWM 为 200，HWM 为 300，PCT 为 50。
2. **sp_setpglockpromote "database", master, 1000, 1100, 45**
为 *master* 数据库设置锁升级阈值。
3. **sp_setpglockpromote "table", "pubs2..titles", 500, 700, 10**

为 *pubs2* 数据库中的 *titles* 表设置锁升级阈值。该命令必须从 *pubs2* 数据库发出。

```
4. sp_setpglockpromote "database", master,  
   @new_hwm=1600
```

将 *master* 数据库的 HWM 阈值更改为 160。阈值以前是用 *sp_setpglockpromote* 设置的。该命令必须从 *master* 数据库发出。

注释

- *sp_setpglockpromote* 为表、数据库或 Adaptive Server 配置锁升级值。

Adaptive Server 将获取表上的页锁，直到锁的数量超过锁升级阈值。*sp_setpglockpromote* 更改对象、数据库或服务器的锁升级阈值。如果 Adaptive Server 成功获取了表锁，将释放页锁。

若表中锁的数目超过了 HWM 阈值，Adaptive Server 将试图升级到表锁。若表中锁的数目低于 LWM，则 Adaptive Server 不会试图升级到表锁。若表中锁的数目介于 HWM 和 LWM 之间且超过了 PCT 阈值，Adaptive Server 将试图升级到表锁。
- 表的锁升级阈值将覆盖数据库或全服务器范围的设置。数据库的锁升级阈值将覆盖全服务器范围的设置。
- Adaptive Server 的锁升级阈值不需要初始化，但必须用 *sp_setpglockpromote* 指定 LWM、HWM 和 PCT 来初始化数据库和表的锁升级阈值，该命令在数据库或表中首次运行时，将为 *sysattributes* 中的对象创建行。阈值初始化后，可分别修改它们，如示例 4 所示。
- 对于表或数据库，*sp_setpglockpromote* 在单个事务中设置 LWM、HWM 和 PCT。如果 *sp_setpglockpromote* 在更新其中的任何一个值时遇到错误，那么所有更改将中止，且事务将回退。对于全服务器范围的更改，可能会发生一个或多个阈值更新失败，而其它阈值却更新成功。任何一个值更新失败时，Adaptive Server 都将返回错误消息。
- 要查看锁升级阈值的全服务器范围的设置，可以使用 *sp_configure* "lock promotion" 查看全部三个阈值。要查看数据库的锁升级设置，可以使用 *sp_helpdb*。要查看表的锁升级设置，可以使用 *sp_help*。

权限

只有系统管理员才能执行 *sp_setpglockpromote*。

使用的表

master.dbo.sysattributes、*master.dbo.sysconfigures*、*sysattributes*

参见

系统过程	sp_configure、sp_dropglockpromote、sp_help、sp_helpdb
------	--

sp_setpsex

功能

为活动的会话设置自定义执行特性。

语法

```
sp_setpsex spid, exeattr, value
```

参数

spid — 是要设置执行变量的会话的 ID。使用 **sp_who** 来查看 *spid*。

exeattr — 标识要设置的执行特性。其值是 **priority** 和 **enginegroup**。

value — 是 *exeattr* 的新值。每个特性的值如下所示：

- 如果 *exeattr* 是 **priority**，则 *value* 是 **HIGH**、**MEDIUM** 或 **LOW**。
- 如果 *exeattr* 是 **enginegroup**，则 *value* 是现有引擎组的名称。

示例

```
1. sp_setpsex 1, "priority", "HIGH"
```

该语句将 ID 为 1 的进程的优先级设置为 **HIGH**。

注释

- 用 **sp_setpsex** 指定的执行特性值仅对当前会话有效，会话终止后不再适用。
- 慎用 **sp_setpsex**，否则会导致性能下降。如果进程未获得 CPU 时间，用 **sp_setpsex** 更改特性 “on the fly” 会有所帮助；而对于其它原因导致的性能问题（例如锁），更改执行特性可能会使问题更严重。
- 由于只能为会话设置执行特性，所以不能为工作进程 *spid* 设置 **sp_setpsex**。
- 除非是管家 *spid*，否则不能为系统 *spid* 设置执行特性。
- 如果相关引擎组中没有联机引擎，**sp_setpsex** 将不起作用。

权限

只有系统管理员才能无限制地执行 **sp_setpsex**。任何用户都可以执行 **sp_setpsex**，以降低该用户所拥有的进程的优先级。

使用的表

sysattributes、 *sysprocesses*

参见

系统过程	sp_addengine、 sp_addexclass、 sp_bindexclass、 sp_clearpsex、 sp_dropengine、 sp_dropexclass、 sp_showcontrolinfo、 sp_showexclass、 sp_showpsex
------	---

sp_set_qplan

功能

更改现有计划的抽象计划的文本，而不更改相关查询。

语法

```
sp_set_qplan id, plan
```

参数

id — 是抽象计划的 ID。

plan — 是新的抽象计划。

示例

```
1. sp_set_qplan 563789159,  
   "( g_join (scan t1) (scan t2))"
```

注释

- 使用 **sp_set_qplan** 更改现有计划的抽象计划。最多可为一个计划指定 255 个字符。如果抽象计划包含的字符数超过了 255，可用 **sp_drop_qplan** 删除旧计划，然后使用 **create plan** 创建查询的新计划。
- 使用 **sp_set_qplan** 更改计划时，并不检查计划的抽象计划语法是否有效。另外，也不检查计划是否与 SQL 文本兼容。应立即运行指定 ID 的查询，检查用 **sp_set_qplan** 修改的所有计划是否正确。
- 要查找某一计划的 ID，可使用 **sp_help_qpgroup**、**sp_help_qplan** 或 **sp_find_qplan**。计划 ID 也可以由 **create plan** 返回并包含在 **showplan** 的输出中。

权限

任何用户都可以执行 **sp_set_qplan**，以更改自己拥有的计划的文本。只有系统管理员或数据库所有者才能更改属于其他用户的计划的文本。

使用的表

sysqueryplans

参见

命令	create plan
系统过程	sp_find_qplan、 sp_help_qplan

sp_setrowlockpromote

功能

为某个数据行锁定表、数据库中所有的数据行锁定表或服务器中所有的数据行锁定表设置或更改行锁升级阈值。

语法

```
sp_setrowlockpromote "server", NULL, new_lwm,  
                      new_hwm, new_pct  
  
sp_setrowlockpromote {"database" | "table"},  
                      objname, new_lwm, new_hwm, new_pct
```

参数

server — 为行锁升级阈值设置全服务器范围的值。

"database" | "table" — 指定是否设置数据库或表的行锁升级阈值。

objname — 是要为其设置行锁升级阈值或 **null**（如果要设置全服务器范围的值）的表名或数据库名。

new_lwm — 指定要为低水平 (LWM) 阈值设置的值。LWM 必须小于或等于高水平 (HWM)。LWM 的最小值为 2。该参数可以为 **null**。

new_hwm — 指定要为高水平 (HWM) 阈值设置的值。HWM 必须大于或等于 LWM。HWM 的最大值为 2,147,483,647。该参数可以为 **null**。

new_pct — 指定要为锁升级百分比 (PCT) 阈值设置的值。PCT 必须介于 1 与 100 之间。该参数可以为 **null**。

示例

1. `sp_setrowlockpromote "database", engdb, 400, 400,95`
为 `engdb` 数据库中所有的数据行锁定表设置行锁升级值。
2. `sp_setrowlockpromote "table", sales, 250, 250, 100`
为 `sales` 表设置行锁升级值。

注释

- **sp_setrowlockpromote** 设置或更改表、数据库或 Adaptive Server 的行锁升级阈值。

Adaptive Server 将获取数据行锁定表上的行锁，直到锁的数量超过锁升级阈值。如果 Adaptive Server 成功获取了表锁，将释放行锁。

若表中的行锁数目超过了 HWM，Adaptive Server 将试图升级到表锁。若表中的行锁数目低于 LWM，则 Adaptive Server 不会试图升级到表锁。若表中的行锁数目位于 HWM 和 LWM 之间，但超过了表中的 PCT 阈值（表中的行数百分比），Adaptive Server 将试图升级到表锁。
- 锁升级总是双重的，也就是说，行锁升级为表锁。Adaptive Server 不会从行锁升级为页锁。
- 表的锁升级阈值将覆盖数据库或全服务器范围的设置。数据库的锁升级阈值将覆盖全服务器范围的设置。
- 要改变数据库的锁升级阈值，则正在使用的数据库必须是 *master* 数据库。要改变数据库中表的锁升级阈值，则正在使用的数据库必须是表驻留的数据库。
- 也可使用 **sp_configure** 来设置全服务器范围的行锁升级阈值。使用 **sp_setrowlockpromote** 来改变全服务器范围的值时，将更改配置参数，并保存配置文件。首次安装 Adaptive Server 时，由配置参数设置的全服务器范围的行锁升级阈值是：

row lock promotion HWM	200
row lock promotion LWM	200
row lock promotion PCT	100

有关详细信息，参见 *系统管理指南*。

- 系统过程 **sp_sysmon** 报告有关行锁升级的信息。
- 数据库级的行锁升级阈值存储在 *master.sysattributes* 表中。如果转储数据库，并且只将其装载到另一个服务器，则必须新的服务器上设置行锁升级阈值。对象级的行锁升级阈值存储在用户数数据库的 *sysattributes* 表中，并包含在转储中。

权限

只有系统管理员可以执行 **sp_setrowlockpromote**。

使用的表

master.dbo.sysattributes、 *master.dbo.sysconfigures*、 *sysattributes*

参见

系统过程	sp_droprowlockpromote
------	-----------------------

sp_setsuspect_granularity

功能

显示或设置用户数据库的恢复故障隔离模式，当检测到数据损坏时，该模式将管理恢复行为。

语法

```
sp_setsuspect_granularity [dbname  
    [, "database" | "page" [, "read_only"]]]
```

参数

- dbname** — 是要显示或设置恢复故障隔离模式的数据库名。对于显示恢复故障隔离模式，缺省值是当前数据库。对于设置恢复故障隔离模式，则必须在 *master* 数据库中指定目标 **dbname**。
- database** — 将整个数据库标记为可疑，如果恢复进程检测到任何可疑的数据，将导致无法访问数据库。
- page** — 仅将损坏页标记为可疑，如果恢复进程检测到数据库中的损坏数据，只是导致无法访问损坏页。其余数据仍可访问。
- read_only** — 指定该参数后，如果恢复进程将任何页标记为可疑，则整个数据库将被标记为 **read only**。

示例

1. sp_setsuspect_granularity

DB Name	Cur. Suspect Gran.	Cfg. Suspect Gran.	Online mode
-----	-----	-----	-----
pubs2	database	database	read/write

显示当前数据库的恢复故障隔离模式。

2. sp_setsuspect_granularity pubs2

显示 *pubs2* 数据库的当前恢复故障隔离模式和已配置的恢复故障隔离模式。

3. **sp_setsuspect_granularity pubs2, "page"**

DB Name	Cur. Suspect Gran.	Cfg. Suspect Gran.
pubs2	database	database

sp_setsuspect_granularity: The new values will become effective during the next recovery of the database 'pubs2'.

下次在 *pubs2* 数据库中运行恢复时，如果检测到任何损坏页，仅将可疑页脱机，而数据库的其余部分仍将联机。

4. **sp_setsuspect_granularity pubs2, "page", "read_only"**

下次在 *pubs2* 数据库中运行恢复时，如果检测到任何损坏页，仅将可疑页脱机，而数据库的其余部分将以只读模式联机。

5. **sp_setsuspect_granularity pubs2, "database"**

下次在 *pubs2* 数据库中运行恢复时，如果检测到任何损坏数据，整个数据库将被标记为可疑并脱机。

注释

- **sp_setsuspect_granularity** 显示并设置恢复故障隔离模式。当恢复进程检测到所需数据已损坏时，该模式将确定恢复进程是将整个数据库标记为可疑还是仅将损坏页标记为可疑。有关详细信息，参见 *系统管理指南*。
- 用户数据库的缺省恢复故障隔离模式是 “**database**”。只能为用户数据库设置恢复故障隔离模式，而不能为系统数据库设置。
- 必须在 *master* 数据库中设置恢复故障隔离模式。
- 因损坏被标记为可疑的数据在 **Adaptive Server** 启动过程中仍保留。重新启动服务器后，被标记为可疑的页将保持脱机状态。
- 将数据库的一部分或全部标记为可疑后，用户将无法访问可疑的数据，除非系统管理员用 **sp_forceonline_db** 和 **sp_forceonline_page** 过程使可疑的数据变得可访问。
- 一般的数据库损坏，例如数据库日志损坏或无法获得某些资源（这些资源不是页所特有的），将使整个数据库被标记为可疑（即使恢复故障隔离模式是 “**page**”）。
- 如果未指定 **page** 或 **database**，**Adaptive Server** 将显示当前设置和已配置的设置。当前设置是指上次在数据库中执行恢复时所用的设置。已配置的设置是指下次在数据库中执行恢复时所用的设置。

- 如果数据库在 **read_only** 模式下联机，用户将无法修改该数据库的任何数据，包括不受可疑页影响而联机的数据。然而，系统管理员可以通过使用 **sp_dboption** 系统过程将 **read only** 设置为 **false**，使数据库变得可写入。在这种情况下，用户能修改联机数据，但是仍无法访问可疑的数据。

权限

只有系统管理员才能执行 **sp_setsuspect_granularity** 来设置恢复故障隔离模式。任何用户都可以执行 **sp_setsuspect_granularity** 来显示设置。

使用的表

master.dbo.sysattributes、 master.dbo.sysdatabases

参见

命令	dump database、 dump transaction、 load database
系统过程	sp_forceonline_db、 sp_forceonline_page、 sp_listsuspect_db、 sp_listsuspect_page、 sp_setsuspect_threshold

sp_setsuspect_threshold

功能

在将整个数据库标记为可疑之前，显示或设置 Adaptive Server 允许的数据库中的最大可疑页数。

语法

```
sp_setsuspect_threshold [dbname [, threshold]]
```

参数

dbname — 是要显示或设置可疑升级阈值的数据库名。缺省值为当前数据库。

threshold — 在将整个数据库标记为可疑之前，指示恢复所允许的最大可疑数据页数。缺省值为 20 页。最小值为 0。

示例

1. **sp_setsuspect_threshold pubs2, 5**

将最大可疑页数设置为 5。可疑页数大于 5 时，恢复进程将整个数据库标记为可疑。

2. **sp_setsuspect_threshold pubs2**

显示 *pubs2* 数据库中可疑升级阈值的当前设置和已配置的设置。

3. **sp_setsuspect_threshold**

显示当前用户数据库中恢复故障隔离阈值的当前设置和已配置的设置。

注释

- 必须在 *master* 数据库中用 **sp_setsuspect_threshold** 设置可疑升级阈值。
- 如果未指定页数，Adaptive Server 将显示当前设置和已配置的设置。当前设置是指上次在数据库中执行恢复时所用的设置。已配置的设置是指下次在数据库中执行恢复时所用的设置。

权限

只有系统管理员才能执行 **sp_setsuspect_threshold** 来设置升级阈值。任何用户都可以执行 **sp_setsuspect_threshold** 来显示当前设置。

使用的表

master.dbo.sysattributes、 *master.dbo.sysdatabases*

参见

系统过程	sp_forceonline_db、 sp_forceonline_page、 sp_listsuspect_db、 sp_listsuspect_page、 sp_setsuspect_granularity
------	---

sp_showcontrolinfo

功能

显示有关引擎组赋值、绑定的客户端应用、登录和存储过程的信息。

语法

```
sp_showcontrolinfo [object_type, object_name,  
                    spid]
```

参数

object_type — AP、LG、PR、EG 或 PS 分别代表应用程序类型、登录类型、存储过程类型、引擎组类型或进程类型。如果未指定 *object_type* (或将 *object_type* 指定为 null)，sp_showcontrolinfo 将显示有关所有类型的信息。

object_name — 是应用程序、登录、存储过程或引擎组的名称。如果将 PS 指定为 *object_type*，则不要指定 *object_name*。如果未指定 *object_name* (或将 *object_name* 指定为 null)，sp_showcontrolinfo 将显示有关所有对象名的信息。

spid — 是 Adaptive Server 的进程 ID。只有将 PS 指定为 *object_type* 时，才指定 *spid*。如果未指定 *spid* (或将 *spid* 指定为 null)，sp_showcontrolinfo 将显示有关所有 *spid* 的信息。使用 sp_who 来查看 *spid*。

示例

1. sp_showcontrolinfo

显示所有用户指派的执行类到对象的绑定。

2. sp_showcontrolinfo 'AP', 'isql'

显示 isql 应用程序的执行类。

3. sp_showcontrolinfo 'PS'

显示指派到引擎组的所有进程的执行类。

4. sp_showcontrolinfo 'PS', null, 7

显示 *spid* 为 7 的进程的执行类。

注释

- 如果使用不带参数的 **sp_showcontrolinfo**，将显示有关所有用户指派的引擎组赋值、绑定的客户端应用、登录和存储过程的信息。如果使用带 **object_type** 参数的 **sp_showcontrolinfo**，将按对象类型分别提供有关绑定到执行类的应用程序、登录或存储过程，引擎组分和会话级特性绑定的信息。有关详细信息，参见 *Performance and Tuning Guide*。
- 除非 **object_type** 是 **PR**，否则都在 **master** 数据库中执行 **sp_showcontrolinfo**。如果 **object_type** 是 **PR**，则在过程驻留的数据库中执行 **sp_showcontrolinfo**。
- 如果 **object_type** 是 **null**，**sp_showcontrolinfo** 将显示匹配其它参数的对象的执行类信息。
- 如果 **object_name** 是 **null**，**sp_showcontrolinfo** 将显示所有应用程序、登录和存储过程的绑定信息。
- 如果 **spid** 是 **null**，**sp_showcontrolinfo** 将显示匹配其它参数的对象的执行类信息。

权限

任何用户都可以执行 **sp_showcontrolinfo**。

使用的表

sysattributes、*syslogins*

参见

系统过程	sp_addengine、sp_addexeclass、 sp_bindexeclass、sp_clearpsexec、 sp_dropengine、sp_dropexeclass、 sp_showexeclass、sp_showpsexec、 sp_unbindexeclass
实用程序	isql

sp_showexeclass

功能

显示执行类特性以及与指定执行类相关的任何引擎组中的引擎。

语法

```
sp_showexeclass [execclassname]
```

参数

execclassname — 是执行类的名称。

示例

1. sp_showexeclass

classname	priority	engine_group	engines
-----	-----	-----	-----
EC1	HIGH	ANYENGINE	ALL
EC2	MEDIUM	ANYENGINE	ALL
EC3	LOW	LASTONLINE	0

显示所有执行类的优先级和引擎组特性值。

2. sp_showexeclass 'EC1'

classname	priority	engine_group	engines
-----	-----	-----	-----
EC1	HIGH	ANYENGINE	ALL

显示执行类 *EC1* 的特性值。

注释

- **sp_showexeclass** 显示执行类特性以及与 *execclassname* 相关的任何引擎组中的引擎。有关详细信息，参见 *Performance and Tuning Guide*。
- 如果 *execclassname* 是 NULL 或空缺，**sp_showexeclass** 将显示所有执行类的优先级和引擎组特性值，包括系统定义类 *EC1*、*EC2* 和 *EC3* 的特性值。

权限

任何用户都可以执行 **sp_showexeclass**。

使用的表

sysattributes、 *sysengines*

参见

系统过程	sp_addengine、 sp_addexeclass、 sp_bindexeclass、 sp_clearpsex、 sp_dropengine、 sp_dropexeclass、 sp_showcontrolinfo、 sp_showpsex、 sp_unbindexeclass
------	---

sp_showplan

功能

为当前 SQL 语句或同一批处理中的上一语句的任何用户连接显示 **showplan** 的输出内容。

语法

```
sp_showplan spid, batch_id output, context_id  
output, stmt_num output
```

要显示当前 SQL 语句的 **showplan** 输出，而不指定 *batch_id*、*context_id* 或 *stmt_num*：

```
sp_showplan spid, null, null, null
```

参数

spid — 是任何用户连接的进程 ID。使用 **sp_who** 来查看 *spid*。

batch_id — 是批处理的唯一、非负的编号。

context_id — 是批处理中执行的每个过程（或触发器）的唯一编号。

stmt_num — 是批处理中当前语句的编号。*stmt_num* 必须是正数。

示例

```
1. declare @batch int  
   declare @context int  
   declare @statement int  
   exec sp_showplan 99, @batch output, @context  
   output, @statement output
```

显示 *spid* 值为 99 的用户会话中运行的当前语句的查询计划，并显示 *batch_id*、*context_id* 和 *statement_id* 参数的值。这些值可用于检索 *spid* 值为 99 的用户会话中，**sp_showplan** 后续迭代中的查询计划。

```
2. sp_showplan 99, null, null, null
```

显示在 *spid* 值为 99 的用户会话中运行的当前语句的 **showplan** 输出。

注释

- **sp_showplan** 显示当前执行的 SQL 语句或同一批处理中上一语句的 **showplan** 输出。
- 要查看同一批处理中上一语句的查询计划，可以用相同的参数值再次执行 **sp_showplan**，但语句号减 1。使用这种方法，可以向前查看语句批处理中的所有语句，直到查询号 1 为止。
- **sp_showplan** 可独立于 Adaptive Server Monitor™ Server 而运行。
- 如果 *context_id* 在 SQL 批处理中大于 0，当前语句将嵌入到从原始 SQL 批处理调用的存储过程（或触发器）中。选择具有相同 *spid* 值的 *sysprocesses* 行，来显示过程 ID 和语句 ID。

权限

只有系统管理员才能执行 **sp_showplan**。

使用的表

无。

参见

系统过程	sp_who
------	--------

sp_showpsexex

功能

显示在 Adaptive Server 上运行的所有客户端会话的执行类、当前优先级和父子关系。

语法

```
sp_showpsexex [spid]
```

参数

spid — 是要报告的 Adaptive Server 会话 ID。 *spid* 必须属于执行 `sp_showpsexex` 的应用程序或登录。使用 `sp_who` 来列出 *spid*。

示例

1. sp_showpsexex

spid	appl_name	login_name	exec_class	current_priority	task_affinity
1	isql	sa	EC1	HIGH	NONE
5		NULL	NULL	LOW	NULL
7	ctisql	sa	EC2	MEDIUM	NONE
8	ctisql	sa	EC2	MEDIUM	NONE

显示所有当前客户端会话的执行类、当前优先级和父子关系。

2. sp_showpsexex 5

显示 *spid* 为 5 的进程的应用程序名、登录名、当前优先级和引擎相关性。

注释

- `sp_showpsexex` 显示所有会话（带有 *spid* 的对象）的执行类、当前优先级和父子关系。有关详细信息，参见 *Performance and Tuning Guide*。
- 如果 *spid* 是 NULL 或缺， `sp_showpsexex` 将报告当前在 Adaptive Server 上运行的所有会话的信息。
- `sp_showpsexex` 不报告有关下列系统进程的信息：死锁、检查点、网络、审计和镜像处理程序。它显示管家 *spid* 的信息。

权限

任何用户都可以执行 `sp_showpsexex`。

使用的表

syslogins、 *sysprocesses*

参见

系统过程	sp_addengine、 sp_addexclass、 sp_bindexclass、 sp_clearpsex、 sp_dropengine、 sp_dropexclass、 sp_showcontrolinfo、 sp_showexclass、 sp_unbindexclass
------	--

sp_spaceused

功能

显示估计的由当前数据库中一个指定表或所有表使用的行数、数据页数、索引大小以及空间大小。

语法

```
sp_spaceused [objname [,1] ]
```

参数

objname — 是要报告其信息的表的名称。如果省略该参数，将显示当前数据库中已使用的空间的摘要。

1 — 分别输出有关表的索引和 *text/image* 存储的信息。

示例

1. sp_spaceused titles

name	rowtotal	reserved	data	index_size	unused
titles	18	46 KB	6 KB	4 KB	36 KB

报告分配（保留）给 *titles* 表的空间大小、数据占用的空间大小、索引占用的空间大小以及可用（未使用的）空间的大小。

2. sp_spaceused titles, 1

index_name	size	reserved	unused
titleidind	2 KB	32 KB	24 KB
titleind	2 KB	16 KB	14 KB

name	rowtotal	reserved	data	index_size	unused
titles	18	46 KB	6 KB	4 KB	36 KB

除了输出 *titles* 表的信息外，还输出表中每个索引的信息。

3. sp_spaceused blurbs,1

index_name	size	reserved	unused
blurbs	0 KB	14 KB	12 KB
tblurbs	14 KB	16 KB	2 KB

name	rowtotal	reserved	data	index_size	unused
blurbs	6	30 KB	2 KB	14 KB	14 KB

显示 *text/image* 页存储占用的空间大小，该空间与表使用的空间是相互独立的。*text/image* 存储的对象名为 “t” 加上表的名称。

4. sp_spaceused

database_name		database_size	

master		5 MB	
reserved	data	index_size	unused
-----	-----	-----	-----
2176 KB	1374 KB	72 KB	730 KB

输出当前数据库中已使用的空间的摘要。

5. sp_spaceused syslogs

name	rowtotal	reserved	data	index_size	unused

syslogs	Not avail.	32 KB	32 KB	0 KB	0 KB

报告保留的空间大小和可供事务日志使用的空间大小。

注释

- **sp_spaceused** 显示估计的由当前数据库中一个指定表或所有表使用的数据页数和空间大小，以及表的行数。**sp_spaceused** 使用 **rowcnt** 内部函数计算 **rowtotal** 值。该函数根据给对象分配的页数来求得每个数据页的平均行数。这个方法很快捷，但是结果是个估计数，而且更新和插入活动都会使实际值发生变化。由于 **update statistics** 命令、**dbcc checktable** 和 **dbcc checkdb** 会更新每页行数的估计值，所以执行了其中一个命令后，**rowtotal** 的值最准确。如果需要精确的行数，请使用 **select count(*)**。
- **sp_spaceused** 报告受表、集群索引和非集群索引影响的空间大小。
- **sp_spaceused** 报告的分配（保留）空间量是数据空间、索引大小空间和可用（未使用的）空间的总计。
- **text** 和 **image** 列使用的空间作为单独的数据库对象存储，在 **index_size** 列中单独报告并包含在表的摘要行中。**index_size** 列中 **text/image** 存储的对象名为 “t” 加上表的名称。
- 用于 **syslogs** 时，**sp_spaceused** 的 **rowtotal** 报告结果是 “not available”。参见示例 5。

权限

任何用户都可以执行 **sp_spaceused**。

使用的表

master.dbo.spt_values、*master.dbo.sysusages*、*sysindexes*、*sysobjects*

参见

分类存储过程	sp_statistics
命令	create index、create table、drop index、drop table
系统过程	sp_help、sp_helpindex

sp_syntax

功能

根据服务器上现有的产品和相应的 **sp_syntax** 脚本，显示 Transact-SQL 语句、系统过程、实用程序和 Adaptive Server 其它例程的语法。

语法

```
sp_syntax word [, mod][, language]
```

参数

word — 是命令或例程的名称或部分名称；例如，“help”可列出提供帮助的所有系统过程。含有空格或 Transact-SQL 保留字时，请用引号将字引起来。

mod — 是一种模块的名称或部分名称，例如 “Transact-SQL” 或 “Utility”。每个 **sp_syntax** 安装脚本添加不同的模块。使用不带任何参数的 **sp_syntax** 可查看服务器上现有的模块。

language — 是要检索的语法说明的语言。**language** 必须是 **syslanguages** 表中的有效语言名。

示例

1. sp_syntax

```
sp_syntax provides syntax help for Sybase products.  
These modules are installed on this Server:
```

```
Module  
-----  
OpenVMS  
Transact-SQL  
UNIX Utility  
System Procedure
```

```
Usage: sp_syntax command [, module [, language]]
```

显示服务器上所有可用的 **sp_syntax** 模块。

2. sp_syntax "disk"

显示包含 “disk” 的所有例程的语法和功能说明。由于 “disk” 是 Transact-SQL 保留字，因此用引号将其引起来。

注释

- **sp_syntax** 的文本在数据库 *sybsyntax* 中。将 **sp_syntax** 和 *sybsyntax* 数据库装载到 **Adaptive Server** 上，这需要用到 **configuration documentation for your platform** 中所述的安装脚本。如果不能访问 **sp_syntax**，请向系统管理员询问有关在服务器上安装它的信息。
- 可在要搜索的命令名中使用通配符。但是，如果查找含有文字“_”的命令或函数，可能会得到意外的结果，因为下划线通配符代表任意单个字符。

权限

任何用户都可以执行 **sp_syntax**。

使用的表

sybsyntax..sybsyntax

参见

系统过程	sp_help、sp_helpdb
------	-------------------

sp_sysmon

功能

显示性能信息。

语法

```
sp_sysmon begin_sample
sp_sysmon { end_sample | interval }
           [, section [, applmon] ]
sp_sysmon { end_sample | interval } [, applmon ]
```

参数

- begin_sample** — 开始采样。指定 **begin_sample** 时不能指定某一部分。
- end_sample** — 结束采样并输出报告。
- interval** — 指定采样的时间间隔。必须使用 HH:MM:SS 格式，例如 “00:20:00”。
- section** — 是由 **sp_sysmon** 输出的某个部分的缩写。表 7-23 列出了报告各部分的值和相应名称。

表 7-23: sp_sysmon 报告各个部分

报告各个部分	参数
应用程序管理	apppgmt
数据缓存管理	dcache
磁盘 I/O 管理	diskio
ESP 管理	esp
索引管理	indexmgmt
内核使用情况	kernel
锁管理	locks
内存管理	memory
元数据缓存管理	mdcache
对执行中 SQL 的监控访问	monaccess
网络 I/O 管理	netio

表 7-23: sp_sysmon 报告的不同部分 (续)

报告的不同部分	参数
并行查询管理	parallel
过程缓存管理	pcache
恢复管理	recovery
任务管理	taskmgmt
事务管理	xactmgmt
事务配置文件	xactsum
工作进程管理	wpm

applmon — 指定是输出应用程序细节、输出应用程序和登录细节还是
不输出应用程序细节。缺省值是省略应用程序细节。表 7-24 中列
出了有效值。

表 7-24: sp_sysmon 中 applmon 参数的值

参数	报告的信息
appl_only	按应用程序名报告 CPU、I/O、优先级更改和违反 资源限制的情况。
appl_and_login	按应用程序名和登录名报告 CPU、I/O、优先级更 改和违反资源限制的情况。
no_appl	跳过按应用程序名或按登录名报告的部分。这是缺 省值。

只有输出完整报告且将 *section* 指定为 *apppmgmt* 时，该参数才有效。

示例

- 1. `sp_sysmon "00:10:00"`
10 分钟后输出监控信息。
- 2. `sp_sysmon "00:05:00", diskio`
5 分钟后只输出 `sp_sysmon` 报告的“磁盘管理”部分。

```
3. sp_sysmon begin_sample
go
execute proc1
go
execute proc2
go
select sum(total_sales) from titles
go
sp_sysmon end_sample, dcache
go
```

开始采样、执行过程和查询、结束采样并仅输出报告的“数据缓存”部分。

```
4. sp_sysmon "00:05:00", @applmon = appl_and_login
```

输出完整报告，且对每个登录都包括应用程序和登录细节。

注释

- **sp_sysmon** 显示有关 Adaptive Server 性能的信息。它将内部计数器设置为 0，然后在服务器上的活动导致计数器递增时按指定间隔等待。间隔结束时，**sp_sysmon** 输出计数器上的值。有关详细信息，参见 *Performance and Tuning Guide*。
- 要只输出报告的某个部分，可对第二个参数使用表 7-24 中列出的值。
- 如果与 **begin_sample** 和 **end_sample** 一起使用批处理模式下的 **sp_sysmon**，则两次执行之间的时间间隔必须至少为 1 秒钟。可使用 **waitfor delay "00:00:01"** 延长批处理的执行时间。
- 采样间隔过程中，以带符号的整数值保存结果。尤其是在有很多 CPU 和大量活动的系统上，这些计数器可能会溢出。如果 **sp_sysmon** 输出的结果中有负数，请缩短采样时间。

权限

只有系统管理员才能执行 **sp_sysmon**。

使用的表

master.dbo.sysconfigures、*master.dbo.syscurconfigs*、
master.dbo.sysdevices、*master.dbo.ssysmonitors*

sp_thresholdaction

功能

将在日志段的可用页数小于最后机会阈值时自动执行（除非该阈值与其它过程相关联）。Sybase 不提供此过程。

语法

达到阈值时，Adaptive Server 按位置将下列参数传递给阈值过程：

```
sp_thresholdaction @dbname,  
                  @segment_name,  
                  @space_left,  
                  @status
```

参数

@dbname — 是达到阈值的数据库的名称。

@segment_name — 是达到阈值的段的名称。

@space_left — 是阈值大小（以 2K 的页表示）。

@status — 对于最后机会阈值是 1；对于所有其它阈值是 0。

示例

```
1. create procedure sp_thresholdaction  
    @dbname varchar(30),  
    @segmentname varchar(30),  
    @space_left int,  
    @status int  
as  
    dump transaction @dbname to tapedump1
```

为最后机会阈值创建阈值过程，该最后机会阈值将事务日志转储到磁带设备。

注释

- **sp_thresholdaction** 必须由数据库所有者（用户数据库中）、或系统管理员（**sybssystemprocs** 数据库中）、或具有 **create procedure** 权限的用户创建。
- 可为数据库中的任何段添加阈值并创建阈值过程。

- 达到最后机会阈值时，Adaptive Server 在阈值事件发生的数据库中搜索 **sp_thresholdaction** 过程。如果那个数据库中不存在该过程，Adaptive Server 将在 **sybsystemprocs** 中搜索。如果 **sybsystemprocs** 中也不存在该过程，将搜索 **master**。如果 Adaptive Server 找不到该过程，它将向错误日志发送错误消息。
- **sp_thresholdaction** 应包含 **dump transaction** 命令以截断事务日志。
- 根据设计，最后机会阈值留出足够的可用空间来记录 **dump transaction** 命令。这样就可能没有足够的空间记录数据库的其它用户事务。只能执行不在事务日志中记录的命令（**select**、快速 **bcp**、**readtext** 和 **writetext**）以及释放额外日志空间（**dump transaction**、**dump database** 和 **alter database**）可能需要的命令。缺省情况下，将挂起其它命令并向错误日志发送消息。要中止这些命令而不是挂起它们，可以使用带 **abort tran on log full** 选项的 **sp_dboption**，紧接着使用 **checkpoint** 命令。

唤醒挂起的进程

- 一旦 **dump transaction** 命令释放了足够的日志空间，将自动唤醒并完成挂起的进程。
- 如果自上次备份后，快速 **bcp**、**writetext** 或 **select into** 导致对数据库的某些更改且这些更改是不记日志的，最后机会阈值过程将不能执行 **dump transaction** 命令。在这种情况下，请使用 **dump database** 复制数据库，然后使用 **dump transaction** 截断事务日志。
- 如果这不能释放足够的空间来唤醒挂起的进程，则可能增加事务日志的大小。使用 **alter database** 命令的 **log on** 选项来分配额外的日志空间。
- 最后一种解决方法是，系统管理员可使用 **sp_who** 决定挂起哪些进程，然后使用 **kill** 命令注销它们。

参见

命令	create procedure 、 dump transaction
系统过程	sp_addthreshold 、 sp_dboption 、 sp_droptreshold 、 sp_helpsegment 、 sp_helpthreshold 、 sp_modifythreshold

sp_transactions

功能

报告有关活动事务的信息。

语法

```
sp_transactions ["xid", xid_value] |
  ["state", {"heuristic_commit" | "heuristic_abort" |
    "prepared" | "indoubt"} [, "xactname"]] |
  ["gtrid", gtrid_value]
```

参数

xid_value — 是 *master.dbo.systransactions* 的 *xactname* 列中的事务名。
gtrid_value — 是由 Adaptive Server 协调的事务的全局事务 ID 名。

示例

1. sp_transactions

xactkey	type		coordinator	starttime
state	connection	dbid	spid	loid
failover	srvname			namelen
xactname				

0x00000b1700040000dd6821390001	Local		None	Jun 1 1999 3:47PM
Begun	Attached	1	1	2
Resident Tx	NULL			17
\$user_transaction				
0x00000b1700040000dd6821390001	Remote		ASTC	Jun 1 1999 3:47PM
Begun	NA	0	8	0
Resident Tx	caserv2			108
00000b1700040000dd6821390001-aa01f04ebb9a-00000b1700040000dd6821390001-a				
a01f04ebb9a-caserv1-caserv1-0002				

显示有关所有活动事务的一般信息。


```
-----
-----
0x00000b1700040000dd6821390001 Local      None      Jun 1 1999 3:47PM
Begun      Attached      1      1      2
Resident Tx      NULL      17
$user_transaction

caserv1
caserv1
```

显示具有指定全局事务 ID 的事务的状态信息。

注释

- **sp_transactions** 转换 *systransactions* 表中的数据来显示有关活动事务的信息。 *systransactions* 本身包含 *syscoordinations* 表中的数据，还包含有关活动事务的内存信息。
- 不带关键字的 **sp_transactions** 显示有关所有活动事务的信息。
- 带有 **xid** 关键字的 **sp_transactions** 只显示指定事务的 *gtrid*、*commit_node* 和 *parent_node* 列。
- 带有 **state** 关键字的 **sp_transactions** 只显示处于指定状态的活动事务的信息。
同时带有 **xid** 和 **xactname** 的 **sp_transactions** 只显示处于指定状态的事务的事务名。
- 带有 **gtrid** 关键字的 **sp_transactions** 只显示具有指定全局事务 ID 的事务的信息。
- **sp_transactions** 替代了 XA-Library 和 XA-Server 产品提供的 **sp_xa_scan_xact** 过程。
- 有关详细信息，参见 *Using Adaptive Server Distributed Transaction Management Features*。

sp_transactions 输出的列说明

- **xactkey** 列显示内部事务关键字， Adaptive Server 使用该关键字唯一地标识事务。
- **type** 列指明事务类型。
 - “Local” 表示用 **begin transaction** 语句在本地 Adaptive Server 上显式启动事务。
 - “Remote” 表示事务是在远程 Adaptive Server 上执行的。
 - “External” 表示事务具有相关的外部协调器。例如，由远程 Adaptive Server、MSDTC 或 X/Open XA 事务管理器协调的事务被标记为 “External”。

- “Dtx_State” 是 Adaptive Server 协调的分布式事务的特殊状态。它表示本地服务器上的事务要么被提交要么被中止，但是 Adaptive Server 已不能在远程参与者上解决该事务的分支。当 Adaptive Server 与其协调的服务器失去联系时，可能会出现这种情况。
- *coordinator* 列说明用来管理分布式事务的方法或协议：

sp_transactions 的 “coordinator” 值	含义
无	事务不是分布式事务且不需要协调协议。
ASTC	使用 Adaptive Server 事务协调服务来协调事务。
XA	符合 X/Open XA 的事务管理器通过 Adaptive Server XA-Library 接口协调事务。这些事务管理器包括 Encina、CICS 和 Tuxedo。
DTC	事务由 MSDTC 协调。
SYB2PC	使用 Sybase 两阶段提交协议协调事务。

- *starttime* 列指明事务启动的时间。
- *state* 列指明 sp_transactions 运行时事务的状态：

sp_transactions 的 “state” 值	含义
Begun	事务已开始但是未执行任何更新。
Done Command	事务已完成更新命令。
Done	X/Open XA 事务已修改完数据。
Prepared	事务已成功就绪。
In Command	事务正在修改数据。
In Abort Cmd	事务中当前命令的执行已中止。
Committed	事务已被成功提交，并且已写入提交日志记录。
In Post Commit	事务已被成功提交，但当前正在释放事务资源。
In Abort Tran	事务被中止。显式命令或系统故障都会导致事务的中止。
In Abort Savept	事务被回退到保存点。

sp_transactions 的 “state” 值	含义
Begun-Detached	事务已经开始，但是当前没有线程附加在上面。
Done Cmd-Detached	事务已经修改完数据，且当前没有线程附加在上面。
Done-Detached	事务不再修改数据，且当前没有线程附加在上面。
Prepared-Detached	事务已成功就绪，且当前没有线程附加在上面。
Heur Committed	已使用 dbcc complete_xact 命令尝试提交事务。
Heur Rolledback	已使用 dbcc complete_xact 命令尝试回退事务。

- **connection** 列指明事务当前是否与线程相关：
 - “Attached” 表示事务有相关的控制线程。
 - “Detached” 表示当前没有与事务相关的线程。某些外部事务管理器（如 CICS 和 TUXEDO）可以使用 X/Open XA 的 “suspend” 和 “join” 语义将不同的线程与同一事务相关联。
- **dbid** 列指明从中启动事务的数据库的 ID。
- **spid** 列指明与事务相关的服务器进程 ID。如果事务是 “Detached”，则该值是 0。
- **loid** 列指明 *master.dbo.systransactions* 的唯一锁所有者 ID。
- **failover** 列指明事务的故障替换状态：
 - “Resident Tx” 表示已启动事务并在同一服务器上执行事务。在正常操作条件下，不使用 Adaptive Server 高可用性功能的系统上将显示 “Resident Tx”。
 - 对辅助协同服务器进行故障替换后，将显示 “Failed-over Tx”。“Failed-over Tx” 表示事务原来在主服务器上启动并达到就绪状态，但被自动迁移到辅助协同服务器上（例如，因为主服务器发生系统故障）。就绪事务的迁移对于外部协调服务来说是透明的。
 - “Tx by Failover-Conn” 表示试图在指定服务器上启动事务，但事务却在辅助协同服务器上启动了。这种情况出现在原始服务器进行故障替换的时候。
- **srvname** 列指明执行事务的远程服务器的名称。该列只对远程事务有意义。对于本地事务和外部事务，**srvname** 为空。

- *namelen* 列指明 *xactname* 值的总长度。
- *xactname* 是事务的名称。对于本地事务，事务名可作为 **begin transaction** 命令的一部分来定义。外部事务管理器提供多种格式的唯一的事务名。例如，符合 X/Open XA 的事务管理器提供的事务 ID (*xid*) 包含全局事务标识符和分支限定符，二者都存储在 *xactname* 中。
- 对于 Adaptive Server 协调的事务，*gtrid* 列将显示全局事务 ID。而事务分支（同一分布式事务的一部分）共享相同的 *gtrid*。可以将特定的 *gtrid* 与 **sp_transactions** *gtrid* 关键字一起使用，以确定同一分布式事务中其它事务分支的状态。

sp_transactions 不能显示具有外部协调器的事务的 *gtrid*。对于由符合 X/Open XA 的事务管理器、MSDTC 或 SYB2PC 来协调的事务，*gtrid* 列将显示外部协调器提供的完整事务名称。
- 对于由 Adaptive Server 协调的事务，*commit_node* 列将指明执行分布式事务最外面的块的服务器。这个最外面的块将最终确定所有下级事务的提交状态。

对于不是由 Adaptive Server 协调的事务，*commit_node* 将显示表 7-25 中所列的一个值。

表 7-25: *commit_node* 和 *parent_node* 的值

值	含义
<i>server_name</i>	提交节点或父节点是具有指定 <i>server_name</i> 的 Adaptive Server。
XATM	提交节点或父节点是符合 X/Open XA 的事务管理器。
MSDTCTM	提交节点或父节点是 MSDTC。
SYB2PCTM	使用 SYB2PC 协议来协调事务。

- 对于由 Adaptive Server 协调的事务，*parent_node* 列指明协调本地服务器上外部事务的服务器。

对于不是由 Adaptive Server 协调的事务，*parent_node* 将显示表 7-25 中所列的一个值。

► 注意

commit_node 和 *parent_node* 的值可以不同，这取决于分布式事务的层次级别。

权限

任何用户都可以执行 **sp_transactions**。

使用的表

spt_values、*systransactions*

参见

系统过程	sp_who、sp_lock
------	----------------

sp_unbindcache

功能

解除数据库、表、索引、 *text* 对象或 *image* 对象与数据缓存之间的绑定。

语法

```
sp_unbindcache dbname [, [owner.]tablename  
[, indexname | "text only"]]
```

参数

dbname — 是要解除绑定的数据库名，或是包含要解除绑定的对象的数据库名。

owner — 是表所有者的名称。如果表属于数据库所有者，可以不指定所有者名称。

tablename — 是要与缓存解除绑定的表名，或是其索引、 *text* 对象或 *image* 对象要与缓存解除绑定的表名。

indexname — 是要与缓存解除绑定的索引的名称。

text only — 解除 *text* 或 *image* 对象与缓存之间的绑定。

示例

1. **sp_unbindcache pubs2, titles**

解除 *titles* 表与其绑定缓存之间的绑定。

2. **sp_unbindcache pubs2, titles, titleidind**

解除 *titleidind* 索引与其绑定缓存之间的绑定。

3. **sp_unbindcache pubs2, au_pix, text**

解除 *au_pix* 表的 *text* 或 *image* 对象与其绑定缓存之间的绑定。

4. **sp_unbindcache pubs2, syslogs**

解除事务日志 *syslogs* 与其缓存之间的绑定。

注释

- 解除数据库或数据库对象与缓存之间的绑定后，缓存的所有后续 I/O 将在缺省数据缓存中执行。缓存中正在解除绑定的所有脏页将被写入磁盘，所有净页将从缓存中清除。有关详细信息，参见 *Performance and Tuning Guide*。
- 解除与缓存之间的绑定将立即生效，不需要重新启动服务器。
- 删除数据库、表或索引时，将自动删除其缓存绑定。
- 要解除数据库的绑定，必须使用 *master* 数据库。对于表、索引、*text* 对象或 *image* 对象，必须使用存储这些对象的数据库。
- 要解除数据库中所有系统表的绑定，必须正在使用该数据库，而且该数据库必须处于单用户模式。使用以下命令：
`sp_dboption db_name, "single user", true`
有关详细信息，参见 `sp_dboption`。
- 以下过程提供其相应对象的绑定信息：`sp_helpdb` 用于数据库，`sp_help` 用于表，`sp_helpindex` 用于索引。
- `sp_helpcache` 将显示绑定到缓存的对象名。
- 解除表或其索引与缓存之间的绑定时，`sp_unbindcache` 需要获取一个排它表锁。解除绑定时，不能读取任何页。如果用户在表上持有锁，那么对该对象发出 `sp_unbindcache` 后，`sp_unbindcache` 任务将休眠，直到这些锁被释放。
- 使用 `sp_bindcache` 或 `sp_unbindcache` 更改对象的缓存绑定时，引用该对象的存储过程将在下一次执行时被重新编译。更改数据库的绑定时，引用该数据库中对象的存储过程将在下一次执行时被重新编译。
- 要解除所有对象与缓存之间的绑定，使用系统过程 `sp_unbindcache_all`。

权限

只有系统管理员才可以执行 `sp_unbindcache`。

使用的表

master..sysattributes、*master..sysdatabases*、*sysindexes*、*sysobjects*

参见

系统过程	<code>sp_bindcache</code> 、 <code>sp_dboption</code> 、 <code>sp_helpcache</code> 、 <code>sp_unbindcache_all</code>
------	--

sp_unbindcache_all

功能

解除所有对象与其缓存之间的绑定。

语法

sp_unbindcache_all *cache_name*

参数

cache_name — 是数据缓存的名称，将解除对象与该缓存之间的绑定。

示例

1. sp_unbindcache_all pub_cache

解除所有数据库、表、索引、 *text* 对象和 *image* 对象与 *pub_cache* 之间的绑定。

注释

- 解除实体与缓存之间的绑定后，该缓存的所有后续 I/O 将在缺省缓存中执行。
- 要解除单个对象与缓存之间的绑定，使用系统过程 **sp_unbindcache**。
- 有关解除缓存绑定的详细信息，参见 **sp_unbindcache**。

权限

只有系统管理员才可以执行 **sp_unbindcache_all**。

使用的表

master..sysattributes、 *master..sysdatabases*、 *sysindexes*、 *sysobjects*

参见

系统过程	sp_bindcache、 sp_helpcache、 sp_unbindcache
------	--

sp_unbindefault

功能

解除创建的缺省值与列或用户定义数据类型之间的绑定。

语法

```
sp_unbindefault objname [, futureonly]
```

参数

objname — 是要解除缺省值绑定的表和列或用户定义数据类型的名称。如果该参数的格式不是 “*Table.column*”，则会将 *objname* 视为一个用户定义的数据类型。解除用户定义数据类型与其缺省值之间的绑定时，对于任何属于该类型的列，如果其缺省值与该用户定义数据类型相同，那么也将解除这些列与其缺省值之间的绑定；如果已经更改了其缺省值，这些列将不受影响。

futureonly — 防止属于指定用户定义数据类型的现有列失去其缺省值。解除列与其缺省值之间的绑定时将忽略该参数。

示例

1. **sp_unbindefault "employees.startdate"**

解除 *employees* 表的 *startdate* 列与其缺省值之间的绑定。

2. **sp_unbindefault ssn**

解除名为 *ssn* 的用户定义数据类型和属于该类型的所有列与其缺省值之间的绑定。

3. **sp_unbindefault ssn, futureonly**

解除用户定义数据类型 *ssn* 与其缺省值之间的绑定，但不影响属于该类型的现有列。

注释

- 使用 **sp_unbindefault** 删除用 **sp_bindefault** 创建的缺省值。使用 **alter table** 删除用 **create table** 或 **alter table** 语句声明的缺省值。
- 对于属于用户定义数据类型的列，除非已更改了其缺省值或第二个可选参数的值为 **futureonly**，否则这些列将失去其当前缺省值。
- 要显示缺省值的文本，执行 **sp_helptext**，并将该缺省值的名称作为参数。

权限

只有对象所有者才可以执行 **sp_unbindefault**。

使用的表

syscolumns、 *sysobjects*、 *sysprocedures*、 *systypes*

参见

命令	create default、 drop default
系统过程	sp_bindefault、 sp_helptext

sp_unbindexclass

功能

删除以前与指定客户端应用、登录或存储过程相关的执行类特性。

语法

```
sp_unbindexclass object_name, object_type, scope
```

参数

object_name — 是要删除与执行类之间的关联的应用程序、登录或存储过程的名称。

object_type — 将应用程序、登录或存储过程的 **object_name** 类型标识为 **ap**、**lg** 或 **pr**。

scope — 是将应用解除绑定的应用程序或登录的名称。它是存储过程的存储过程所有者名（用户名）。

示例

1. sp_unbindexclass 'sa', 'lg', 'isql'

删除属于应用程序 **isql** 的“**sa**”登录与执行类之间的关联。“**sa**”会自动绑定到另一个执行类，这取决于其它绑定规范、过程和范围规则。如果不适用其它绑定，对象将绑定到缺省执行类，**EC2**。

注释

- 参数必须与 **sysattributes** 系统表中的现有条目相匹配。
- 如果将范围指定为空值，**Adaptive Server** 将其范围为空的对象解除绑定（如果有的话）。
- 范围为空值不表示解除绑定将应用于所有的绑定对象。
- 解除执行类与存储过程之间的绑定时，必须使用存储过程所有者的名称（用户名）作为 **scope** 参数。
- 解除绑定前后都可以删除存储过程。
- 如果用户拥有绑定到数据库中某个执行类的存储过程，就不能从该数据库删除该用户。
- 删除数据库中类型为 **PR** 的对象之前，应先将其解除绑定。
- 如果相关引擎组没有联机引擎，且活动进程绑定到相关执行类，那么解除绑定将失败。

- 由于过程规则和范围规则，被解除绑定的执行类对称为 *object_name* 的对象可能有效或无效。对象会自动绑定到另一个执行类，这取决于其它绑定规范、过程和范围规则。如果不适用其它绑定，对象将绑定到缺省执行类， *EC2*。

权限

只有系统管理员才可以执行 **sp_unbindexclass**。

使用的表

sysattributes、 *syslogins*

参见

系统过程	sp_addengine、 sp_addexclass、 sp_bindexclass、 sp_clearpsex、 sp_dropengine、 sp_dropexclass、 sp_showcontrolinfo、 sp_showexclass、 sp_showpsex
实用程序	isql

sp_unbindmsg

功能

解除约束与用户定义的消息之间的绑定。

语法

```
sp_unbindmsg constrname
```

参数

constrname — 是要解除消息绑定的约束的名称。

示例

```
1. sp_unbindmsg positive_balance
```

解除约束 *positive_balance* 与用户定义的消息之间的绑定。

注释

- 一个约束只能绑定一条消息。要更改绑定到约束的消息，使用 **sp_bindmsg**；新的消息号将替换任何现有的绑定消息。不必要事先使用 **sp_unbindmsg**。
- 要从 *sysusermessages* 表检索消息文本，执行 **sp_getmessage**。

权限

只有对象所有者才可以执行 **sp_unbindmsg**。

使用的表

sysconstraints、*sysobjects*

参见

系统过程	sp_addmessage、sp_bindmsg、sp_getmessage
------	--

sp_unbindrule

功能

从列或用户定义的数据类型将规则解除绑定。

语法

```
sp_unbindrule objname [, futureonly]
```

参数

objname — 是要解除规则绑定的表和列或用户定义数据类型的名称。如果该参数的格式不是 “*Table.column*”，则会将 *objname* 视为一个用户定义的数据类型。解除用户定义的数据类型与规则之间的绑定时，也将解除属于该类型的列与该规则之间的绑定。已绑定不同规则的列将不受影响。

futureonly — 防止属于指定用户定义数据类型的列失去其规则。解除列与规则之间的绑定时将忽略该参数。

示例

1. **sp_unbindrule "employees.startdate"**

解除 *employees* 表的 *startdate* 列与其规则之间的绑定。

2. **sp_unbindrule def_ssn**

解除名为 *def_ssn* 的用户定义数据类型和属于该类型的所有列与其规则之间的绑定。

3. **sp_unbindrule ssn, futureonly**

用户定义数据类型 *ssn* 不再有规则，但是现有 *ssn* 列不受影响。

注释

- 执行 **sp_unbindrule** 可将规则从当前数据库的列或用户定义数据类型中删除。如果不想解除现有 *objname* 列与其规则之间的绑定，应使用 *futureonly* 作为第二个参数。
- 不能使用 **sp_unbindrule** 解除绑定的检查约束。应使用 **alter table** 删除该约束。
- 要解除表中的列与其规则之间的绑定，请以 “*table.column*” 格式指定 *objname* 参数。
- 对于所有属于该用户定义数据类型的现有列，除非已对其规则进行了更改或第二个可选参数的值为 **futureonly**，否则将解除这些列与其规则之间的绑定。

- 要显示规则的文本，执行 **sp_helptext**，并将该规则的名称作为参数。

权限

只有对象所有者才可以执行 **sp_unbindrule**。

使用的表

syscolumns、*sysconstraints*、*sysobjects*、*sysprocedures*、*systypes*

参见

命令	create rule、drop rule
系统过程	sp_bindrule、sp_helptext

sp_volchanged

功能

通知 Backup Server 操作员已在转储或装载期间执行了请求的卷处理。

语法

```
sp_volchanged session_id, devname, action  
[, fname [, vname]]
```

参数

session_id — 标识请求卷更改的 Backup Server。使用 Backup Server 的卷更改请求中指定的 **@session_id** 参数。

devname — 是装入新卷的设备。使用 Backup Server 的卷更改请求中指定的 **@devname** 参数。如果 Backup Server 与 Adaptive Server 安装在不同的计算机上，请使用以下格式：

```
device at backup_server_name
```

action — 表示 Backup Server 是否应该 **abort**、**proceed** 或 **retry** 转储或装载。

fname — 是要装载的文件。如果不用 **sp_volchanged** 指定文件名，Backup Server 将装载 **load** 命令中的 **file = filename** 参数。如果 **sp_volchanged** 和装载命令都没有指定要装载的文件，Backup Server 装载磁带上的第一个文件。

vname — 是出现在 ANSI 磁带标签上的卷名。Backup Server 在覆盖现有转储、转储到崭新的磁带中或转储到一个内容不可识别的磁带中时，将卷名写入 ANSI 磁带标签中。如果未用 **sp_volchanged** 指定 **vname**，Backup Server 将使用转储命令中指定的 **dumpvolume** 值。如果 **sp_volchanged** 或转储命令都未指定卷名，Backup Server 将保留 ANSI 磁带标签空白的名称域。

装载时，Backup Server 使用 **vname** 来确认是否已经装入了正确的磁带。如果未用 **sp_volchanged** 指定 **vname**，Backup Server 将使用装载命令中指定的 **dumpvolume**。如果 **sp_volchanged** 或装载命令都未指定卷名，Backup Server 在装载转储之前，不检查 ANSI 磁带标签的名称域。

示例**1. sp_volchanged 8, "/dev/nrmt4", RETRY**

下列 Backup Server 的消息表示尚未到达所装入磁带的到期日：

```
Backup Server: 4.49.1.1: OPERATOR: Volume to be overwritten on
'/dev/rmt4' has not expired: creation date on this volume is
Sunday, Nov. 15, 1992, expiration date is Wednesday, Nov. 25,
1992.
```

```
Backup Server: 4.78.1.1: EXECUTE sp_volchanged
@session_id = 8,
@devname = '/auto/remote/pubs3/SERV/Masters/testdump',
@action = { 'PROCEED' | 'RETRY' | 'ABORT' }
```

操作员更改磁带，然后发出示例 1 中的命令。

注释

- 如果 Backup Server 检测到当前装入的卷有问题，它会请求卷更改：
 - 在 OpenVMS 系统上，Backup Server 将卷更改消息发送到其运行机器上的操作员终端。使用转储或装载命令的 **with notify = client** 选项将其它 Backup Server 消息传送到启动 dump 或 load 请求的终端会话。
 - 在 UNIX 系统上，Backup Server 将消息发送到启动转储或装载请求的客户端。使用转储或装载命令的 **with notify = operator_console** 选项将消息传送到启动 Backup Server 的终端。
 - 装入另一个卷后，操作员执行任何 Adaptive Server（可与执行转储或装载的 Backup Server 通信）的 sp_volchanged。操作员不必登录到开始转储或装载的 Adaptive Server。
- 在 OpenVMS 系统上，操作系统（而不是 Backup Server）在检测到卷尾或指定的驱动器脱机时，请求卷更改。操作员使用 OpenVMS 的 REPLY 命令答复这些消息。
- 在 UNIX 系统中，Backup Server 在磁带容量满后请求卷更改。操作员装入另一个磁带并执行 sp_volchanged。表 7-26 说明了这一进程。

表 7-26：在 UNIX 系统中更改磁带的卷

序列	操作员，使用 <i>isql</i>	Adaptive Server	Backup Server
1	发出 dump database 命令		
2		向 Backup Server 发出转储请求	
3			接收到来自 Adaptive Server 的转储请求消息 将装入磁带的消息发送给操作员 等待操作员的答复
4	接收到来自 Backup Server 的卷更改请求 装入磁带 执行 sp_volchanged		
5			检查磁带 如果磁带没有问题，则开始转储 磁带满后，向操作员发出卷更改请求
6	接收到来自 Backup Server 的卷更改请求 装入磁带 执行 sp_volchanged		
7			继续转储 转储操作完成后，将消息发送给操作员和 Adaptive Server
8	接收到转储已完成的消息 删除磁带并给磁带标号	接收到转储已完成的消息 释放锁 完成 dump database 命令	

权限

任何用户都可以执行 **sp_volchanged**。

使用的表

master..sysdevices、 *sysobjects*

参见

命令	dump database、 dump transaction、 load database、 load transaction
实用程序	isql

sp_who

功能

报告所有当前 Adaptive Server 用户和进程的信息，或有关特定用户和进程的信息。

语法

```
sp_who [loginname | "spid"]
```

参数

loginname — 是要报告其相关信息的 Adaptive Server 用户登录名。

spid — 是要报告其相关信息的进程号。请用引号将进程号引起（Adaptive Server 将其视为 *char* 类型）。

示例

1. sp_who

fid	spid	status	loginname	origname	hostname	blk_spid	dbname
		cmd		blk_xloid			
---	---	-----	-----	-----	-----	-----	-----
0	1	recv sleep	bird	bird	jazzy	0	master
		AWAITING COMMAND		0x0000ed92			
0	2	sleeping	NULL	NULL		0	master
		NETWORK HANDLER		0x0000ed92			
0	3	sleeping	NULL	NULL		0	master
		MIRROR HANDLER		0x0000ed92			
0	4	sleeping	NULL	NULL		0	master
		AUDIT PROCESS		0x0000ed92			
0	5	sleeping	NULL	NULL		0	master
		CHECKPOINT SLEEP		0x0000ed92			
0	6	recv sleep	rose	rose	petal	0	master
		AWAITING COMMAND		0x0000ed92			
0	7	sleeping	NULL	NULL	actor	0	sybsystemdb
		ASTC HANDLER		0x0000ed92			
0	8	running	robert	sa	helos	0	master
		SELECT		0x0000ed92			
0	9	send sleep	daisy	daisy	chain	0	pubs2
		SELECT		0x0000ed92			
0	10	alarm sleep	lily	lily	pond	0	master
		WAITFOR		0x0000ed92			
0	11	lock sleep	viola	viola	cello	8	pubs2
		SELECT		0x0000ed92			

报告 Adaptive Server 上正在运行的进程的信息。进程 11（对表执行 **select**）被进程 8（依次对同一表执行 **begin transaction** 和 **insert**）阻塞。

对于进程 8，当前的 *loginame* 是 “robert”，但是初始的 *loginame* 是 “sa”。登录 “sa” 执行 **set proxy** 命令来充当用户 “robert”。

2. sp_who victoria

报告用户 “victoria” 运行的进程的信息。

3. sp_who "17"

报告编号为 17 的 Adaptive Server 进程的信息。

4. sp_who

fid	spid	status	loginame	origname	hostname	blk_spid	dbname
	cmd		block_xloid				
0	1	running	sa	sa	helos	0	master
	SELECT		0				
0	2	sleeping	NULL	NULL		0	master
	NETWORK HANDLER		0				
0	3	sleeping	NULL	NULL		0	master
	DEADLOCK TUNE		0				
0	4	sleeping	NULL	NULL		0	master
	MIRROR HANDLER		0				
0	5	sleeping	NULL	NULL	actor	0	master
	ASTC HANDLER		0				
0	6	sleeping	NULL	NULL		0	master
	CHECKPOINT SLEEP		0				
0	5	sleeping	NULL	NULL		0	master
	HOUSEKEEPER		0				

报告 Adaptive Server 上正在运行的进程的信息。虽然除 **sp_who** 外没有运行任何用户进程，服务器仍然显示活动。空闲周期过程中，管家任务将脏缓冲移至缓冲清洗区。

注释

- **sp_who** 报告有关指定用户或 **Adaptive Server** 进程的信息。没有参数时，**sp_who** 将报告所有数据库中哪些用户正在运行什么进程。
- 如果启用已镜像磁盘或远程过程调用，镜像处理程序和节点处理器也将出现在 **sp_who** 的报告中。
- “**spid**” 列包含在 **Transact-SQL kill** 命令中使用的进程标识号。“**blk_spid**” 列包含阻塞进程的进程 ID（如果有的话）。阻塞进程当前占有其它进程所需的资源，它可能受影响或拥有排它锁。“**block_xloid**” 列标识出阻塞事务的唯一的锁所有者 ID。“**fid**” 列标识出锁所属的系列（包括协调进程及其工作进程），有关详细信息，参见 **sp_familylock**。
- 在单引擎服务器上运行 **sp_who** 可显示状态为 “**running**” 的 **sp_who** 进程和状态为 “**runnable**” 或处于休眠状态之一的其它进程。在多引擎服务器中，每个引擎可能都有一个状态为 “**running**” 的进程。
- **sp_who** 在所有系统进程的 “**loginame**” 列中都报告 **NULL**。
- 条件语句（例如 **if** 或 **while** 循环）的求值将在 “**cmd**” 列中显示为 “**COND**”。
- 系统管理员可使用 **kill** 命令删除许多进程。

权限

任何用户都可以执行 **sp_who**。

使用的表

master..sysprocesses

参见

命令	kill
系统过程	sp_lock

8

分类存储过程

本章说明分类存储过程，而分类存储过程用于在表格形式的系统表中检索信息。

表 8-1 列出了本章所涵盖的分类存储过程。

表 8-1：分类存储过程

过程	说明
sp_column_privileges	返回表或视图中一列或多列的权限信息。
sp_columns	返回可以存储在一列或多列中的数据的类型信息。
sp_databases	返回 Adaptive Server 中的数据库列表。
sp_datatype_info	返回有关特定数据类型或所有支持的数据类型的信息。
sp_fkeys	返回有关用 create table 或 alter table 命令在当前数据库中所创建外键约束的信息。
sp_pkeys	返回有关用 create table 或 alter table 命令为单个表创建的主键约束的信息。
sp_server_info	返回 Adaptive Server 特性名和当前值的列表。
sp_special_columns	返回唯一标识表或视图中的行的最佳列集；还可以返回这样一些列的列表：当通过事务更新行中的任何值时，将自动更新这些列。
sp_sproc_columns	返回有关存储过程的输入和返回参数的信息。
sp_statistics	返回单个表上索引的列表。
sp_stored_procedures	返回有关一个或多个存储过程的信息。
sp_table_privileges	返回表或视图中所有列的权限信息。
sp_tables	返回可以显示在 from 子句中的对象的列表。

分类存储过程简介

分类存储过程用于在表格形式的系统表中检索信息。

分类存储过程是在安装时通过 **installmaster** 创建的，它位于 **sybssystemprocs** 数据库中并由系统管理员所有。

许多分类存储过程都可以从任何数据库运行。如果从除 **sybssystemprocs** 之外的数据库运行分类存储过程，分类存储过程将从其执行位置的数据库中的系统表检索信息。

所有分类存储过程都在隔离级别 1 上执行。

所有分类存储过程都会报告返回状态。例如：

```
return status = 0
```

表示该过程已成功执行。本书中的示例不包括返回状态。

指定可选参数

如果分类存储过程的参数值中包含标点符号或嵌入式空格，或者参数值为保留字，则必须用单或双引号将其引起来。如果参数是被数据库名或所有者名限定的对象名，则用单或双引号将整个名称引起来。

► 注意

不要用分隔标识符作为分类存储过程参数。那样做可能会导致意外结果。

在很多情况下，以下面的格式为分类存储过程提供参数较为方便：

```
@parametername = value
```

而提供所有参数则不方便。语法语句中的参数名与过程定义的参数名是相匹配的。

例如，`sp_columns` 的语法为：

```
sp_columns table_name [, table_owner]
[, table_qualifier] [, column_name]
```

要用 `sp_columns` 查找有关特定列的信息，可以使用：

```
sp_columns publishers, @column_name = "pub_id"
```

它提供的信息与指定了所有参数的命令所提供的信息相同：

```
sp_columns publishers, "dbo", "pubs2", "pub_id"
```

还可以使用 “Null” 作为占位符：

```
sp_columns publishers, null, null, "pub_id"
```

如果指定的参数数量超出系统过程预期的数量，Adaptive Server 将忽略多余的参数。

模式匹配

Adaptive Server 通过常规表达式提供范围广泛的模式匹配。然而，为了获得最大互用性，将只采用 SQL 标准模式匹配（% 和 _ 通配符）。

系统过程表

这些分类存储过程	使用这些分类存储过程表
sp_columns	spt_datatype_info
sp_datatype_info	spt_datatype_info_ext
sp_special_columns	spt_server_info
sp_sproc_columns	

在 *sybsystemprocs* 数据库中，用于将内部系统值（如状态位）转换为
人可以阅读的格式。

分类存储过程 **sp_column_privileges** 和 **sp_table_privileges** 将创建临时表，
然后将其删除。

ODBC 数据类型

表 8-2 和表 8-3 列出了 “Data_type” 列中的 **sp_columns** 和
sp_sproc_columns 返回的数据类型代码编号和匹配的数据类型名称。
说明的来源是 Open Database Connectivity (ODBC) 应用程序编程接
口 (API)。

表 8-2: ODBC 数据类型代码编号

名称	类型
char	1
decimal	3
double precision	8
float	6
integer	4
numeric	2
real	7
smallint	5
varchar	12

表 8-3：扩展数据类型代码编号

名称	类型
<i>bigint</i>	-5
<i>binary</i> （位数据类型）	-2
<i>bit</i>	-7
<i>date</i>	9
<i>long varbinary</i>	-4
<i>long varchar</i>	-1
<i>time</i>	10
<i>timestamp</i>	11
<i>tinyint</i>	-6
<i>binary</i> （位变化数据类型）	-3

sp_column_privileges

功能

返回表或视图中一列或多列的权限信息。

语法

```
sp_column_privileges table_name [, table_owner
                                [, table_qualifier [, column_name]]]
```

参数

- table_name* — 是表名。不支持在模式匹配中使用通配符。
- table_owner* — 是表所有者的名称。不支持在模式匹配中使用通配符。如果不指定表所有者， **sp_column_privileges** 将搜索当前用户拥有的表，然后搜索数据库所有者拥有的表。
- table_qualifier* — 是数据库的名称。值为当前数据库的名称和 **null**。
- column_name* — 是要显示其权限的列的名称。使用通配符可以请求多于一列的信息。如果不指定列名，将返回指定表中所有列的权限信息。

示例

```
1. sp_column_privileges discounts, null, null,
   discounttype
```

table_qualifier	table_owner	table_name	column_name
grantor	grantee	privilege	is_grantable
-----	-----	-----	-----
pubs2	dbo	discounts	discounttype
dbo	dbo	SELECT	YES
pubs2	dbo	discounts	discounttype
dbo	dbo	UPDATE	YES
pubs2	dbo	discounts	discounttype
dbo	dbo	REFERENCE	YES
pubs2	dbo	discounts	discounttype
dbo	guest	SELECT	NO
pubs2	dbo	discounts	discounttype
dbo	guest	UPDATE	NO
pubs2	dbo	discounts	discounttype
dbo	guest	REFERENCE	NO

注释

- 表 8-4 对结果集进行了说明：

表 8-4: sp_column_privileges 结果集

列	数据类型	说明
<i>table_qualifier</i>	<i>varchar(32)</i>	数据库的名称，该数据库中存储了为 <i>table_name</i> 参数指定的表。
<i>table_owner</i>	<i>varchar(32)</i>	表所有者。如果没有为 <i>table_owner</i> 参数指定值，该值为当前所有者或 <i>dbo</i> 。
<i>table_name</i>	<i>varchar(32)</i>	为 <i>table_name</i> 参数指定的名称。该值不能为 NULL。
<i>column_name</i>	<i>varchar(32)</i>	指定的列名。如果没有在该语句中指定列名，结果中将包含指定表中的所有列。
<i>grantor</i>	<i>varchar(32)</i>	数据库用户的名称，他已将 <i>column_name</i> 权限授予 <i>grantee</i> 。该值不能为 NULL。
<i>grantee</i>	<i>varchar(32)</i>	数据库用户的名称，他已被 <i>grantor</i> 授予 <i>column_name</i> 权限。该值不能为 NULL。
<i>privilege</i>	<i>varchar(32)</i>	标识列的权限。可以为下列权限之一： SELECT — 允许被授权者检索列中的数据。 UPDATE — 允许被授权者更新列中的数据。 REFERENCE — 允许被授权者在约束（如唯一、参照或表检查约束）中引用列。
<i>is_grantable</i>	<i>varchar(3)</i>	指示是否允许被授权者将权限授予其他用户。该值为 YES、NO 和 NULL。

权限

任何用户都可以执行 **sp_column_privileges**。

使用的表

syscolumns、*sysobjects*、*sysusers*

参见

系统过程	sp_help 、 sp_helprotect
------	---------------------------------------

sp_columns

功能

返回可以存储在一列或多列中的数据的类型信息。

语法

```
sp_columns table_name [, table_owner ]
           [, table_qualifier] [, column_name]
```

参数

table_name — 是表或视图的名称。使用通配符可以请求多个表的信息。

table_owner— 是表或视图的所有者。使用通配符可以请求被多个用户所拥有的表的信息。如果不指定表所有者， **sp_columns** 将搜索当前用户拥有的表，然后搜索数据库所有者拥有的表。

table_qualifier—是数据库的名称。该值可以是当前数据库或 NULL。

column_name — 是要从中获取信息的列的名称。使用通配符可以请求多个列的信息。

示例

```
1. sp_columns "publishers", null, null, "p%"
```

table_qualifier	table_owner			
table_name	column_name			
data_type	type_name	precision	length	
scale	radix	nullable		
remarks				

ss_data_type colid

```
pubs2                                dbo
publishers                          pub_id
      1 char                        NULL          4
NULL    NULL                      0
```

```

NULL
          47      1
pubs2          dbo
  publishers          pub_name
        12 varchar          NULL          40
      NULL  NULL          1
      NULL
          39      2
```

显示 *publishers* 表中所有以 “p” 开头的列的信息。

2. **sp_columns "s%", null, null, "st%"**

显示以 “s” 开头的表中以 “st” 开头的所有列的信息。

注释

- 表 8-5 显示了结果集：

表 8-5: sp_columns 结果集

列	数据类型	说明
<i>table_qualifier</i>	<i>varchar(32)</i>	数据库的名称，该数据库中存储了为 <i>table_name</i> 参数指定的表。
<i>table_owner</i>	<i>varchar(32)</i>	表所有者。如果没有为 <i>table_owner</i> 参数指定值，该值为当前所有者或 <i>dbo</i> 。
<i>table_name</i>	<i>varchar(32)</i>	NOT NULL。
<i>column_name</i>	<i>varchar(32)</i>	NOT NULL。
<i>data_type</i>	<i>smallint</i>	ODBC 数据类型的整数代码。如果该数据类型无法映射到 ODBC 类型，则它为 NULL。
<i>type_name</i>	<i>varchar(30)</i>	表示数据类型的字符串。基础 DBMS 说明这种数据类型名。
<i>precision</i>	<i>int</i>	有效数字的位数。
<i>length</i>	<i>int</i>	用字节表示的数据类型的长度。
<i>scale</i>	<i>smallint</i>	小数点右侧的位数。
<i>radix</i>	<i>smallint</i>	数值数据类型的基。
<i>nullable</i>	<i>smallint</i>	值为 1 表示可以出现 NULL；值为 0 表示 NOT NULL。
<i>remarks</i>	<i>varchar(254)</i>	
<i>ss_data_type</i>	<i>smallint</i>	一种 Adaptive Server 数据类型。
<i>colid</i>	<i>tinyint</i>	添加到结果集的列。

表 8-5: sp_columns 结果集 (续)

列	数据类型	说明
<i>column_def</i>	<i>varchar(255)</i>	
<i>sql_data_type</i>	<i>smallint</i>	
<i>sql_datetime_sub</i>	<i>smallint</i>	
<i>char_octet_length</i>	<i>int</i>	
<i>ordinal_position</i>	<i>int</i>	
<i>is_nullable</i>	<i>varchar(3)</i>	

- 对于定义为 *double precision* 的列，**sp_columns** 报告 *type_name* 为浮点型，而报告 *data_type* 为 6。Adaptive Server *double precision* 数据类型是一种浮点执行类型，它支持 ODBC 规范中指定的值的范围。

权限

任何用户都可以执行 **sp_columns**。

使用的表

systcolumns、*sysobjects*、*systypes*、*sybsystemprocs..spt_datatype_info*

参见

系统过程	sp_help
------	----------------

sp_databases

功能

返回 Adaptive Server 中的数据库列表。

语法

```
sp_databases
```

参数

无。

示例

1. sp_databases

database_name	database_size	remarks
-----	-----	-----
master	5120	NULL
model	2048	NULL
mydb	2048	NULL
pubs2	2048	NULL
sybsecurity	5120	NULL
sybsystemprocs	16384	NULL
tempdb	2048	NULL

注释

- 表 8-6 对结果集进行了说明：

表 8-6: sp_databases 结果集

列	数据类型	说明
database_name	char(32)	非空的数据库名。
database_size	int	用千字节表示的数据库大小。
remarks	varchar(254)	Adaptive Server 始终返回 NULL。

权限

任何用户都可以执行 sp_databases。

使用的表

master..sysdatabases、 master..sysusages、 sysobjects

参见

系统过程	sp_helpdb
------	-----------

sp_datatype_info

功能

返回有关特定 ODBC 数据类型或所有 ODBC 数据类型的信息。

语法

```
sp_datatype_info [data_type]
```

参数

data_type — 是指定的 ODBC 数据类型有关返回信息的代码编号。数据类型代码在表 8-2 和第 8-4 页的表 8-3 中列出。

注释

- 表 8-7 对结果集进行了说明：

表 8-7: sp_datatype_info 结果集

列	数据类型	说明
<i>type_name</i>	<i>varchar(30)</i>	与 DBMS 有关的数据类型名（与 sp_columns 结果集中的 <i>type_name</i> 列相同）。
<i>data_type</i>	<i>smallint</i>	该类型的所有列都映射到的 ODBC 类型代码。
<i>precision</i>	<i>int</i>	数据源上数据类型的最大精度。如果精度不适用于数据类型，则返回零。
<i>literal_prefix</i>	<i>varchar(32)</i>	在文字开头使用的字符。例如，对于字符类型为单引号 (')，而对于二进制为 0x。
<i>literal_suffix</i>	<i>varchar(32)</i>	在文字结尾使用的字符。例如，对于字符类型为单引号 (')，而对于二进制则不必加任何符号。
<i>create_params</i>	<i>varchar(32)</i>	该数据类型的创建参数说明。
<i>nullable</i>	<i>smallint</i>	值为 1 表示在创建该数据类型时，允许空值；值为 0 则表示不允许空值。
<i>case_sensitive</i>	<i>smallint</i>	值为 1 表示该类型的所有列都是区分大小写的（对于归类）； 0 表示不区分大小写。
<i>searchable</i>	<i>smallint</i>	值为 1 表示该类型的所有列都可以在 where 子句中使用。
<i>unsigned_attribute</i>	<i>smallint</i>	值为 1 表示该数据类型是无符号的； 0 表示该数据类型是有符号的。

表 8-7: sp_datatype_info 结果集 (续)

列	数据类型	说明
<i>money</i>	<i>smallint</i>	值为 1 表示该数据类型是 money 数据类型； 0 表示该数据类型不是 money 数据类型。
<i>auto_increment</i>	<i>smallint</i>	值为 1 表示该数据类型是自动递增的； 0 表示该数据类型不会自动递增。
<i>local_type_name</i>	<i>varchar(128)</i>	该数据类型与数据源相关的名称的本地化版本。
<i>sql_data_type</i>	<i>smallint</i>	
<i>sql_datetime_sub</i>	<i>smallint</i>	
<i>num_prec_radix</i>	<i>smallint</i>	
<i>interval_precision</i>	<i>smallint</i>	

权限

任何用户都可以执行 **sp_datatype_info**。

使用的表

sybsystemprocs..spt_datatype_info、 *systypes*、 *sysdatabases*、
sysmessages、 *sysprocesses*

参见

系统过程	sp_help
------	----------------

sp_fkeys

功能

返回有关用 **create table** 或 **alter table** 命令在当前数据库中创建的外键约束的信息。

语法

```
sp_fkeys pktable_name [, pktable_owner]
        [, pktable_qualifier] [, fktable_name]
        [, fktable_owner] [, fktable_qualifier]
```

参数

pktable_name — 是主键表的名称。不支持在模式匹配中使用通配符。必须指定 **pktable_name** 或 **fktable_name**，或指定两者。

pktable_owner — 是主键表所有者的名称。不支持在模式匹配中使用通配符。如果不指定表所有者，**sp_fkeys** 将搜索当前用户拥有的表，然后搜索数据库所有者拥有的表。

pktable_qualifier — 是包含主键表的数据库的名称。该值可以是当前数据库或 **NULL**。

fktable_name — 是外键表的名称。不支持在模式匹配中使用通配符。必须给出 **fktable_name** 或 **pktable_name**，或给出两者。

fktable_owner — 是外键表所有者的名称。不支持在模式匹配中使用通配符。如果不指定 **fktable_owner**，**sp_fkeys** 将搜索当前用户拥有的表，然后搜索数据库所有者拥有的表。

fktable_qualifier — 是包含外键表的数据库的名称。该值可以是当前数据库或 **null**。

注释

- **sp_fkeys** 返回有关用 **create table** 或 **alter table** 命令在当前数据库中创建的外键约束的信息。外键是表中在逻辑上依赖于其它表的 **primary key** 列的键列。

- 表 8-8 对结果集进行了说明：

表 8-8: sp_fkeys 结果集

列	数据类型	说明
<i>pktable_qualifier</i>	<i>varchar(32)</i>	包含主键表的数据库。
<i>pktable_owner</i>	<i>varchar(32)</i>	主键表的所有者。
<i>pktable_name</i>	<i>varchar(32)</i>	NOT NULL。
<i>pkcolumn_name</i>	<i>varchar(32)</i>	NOT NULL。
<i>fktable_qualifier</i>	<i>varchar(32)</i>	包含外键表的数据库。
<i>fktable_owner</i>	<i>varchar(32)</i>	外键表的所有者。
<i>fktable_name</i>	<i>varchar(32)</i>	NOT NULL。
<i>fkcolumn_name</i>	<i>varchar(32)</i>	NOT NULL。
<i>key_seq</i>	<i>smallint</i>	NOT NULL。列在多列主键中的序列号。
<i>update_rule</i>	<i>smallint</i>	当 SQL 操作为 UPDATE 时对外键执行的操作。该列将返回零。
<i>delete_rule</i>	<i>smallint</i>	当 SQL 操作为 DELETE 时对外键执行的操作。该列将返回零。

- 主键和外键都必须已经在 **create table** 或 **alter table** 语句中声明。
- 如果提供了主键表名，但外键表名为 NULL，**sp_fkeys** 将把所有包含外键的表返回到给定表。如果提供了外键表名，但主键表名为 NULL，**sp_fkeys** 将把所有通过主键/外键关系关联的表返回到外键表中的外键。
- **sp_fkeys** 不会返回有关用 **sp_commonkey**、**sp_foreignkey** 或 **sp_primarykey** 系统过程声明的键的信息。

权限

任何用户都可以执行 **sp_fkeys**。

使用的表

sysobjects、*sysreferences*

参见

命令	alter table 、 create table
系统过程	sp_helpkey

sp_pkeys

功能

返回有关用 **create table** 或 **alter table** 命令为单个表创建的主键约束的信息。

语法

```
sp_pkeys table_name [, table_owner]
        [, table_qualifier]
```

参数

- table_name* — 是表名。不支持在模式匹配中使用通配符。
- table_owner* — 是表所有者的名称。不支持在模式匹配中使用通配符。如果不指定 *table_owner*, **sp_pkeys** 将搜索当前用户拥有的表, 然后搜索数据库所有者拥有的表。
- table_qualifier* — 是包含表的数据库的名称。该值可以是当前数据库或 NULL。

注释

- 表 8-9 对结果集进行了说明:

表 8-9: sp_pkeys 结果集

列	数据类型	说明
<i>table_qualifier</i>	<i>varchar</i> (32)	数据库名。该域可以为 NULL。
<i>table_owner</i>	<i>varchar</i> (32)	表所有者。如果没有为 <i>table_owner</i> 参数指定值, 该值为当前所有者或 <i>dbo</i> 。
<i>table_name</i>	<i>varchar</i> (32)	NOT NULL。
<i>column_name</i>	<i>varchar</i> (32)	NOT NULL。
<i>key_seq</i>	<i>smallint</i>	NOT NULL。列在多列主键中的序列号。

- 主键必须已经用 **create table** 或 **alter table** 语句进行声明, 而不用 **sp_primarykey** 系统过程来声明。
- 术语 **primary key** 是指表的逻辑主键。Adaptive Server 期望每个逻辑主键都具有在其上定义的唯一索引, 并且该唯一索引还可以返回到 **sp_statistics**。

权限

任何用户都可以执行 **sp_pkeys**。

使用的表

sysindexes、 *sysobjects*

参见

命令	alter table、 create table
系统过程	sp_helpkey

sp_server_info

功能

返回 Adaptive Server 特性名和当前值的列表。

语法

```
sp_server_info [attribute_id]
```

参数

attribute_id — 是服务器特性的整数 ID。

示例

```
1. sp_server_info 12
```

attribute_id	attribute_name	attribute_value
12	MAX_OWNER_NAME_LENGTH	0

```
2. sp_server_info
```

返回通过强制行说明的服务器特性及其值的列表。

注释

- 表 8-10 对结果集进行了说明：

表 8-10: sp_server_info 结果集

列	数据类型	说明
<i>attribute_id</i>	<i>int</i>	NOT NULL。
<i>attribute_name</i>	<i>varchar(60)</i>	NOT NULL。
<i>attribute_value</i>	<i>varchar(255)</i>	

- 表 8-11 显示了结果集中的强制行：

表 8-11: sp_server_info 返回的强制结果

ID	服务器特性名	说明	值
1	DBMS_NAME	DBMS 的名称。	SQL_SERVER
2	DBMS_VER	DBMS 的版本。	@@version
6	DBE_NAME	未使用	

表 8-11: sp_server_info 返回的强制结果 (续)

ID	服务器特性名	说明	值
10	OWNER_TERM	表所有者使用的 Adaptive Server 术语 (三部分名称中的第二部分)。	owner
11	TABLE_TERM	表的 Adaptive Server 术语 (三部分名称中的第三部分)。	table
12	MAX_OWNER_NAME_LENGTH	表所有者名称的最大长度 (三部分名称中的第二部分)。	30
16	IDENTIFIER_CASE	数据库中用户定义名称 (表名、列名和存储过程名) 是否区分大小写 (这些对象显示在系统目录中的大小写形式)。	MIXED
15	COLUMN_LENGTH	列名的最大字符数。	30
13	TABLE_LENGTH	表名的最大字符数。	30
100	USERID_LENGTH	用户名的最大字符数。	30
17	TX_ISOLATION	服务器假定的最初事务隔离级别, 这与 SQL92 中定义的隔离级别是相对应的。	4
18	COLLATION_SEQ	该服务器假定的字符集排序。	
14	MAX_QUAL_LENGTH	表限定符名称的最大长度 (三部分表名中的第一部分)。	30
101	QUALIFIER_TERM	表限定符的 Adaptive Server 术语 (三部分名称中的第一部分)。	database
19	SAVEPOINT_SUPPORT	基础 DBMS 是否支持指定的保存点?	Y
20	MULTI_RESULT_SETS	基础 DBMS 或网关本身是否支持多结果集 (在多个结果集返回到客户端的情况下, 是否可以通过网关发送多个语句)?	Y
102	NAMED_TRANSACTIONS	基础 DBMS 是否支持指定的事务?	Y
103	SPROC_AS_LANGUAGE	是否可以将存储过程作为语言事件执行?	Y
103	REMOTE_SPROC	是否可以通过 DB-Library 中的远程存储过程 API 执行存储过程?	Y
22	ACCESSIBLE_TABLES	在 sp_tables 存储过程中, 网关是否只返回当前用户 (即至少具有对表的 select 权限的用户) 可以访问的表、视图等?	Y
104	ACCESSIBLE_SPROC	在 sp_stored_procedures 存储过程中, 网关是否只返回当前用户可以执行的存储过程?	Y
105	MAX_INDEX_COLS	DBMS 索引的最大列数。	32
106	RENAME_TABLE	是否可以重命名表?	Y

表 8-11: sp_server_info 返回的强制结果 (续)

ID	服务器特性名	说明	值
107	RENAME_COLUMN	是否可以重命名列?	Y
108	DROP_COLUMN	是否可以删除列?	Y
109	INCREASE_COLUMN_LENGTH	是否可以增加列的大小?	N
110	DDL_IN_TRANSACTION	事务中是否可以出现 DDL 语句?	Y
111	DESCENDING_INDEXES	是否支持降序索引?	Y
112	SP_RENAME	是否可以重命名存储过程?	Y
500	SYS_SPROC_VERSION	当前执行的分类存储过程的版本。	01.01.2822

权限

任何用户都可以执行 **sp_server_info**。

使用的表

sybsystemprocs..spt_server_info、*sysobjects*

参见

分类存储过程	sp_stored_procedures 、 sp_tables
--------	--

sp_special_columns

功能

返回唯一标识表或视图中的行的最佳列集；还可以返回 *timestamp* 列的列表。当事务更新行中的任何值时，将自动生成这些列的值。

语法

```
sp_special_columns table_name [, table_owner]
                    [, table_qualifier] [, col_type]
```

参数

- table_name* — 是表或视图的名称。不支持在模式匹配中使用通配符。
- table_owner* — 是表或视图所有者的名称。不支持在模式匹配中使用通配符。如果不指定表所有者，**sp_special_columns** 将搜索当前用户拥有的表，然后搜索数据库所有者拥有的表。
- table_qualifier* — 是数据库的名称。该值可以是当前数据库或 NULL。
- col_type* — 为 **R** 时，返回有关列的信息，这些列的值唯一标识表中的任一行；为 **V** 时，返回有关 *timestamp* 列的信息，每次插入或更新新行时，**Adaptive Server** 都将生成这些列的值。

示例

1. sp_special_columns systypes

scope	column_name	data_type	type_name	precision
	length	scale		

0	name	12	varchar	30
	30	NULL		

2. sp_special_columns @table_name=authors, @col_type=R

scope	column_name	data_type	type_name	precision
	length	scale		

0	au_id	12	varchar	11
	11	NULL		

注释

- 表 8-12 对结果集进行了说明：

表 8-12: sp_special_columns 结果集

列	数据类型	说明
<i>scope</i>	<i>int</i>	NOT NULL。行 ID 的实际范围。 Adaptive Server 始终返回 0。
<i>column_name</i>	<i>varchar(30)</i>	NOT NULL。列标识符。
<i>data_type</i>	<i>smallint</i>	ODBC 数据类型的整数代码。如果数据类型无法映射到 ANSI/ISO 类型，则该值为 NULL。 <i>type_name</i> 列中返回的本机数据类型名称。 (参见 ODBC 数据类型第 8-3 页的表 8-2。)
<i>type_name</i>	<i>varchar(13)</i>	数据类型的字符串表示。这是基础 DBMS 表示的数据类型名。
<i>precision</i>	<i>int</i>	有效数字的位数。
<i>length</i>	<i>int</i>	用字节表示的数据类型的长度。
<i>scale</i>	<i>smallint</i>	小数点右侧的位数。

权限

任何用户都可执行 **sp_special_columns**。

使用的表

sybsystemprocs..spt_datatype_info、*syscolumns*、*sysindexes*、*sysobjects*、*systypes*、*sysusers*

参见

数据类型	时间戳数据类型
系统过程	sp_help

sp_sproc_columns

功能

返回有关存储过程的输入和返回参数的信息。

语法

```
sp_sproc_columns procedure_name [, procedure_owner]
                [, procedure_qualifier] [, column_name]
```

参数

- procedure_name* — 是存储过程的名称。不支持在模式匹配中使用通配符。
- procedure_owner* — 是存储过程的所有者。不支持在模式匹配中使用通配符。如果不指定过程的所有者，**sp_sproc_columns** 将搜索当前用户拥有的过程，然后搜索数据库所有者拥有的过程。
- procedure_qualifier* — 是数据库的名称。该值可以是当前数据库或 NULL。
- column_name* — 是要从中获取信息的参数的名称。如果不提供参数名，**sp_sproc_columns** 将返回有关存储过程的所有输入和返回参数的信息。

注释

- 表 8-13 对结果集进行了说明：

表 8-13: sp_sproc_columns 结果集

列	数据类型	说明
<i>procedure_qualifier</i>	<i>varchar(30)</i>	
<i>procedure_owner</i>	<i>varchar(30)</i>	
<i>procedure_name</i>	<i>varchar(41)</i>	NOT NULL。
<i>column_name</i>	<i>varchar(30)</i>	NOT NULL。
<i>column_type</i>	<i>smallint</i>	
<i>data_type</i>	<i>smallint</i>	ODBC 数据类型的整数代码。如果数据类型无法映射到 ANSI/ISO 类型，则该值为 NULL。 <i>type_name</i> 列中返回的本机数据类型名称。

表 8-13: sp_sproc_columns 结果集 (续)

列	数据类型	说明
<i>type_name</i>	<i>char</i> (30)	数据类型的字符串表示。这是基础 DBMS 表示的数据类型名。
<i>precision</i>	<i>int</i>	有效数字的位数。
<i>length</i>	<i>int</i>	用字节表示的数据类型的长度。
<i>scale</i>	<i>smallint</i>	小数点右侧的位数。
<i>radix</i>	<i>smallint</i>	数值类型的基。
<i>nullable</i>	<i>smallint</i>	值为 1 表示在创建该数据类型时，允许空值；值为 0 则表示不允许空值。
<i>remarks</i>	<i>varchar</i> (254)	NULL。
<i>ss_data_type</i>	<i>tinyint</i>	一种 Adaptive Server 数据类型。
<i>colid</i>	<i>tinyint</i>	添加到结果集的 Adaptive Server 特定列。

- 对于定义为 *double precision* 的参数，**sp_sproc_columns** 报告 *type_name* 为浮点型，而报告 *data_type* 为 6。Adaptive Server *double precision* 数据类型是一种浮点执行类型，它支持 ODBC 规范中指定的值的范围。

权限

任何用户都可执行 **sp_sproc_columns**。

使用的表

sysprocs..spt_datatype_info、*syscolumns*、*sysobjects*、*sysprocedures*、*systypes*

参见

系统过程	sp_help、sp_helptext
------	---------------------

sp_statistics

功能

返回单个表上的索引列表。

语法

```
sp_statistics table_name [, table_owner]
               [, table_qualifier] [, index_name] [, is_unique]
```

参数

- table_name* — 是表名。不支持在模式匹配中使用通配符。
- table_owner* — 是表的所有者。不支持在模式匹配中使用通配符。如果不指定 *table_owner*，**sp_statistics** 将搜索当前用户拥有的表，然后搜索数据库所有者拥有的表。
- table_qualifier* — 是数据库的名称。该值可以是当前数据库或 NULL。
- index_name* — 是索引名。不支持在模式匹配中使用通配符。
- is_unique* — 为 **Y** 时，只返回唯一索引；相反，为 **N** 时，既返回唯一索引，又返回非唯一索引。

示例

1. sp_statistics publishers

table_qualifier			table_owner		
table_name			non_unique		
index_qualifier			index_name		
type	seq_in_index	column_name	collation		
cardinality	pages				

-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		
-----			-----		

注释

- 表 8-14 对结果集进行了说明：

表 8-14: sp_statistics 结果集

列	数据类型	说明
<i>table_qualifier</i>	<i>varchar</i> (32)	数据库名。该域可以为 NULL。
<i>table_owner</i>	<i>varchar</i> (32)	
<i>table_name</i>	<i>varchar</i> (32)	NOT NULL。
<i>non_unique</i>	<i>smallint</i>	NOT NULL。值为 0 表示唯一，值为 1 表示不唯一。
<i>index_qualifier</i>	<i>varchar</i> (32)	
<i>index_name</i>	<i>varchar</i> (32)	
<i>type</i>	<i>smallint</i>	NOT NULL。值为 0 表示集群，值为 2 表示散列，值为 3 表示其它。
<i>seq_in_index</i>	<i>smallint</i>	NOT NULL。
<i>column_name</i>	<i>varchar</i> (32)	NOT NULL。
<i>collation</i>	<i>char</i> (1)	值为 A 表示升序；值为 D 表示降序；值为 NULL 表示不适用。
<i>cardinality</i>	<i>int</i>	表中的行数或索引中的唯一值数。
<i>pages</i>	<i>int</i>	存储索引或表的页数。

- 结果集中的索引按升序显示，其顺序按 *non-unique*、*type*、*index_name* 和 *seq_in_index* 列进行排序。
- 索引类型 *hashed* 接受精确匹配或范围搜索，但涉及模式匹配的搜索不使用索引。

权限

任何用户都可以执行 **sp_statistics**。

使用的表

systcolumns、*sysindexes*、*sysobjects*

参见

系统过程	sp_help 、 sp_helpindex
------	--------------------------------------

sp_stored_procedures

功能

返回有关一个或多个存储过程的信息。

语法

```
sp_stored_procedures [sp_name [, sp_owner  
                        [, sp_qualifier]]]
```

参数

- sp_name* — 是存储过程的名称。使用通配符可以请求多个存储过程的信息。
- sp_owner* — 是存储过程的所有者。使用通配符可以请求被多个用户所拥有的过程的信息。
- sp_qualifier* — 是数据库的名称。该值可以是当前数据库或 NULL。

注释

- sp_stored_procedures** 只返回有关当前数据库中存储过程的信息。
- 表 8-15 显示了结果集：

表 8-15: sp_stored_procedures 结果集

列	数据类型	说明
<i>procedure_qualifier</i>	<i>varchar</i> (30)	数据库的名称。
<i>procedure_owner</i>	<i>varchar</i> (30)	
<i>procedure_name</i>	<i>varchar</i> (41)	NOT NULL。
<i>num_input_params</i>	<i>int</i>	NOT NULL。始终返回 -1。
<i>num_output_params</i>	<i>int</i>	NOT NULL。值 ≥ 0 将显示参数的个数；值为 -1 表示参数的个数不确定。
<i>num_result_sets</i>	<i>int</i>	NOT NULL。始终返回 -1。
<i>remarks</i>	<i>varchar</i> (254)	NULL。

- sp_stored_procedures** 可以返回当前用户对其没有执行权限的存储过程的名称。但是，如果 **sp_server_info** 结果集中的服务器特性 *accessible_sproc* 为 “Y”，将只返回当前用户可以执行的存储过程。

权限

任何用户都可以执行 **sp_stored_procedures**。

使用的表

sysobjects、*sysprocedures*、*sysprotects*、*sysusers*

参见

系统过程	sp_help、sp_helptext
------	---------------------

sp_table_privileges

功能

返回表或视图中所有列的权限信息。

语法

```
sp_table_privileges table_name [, table_owner
                             [, table_qualifier]]
```

参数

- table_name* — 是表名。不支持在模式匹配中使用通配符。
- table_owner* — 是表所有者的名称。不支持在模式匹配中使用通配符。如果不指定表所有者， **sp_table_privileges** 将搜索当前用户拥有的表，然后搜索数据库所有者拥有的表。
- table_qualifier* — 是数据库的名称。该值可以是当前数据库或 NULL。

注释

- 表 8-16 显示了结果集：

表 8-16: sp_table_privileges 结果集

列	数据类型	说明
<i>table_qualifier</i>	<i>varchar</i> (32)	数据库的名称。该域可以为 NULL。
<i>table_owner</i>	<i>varchar</i> (32)	
<i>table_name</i>	<i>varchar</i> (32)	NOT NULL。
<i>grantor</i>	<i>varchar</i> (32)	NOT NULL。
<i>grantee</i>	<i>varchar</i> (32)	NOT NULL。

表 8-16: sp_table_privileges 结果集 (续)

列	数据类型	说明
<i>privilege</i>	<i>varchar(32)</i>	标识表的权限。可以为下列权限之一： SELECT — 允许被授权者检索表的一列或多列中的数据。 INSERT — 允许被授权者将包含一列或多列数据的新行插入表。 UPDATE — 允许被授权者更新表的一列或多列中的数据。 DELETE — 允许被授权者将数据行从表中删除。 REFERENCE — 允许被授权者在约束中引用表的一列或多列。
<i>is_grantable</i>	<i>varchar(3)</i>	指示是否允许被授权者将权限授予其他用户。该值为 YES、NO 和 NULL。

权限

任何用户都可以执行 **sp_table_privileges**。

使用的表

sysobjects、*sysusers*

参见

系统过程	sp_help 、 sp_helprotect
------	---------------------------------------

sp_tables

功能

返回可以显示在 **from** 子句中的对象的列表。

语法

```
sp_tables [table_name] [, table_owner]
          [, table_qualifier][, table_type]
```

参数

table_name — 是表名。使用通配符可以请求多个表的信息。

table_owner — 是表的所有者。使用通配符可以请求多个表的信息。

table_qualifier — 是数据库的名称。可以接受的值为当前数据库的名称和 NULL。

table_type — 是值的列表，用逗号分隔，给出了指定表类型的所有表的信息，其中包括以下内容：

```
"'TABLE', 'SYSTEM TABLE', 'VIEW'"
```

► 注意

用单引号将每种表类型引起来，然后用双引号将整个参数引起来。以大写格式输入表类型。

示例

```
1. sp_tables @table_type = "'TABLE', 'VIEW'"
```

该过程将返回当前数据库中类型 **TABLE** 和 **VIEW** 的所有表的信息，并排除系统表的信息。

注释

- **Adaptive Server** 不必检查对 *table_name* 的读写权限。不保证对表的访问，即使您可以显示其信息，也是如此。
- 结果集包含表、视图以及 **DBMS** 产品网关的同义词和别名。
- 如果 **sp_server_info** 结果集中的服务器特性 *accessible_tables* 为“Y”，将只返回当前用户可以访问的表。

- 表 8-17 对结果集进行了说明：

表 8-17: sp_tables 结果集

列	数据类型	说明
<i>table_qualifier</i>	<i>varchar</i> (30)	数据库名。该域可以为 NULL。
<i>table_owner</i>	<i>varchar</i> (30)	
<i>table_name</i>	<i>varchar</i> (30)	NOT NULL。表名。
<i>table_type</i>	<i>varchar</i> (32)	NOT NULL。下面中的一个: 'TABLE'、 'VIEW'、'SYSTEM TABLE'。
<i>remarks</i>	<i>varchar</i> (254)	NULL

权限

任何用户都可以执行 **sp_tables**。

使用的表

sysdatabases、*sysobjects*、*sysprotects*、*sysusers*

参见

系统过程	sp_help
分类存储过程	sp_server_info

9

系统扩展存储过程

本章介绍系统扩展存储过程 (ESP)，它们是 Sybase 提供的扩展存储过程。

表 9-1 列出本章中介绍的系统扩展存储过程。

表 9-1：系统扩展存储过程

过程	说明	平台
xp_cmdshell	在运行 Adaptive Server 的主机系统上执行本地操作系统命令。	所有支持 DLL 的平台
xp_deletemail	从 Adaptive Server 收件箱中删除消息。	仅对 NT
xp_enumgroups	显示特定 Windows NT 域的组。	仅对 NT
xp_findnextmsg	在 Adaptive Server 收件箱中检索下一个消息的消息标识符。	仅对 NT
xp_logevent	用于在 Windows NT 事件日志中记录用户定义的事件。	仅对 NT
xp_readmail	从 Adaptive Server 收件箱中阅读消息。	仅对 NT
xp_sendmail	使用 MAPI 接口将消息发送给指定收件人。	仅对 NT
xp_startmail	启动 Adaptive Server 邮件会话。	仅对 NT
xp_stopmail	停止 Adaptive Server 邮件会话。	仅对 NT

简介

系统扩展存储过程是在安装时通过 **installmaster** 创建的，它位于 **sybssystemprocs** 数据库中并由系统管理员所拥有。它们可从任何数据库运行。

对系统 ESP 的权限

因为系统扩展存储过程位于 **sybssystemprocs** 数据库中，所以也在该数据库设置其权限。

具有 **sa_role** 的用户对系统 ESP 具有缺省执行权限。这些系统管理员可以将执行权限授予其他用户。

与系统 ESP 相关的 DLL

通过在 *sybserverprocs* 数据库中运行 `sp_helpextendedproc`，可以获得与系统 ESP 相关的 DLL 的名称。

使用系统 ESP

系统 ESP 所遵循的调用约定与常规系统过程相同。

对于系统 ESP 的唯一的额外要求是必须运行 **Open Server** 应用程序 **XP Server**。**Adaptive Server** 在首次调用 ESP 时启动 **XP Server**。在关闭 **Adaptive Server** 之前，**XP Server** 将持续运行。

xp_cmdshell

功能

在运行 Adaptive Server 的主机系统上执行本机操作系统命令。

语法

```
xp_cmdshell command [, no_output]
```

参数

command — 是操作系统命令字符串；最大长度为 255 个字节。

no_output — 如指定，将取消该命令的输出。

示例

1. **xp_cmdshell 'copy C:\log A:\log.0102', no_output**

自动将 C 驱动器上的名为 *log* 的文件复制到 A 驱动器上名为 *log.0102* 的文件。

2. **xp_cmdshell 'date'**

执行操作系统的 **date** 命令，将当前日期作为数据行返回。

注释

- **xp_cmdshell** 以一系列文本行的方式返回所有输出，包括操作系统错误。
- 从 XP Server 的当前目录运行 **xp_cmdshell**。
- 返回的输出的列宽为 80 个字符。该输出未经格式化。
- **xp_cmdshell** 无法执行需要与用户进行交互的命令，如 “login”。
- 通过 **xp_cmdshell** 执行操作系统命令所在的用户环境由 **xp_cmdshell context** 配置参数的值控制。如果将该参数设置为 1（缺省值），**xp_cmdshell** 将限制权限，只授予在操作系统级具有系统管理特权的用户。如果将该参数设置为 0，**xp_cmdshell** 将使用运行 Adaptive Server 的操作系统帐户的安全环境。因此，如果使用将 **xp_cmdshell context** 配置参数设置为 0 的 **xp_cmdshell**，任何用户都可以使用运行 Adaptive Server 的帐户的权限来执行操作系统命令。该帐户的限制可能要少于用户自己帐户的限制。

有关 **xp_cmdshell context** 的详细信息，参见 *系统管理指南*。

- 无论 **xp_cmdshell context** 的值如何，如果执行 **xp_cmdshell** 的用户不是系统管理员（不具有 **sa_role**），则系统管理员必须已授予该用户明确的权限，以便执行 **xp_cmdshell**。例如，以下语句授予“joe”执行 **xp_cmdshell** 的权限：

```
grant execute on xp_cmdshell to joe
```

权限

缺省情况下，只有系统管理员才能执行 **xp_cmdshell**。系统管理员可以将执行权限授予其他用户。

参见

系统过程	sp_configure
------	--------------

xp_deletemail

(仅适于 Windows NT)

功能

从 Adaptive Server 收件箱中删除消息。

语法

```
xp_deletemail [msg_id]
```

参数

msg_id — 是要删除的邮件消息的消息标识符。

示例

```
1. declare @cur_msg_id binary(255)
   xp_deletemail @msg_id = @cur_msg_id
```

从 Adaptive Server 收件箱中删除带有 *cur_msg_id* 变量所指定的消息标识符的消息。

```
2. xp_deletemail
```

从 Adaptive Server 收件箱中删除第一条消息。

注释

- 使用 `xp_findnextmsg` 获得 *msg_id*。
- 如果未使用 *msg_id* 参数，缺省情况下将把收件箱中的第一个消息作为要删除的消息。

权限

缺省情况下，只有系统管理员才能执行 `xp_deletemail`。系统管理员可以将此权限授予其他用户。

参见

系统 ESP	xp_findnextmsg、xp_startmail
系统过程	sp_processmail

xp_enumgroups

(仅适于 Windows NT)

功能

显示指定 Windows NT 域的组。

语法

```
xp_enumgroups [domain_name]
```

参数

domain_name — 是要为其列出用户组的 Windows NT 域。

示例

1. xp_enumgroups

列出运行 XP Server 的 Windows NT 计算机上的所有用户组。

2. xp_enumgroups 'PCS'

列出 PCS 域中的所有用户组。

注释

- 如果未传递任何参数，**xp_enumgroups** 将显示所有本地用户组。
- **域**是指定的一组计算机，它们共享同一公用用户帐户数据库和安全性策略。
- 如果返回状态为 **0**，表示成功；如果返回状态为 **1**，则表示失败。

权限

缺省情况下，只有系统管理员才能执行 **xp_enumgroups**。系统管理员可以将此权限授予其他用户。

xp_findnextmsg

(仅适于 Windows NT)

功能

从 Adaptive Server 收件箱中检索下一个消息标识符。

语法

```
xp_findnextmsg @msg_id = @msg_id output [, type]
               [, unread_only = {true | false}]
```

参数

msg_id — 需要输入，指定紧邻要检索的消息标识符的上一个消息标识符。将检索到的消息标识符放置于 **msg_id** 输出参数中，该参数必须为二进制类型。

type — 是基于 MAPI 邮件定义的输入消息类型。唯一支持的消息类型是 **CMC:IPM**。如果值为 **NULL** 或无值，将缺省设置为 **CMC:IPM**。

unread_only — 如果此参数设置为 **true**，**xp_findnextmsg** 只考虑未读消息。如果此参数设置为 **false**，则在检索下一个消息标识符时，**xp_findnextmsg** 将考虑所有消息（包括已读取的和未读取的）。缺省值是 **true**。

示例

1. **xp_findnextmsg @msg_id = @out_msg_id output**

通过 **@out_msg_id** 输出变量返回 **@out_msg_id** 所指定的消息后的下一个未读取消息的消息标识符。

2. **xp_findnextmsg @msg_id = @out_msg_id output, NULL, @unread_only = false**

通过 **@out_msg_id** 输出变量返回 **@out_msg_id** 所指定的消息的下一个消息的消息标识符。该消息可能已读取，也可能尚未读取。

注释

- 如果 **xp_findnextmsg** 在收件箱中找不到其它消息，将返回状态 1。
- **xp_deletemail** 和 **xp_readmail** 将使用 **xp_findnextmsg** 所返回的消息标识符。

权限

缺省情况下，只有系统管理员才能执行 **xp_findnextmsg**。系统管理员可以将此权限授予其他用户。

参见

系统 ESP	xp_deletemail、xp_readmail、xp_startmail
系统过程	sp_processmail

xp_logevent

(仅适于 Windows NT)

功能

用于将 Adaptive Server 中用户定义的事件记录在 Windows NT 事件日志中。

语法

`xp_logevent error_number, message [, type]`

参数

- error_number* — 是用户指派的错误编号。它必须大于或等于 50000。
- message* — 是显示在事件查看器的说明域中的消息的文本。消息的最大长度是 255 个字节。消息用引号引起。
- type* — 说明事件的紧急程度。其值可以是 **informational**、**warning** 和 **error**。缺省值为 **informational**。值用引号引起。

示例

1. `xp_logevent 55555, 'Email message deleted.'`
编号为 55555 的信息性事件将记录在 Windows NT 事件日志中。事件详细信息窗口中的说明文本为 “Email message deleted”。
2. `xp_logevent 66666, 'DLL not found.', 'error'`
编号为 66666 的错误事件将记录在 Windows NT 事件日志中。事件详细信息窗口中的说明文本为 “DLL not found”。

注释

- 下表说明 `xp_logevent` 所生成的事件的缺省事件详细信息：

详细信息	值
用户	N/A
计算机	运行 XP Server 的计算机的名称
事件 ID	12
源	Adaptive Server 的名称
种类	用户

权限

只有系统管理员才能执行 `xp_logevent`。

xp_readmail

(仅适于 Windows NT)

功能

从 Adaptive Server 收件箱中阅读消息。

语法

```
xp_readmail [msg_id]
    [, recipients output]
    [, sender output]
    [, date_received output]
    [, subject output]
    [, cc output]
    [, message output]
    [, attachments output]
    [, suppress_attach = {true | false}]
    [, peek = {true | false}]
    [, unread = {true | false}]
    [, msg_length output]
    [, bytes_to_skip [output]]
    [, type [output]]
```

参数

msg_id— 指定 **xp_readmail** 所读取的消息的消息标识符。如果未使用 **msg_id** 参数, 则在 **unread** 为 **true** 的情况下, 消息将缺省设置为收件箱中的第一个未读消息; 而在 **unread** 为 **false** 情况下, 消息将缺省设置为收件箱中的第一个消息。

recipients— 是以分号分隔的消息收件人的列表。

sender— 是消息的发送者。

date_received— 是接收消息的日期。

subject— 是消息的标题。

cc— 是要将该消息抄送 (cc'd) 给的收件人的列表 (以分号分隔)。

message— 是消息正文的文本。如果从 **msg_length** 输出参数所获得的消息正文的长度大于 255, 则使用 **byte_to_skip** 和 **msg_length** 参数按 255 个字节为单位来读取该消息。

attachments— 是附件的临时路径的列表 (以分号分隔)。如果 **suppress_attach** 为 **true**, 将忽略 **attachments**。

suppress_attach — 如果设置为 **true**，将阻止为附件创建临时文件。
缺省值是 **true**。

peek — 如果设置为 **false**，将在读取该消息后将其标记为“未读”。
如果设置为 **true**，将把该消息标记为未读消息，即使该消息已被读取。缺省值是 **false**。

unread_only — 如果设置为 **true**，**xp_readmail** 将只考虑未读消息。如果设置为 **false**，**xp_readmail** 将考虑所有消息，无论标记为“已读”还是“未读”。缺省值是 **true**。

msg_length — 是消息的总长度，以字节为单位。如果同 **bytes_to_skip** 参数一起使用，**xp_readmail** 可以按 255 个字节为单位来读取消息。

bytes_to_skip — 需要输入，如果非 0，则指定在将消息的下一批 255 个字节读入 **message** 输出参数前要跳过的字节数。如果是输出，包含消息中的偏移（**bytes_to_skip** 的先前值加上调用所输出的 **msg_length**），将从这个位置开始读取下一批 255 个字节。

type — 是基于 MAPI 邮件定义的消息类型。唯一支持的消息类型是 **CMC:IPM**。如果值为 NULL 或无值，将缺省设置为 **CMC:IPM**。

示例

```
1. declare @msgid binary(255)
   declare @originator varchar(20)
   declare @mess varchar(255)
   exec xp_findnextmsg @msg_id = @msgid output
   exec xp_readmail @msg_id = @msgid,
   @sender = @originator output,
   @message = @mess output
```

xp_readmail 读取收件箱中的第一个未读消息。它从 **@msgid** 变量获得该消息的消息标识符（消息标识符已由 **xp_findnextmsg** ESP 存储在该变量中）。**xp_readmail** 将发件人的名字存储在 **@originator** 变量中，将消息正文存储在 **@mess** 变量中。

```
2. declare @msgid binary(255)
   declare @mess varchar(255)
   declare @len int
   declare @skip int = 0
   exec xp_findnextmsg @msgid output
   exec xp_readmail @msg_id = @msgid,
   @message = @mess output
   @msg_length = @len output,
   @bytes_to_skip = @skip output
   print @mess
```

```
if (@len > 255)
begin
    while (@skip < @len)
    begin
        xp_readmail @msg_id = @msgid,
        @message = @mess output,
        @bytes_to skip = @skip output
        print @mess
    end
end
```

读取由 **xp_findnextmsg** 输出的消息标识符所对应的消息的前 255 个字符。如果该消息的总长度超过 255 个字节，则依次读取下一批 255 个字节，并以此类推直到读取了所有字节。

注释

- **xp_readmail** 从 Adaptive Server 收件箱读取消息。
- 要获得收件箱中下一个消息的消息标识符，请使用 **xp_findnextmsg**。

权限

缺省情况下，只有系统管理员才能执行 **xp_readmail**。系统管理员可以将此权限授予其他用户。

参见

系统 ESP	xp_deletemail、xp_findnextmsg、xp_sendmail、xp_startmail
系统过程	sp_processmail

xp_sendmail

(仅适于 Windows NT)

功能

将消息发送给指定收件人。消息可以是文本，也可以是 Transact-SQL 查询的结果。

语法

```
xp_sendmail recipient [; recipient] . . .  
    [, subject]  
    [, cc_recipient] . . .  
    [, bcc_recipient] . . .  
    [, {query | message}]  
    [, attachname]  
    [, attach_result = {true | false}]  
    [, echo_error = {true | false}]  
    [, include_file [, include_file] . . .]  
    [, no_column_header = {true | false}]  
    [, width]  
    [, separator]  
    [, dbuser]  
    [, dbname]  
    [, type]  
    [, include_query = {true | false}]
```

参数

recipient — 是要接收该消息的用户的电子邮件地址。要求至少有一个收件人。使用分号来分隔多个收件人。

subject — 是可选的消息标题。如不使用此参数，缺省情况下为“Sybase SQL Server Message”。

cc_recipient — 是要将消息抄送 (cc'd) 给的收件人的列表（以分号分隔）。

bcc_recipient — 是要将消息密送 (bcc'd) 给的收件人的列表（以分号分隔）。

query — 是一个或多个 Transact-SQL 语句。其结果将发送给消息收件人。如果使用 *query*，则不能使用 *message*。

message — 是所发送的消息的文本。如果使用 *message*，则不能使用 *query*。

attachname — 是包含查询结果的文件的名称，如果使用 **query** 参数，该文件将作为消息的附件包括在消息中。如果使用 **attachname**，则必须将 **attach_result** 设置为 **true**。如果 **attach_result** 为 **true** 而未指定 **attachname**，则所附文件的生成文件名的前缀为 **syb**，然后是 5 个随机数字，之后的扩展名为 “.txt”，例如 **syb84840.txt**。如果使用 **message** 参数，将忽略此参数。

attach_result — 如果设置为 **true**，则将查询的结果以附件形式随消息发送。如果设置为 **false**，将直接在消息正文中发送结果。缺省值是 **false**。如果使用 **message** 参数，将忽略此参数。

echo_error — 如果设置为 **true**，将把消息（包括相关消息的行数）随 Adaptive Server 查询结果一并发送。如果设置为 **false**，则不发送 Adaptive Server 消息。缺省值是 **true**。如果使用 **message** 参数，将忽略此参数。

include_file — 是要作为消息附件包括的文件的列表，以分号分隔。这些文件可以通过文件名、路径名或相对路径名来指定，它们可以是文本文件也可以是二进制文件。

no_column_header — 如果设置为 **true**，则列标题将同查询结果一起发送。如果设置为 **false**，将不发送列标题。缺省值是 **false**。如果使用 **message** 参数，将忽略此参数。

no_output — 如果设置为 **true**，则不向发送该邮件的会话发送任何输出。如果设置为 **false**，发送该邮件的会话将接收输出。缺省值是 **false**。如果使用 **message** 参数，将忽略此参数。

width — 指定通过消息发送查询结果时结果集的宽度（以字符为单位）。**width** 与 **isql** 中的 **lw** 选项相同。如果达到指定的 **width**，结果行将由换行符进行折行。缺省值为 80 个字符。如果使用 **message** 参数，将忽略此参数。

separator — 指定通过消息发送查询结果时要用作列分隔符的字符。**separator** 与 **isql** 中的 **ls** 选项相同。缺省值为制表符。如果使用 **message** 参数，将忽略此参数。

dbuser — 指定数据库用户名，在使用 **query** 参数的情况下，会假定该用户名作为用户环境，以便执行查询。缺省值为 “**guest**”。如果使用 **message** 参数，将忽略此参数。

dname — 指定数据库名称，在使用 **query** 参数的情况下，会假定该名称作为数据库环境，以便执行查询。缺省值是 “**master**”。如果使用 **message** 参数，将忽略此参数。

type — 是基于 MAPI 邮件定义的输入消息类型。唯一支持的消息类型是 **CMC:IPM**。如果值为 **NULL** 或无值，将缺省设置为 **CMC:IPM**。

include_query— 如果设置为 **true**，**query** 参数中使用的查询将被添加到结果集。如果设置为 **false**，则不添加查询。缺省值为 **false**。如果使用 **message** 参数，将忽略 **include_query**。

示例

```
1. xp_sendmail @recipient = "sally";"ramon",
   @subject = "Adaptive Server Backup Status",
   @message = "Adaptive Server Backup for SERVER2 is
   complete.",
   @copy_recipient="admin"
```

xp_sendmail 将关于 Adaptive Server 备份状态的文本消息发送到 “sally” 和 “ramon”，并将副本发送到 “admin” 组。

```
2. xp_sendmail "peter",
   @query = "select * from authors",
   @attachname = "au_list.res",
   @attach_result= true
```

将对于 *authors* 表的查询结果发送给 “peter”。该查询结果位于消息的附件中，该附件包含名为 *au_lis.res* 的文件，这个文件所在的目录就是执行服务器的目录。

注释

- 如果使用 **query** 参数，以下参数与通过消息发送的查询结果相关联。如果使用的是 **message** 参数，它们将被忽略。**attachname**、**attach_result**、**echo_error**、**no_column_header**、**no_output**、**width**、**separator**、**dbuser**、**dname** 和 **include_query**。

权限

缺省情况下，只有系统管理员才能执行 **xp_sendmail**。系统管理员可以将此权限授予其他用户。

参见

系统 ESP	xp_deletemail、xp_findnextmsg、xp_readmail、xp_startmail
系统过程	sp_processmail
实用程序	isql

xp_startmail

(仅适于 Windows NT)

功能

启动 Adaptive Server 邮件会话。

语法

```
xp_startmail [mail_user] [, mail_password]
```

参数

mail_user — 是 Adaptive Server 用于登录到 Windows NT 邮件系统中的邮件配置文件名称。如果不使用 *mail_user*, **xp_startmail** 将使用曾用于设置 SybmailAdaptive Server 帐户的邮件用户名。

mail_password — 是 Adaptive Server 用于登录到 Windows NT 邮件系统中的邮件口令。如果不使用 *mail_password*, **xp_startmail** 将使用曾用于设置 SybmailAdaptive Server 帐户的邮件口令。

示例

1. xp_startmail

使用 Sybmail 用户帐户的邮件用户名和口令启动 Adaptive Server 邮件会话。

2. xp_startmail "mailuser", "tre55uu"

使用配置文件名 “mailuser” 及其相关的口令来启动 Adaptive Server 邮件会话。

注释

- 如果一个 Adaptive Server 邮件会话已在运行, **xp_startmail** 将不会再启动另一个会话。
- 要执行任何与 Sybmail 相关的系统 ESP 或 **sp_processmail** 存储过程以前, 必须先启动 Adaptive Server 邮件会话, 方法是明确调用 **xp_startmail** 或配置 Adaptive Server 从而在启动时自动启动 Adaptive Server 邮件会话。有关在启动时自动开始 Adaptive Server 邮件会话的信息, 参见 *系统管理指南* 中的 **start mail session**。
- 如果 Windows NT **automail** 会话未启动, 必须将 *mail_user* 和 *mail_password* 参数与 **xp_startmail** 一同使用。

- 要从 *fullname* 域查看 Sybmail 用户帐户的缺省 *mail_user* 值，按如下方式使用 **sp_displaylogin** 系统过程：
sp_displaylogin sybmail

权限

缺省情况下，只有系统管理员才能执行 **xp_startmail**。系统管理员可以将此权限授予其他用户。

参见

系统 ESP	xp_stopmail
系统过程	sp_configure

xp_stopmail

(仅适于 Windows NT)

功能

停止 Adaptive Server 邮件会话。

语法

`xp_stopmail`

参数

无。

示例

1. `xp_stopmail`

停止 Adaptive Server 邮件会话。

注释

- 在使用 `xp_stopmail` 终止 Adaptive Server 邮件会话后，将无法执行与 Sybmail 相关的系统 ESP 和 `sp_processmail` 存储过程。

权限

缺省情况下，只有系统管理员才能执行 `xp_stopmail`。系统管理员可以将此权限授予其他用户。

参见

系统 ESP	<code>xp_startmail</code>
系统过程	<code>sp_processmail</code>

10

dbcc 存储过程

本章描述 **dbcc** 存储过程。这些过程仅访问 **dbccdb** 数据库或替代数据库 **dbccalt** 中的表。有关设置 **dbccdb** 或 **dbccalt** 的详细信息，参见 *系统管理指南*。有关这些数据库中所使用的表的信息，参见 *Adaptive Server Enterprise 参考手册* 中的第 12 章 “**dbccdb** 表”。

表 10-1 列出本章说明的 **dbcc** 存储过程。有关 **dbcc** 系统过程 **sp_plan_dbccdb** 的详细信息，参见 **sp_plan_dbccdb**。有关该系统过程和 **dbcc** 存储过程的详细信息，参见 *系统管理指南*。

表 10-1: dbcc 存储过程

过程名称	说明
sp_dbcc_alterws	将指定工作空间的大小更改为指定值，并初始化该工作空间。
sp_dbcc_configreport	生成说明指定数据库的 dbcc checkstorage 操作所使用的配置信息的报告。
sp_dbcc_createws	在指定段与数据库上创建指定类型与大小的工作空间。
sp_dbcc_deletedb	从 dbccdb 删除与指定目标数据库相关的所有信息。
sp_dbcc_deletehistory	删除在指定日期和时间前在目标数据库上执行 dbcc checkstorage 操作的结果。
sp_dbcc_differentialreport	生成突出显示在两个 dbcc 操作之间进行的 I/O 统计信息更改和故障报告。
sp_dbcc_evaluatedb	重新计算目标数据库的配置信息并将其与当前的配置信息进行比较。
sp_dbcc_faultreport	生成涉及 dbcc checkstorage 操作的故障统计信息报告，这些操作是在指定的日期对目标数据库上的特定对象执行的。
sp_dbcc_fullreport	在指定的 <i>date</i> 或在此之前为 <i>database..object_name</i> 运行 sp_dbcc_summaryreport 、 sp_dbcc_configreport 、 sp_dbcc_statisticsreport 和 sp_dbcc_faultreportshort 。
sp_dbcc_runcheck	在指定的数据库上运行 dbcc checkstorage ，然后运行 sp_dbcc_summaryreport 或者指定的报告。
sp_dbcc_statisticsreport	生成关于目标数据库中的指定对象的分配统计信息报告。

表 10-1: dbcc 存储过程 (续)

过程名称	说明
sp_dbcc_summaryreport	生成指定数据库的总结报告。
sp_dbcc_updateconfig	用目标数据库的配置信息更新 <i>dbccdb</i> 中的 <i>dbcc_config</i> 表。

指定对象名和日期

有几个 **dbcc** 存储过程使用参数来表示对象名和日期。本节提供有关指定对象名和日期的重要信息。

指定对象名

对象名只指定用于生成报告的表或索引的名称。当指定对象名时，还必须指定数据库名 (*dbname*)。您不能指定对象的所有者。如果指定的对象名在目标数据库中不是唯一的，则系统过程会对所有具有指定名称的系统过程生成报告。

指定日期

使用以下语法来指定日期和时间 (可选)：

mm/dd/yy[:hh:mm:ss]

采用 24 小时计时制。

指定日期时，系统过程将对其做如下解释：

- 如果日期和时间都已指定，则为报告选择在指定的日期和时间完成的 **dbcc** 操作。
- 如果指定日期是当前日期，但未指定时间，则时间将自动设置为当前时间。为报告选择在过去 24 小时内完成的、结束时间离当前时间最近的 **dbcc** 操作。
- 如果指定的日期不是当前日期，并且没有指定时间，则时间就自动设置为 “23:59:59”。为报告选择完成操作的结束日期和时间离指定的日期和系统提供的时间最近的 **dbcc checkstorage** 操作。

例如，假设最近的 **dbcc checkstorage** 操作在 1997 年 3 月 4 日的 10:20:45 完成。

如果将日期指定为 “**03/04/97**”，则系统过程就将该日期解释为 **03/04/97:23:59:59**。该日期和时间与实际的结束日期和时间 **03/04/97:10:20:45** 进行比较。

如果将日期指定为 “**03/04/97:10:00:00**”，则不会为报告选择在 **10:20:45** 完成的操作，因为只有在指定时间或在此之前完成的操作才符合标准。

如果将日期指定为 “**03/06/97**”，则不生成报告，因为最近完成操作的时间与该时间相比超过了 **24** 小时。

sp_dbcc_alterws

功能

将指定工作空间的大小更改为指定值，并初始化该工作空间。

语法

```
sp_dbcc_alterws dbname, wsname, "wssize[K|M]"
```

参数

dbname — 工作空间所在的数据库名称。指定 **dbccdb** 或 **dbccalt**。

wsname — 指定要变更的工作空间的名称。

wssize — 新的工作空间大小，用 **K**（千字节）或 **M**（兆字节）来指定。如果没有指定 **K** 或 **M**，**wssize** 就指定页数。页大小视平台而定。工作空间的最小大小是 24 页。

示例

```
1. sp_dbcc_alterws dbccdb, scan_ws_000001, "30M"
```

```
Workspace scan_ws_000001 has been altered  
successfully to size 30MB
```

将 **dbccdb** 中 **scan_ws_000001** 工作空间的大小更改为 30MB。

注释

- **sp_dbcc_alterws** 将指定工作空间的大小更改为指定值，并初始化该工作空间。
- 要获得最佳性能，确保在变更工作空间前已经配置了至少为 16K 的缓冲池。
- 在变更工作空间前，使用 **sp_plan_dbccdb** 确定大小估计值。
- 工作空间必须存在才可以进行变更。有关创建工作空间的信息，参见 **sp_dbcc_createws**。
- 要删除工作空间，在 **dbccdb** 发出：

```
drop table workspace_name
```
- 有关 **scan** 和 **text** 工作空间以及 **dbccalt** 数据库的信息，参见 *系统管理指南*。

权限

只有系统管理员或数据库所有者才能运行 **sp_dbcc_alterws**。

使用的表

master..sysdatabases、 syssegments、 sysobjects

参见

命令	dbcc
dbcc 存储过程	sp_dbcc_createws、 sp_dbcc_evaluatedb
系统过程	sp_plan_dbccdb、 sp_help、 sp_helpdb

sp_dbcc_configreport

功能

生成说明指定数据库的 **dbcc checkstorage** 操作所使用的配置信息的报告。

语法

```
sp_dbcc_configreport [dbname]
```

参数

dbname — 指定数据库的名称。如果没有指定 *dbname*，则报告将所有数据库的信息包含在 *dbccdb..dbcc_operation_log* 中。

示例

1. sp_dbcc_configreport

Reporting configuration information of database sybsystemprocs.

Parameter Name	Value	Size
database name	sybsystemprocs	51200K
dbcc named cache	default data cache	1024K
text workspace	textws_001 (id = 544004969)	128K
scan workspace	scanws_001 (id = 512004855)	1024K
max worker processes	1	
operation sequence number	2	

为 *sybsystemprocs* 数据库生成与 **dbcc** 有关的配置信息报告。在适用情况下，“Value”列列出对象名，并列出大小。

注释

- **sp_dbcc_configreport** 生成报告，用来说明指定数据库的 **dbcc** 操作所使用的配置信息。这些信息将存储在 *dbcc_config* 表中。
- 要计算最新的配置参数值，先运行 **sp_dbcc_updateconfig**，然后运行 **sp_dbcc_configreport**。
- 要更改工作空间的配置值，请使用 **sp_dbcc_alterws**。

权限

任何用户都可以运行 **sp_dbcc_configreport**。

使用的表

*master..sysdatabases、 dbccdb..dbcc_operation_log、
dbccdb..dbcc_operation_results、 dbccdb..dbcc_config*

参见

命令	dbcc
dbcc 存储过程	sp_dbcc_alterws、 sp_dbcc_fullreport、 sp_dbcc_statisticsreport、 sp_dbcc_summaryreport、 sp_dbcc_updateconfig

sp_dbcc_createws

功能

在指定段和数据库上创建指定类型、指定大小的工作空间。

语法

```
sp_dbcc_createws dbname, segname, [wsname], wstype,  
"wssize[K|M]"
```

参数

dbname — 要在其中创建工作空间的数据库的名称。值为 *dbccdb* 和 *dbccalt*。

segname — 工作空间的段名。

wsname — 是工作空间名。如果值为空，**sp_dbcc_createws** 就分别为 *scan* 工作空间和 *text* 工作空间生成名称 *scan_ws_nnnnnnn* 和 *text_ws_nnnnnnn*，其中 *nnnnnnn* 是唯一的六位数字。

wstype — 指定要创建的工作空间的类型。值为 *scan* 和 *text*。

wssize — 是工作空间大小，用 **K**（千字节）或 **M**（兆字节）指定。如果没有指定 **K** 或 **M**，*wssize* 就指定页数。工作空间的最小大小是 24 页。

示例

```
1. sp_dbcc_createws dbccdb, scanseg, scan_ws_pubs2,  
scan, "10M"
```

在 *dbccdb* 中的 *scanseg* 段上创建一个名为 *scan_ws_pubs2* 的、大小为 10MB 的 *scan* 工作空间。

```
2. sp_dbcc_createws dbccdb, textseg, text, "14M"
```

在 *dbccdb* 中的 *textseg* 段上创建一个名为 *text_ws_000001* 的、大小为 14MB 的 *scan* 工作空间。

注释

- **sp_dbcc_createws** 创建指定名称和大小的工作空间并对其进行初始化。
- 在创建工作空间前，用 **sp_addsegment** 创建段。
- 在创建工作空间前，确保配置至少 16K 的缓冲池以便获得最佳性能。

- 使用 **sp_plan_dbccdb** 确定大小估计值。
- 在创建工作空间后，运行 **sp_dbcc_updateconfig** 将新的配置信息记录在 *dbcc_config* 中。
- 每个工作空间必须有一个唯一名称。
- 要删除工作空间，在 *dbccdb* 发出：
drop table workspace_name
- 有关 *scan* 和 *text* 工作空间的详细信息，参见 *系统管理指南*。
- 有关 *dbccalt* 数据库的信息，参见 *系统管理指南*。

权限

只有系统管理员或数据库所有者才能够运行 **sp_dbcc_createws**。

使用的表

master..sysdatabases、*syssegments*、*sysobjects*

参见

命令	dbcc
dbcc 存储过程	sp_dbcc_alterws、sp_dbcc_evaluatedb
系统过程	sp_addsegment、sp_plan_dbccdb、sp_help、sp_helpsegment

sp_dbcc_deletedb

功能

从 *dbccdb* 删除与指定的目标数据库相关的所有信息。

语法

```
sp_dbcc_deletedb [dbname]
```

参数

dbname — 指定要删除配置信息的目标数据库的名称。如果没有指定 *dbname* 的值，Adaptive Server 删除 *dbccdb..dbcc_config* 中所有数据库的数据。如果目标数据库是 *dbccdb*，并且 *dbccalt* 存在，Adaptive Server 将删除 *dbccalt* 的数据。

示例

1. sp_dbcc_deletedb "engdb"

```
All information for database engdb has been deleted
from dbccdb.
```

从 *dbccdb* 删除 *engdb* 数据库的所有信息。

注释

- **sp_dbcc_deletedb** 从 *dbccdb* 删除与指定目标数据库有关的所有信息，包括配置信息和先前 **dbcc checkstorage** 操作的结果。
- 如果删除的数据库是 *dbccdb*，并且 *dbccalt* 数据库存在，**sp_dbcc_deletedb** 从 *dbccalt* 删除 *dbccdb* 的配置信息和结果。
- 要删除在指定日期前创建的 **dbcc checkstorage** 操作的结果，使用 **sp_dbcc_deletehistory**。
- 有关 *dbccalt* 数据库的信息，参见 *系统管理指南*。

权限

只有系统管理员或数据库所有者才能够运行 **sp_dbcc_deletedb**。

使用的表

master..sysdatabases、*dbccdb..dbcc_config*、
dbccdb..dbcc_operation_log、*dbccdb..dbcc_operation_results*、
dbccdb..dbcc_counters、*dbccdb..dbcc_faults*、
dbccdb..dbcc_fault_params

参见

命令	dbcc
dbcc 存储过程	sp_dbcc_deletehistory、 sp_dbcc_evaluatedb
系统过程	sp_plan_dbccdb

sp_dbcc_deletehistory

功能

删除在指定日期和时间前在目标数据库上执行 **dbcc checkstorage** 操作的结果。

语法

```
sp_dbcc_deletehistory [cutoffdate [, dbname]]
```

参数

cutoffdate — 删除在该日期或在此之前的日期创建的所有条目。该参数属于 *datetime* 类型。如果没有指定日期，则只保留最后一个操作的结果。有关详细信息，参见第 10-2 页的“指定日期”。

dbname — 指定必须删除其数据的数据库名称。如果未指定，**sp_dbcc_deletehistory** 删除 *dbccdb..dbcc_config* 中的所有数据库的历史记录信息。

示例

```
1. sp_dbcc_deletehistory "03/04/1997", "pubs2"
```

删除在 1997 年 3 月 4 日或在此之前在 *pubs2* 数据库上执行的所有操作的结果。

注释

- **sp_dbcc_deletehistory** 删除在指定日期和时间前在目标数据库上执行 **dbcc checkstorage** 操作的结果。
- 如果目标数据库是 *dbccdb*，并且 *dbccalt* 数据库存在，**sp_dbcc_deletehistory** 从 *dbccalt* 删除 *dbccdb* 的历史数据。
- 为 **cutoffdate** 指定的值将与每个 **dbcc** 操作的结束时间进行比较。
- 要查看运行 **dbcc checkstorage** 的日期以便选择 **cutoffdate** 的值，运行 **sp_dbcc_summaryreport**。
- 有关 *dbccalt* 数据库的信息，参见 *系统管理指南*。

权限

只有系统管理员或数据库所有者才能够在指定的数据库上运行 **sp_dbcc_deletehistory**。只有系统管理员无需指定数据库名就能够运行 **sp_dbcc_deletehistory**。

使用的表

*master..sysdatabases、 master..sysdevices、 master..sysusages、
dbccdb..dbcc_operation_log、 dbccdb..dbcc_operation_results、
dbccdb..dbcc_counters、 dbccdb..dbcc_faults、
dbccdb..dbcc_fault_params*

参见

命令	dbcc
dbcc 存储过程	sp_dbcc_deletedb、 sp_dbcc_evaluatedb
系统过程	sp_plan_dbccdb

sp_dbcc_differentialreport

功能

生成重点突出两个 **dbcc** 操作之间 I/O 统计信息的更改和发生的故障的报告。

语法

```
sp_dbcc_differentialreport [dbname [, objectname]],  
    [db_op] [, "date1" [, "date2"]]
```

参数

dbname — 指定数据库的名称。如果没有指定 **dbname**，则报告将所有数据库的信息包含在 **dbccdb..dbcc_operation_log** 中。

objectname — 指定要为其生成报告的表或索引的名称。如果没有指定 **object_name**，则报告目标数据库中所有对象的统计信息。

db_op — 指定报告要使用的数据源。仅有的值是 **checkstorage**。根据 **db_op** 指定的数据在 **date1** 和 **date2** 为目标数据库中的指定对象生成报告。如果没有指定日期，则比较 **db_op** 类型的最后两个操作。

date1 — 指定要比较的 **dbcc checkstorage** 操作的第一个日期。

date2 — 指定要比较的 **dbcc checkstorage** 操作的最后一个日期。

示例

```
1. sp_dbcc_differentialreport master, sysprocedures,  
    checkstorage, "05/01/97", "05/04/97"
```

生成报告，说明在 1997 年 5 月 1 日到 5 月 4 日之间在 **sysprocedures** 表中 I/O 统计信息的更改和出现的故障。

注释

- **sp_dbcc_differentialreport** 生成报告，重点突出在两次 **dbcc** 操作之间 I/O 统计信息的更改和发生故障的报告。它对从 **dbcc checkstorage** 的两个实例报告的计数器值进行比较。只有已被更改的值才会报告。
- 如果只指定了一个日期，则将指定日期选定的 **dbcc checkstorage** 操作的结果与在进行所选操作前的 **dbcc checkstorage** 操作结果进行比较。
- 如果没有指定日期，则比较最后两个 **dbcc checkstorage** 操作的结果。

- 如果 `sp_dbcc_differentialreport` 为 *object_name* 返回一个数字，它表示在完成 `dbcc checkstorage` 操作后将对象删除。
- 如果在两个指定的操作之间没有发生任何更改，`sp_dbcc_differentialreport` 就不生成报告。

权限

任何用户都可以运行 `sp_dbcc_differentialreport`。

使用的表

master..sysdatabases、*dbccdb..dbcc_operation_log*、*dbccdb..dbcc_operation_results*、*dbccdb..dbcc_counters*

参见

命令	dbcc
dbcc 存储过程	sp_dbcc_fullreport、sp_dbcc_statisticsreport、sp_dbcc_summaryreport、sp_dbcc_updateconfig

sp_dbcc_evaluatedb

功能

重新计算目标数据库的配置信息并将之与当前的配置信息进行比较。

语法

```
sp_dbcc_evaluatedb [dbname]
```

参数

dbname — 指定目标数据库的名称。如果没有指定 *dbname*, **sp_dbcc_evaluatedb** 就比较 *dbcc_config* 表中列出的所有数据库。

示例

1. sp_dbcc_evaluatedb

Recommended values for workspace size, cache size and worker process count are:

```
Database name : sybsystemprocs
current scan workspace size : 400K          suggested scan workspace
size : 272K
current text workspace size : 208K          suggested text workspace
size : 208K
current cache size : 1024K          suggested cache size : 640K
current process count : 1          suggested process count : 1
```

重新计算当前数据库 *sybsystemprocs* 的配置信息，并为某些参数建议新值。

注释

- **sp_dbcc_evaluatedb** 重新计算目标数据库的配置信息并将该数据与当前的配置信息进行比较。它使用在 *dbcc_counters* 表中记录的目标数据库的计数器值。
- 缓存大小为缓存中 6K 缓冲池的大小。对于 2K 缓冲池，该缓存的最小大小必须是推荐值加上 512。
- 当目标数据库的大小和数据分布模式变化时，运行 **sp_dbcc_evaluatedb** 以优化配置信息。
- 第一次收集目标数据库的配置信息时，使用 **sp_plan_dbccdb**。
- 要确保求出最新的配置参数值，先运行 **sp_dbcc_updateconfig** 然后运行 **sp_dbcc_evaluatedb**。

权限

只有系统管理员或数据库所有者才能够运行 **sp_dbcc_evaluatedb**。只有系统管理员无需指定数据库名就能够运行 **sp_dbcc_evaluatedb**。

使用的表

master..sysdatabases、 *master..sysdevices*、 *master..sysusages*、
dbccdb..dbcc_counters、 *dbccdb..dbcc_config*

参见

命令	dbcc
dbcc 存储过程	sp_dbcc_updateconfig
系统过程	sp_plan_dbccdb

sp_dbcc_faultreport

功能

生成在指定的日期针对目标数据库上的特定对象执行的 **dbcc checkstorage** 操作的故障统计信息报告。

语法

```
sp_dbcc_faultreport [report_type [, dbname  
                    [, objectname [, date ]]]]
```

参数

report_type — 指定故障报告的类型。有效值为 **short** 和 **long**。缺省值为 **short**。

dbname — 指定目标数据库的名称；例如， **master.sysdatabases**。
如果没有指定 **dbname**，则报告将所有数据库的信息包含在 **dbccdb..dbcc_operation_log** 中。

object_name — 指定要为其生成报告的表或索引的名称。如果没有指定 **object_name**，则报告目标数据库中所有对象的统计信息。

date — 指定 **dbcc checkstorage** 操作结束的精确日期和时间。可以在 **dbcc_operation_log.finish** 中找到该值。也可以创建该值，方法是在 **sp_dbcc_summaryreport** 输出中将 **start time** 的日期和 **end time** 的小时数和分钟数进行组合。如果没有指定 **date**， **Adaptive Server** 就使用最近操作的日期。

示例

1. sp_dbcc_faultreport "short"

Database Name : sybsystemprocs

Table Name	Index	Type	Code	Description	Page Number
sysprocedures	0	100031	page not allocated		5702
sysprocedures	1	100031	page not allocated		14151
syslogs	0	100022	chain start error		24315
syslogs	0	100031	page not allocated		24315

生成在 **sybsystemprocs** 数据库的表中发现故障的短报告。报告包括表名、故障发生所在的索引号、故障类型代码、故障简要说明和故障所在的页号。

2. sp_dbcc_faultreport "long"

```
Generating 'Fault Report' for object sysprocedures in database
sybsystemprocs.

Type Code: 100031; Soft fault, possibly spurious
Page reached by the chain is not allocated.
page id: 14151
page header: 0x00003747000037880000374600000005000648B803EF0001000103FE
0080000F
Header for 14151, next 14216, previous 14150, id = 5:1
time stamp = 0x0001000648B8, next row = 1007, level = 0
free offset = 1022, minlen = 15, status = 128(0x0080)
.
.
.
```

生成在 *sybsystemprocs* 数据库的表中发现故障的长报告。此例显示长报告输出的第一部分。完整的报告则为 **dbcc checkstorage** 在其中发现故障的**目标数据库**中的每个对象都输出此信息。在 “page header” 域下显示的长字符串数字后的数据 (“Header for 14151, next 14216, previous 14150 ...”) 描述 “page header” 字符串的组成部分。

注释

- **sp_dbcc_faultreport** 生成说明目标数据库中指定对象的所有故障的报告。
- 如果 **sp_dbcc_faultreport** 为 *object_name* 返回一个数字，则表示在完成 **dbcc checkstorage** 操作后将对象删除。
- 有关故障 ID 的信息，参见在*系统管理指南*中说明的 *type_code* 列。
- 有关故障状态的信息，参见 *系统管理指南*。

权限

任何用户都可以运行 **sp_dbcc_faultreport**。

使用的表

master..sysdatabases、*dbccdb..dbcc_operation_log*、
dbccdb..dbcc_operation_results、*dbccdb..dbcc_faults*、
dbccdb..dbcc_fault_params

参见

命令	dbcc
dbcc 存储过程	sp_dbcc_fullreport、sp_dbcc_statisticsreport、 sp_dbcc_summaryreport、sp_dbcc_updateconfig

sp_dbcc_fullreport

功能

在指定的 *date* 或在此之前为 *database..object_name* 运行 **sp_dbcc_summaryreport**、**sp_dbcc_configreport**、**sp_dbcc_statisticsreport** 和 **sp_dbcc_faultreport short**。

语法

```
sp_dbcc_fullreport [dbname [, objectname [, date]]]
```

参数

dbname — 指定数据库的名称。如果没有指定 *dbname*，则报告将所有数据库的信息包含在 *dbccdb..dbcc_operation_log* 中。

object_name — 指定要为其生成报告的表或索引的名称。如果没有指定 *object_name*，则报告目标数据库中所有对象的统计信息。

date — 指定执行 **dbcc checkstorage** 操作的日期。如果没有指定 *date*，则使用最后一个操作的日期。

示例

1. **sp_dbcc_fullreport master, sysprocedures**

为运行在 *master* 数据库的 *sysprocedures* 表中最近的 **dbcc checkstorage** 操作运行 **sp_dbcc_summaryreport**、**sp_dbcc_configreport**、**sp_dbcc_statisticsreport** 和 **sp_dbcc_faultreport short**。

注释

- **sp_dbcc_fullreport** 在指定日期或在此之前的日期为指定的数据库对象运行 **sp_dbcc_summaryreport**、**sp_dbcc_configreport**、**sp_dbcc_statisticsreport** 和 **sp_dbcc_faultreport short**。

权限

任何用户都可以运行 **sp_dbcc_fullreport**。

使用的表

master.sysdatabases、*dbccdb..dbcc_operation_log*、*dbccdb..dbcc_operation_results*、*dbccdb..dbcc_faults*、*dbccdb..dbcc_fault_params*、*dbccdb..dbcc_counters*

参见

命令	dbcc
dbcc 存储过程	sp_dbcc_statisticsreport、 sp_dbcc_summaryreport、sp_dbcc_updateconfig

sp_dbcc_runcheck

功能

在指定数据库上运行 **dbcc checkstorage**，然后运行 **sp_dbcc_summaryreport** 或者指定的报告。

语法

```
sp_dbcc_runcheck dbname [, user_proc]
```

参数

dbname — 指定要在其中执行检查的数据库名称。

user_proc — 指定 **dbcc** 存储过程或用户创建的替代 **sp_dbcc_summaryreport** 运行的存储过程的名称。

示例

```
1. sp_dbcc_runcheck "engdb"
```

检查数据库 *engdb* 并根据所找到的信息生成总结报告。

```
2. sp_dbcc_runcheck "pubs2", sp_dbcc_fullreport
```

检查数据库 *pubs2* 并生成完整的报告。

注释

- **sp_dbcc_runcheck** 在指定的数据库上运行 **dbcc checkstorage**。
- 在完成 **dbcc checkstorage** 操作后，**sp_dbcc_runcheck** 运行 **sp_dbcc_summaryreport** 以生成总结报告。如果为 *dbcc_report* 指定其它生成报告的 **dbcc** 存储过程，则 **sp_dbcc_runcheck** 运行该存储过程而不运行 **sp_dbcc_summaryreport**。有关 *dbccdb* 提供的所有生成报告的存储过程的简要说明和示例，参见 *系统管理指南*。
- 可以自己编写生成报告的存储过程并为 *user_proc* 指定它的名称。该存储过程必须是独立的。**sp_dbcc_runcheck** 不能将任何参数传递给 Adaptive Server。

权限

只有系统管理员或数据库所有者才能够运行 **sp_dbcc_runcheck**。

使用的表

master..sysdatabases、*dbccdb..dbcc_config*、*dbccdb..dbcc_counters*、*dbccdb..dbcc_fault_params*、*dbccdb..dbcc_faults*、*dbccdb..dbcc_operation_log*、*dbccdb..dbcc_operation_results*

参见

命令	dbcc
dbcc 存储过程	sp_dbcc_summaryreport

sp_dbcc_statisticsreport

功能

生成关于目标数据库中的指定对象的分配统计信息报告。

语法

```
sp_dbcc_statisticsreport [dbname [, objectname  
                        [, date]]]
```

参数

- dbname* — 指定目标数据库。如果没有指定 *dbname*，则报告将所有数据库的信息包含在 *dbccdb..dbcc_operation_log* 中。
- objectname* — 指定要为其生成报告的表或索引的名称。如果没有指定 *object_name*，Adaptive Server 报告目标数据库中所有对象的统计信息。
- date* — 指定执行 *dbcc checkstorage* 操作的日期。如果没有指定 *date*，Adaptive Server 就使用最近操作的日期。

示例

```
1. sp_dbcc_statisticsreport 'sybsystemprocs',  
   'sysobjects'
```

Statistics Report on object sysobjects in database sybsystemprocs

Parameter Name	Index Id	Value
count	0	241.0
max size	0	99.0
max count	0	22.0
bytes data	0	19180.0
bytes used	0	22113.0
count	1	14.0
max size	1	9.0
max level	1	0.0
max count	1	14.0
bytes data	1	56.0
bytes used	1	158.0
count	2	245.0
max level	2	1.0
max size	2	39.0
max count	2	71.0
bytes data	2	4377.0
bytes used	2	6995.0

Parameter Name	Index Id	Partition	Value	Dev_name
-----	-----	-----	-----	-----
page gaps	0	1	13.0	master
pages used	0	1	15.0	master
extents used	0	1	3.0	master
overflow pages	0	1	0.0	master
pages overhead	0	1	1.0	master
pages reserved	0	1	7.0	master
page extent gaps	0	1	11.0	master
ws buffer crosses	0	1	2.0	master
page extent crosses	0	1	11.0	master
pages used	1	1	2.0	master
extents used	1	1	1.0	master
overflow pages	1	1	0.0	master
pages overhead	1	1	1.0	master
pages reserved	1	1	6.0	master
page extent gaps	1	1	0.0	master
ws buffer crosses	1	1	0.0	master
page extent crosses	1	1	0.0	master
page gaps	2	1	4.0	master
pages used	2	1	6.0	master
extents used	2	1	1.0	master
overflow pages	2	1	0.0	master
pages overhead	2	1	1.0	master
pages reserved	2	1	2.0	master
page extent gaps	2	1	0.0	master
ws buffer crosses	2	1	0.0	master
page extent crosses	2	1	0.0	master

在 *master* 数据库的 *sysobjects* 表中生成统计信息报告。

注释

- **sp_dbcc_statisticsreport** 生成关于目标数据库中指定对象的分配统计报告。它使用来自 *dbcc_counters* 表的数据，该表存储着目标数据库中的每个对象页使用情况信息和错误统计信息。
- 如果 **sp_dbcc_statisticsreport** 为 *object_name* 返回一个数字，它表示在完成 **dbcc checkstorage** 操作后已经将对象删除。
- **sp_dbcc_statisticsreport** 报告记录在 *dbcc_counters* 表中数据类型为 5000 到 5019 的值。参见 *系统管理指南*。

对于 *bytes data*、*bytes used* 和 *overflow pages*，**sp_dbcc_statisticsreport** 报告所有分区和设备的值的总和。

对于 *count*、*max count*、*max size* 和 *max level*，**sp_dbcc_statisticsreport** 报告所有分区和设备的最大值。

sp_dbcc_statisticsreport 报告目标数据库中对象所使用每个设备和分区的信息，包括：

- *extents used*
- *io errors*
- *page gaps*
- *page extent crosses*
- *page extent gaps*
- *page format errors*
- *pages reserved*
- *pages overhead*
- *pages misallocated*
- *pages not allocated*
- *pages not referenced*
- *pages used*

page gaps、*page extent crosses* 和 *page extent gaps* 表明对象的数据页在数据库设备中的分布方式。如果值很大，则说明在使用较大的缓存大小和数据预取方面效率较低。

- 如果同一天在目标数据库上运行多个 **dbcc checkstorage** 操作，**sp_dbcc_statisticsreport** 生成报告将基于最后一个在指定时间前完成的 **dbcc checkstorage** 操作的结果。

权限

任何用户都可以运行 **sp_dbcc_statisticsreport**。

使用的表

master..sysdatabases、*dbccdb..dbcc_operation_log*、
dbccdb..dbcc_operation_results、*dbccdb..dbcc_counters*

参见

命令	dbcc
dbcc 存储过程	sp_dbcc_fullreport、sp_dbcc_summaryreport、sp_dbcc_updateconfig

sp_dbcc_summaryreport

功能

生成指定数据库的总结报告。

语法

```
sp_dbcc_summaryreport [dbname [, op_name]]
```

参数

dbname — 指定要为其生成报告的数据库名称。如果没有指定 *dbname*，则 **sp_dbcc_summaryreport** 将在 *dbccdb..dbcc_operation_log* 中生成所有数据库的报告，日期为 *date* 选项指定的日期和时间或在此之前的日期和时间。

op_name — 指定操作。*optype* 可以是 **checkstorage**（缺省值）或 **checkverify**，或者二者均是。如果未指定 *op_name*，将对所有操作生成报告。

示例

1. sp_dbcc_summaryreport

DBCC Operation : checkstorage

Database Name	Start time	End Time	Operation ID
Hard Faults	Soft Faults	Text Columns	Abort Count
User Name			
-----	-----	-----	-----
-----	-----	-----	-----
sybsystemprocs	05/11/1999 14:53:11	14:53:32:163	1
0	0	0	0
sa			
sybsystemprocs	05/11/1999 14:55:06	14:55:29:200	2
0	0	0	0
sa			
sybsystemprocs	05/11/1999 14:56:10	14:56:27:750	3
0	0	0	0

```
sa

DBCC Operation : checkverify

Database Name      Start time      End Time      Operation ID
Hard Faults      Soft Faults      User Name
-----
sybssystemprocs    05/11/1999 14:55:29  14:55:29:310      2
0                                     0 sa
```

对 **sybssystemprocs** 数据库生成总结报告，该报告提供有关所有执行的 **dbcc checkstorage** 和 **dbcc checkverify** 操作的信息。

2. sp_dbcc_summaryreport "testdb"

```
DBCC Operation : checkstorage

Database Name      Start time      End Time      Operation ID
Hard Faults      Soft Faults      Text Columns      Abort Count
User Name
-----
testdb            05/11/1999 14:55:29  14:55:49:903      1
0                                     0
sa
testdb            05/11/1999 14:55:50  14:56:9:546
2                                     0
0                                     0
sa
testdb            05/11/1999 14:56:28  14:56:40:666
3                                     0
0                                     0
sa
```

为用户数据库 **testdb** 生成总结报告，该报告提供有关所有执行的 **dbcc checkstorage** 操作的信息。**dbcc checkstorage** 是在此数据库上运行的唯一操作，因此报告中不会出现 **dbcc checkverify** 信息。

3. sp_dbcc_summaryreport null, "checkverify"

```
DBCC Operation : checkverify

Database Name      Start time      End Time      Operation ID
Hard Faults      Soft Faults      User Name
-----
sybssystemprocs    05/11/1999 14:55:29  14:55:29:310      2
0                                     0 sa
```

为 **master** 数据库生成总结报告，该报告提供有关所有执行的 **dbcc checkverify** 操作的信息。由于 **dbcc checkverify** 是指定的操作，因此报告中不包含 **dbcc checkstorage** 信息。

4. sp_dbcc_summaryreport sybssystemprocs,
"checkstorage"

```
DBCC Operation :checkstorage
```

Database Name	Start time	End Time	Operation ID		
Hard Faults	Soft	Faults	Text Columns	Abort	Count
User Name					

sybssystemprocs	05/11/1999	14:53:11	14:53:32:163		1
	0	0	0	0	
sa					
sybssystemprocs	05/11/1999	14:55:06	14:55:29:200		2
	0	0	0	0	
sa					
sybssystemprocs	05/11/1999	14:56:10	14:56:27:750		3
	0	0	0	0	
sa					

为 sybssystemprocs 数据库生成总结报告，该报告提供有关所有执行的 dbcc checkstorage 操作的信息。由于 dbcc checkstorage 是指定的操作，因此报告中不包含 dbcc checkverify 信息。

注释

- sp_dbcc_summaryreport 在指定的数据库上生成 checkstorage 或 checkverify 操作的总结报告，或者这两个操作的报告。
- 该报告指名被检查的数据库的名称、dbcc checkstorage 运行的开始和结束时间以及找到的软硬故障的数目。
- “Operation ID” 列包含编号，标识在特定时间在给定数据库上执行 dbcc checkstorage 操作的结果。报告中提供的编号来自 dbcc_operation_log 表的 opid 列。有关详细信息，参见系统管理指南。
- “Text Columns” 列显示运行期间 dbcc checkstorage 找到的非空文本列的数目。
- “Abort Count” 列显示包含错误的表的数目，这些错误导致了 dbcc checkstorage 中止在表上进行的检查。有关错误的详细信息，运行 sp_dbcc_faultreport。

权限

任何用户都可以运行 sp_dbcc_summaryreport。

使用的表

*master..sysdatabases、 dbccdb..dbcc_operation_log、
dbccdb..dbcc_operation_results*

参见

命令	dbcc
dbcc 存储过程	sp_dbcc_fullreport、 sp_dbcc_statisticsreport、 sp_dbcc_updateconfig

sp_dbcc_updateconfig

功能

用目标数据库的配置信息更新 *dbccdb* 中的 *dbcc_config* 表。

语法

```
sp_dbcc_updateconfig dbname, type, "str1" [, "str2"]
```

参数

dbname — 是要为其更新配置信息的目标数据库的名称。

type — 指定 *dbcc_types* 表的类型名称。第 10-32 页的表 10-2 显示 *type* 的有效值。

str1 — 指定要在 *dbcc_config* 表中更新的指定 *type* 的第一个配置值。第 10-32 页的表 10-2 说明指定 *type* 的 *str1* 预期值。

str2 — 指定要在 *dbcc_config* 表中更新的指定 *type* 的第二个配置值。第 10-32 页的表 10-2 说明指定 *type* 的 *str2* 预期值。

示例

1. `sp_dbcc_updateconfig pubs2, "max worker processes", "4"`

当检查 *pubs2* 数据库时为 *dbcc checkstorage* 提供最多数目的工作进程来更新 *dbcc_config*。新工作进程最大数目为 4。

2. `sp_dbcc_updateconfig pubs2, "dbcc named cache", pubs2_cache, "10K"`

用名称为 “*pubs2_cache*” 的 *dbcc* 的大小更新 *dbcc_config*。新的大小为 10K。

3. `sp_dbcc_updateconfig pubs2, "scan workspace", scan_pubs2`

用 *pubs2* 数据库的 *scan* 工作空间的新名称来更新 *dbcc_config*。新的名称是 *scan_pubs2*。该更新是在用 *sp_dbcc_alterws* 更改 *scan* 工作空间的名称后进行的。

4. `sp_dbcc_updateconfig pubs2, "text workspace", text_pubs2`

用 *pubs2* 数据库的 *text* 工作空间的新名称来更新 *dbcc_config*。新的名称是 *scan_pubs2*。该更新是在用 *sp_dbcc_alterws* 更改 *text* 工作空间的名称后进行的。

5. `sp_dbcc_updateconfig pubs2, "OAM count threshold", 5`

用 *pubs2* 数据库的 OAM 计数阈值来更新 *dbcc_config*。新值为 5。

6. `sp_dbcc_updateconfig pubs2, "IO error abort", 3`

用 *pubs2* 数据库的 I/O 错误中止值来更新 *dbcc_config*。新值为 3。

7. `sp_dbcc_updateconfig pubs2, "linkage error abort", 8`

用 *pubs2* 数据库的链接错误中止值来更新 *dbcc_config*。新值为 8。

注释

- `sp_dbcc_updateconfig` 为目标数据库更新 *dbcc_config* 表。
- 如果目标数据库的名称为 *dbccdb*，并且数据库 *dbccalt* 存在，`sp_dbcc_updateconfig` 更新 *dbccalt* 中的 *dbcc_config* 表。
- 如果在 *dbcc_config* 中没有找到目标数据库的名称，`sp_dbcc_updateconfig` 添加该名称并在更新其它配置信息前将操作顺序号设置为 0。
- 如果指定 *type* 的预期值是一个数字，则 `sp_dbcc_updateconfig` 将把您为 *str1* 和 *str2* 提供的值转换为数字。
- 表 10-2 显示用于 *type* 的有效类型名称和 *str1* 或 *str2* 的预期值。

表 10-2: 类型名称和预期值

<i>type</i> 名称	<i>str1</i> 或 <i>str2</i> 的预期值
dbcc named cache	<i>str1</i> 指定的缓存名称和 <i>str2</i> 指定的新大小（以千字节或兆字节计）或 2K 页页数。
IO error abort	新的错误计数，由 <i>str1</i> 指定。该值必须是大于 0 的数。 <i>str2</i> 不用于此类型。
linkage error abort	在 <i>str1</i> 中指定的新链接错误计数。该值必须是大于 0 的数。 <i>str2</i> 不用于此类型。
max worker processes	新的工作进程数，由 <i>str1</i> 指定。该值必须是大于 0 的数。 <i>str2</i> 不用于此类型。
OAM count threshold	新的阈值计数，由 <i>str1</i> 指定。该值必须是大于 0 的数。 <i>str2</i> 不用于此类型。
scan workspace	<i>scan</i> 工作空间的新名称，由 <i>str1</i> 指定。 <i>str2</i> 不用于此类型。
text workspace	<i>text</i> 工作空间的新名称，由 <i>str1</i> 指定。 <i>str2</i> 不用于此类型。

- 有关 *type* 名称和值的详细信息，参见 *系统管理指南*。

权限

只有系统管理员或数据库所有者才能够运行 **sp_dbcc_updateconfig**。

使用的表

master..sysdatabases、*dbccdb..dbcc_types*、*dbccdb..dbcc_config*

参见

命令	dbcc
dbcc 存储过程	sp_dbcc_evaluatedb
系统过程	sp_plan_dbccdb、sp_help

如需索引，参见第四卷“表格和参考手册索引”。

第四卷“表格和参考手册索引”包括 *Adaptive Server Enterprise 参考手册* 中所有卷的索引条目。

